

UNIVERSIDADE FEDERAL DE SANTA CATARINA

SUSAN MOLLER FERREIRA

AVALIAÇÃO EXPERIMENTAL DO PADRÃO 802.1X PARA PROVER
MOBILIDADE E SEGURANÇA EM REDES DE COMPUTADORES

FLORIANÓPOLIS (SC)
2004

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMATICA E ESTATÍSTICA
CURSO DE CIÊNCIAS DA COMPUTAÇÃO

SUSAN MOLLER FERREIRA

AValiação EXPERIMENTAL DO PROTOCOLO 802.1X PARA PROVER
MOBILIDADE E SEGURANÇA EM REDES DE COMPUTADORES

**Trabalho de Conclusão de Curso
apresentado ao Curso de Ciência da
Computação, como requisito à
obtenção do título de Bacharel em
Ciência da Computação, da
Universidade Federal de Santa
Catarina.**

**ORIENTADOR: GUILHERME ELISEU RHODEN
COORIENTADOR: CARLOS BECKER WESTPHALL
BANCA: EDISON TADEU LOPES MELO**

FLORIANÓPOLIS (SC)
2004

SUSAN MOLLER FERREIRA

**AValiação EXPERIMENTAL DO PROTOCOLO 802.1X PARA PROVER
MOBILIDADE E SEGURANÇA EM REDES DE COMPUTADORES**

**Trabalho de Conclusão de Curso apresentado ao Curso de Ciência da Computação,
como requisito à obtenção do título de Bacharel em Ciência da Computação, da
Universidade Federal de Santa Catarina.**

Orientador: Guilherme Eliseu Rhoden, Mestre

Coorientador: Carlos Becker Westphall, Doutor

Banca: Edison Tadeu Lopes Melo, Mestre

Ao meu pai Celso e a meu avô Joca que colaboraram muito para eu ser a pessoa que sou hoje e adorariam ver a realização deste trabalho. Vocês estarão sempre presentes.

Agradecimentos

Gostaria de agradecer ao Guilherme Rhoden pela orientação, por toda a ajuda e pela paciência com minhas “listas de perguntas”. Ao Edison Melo por compor a banca e pelos valiosos conselhos. Ao Fernando Cerutti pela ajuda, sempre bem humorada, com “todos aqueles conceitos e normas”. E a todos amigos e colegas do NPD que colaboram diariamente com conversas, conselhos e sorrisos.

Agradeço também ao professor Westphall pela co-orientação e por, com sua experiência, ter ajudado sempre que foi preciso. Não esquecendo do professor Olinto por ter sido um bom orientador e amigo nas primeiras, e grandes, dúvidas sobre possíveis temas de projeto de conclusão de curso.

Aos meus amigos pelo estímulo, conversas, músicas, brincadeiras e risadas. Ao meu namorado Carlos pela paciência e pelo esforço em se manter acordado mesmo escutando a mesma coisa muitas e muitas vezes. A meu irmão Celso por todo o apoio, encorajamento e pelas longas conversas.

E principalmente a minha família, que mesmo muitas vezes não podendo estar tão perto quanto gostaria, se faz sempre presente através dos ensinamentos passados por toda a minha vida. Agradeço especialmente a minha mãe pelo exemplo de força e coragem e as minhas avós Oma e Maria.

RESUMO

Este trabalho tem o intuito de realizar um estudo sobre o padrão 802.1x, objetivando através deste, a obtenção de maior segurança e a possibilidade de mobilidade aos usuários da rede. Inicialmente é realizada uma revisão de alguns conceitos de segurança evidenciando a sua importância em redes de computadores, seguindo-se com um estudo aprofundado sobre o padrão 802.1x. A seguir são apresentadas as tecnologias Radius e LDAP utilizadas no ambiente de testes descrito da seção seguinte. Através do estudo teórico realizado e da implantação do ambiente de teste pôde ser obtido um ambiente que possibilite a mobilidade do usuário na rede e uma melhora na segurança e gerência de redes através da integração entre as tecnologias acima citadas.

Palavras-chave: 802.1x; Segurança em Redes de Computadores; Mobilidade; Radius; LDAP; EAP; EAPOL.

ABSTRACT

This work has the intention to realize a study about the 802.1x standard, objectifying through this, the improvement of security and the mobility to networks users. Initially, it is carried through a revision of some concepts of security, evidencing its importance in computer networks. It is followed by a deep study on the 802.1x standard and the presentation of the technologies RADIUS and LDAP used in the environment of tests, which is described in this study. Through the studies and tests realized, it can be concluded that the integration of the technologies cited above makes possible the mobility of users in the network and an improvement in the security and management of them.

Palavras-chave: 802.1x; Security of computers networks; Mobility; RADIUS; LDAP; EAP; EAPOL.

SUMÁRIO

Lista de Figuras.....	x
Lista de Tabelas.....	xi
Lista de Siglas.....	xii
1. INTRODUÇÃO.....	1
1.1 Proposta do Trabalho.....	2
1.2 Objetivo.....	2
1.3 Justificativa.....	3
1.4 Áreas de Conhecimento.....	4
1.5 Organização do Trabalho.....	5
2. Segurança em Redes de Computadores.....	6
2.1 Introdução.....	6
2.2 Ameaças.....	7
2.3 Políticas e serviços de segurança.....	7
2.3.1 Autenticação.....	8
2.3.2 Controle de Acesso.....	9
3. Padrão 802.1x.....	11
3.1 Introdução.....	11
3.2 Definições.....	12
3.3 Princípios de operação.....	14
3.3.1 Entidade de acesso à porta.....	14
3.3.2 Acesso controlado e não controlado.....	15
3.3.3 Controle Unidirecional e Bidirecional.....	19
3.4 EAP.....	20
3.5 EAPOL.....	23
3.5.1 Formato do Frame e definição dos campos.....	23
3.5.2 Validação de frames EAPOL recebidos.....	25
3.6 Exemplo de comunicação do EAP.....	25
4. Tecnologias utilizadas.....	28
4.1 Radius.....	28
4.1.1 Mobilidade.....	29
4.2 LDAP.....	30
5. Ambiente de Teste.....	33
5.1 Funcionamento do ambiente.....	33
5.2 EAP – MD5.....	35
5.3 Recursos de Hardware.....	36
5.3.1 Computadores.....	36
5.3.2 Switches.....	37
5.4 Suplicante.....	37
5.4.1 MS Windows XP.....	38
5.4.2 MS Windows 2000.....	39
5.4.3 Outros clientes.....	39
5.5 Autenticador.....	40
5.5.1 Configuração básica.....	40
5.5.2 Funcionalidades adicionais.....	41
5.6 Servidor de Autenticação.....	44
5.6.1 MS Radius.....	44

5.6.2	FreeRADIUS	45
5.7	OpenLDAP.....	47
5.8	Resultados obtidos.....	49
6.	Conclusões e Trabalhos Futuros	51
6.1	Conclusão.....	51
6.2	Trabalhos Futuros.....	52
7.	REFERÊNCIAS BIBLIOGRÁFICAS	54
8.	ANEXOS	57
	ANEXO A – Arquivos de Configuração do FreeRADIUS.....	57
	ANEXO B – Log do switch Cisco Catalyst 3750 utilizando mapeamento de vlans.....	69
	ANEXO C – Configuração e dados do Enterasys Matrix E-Series.....	72
	ANEXO D – Radius.schema	74
	ANEXO E – Artigo.....	85

Lista de Figuras

Figura 1 - Cenário do ambiente 802.1x.....	12
Figura 2 - Portas controladas não controladas.....	15
Figura 3 - Efeito do estado de autorização em portas controladas	16
Figura 4 - Uso da porta controlada e não controlada.....	17
Figura 5 - Relação entre o suplicante, autenticador e servidor de autenticação	18
Figura 6 - Sistemas representando papéis de autenticador e suplicante	19
Figura 7 - Formato do frame EAP	21
Figura 8 - Formato do frame EAPOL para 802.3.....	24
Figura 9 - Autenticação com sucesso.....	26
Figura 10 - Autenticação rejeitada.....	27
Figura 11 - Modelo de arvore LDAP.....	32
Figura 12 - Modelo do ambiente de teste.....	33
Figura 13 - Interface de configuração do 802.1x, Windows XP	38
Figura 14 - Ambiente com autenticação <i>multi-host</i>	42
Figura 15 - Interface do IAS no Windows 2000 Server.....	45
Figura 16 - Interface LDAP Browser.....	48
Figura 17 - Árvore do OpenLDAP utilizada na implementação do sistema.....	49

Lista de Tabelas

Tabela 1 - Tipos de EAP	22
Tabela 2 - Relação entre sistemas operacionais e clientes 802.1x compatíveis.....	39

Lista de Siglas

EAP - Extensible Authentication Protocol

EAPOL - EAP over LANs

IAS – Internet Authentication Service

IEEE - Institute of Electrical and Electronics Engineers

LAN - Local Area Network

LDAP - Lightweight Directory Access Protocol

MAC – Media Access Control

MIB – Management Information Base

OTP – One Time Password

PAC - Protected Access Credentials

PAE - Port Access Entity

PEAP – Protected Extensible Authentication Protocol

PDU - Protocol Data Unit

Port - Network Access Port

PPP – Point-to-Point Protocol

RADIUS - Remote Authentication Dial in User Service

RAS - Remote Access Service

TLS - Transport Level Security

UDP – User Datagram Protocol

VPN - Virtual Private Network

1. INTRODUÇÃO

Nos últimos anos as redes de computadores têm se desenvolvido rapidamente. Empresas, lojas, escolas e outras instituições cada vez mais utilizam redes locais em seus departamentos. Como um grande fluxo de pessoas normalmente transita por essas instituições gerou-se uma preocupação maior com a segurança da rede.

Qualquer pessoa que tenha acesso a um computador da instituição tem livre acesso à rede local. Esta é uma deficiência que pode ser muito prejudicial, pois um usuário mal intencionado pode tentar acessar dados confidenciais, iniciar um ataque ou ainda contaminar com vírus existente em seu laptop todas as estações da rede. Assim criou-se uma necessidade de restringir o acesso a rede apenas a pessoas autorizadas.

Outra limitação observada é a falta de mobilidade da rede. Normalmente um funcionário acessa a rede de seu departamento com políticas de controle de acesso que permitem ou não que este acesse determinadas máquinas. Mas muitas vezes em uma empresa as atividades de um departamento são integradas a outros departamentos, assim um funcionário nem sempre vai utilizar o rede em uma estação localizada em seu departamento de origem. Logo é uma grande necessidade das instituições que os funcionários possam utilizar a rede através de qualquer ponto de acesso e obter suas próprias políticas de controle de acesso.

Reconhecendo essas deficiências das redes de computadores, geralmente implementadas nas corporações, este trabalho tem o objetivo de solucioná-las através da utilização do padrão 802.1x. Este tem como promessa aumentar a segurança em redes de computadores e facilitar implementações que propiciem

mobilidade na rede.

De modo a proporcionar maior flexibilidade de uso, este trabalho propões a integração do padrão 802.1x ao serviço de diretórios LDAP e ao servidor de autenticação Radius. Esta integração foi experimentada no ambiente de testes utilizado para validar a implementação. Mais informações sobre estas tecnologias e sua integração serão apresentadas posteriormente.

1.1 Proposta do Trabalho

Este trabalho tem como proposta realizar um estudo sobre o padrão 802.1x, analisando seu funcionamento e suas características. Com base nesta pesquisa será realizada uma avaliação experimental deste padrão, visando prover mobilidade e aumentar a segurança em rede de computadores.

1.2 Objetivo

Objetivo Geral

- Avaliação do padrão 802.1x.

Objetivos Específicos

- Avaliar a importância de mobilidade e segurança em redes de computadores;
- Estudar o padrão 802.1x;
- Estudo e integração dos padrões Radius, LDAP e 802.1x;
- Montagem de ambiente de teste utilizando as tecnologias acima;

- Avaliar a funcionalidade deste ambiente observando aspectos de segurança e mobilidade;
- Avaliar a aderência de alguns equipamentos disponíveis no ambiente de testes ao padrão 802.1x.

1.3 Justificativa

A segurança dos dados e equipamentos, que estão conectados a uma rede de computadores, é uma grande preocupação da população nos dias de hoje. Equipes de diversas áreas dependem de uma rede estável e confiável para obterem bons resultados em seus trabalhos. Percebendo esta necessidade, este estudo tem como uma de suas metas aumentar a segurança em redes de computadores através da implementação do controle de acesso em nível de usuário ao contrario da prática atual onde o controle de acesso ocorre em nível de estação de trabalho. Segundo Rhoden (2002), a segurança em redes de computadores está sendo considerada um fator primordial para um melhor funcionamento de uma rede. Logo melhorias na segurança de uma rede resultarão em uma melhor qualidade de trabalho para seus usuários.

A realização de uma pesquisa bibliográfica sobre o padrão 802.1x resultará em um documento com informações tanto sobre essa tecnologia quanto sobre outras tecnologias a ela relacionadas. Podendo assim, servir como base para outros projetos, visto que trata de um tema ainda pouco explorado.

1.4 Áreas de Conhecimento

- Redes de Computadores;
- Segurança em redes de computadores;
- Mobilidade em redes de computadores;
- Gerência de redes de computadores;
- Padrão 802.1x;
- Sistema de autenticação através do Radius;
- Serviços de diretórios baseados no padrão LDAP.

1.5 Organização do Trabalho

Este trabalho está dividido em 6 capítulos:

- Capítulo 1 – Trás informações sobre o tema, os objetivos pretendidos bem como as justificativas para o estudo;
- Capítulo 2 – Faz uma revisão de conceitos sobre segurança em redes de computadores, evidenciando sua necessidade;
- Capítulo 3 – Mostra um estudo detalhado sobre o padrão 802.1x, observando seu funcionamento, características e os avanços que serão obtidos através da utilização deste em redes de computadores;
- Capítulo 4 – Apresenta sucintamente os padrões LDAP e Radius, detalhando as adaptações necessárias para a integração dessas tecnologias com o 802.1x;
- Capítulo 5 – Este capítulo é dedicado ao ambiente de avaliação experimental do padrão 802.1x, mostrando detalhes sobre sua implementação e análise dos resultados obtidos.
- Capítulo 6 – Finalizando, são apresentadas as conclusões e sugestões para trabalhos futuros.

2. Segurança em Redes de Computadores

2.1 Introdução

Este capítulo tem como objetivo dar uma introdução ao tema segurança em redes de computadores, facilitando assim a compreensão dos estudos realizados posteriormente. São apresentados algumas definições e conceitos básicos que serão futuramente utilizados no trabalho.

Segundo Matos (1999), enquanto os computadores eram vistos como equipamentos isolados, a grande preocupação era proteger-se de intrusos no prédio onde o mesmo se situava. No entanto, a possibilidade de distribuir informações através de computadores espalhados em uma rede, trouxe preocupações com o risco de acessos não autorizados aos dados. Assim não bastava impedir o acesso físico aos equipamentos, criando uma necessidade de proteger os computadores de acessos indevidos pela rede.

O termo segurança é usado com o significado de minimizar a vulnerabilidade de bens (qualquer coisa de valor) e recursos. Vulnerabilidade é qualquer fraqueza que pode ser explorada para se violar um sistema ou as informações que ele contém. (ISO, 1989).

A segurança está relacionada à necessidade de proteção contra o acesso ou manipulação, intencional ou não de informações confidenciais por elementos não autorizados, e a utilização não autorizada do computador ou de seus dispositivos periféricos (Soares, 1995).

2.2 Ameaças

O conceito de ameaça é utilizado para definir uma possível violação de segurança. Uma ameaça a um sistema computacional consiste em uma ação possível que, uma vez concretizada, poderá produzir efeitos indesejáveis sobre os dados ou recursos do sistema (Obelheiro, 2001). A consolidação de uma ameaça pode se realizar através da exploração de alguma vulnerabilidade (falha) do sistema.

Segundo Soares (1995) as ameaças podem ser classificadas como acidentais (sem intenção premeditada) ou intencionais. A realização de uma ameaça intencional configura um ataque.

As principais ameaças às redes de computadores são:

- Destruição de informação ou de outros recursos;
- Modificação ou deturpação da informação;
- Roubo, remoção ou perda de informação ou de outros recursos;
- Revelação de informação, e
- Interrupção de serviços.

2.3 Políticas e serviços de segurança

Como consequência do grande número de ameaças que um sistema esta exposto, percebeu-se a necessidade de estabelecer medidas preventivas de segurança. Com este intuito foram definidas políticas de segurança.

Políticas de segurança definem regras que regulamentam como a organização gerenciará e protegerá suas informações (Silva, 2003). É importante

estabelecer uma política de segurança, pois desta forma, as medidas aplicadas podem cessar tentativas de ataque ou pelo menos diminuí-las consideravelmente.

Segundo Stallings (1998, p. 9) os serviços de segurança podem ser classificados como:

- Confidencialidade - impede que certas informações sejam acessadas por pessoas não autorizadas;
- Autenticação – assegura que a origem da informação seja corretamente identificada;
- Integridade – protege a informação contra modificações não autorizadas;
- Não Repúdio – impede que o responsável pelo envio ou recebimento de determinada informação possa negar a transmissão;
- Controle de Acesso – restringi/permite o uso de recursos a determinados usuários, e
- Disponibilidade – requer que os sistemas computacionais estejam disponíveis a usuários autorizados quando necessários.

Alguns dos benefícios providos pelo padrão 802.1x estão relacionados aos serviços de autenticação e controle de acesso, logo a compreensão desses serviços é de suma importância para este estudo. Uma visão mais detalhada destes é apresentada a seguir.

2.3.1 Autenticação

O processo de autenticação é responsável por garantir a procedência da informação. A autenticação é realizada no início da conexão, identificando se a

origem e/ou destino da chamada é verdadeira (FILHO, 2000). As técnicas de autenticação podem ser simples ou complexas, dependendo do ambiente onde se dará a autenticação.

Segundo Fernandes (2000) e Soares (1995) existem três métodos básicos pra verificar a identidade de um usuário:

- Algo conhecido pelo usuário – São usadas senhas de acesso. É o método mais utilizado, mas não o mais seguro. Deve ser usado em sistemas em que os parceiros e os meios de comunicação são seguros.
- Algo de posse do usuário – São empregados métodos de criptografia, como por exemplo, o uso de chaves criptográficas. É indicado quando a entidade confia no parceiro, mas não no meio de comunicação.
- Algo do próprio usuário – São utilizados mecanismos como impressões digitais, padrões de retina e assinaturas digitais. Este é o método mais seguro, e também o mais dispendioso.

O padrão 802.1x prove uma maior segurança a uma rede de computadores através da autenticação de seus usuários, podendo utilizar os três métodos de autenticação acima citados. A seleção do método deve ser feita analisando-se o nível de segurança desejado e a complexidade para sua implementação.

2.3.2 Controle de Acesso

O controle de acesso é um mecanismo utilizado para restringir o uso de um recurso a usuários não autorizados. Assim apenas usuários previamente cadastrados e autorizados têm permissão para acessar o recurso ou serviço.

Segundo Silva (2003), deve haver uma preocupação com o acesso de pessoas a ambientes de uma organização. Em instituições públicas de ensino, por exemplo, este controle é quase inexistente, apenas poucas pessoas controlam grandes áreas de prédios e o acesso pode ser facilitado pelo grande fluxo por corredores e salas.

O controle de acesso deve compreender as instalações físicas e lógicas da empresa. Tanto acessos físicos como virtuais devem ser monitorados e controlados. Este controle deve ser baseado na aplicação de regras, que limitam o acesso de uma entidade as informações e recursos, com base na comparação do seu nível de autorização relativo a essa informação ou recurso.

O padrão 802.1x permite o controle de acesso à rede, permitindo o acesso aos seus serviços apenas a usuários previamente autorizados.

3. Padrão 802.1x

3.1 Introdução

Reconhecendo a necessidade de um novo mecanismo para autenticação a IEEE (*Institute of Electrical and Electronics Engineers*) aprovou o padrão 802.1x em junho de 2001, que tem como promessa aprimorar a segurança em redes de computadores.

Vários métodos de autenticação já foram aplicados no controle de acesso à rede, sendo a maioria baseado na autenticação de dispositivos não autenticando seus usuários. Para assegurar que uma LAN está sendo utilizada apenas por usuários autorizados, o padrão IEEE 802.1x define um novo tipo de segurança de acesso, que requisita a todos os usuários uma prévia autenticação antes da disponibilização dos recursos e serviços da rede. Este controle de acesso é realizado com base em portas.

Segundo o padrão IEEE 802.1X, o controle de acesso à rede baseado em portas provê autenticação e autorização de equipamentos conectados a porta de uma LAN com características de conexão ponto-a-ponto, prevenindo o acesso a essa porta em casos de falha dos processos de autenticação e autorização. Uma porta é definida como um único ponto de conexão com uma LAN.

Adicionando-se a segurança, outro progresso que pode ser obtido através do 802.1x é a mobilidade em redes de computadores. A grande vantagem da mobilidade é a possibilidade de um usuário obter o mesmo nível e perfil de acesso à rede independente do local e da estação onde irá efetuar o acesso. Por exemplo, um empregado pode acessar a rede através de um ponto fora de seu departamento

utilizando suas próprias configurações. Estas configurações podem incluir seu endereço IP, políticas de *firewall*, permissão de acesso a máquinas restritas a sua rede interna, entre outras. Para a obtenção da mobilidade é necessário a integração do 802.1x ao Radius. Mais detalhes sobre sua implementação serão apresentados na seção 4.1.

É mostrada a seguir uma descrição mais detalhada do padrão 802.1x, tendo como base à norma IEEE 802.1x.

3.2 Definições

Para facilitar a compreensão inicialmente é feita a definição de alguns termos utilizados neste documento. Estes são representados na Figura 1, onde é mostrada a integração entre alguns equipamentos formando um ambiente de autenticação.

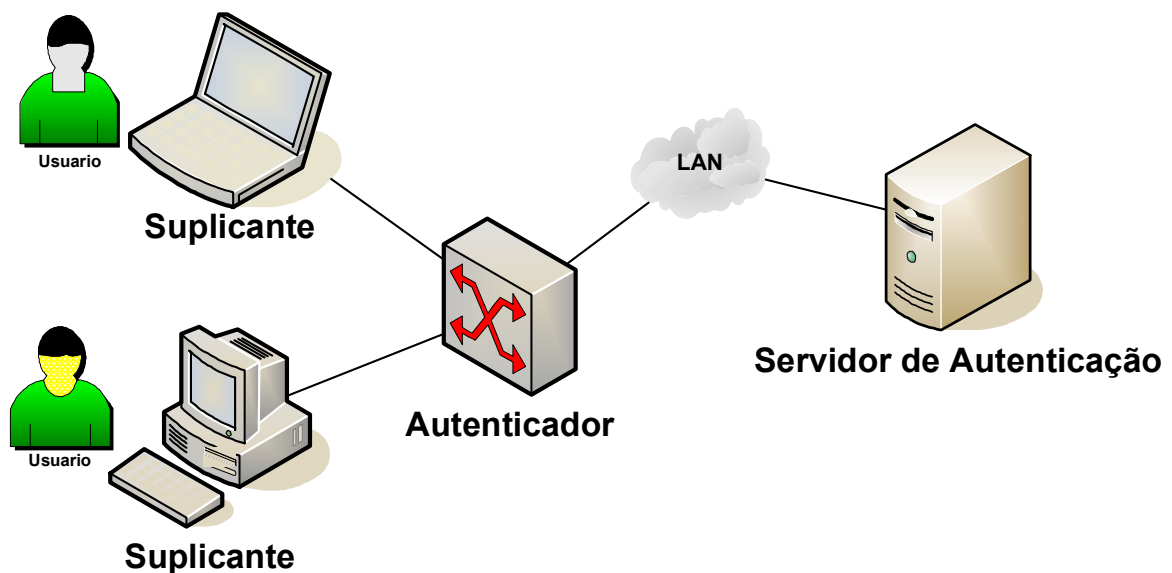


Figura 1 - Cenário do ambiente 802.1x

- **Suplicante (*Supplicant*):** É uma entidade numa ponta de um segmento de LAN ponto-a-ponto que está sendo autenticada por um autenticador conectado a outra ponta do segmento.
- **Autenticador (*authenticator*):** É uma entidade numa ponta de um segmento de LAN ponto-a-ponto que facilita a autenticação de uma entidade conectada a outra ponta;
- **Servidor de autenticação (*authentication server*):** É a entidade que provê o serviço de autenticação ao autenticador. As funções do servidor de autenticação podem estar anexadas ao autenticador, ou podem ser acessadas remotamente pela rede. Baseado nas credenciais do suplicante, o servidor de autenticação determina se o suplicante está autorizado a acessar os serviços providos pelo autenticador.
- **Porta de acesso à rede:** Ponto de conexão de um sistema a LAN. Pode ser uma porta física, por exemplo um host conectado fisicamente a um segmento de rede, ou lógica, por exemplo uma associação IEEE 802.11 entre uma máquina e um ponto de acesso. A porta de acesso à rede será referenciada como porta neste documento.
- **Entidade de acesso à porta (*Port Access Entity - PAE*):** Entidade do protocolo associada a uma porta. Pode suportar funcionalidades do protocolo associadas com o autenticador, com o suplicante ou com ambos.
- **Sistema:** Equipamento que está conectado a uma LAN através de uma ou mais portas.

3.3 Princípios de operação

A porta de um sistema pode acessar e oferecer serviços providos por outros sistemas conectados a LAN. O controle de acesso baseado em porta permite que as operações das portas sejam controladas para que apenas sistemas autorizados possam acessar os seus serviços.

3.3.1 Entidade de acesso à porta

A entidade de acesso à porta (PAE) opera os algoritmos e protocolos associados com os mecanismos de autenticação de determinada porta no sistema.

Uma PAE pode assumir um dos seguintes papéis em uma interação de controle de acesso:

- **Autenticador:** É a porta que requer autenticação antes de permitir acesso aos serviços disponíveis através dela. Ela é responsável pela comunicação com o suplicante e por repassar as informações do suplicante ao servidor de autenticação.
- **Suplicante:** É a porta que deseja acessar os serviços oferecidos pelo sistema de autenticação. Ela é responsável por responder as informações requisitadas pelo autenticador.

Uma porta pode assumir o papel de suplicante em alguns processos de autenticação e o papel de autenticador em outros. A PAE de autenticação controla o estado de autorizado e não autorizado das portas dependendo do resultado do processo de autenticação.

3.3.2 Acesso controlado e não controlado

A operação de controle de acesso baseado em portas cria dois diferentes pontos de acesso para conectar o sistema de autenticação a LAN. Um ponto de acesso, a porta não controlada, permite a troca de *Protocol Data Unit* (PDU) entre o sistema e outros sistemas da LAN, independente da autorização. O outro ponto, a porta controlada, permite a troca de PDU somente se o estado da porta for autorizado. Conforme ilustrado na Figura 2, as portas controladas e não controladas fazem parte do mesmo ponto de acesso a LAN, deste modo todos os *frames* recebidos pela porta física são encaminhados a ambas as portas.

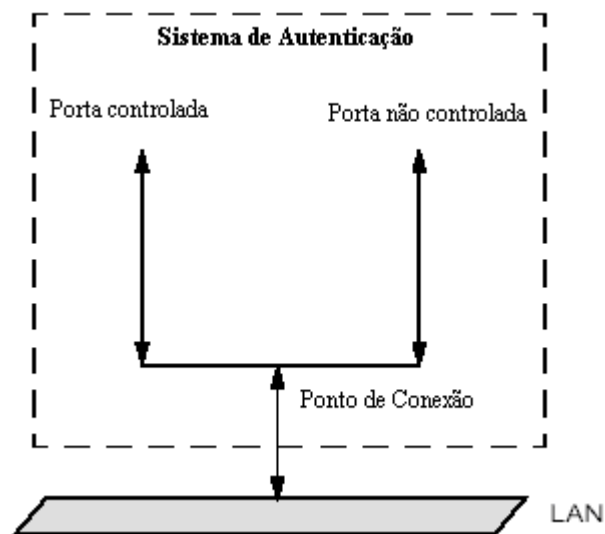


Figura 2 - Portas controladas não controladas.
Fonte: IEEE 802.1x.

Alguns parâmetros definem o estado das portas do sistema. A Figura 3 ilustra o efeito do *AuthControlledPortStatus* em uma porta controlada que permite ou não o fluxo de PDUs por esta porta. No sistema de autenticação 1, o *AuthControlledPortStatus* associado à porta controlada é não autorizado, ou seja,

ela não esta habilitada. Já no sistema 2 o *AuthControlledPortStatus* é autorizado, logo a porta está habilitada.

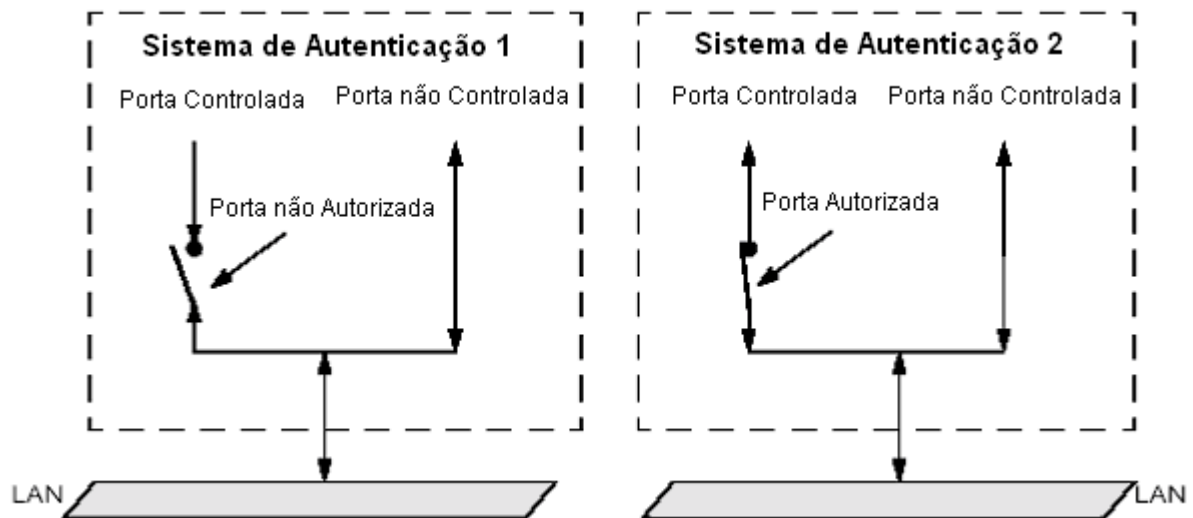


Figura 3 - Efeito do estado de autorização em portas controladas.
Fonte: IEEE 802.1x

Em conjunto ao *AuthControlledPortStatus* é definido um outro parâmetro. O *AuthControlledPortControl* permite controle administrativo sobre o estado de autorização das portas. Seus possíveis valores são:

- *ForceUnauthorized* - força a PAE de autenticação marcar o *AuthControlledPortStatus* como não autorizado. Assim a porta controlada fica incondicionalmente não autorizada.
- *ForceAuthorized* - força a PAE de autenticação marcar o *AuthControlledPortStatus* como autorizado. Assim a porta controlada fica incondicionalmente autorizada.
- *Auto* - permite que a PAE de autenticação marque o valor da *AuthControlledPortStatus* de acordo com o resultado dos processos de autenticação entre a PAE suplicante, a PAE de autenticação e o servidor de autenticação. Sendo este o padrão de inicialização do sistema.

Além do controle individual sobre cada porta, é possível o controle das portas do sistema como um todo. Podendo assim, o valor do *AuthControlledPortControl* de cada porta pode ser alterado por um parâmetro do sistema, o *SystemAuthControl*. Este parâmetro controla o processo de autenticação, podendo assumir os seguintes valores:

- *Enabled* – Habilita a autenticação no sistema. O estado de cada porta obedece ao *AuthControlledPortControl*.
- *Disabled* – Desabilita a autenticação para todas as portas, forçando-as a permanecer no estado autorizado. Este é o valor padrão na inicialização do sistema.

As portas não controladas são utilizadas pela PAE de autenticação para a troca de informações (protocolos) de autenticação com o suplicante. O uso de portas controladas e não controladas pode ser observado na Figura 4. Esta mostra também a habilidade da PAE de autenticação em alterar o estado de sua porta controlada dependendo do resultado do processo de autenticação.

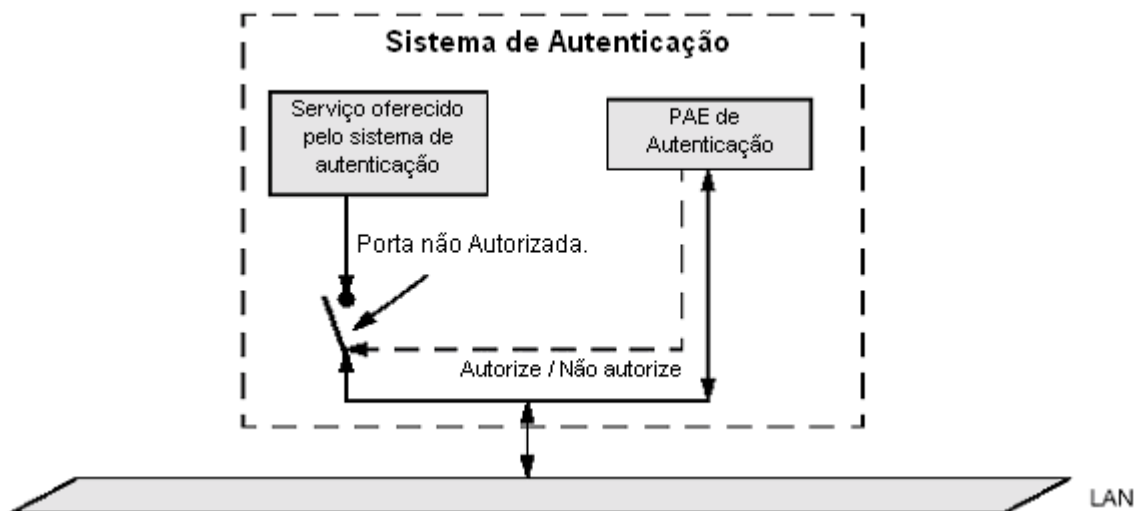


Figura 4 - Uso da porta controlada e não controlada.
Fonte: IEEE 802.1x

A relação entre o suplicante, autenticador e servidor de autenticação pode ser observada na Figura 5. Na ilustração a porta controlada do autenticador não está autorizada, impedindo assim o acesso a serviços por ele oferecidos. A porta não controlada é usada pela PAE de autenticação para se comunicar com a PAE suplicante, utilizando o protocolo EAPOL¹, e com o Servidor de autenticação utilizando EAP¹.

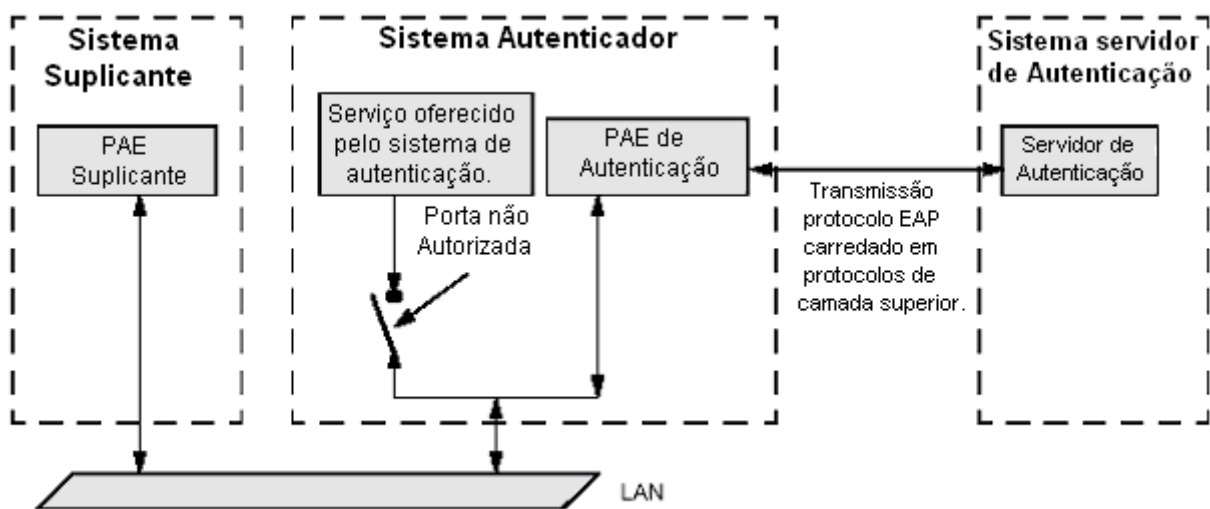


Figura 5 - Relação entre o suplicante, autenticador e servidor de autenticação
Fonte: IEEE 802.1x

A comunicação entre o autenticador e o servidor de autenticação pode ser feita através da LAN, ou por outro canal de comunicação. No caso em que o servidor de autenticação e o autenticador estão juntos, a troca de EAP entre as duas entidades é desnecessária.

A Figura 6 ilustra uma situação em que dois sistemas A e B podem representar os papéis de suplicante e autenticador quando necessário. Se o sistema A quiser utilizar algum serviço provido pelo sistema B, este deverá assumir o

¹ Estes protocolos serão definidos nas seções 3.4 e 3.5.

papel de suplicante e o B de autenticador. No caso em que B necessite de algum serviço provido por A os papéis são invertidos. Esta situação pode ser exemplificada considerando-se os dois sistemas como uma ponte. Quando eles forem inicialmente conectados, cada um vai requerer que a outra ponte seja autenticada e autorizada antes de iniciar a transmissão de seus *frames*.

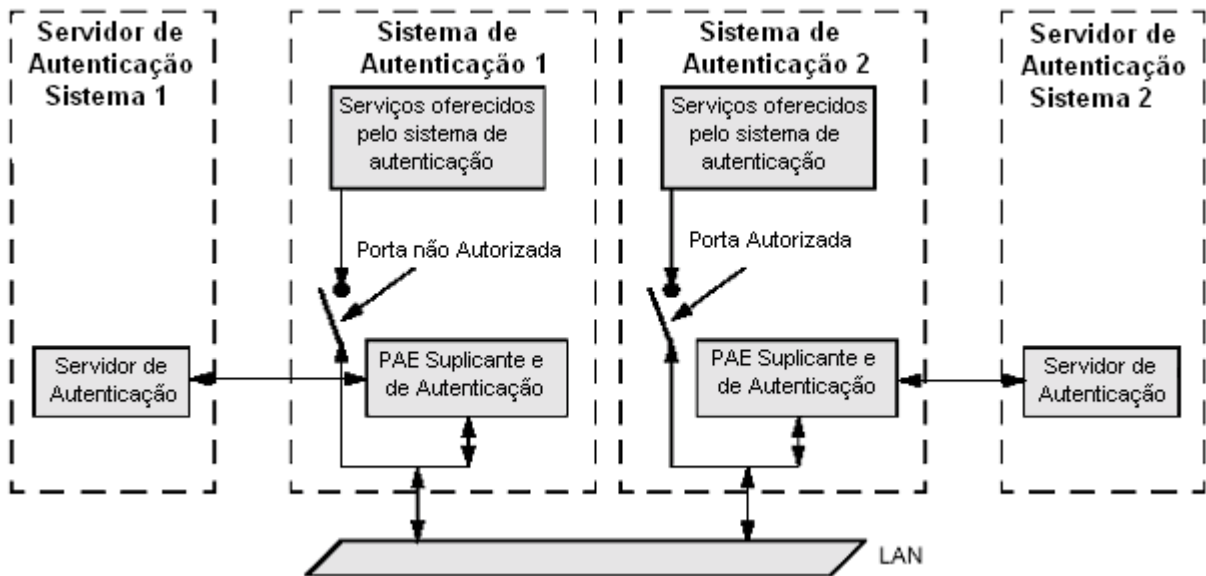


Figura 6 - Sistemas representando papéis de autenticador e suplicante.
Fonte: IEEE 802.1x

3.3.3 Controle Unidirecional e Bidirecional

Cada porta controlada tem dois parâmetros de controle de direção associados a ela, o *AdminControlledDirections* e o *OperationalControlledDirections*. Estes determinam se quando a porta não está autorizada ela desabilita somente a recepção de *frames*, ou também o envio. Os parâmetros funcionam da seguinte maneira:

- *AdminControlledDirections = Both*. – Indica que o controle exercido deve ser bidirecional, ou seja, o controle é exercido no tráfego de entrada e saída da porta. O valor *OperControlledDirections* também é marcado como *Both*.

- *AdminControlledDirections* = *In*. – Indica que o controle exercido deve ser unidirecional, ou seja, o controle é exercido somente no tráfego de entrada da porta. O valor *OperControlledDirections* na inicialização é marcado como *In*, podendo ser alterado dependendo da alteração de algumas condições no sistema.

A comunicação entre as PAEs do suplicante e autenticador é realizada através do protocolo EAPOL, já a comunicação entre autenticador e servidor de autenticação através do EAP. As seções seguintes apresentam as definições desses protocolos.

3.4 EAP

De acordo com as RFCs 2284 e 3748 o *Point-to-point Extensible Authentication Protocol* (EAP) é um protocolo geral, para autenticação ponto a ponto.

Rodando na camada de enlace o EAP não requer o uso de IP e tem seu próprio suporte a eliminação de duplicação e retransmissão de pacotes.

O grande benefício do EAP é a sua flexibilidade, podendo suportar múltiplos mecanismos de autenticação. Outra vantagem é que os métodos de autenticação podem ser implementados em um servidor, assim o autenticador não precisa entender cada pacote que chega até ele, podendo apenas retransmiti-los. Isso evita que o autenticador tenha que ser atualizado para suportar cada novo método de autenticação.

Conforme definido pela norma da IEEE (802.1x-2001) a Figura 7 representa o frame do EAP.

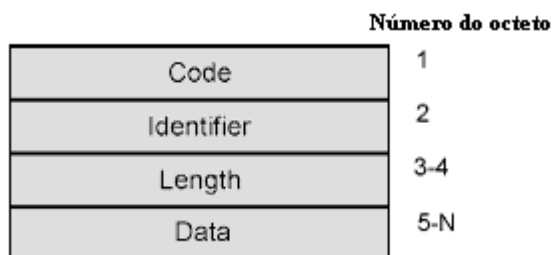


Figura 7 - Formato do frame EAP.
Fonte: IEEE 802.1x

Os campos observados na Figura 7 têm o seguinte significado:

- *Code* – define o tipo do pacote EAP. Podendo assumir os valores: *Request, Response, Success e Failure*.
- *Identifier* – Este campo é usado para garantir que a resposta recebida é corresponde à requisição enviada.
- *Length* – Indica o tamanho do pacote EAP
- *Data* – Contém os dados do pacote EAP, seu formato depende do tipo de pacote definido no campo *Code*.

Existem alguns métodos de autenticação definidos em RFCs. Segundo a RFC 3748 o método Tipo Expandido permite que cada fabricante desenvolva seu próprio tipo expandido de EAP, conforme as suas necessidades. Por essa razão existem diversas implementações de mecanismos de autenticação suportados pelo EAP.

A Tabela 1 foi realizada cruzando informações de Allen, INTEROP (a) , GAST, GEORGE (a) e GEORGE (b). Esta mostra algumas características dos tipos mais comuns de EAP.

	Estrutura Básica	Direção de Autenticação	Certificado de Servidor	Certificado de Cliente	Dificuldade de Implementação	Segurança
EAP-MD5	Autenticação através de <i>Challenge-based password</i>	Autentica apenas cliente	Não	Não	Simples	Ruim para redes sem fio.
LEAP	Autenticação via <i>hash</i> de senha e usuário.	Mutua	Não	Não	Médio	Bom, se a senha for boa.
PEAP	Autenticação do servidor via certificado e cliente por outro método EAP.	Mutua	Sim	Não	Médio	Bom
TTLS	Autenticação do servidor via certificado e cliente por outro método.	Mutua	Sim	Não	Médio	Bom
EAP-FAST	Autenticação mutua através de PAC e autenticação do cliente por senha.	Mutua	Sim	Não	Simples	Ótimo
EAP-TLS	Autenticação de cliente e servidor através de certificado.	Mutua	Sim	Sim	Difícil	Ótimo

Tabela 1 - Tipos de EAP

Conforme observado na Tabela 1 os tipos de EAP diferem na maneira de autenticação e conseqüentemente no nível de segurança e dificuldade de implementação.

Como pode ser observado na Tabela 1 o INTEROP (a) classifica o EAP-Fast como um método de fácil implementação e altamente seguro. Neste caso este método seria considerado o melhor entre os listados, pois tem a grande vantagem

da segurança sem o problema da dificuldade de implementação. Porém existem controvérsias quanto a isso. O EAP-Fast é um método novo, ainda pouco testado e não suportado pela maioria das ferramentas de autenticação. Segundo INTEROP (a), o EAP-Fast foi desenvolvido pela Cisco com a intenção de ser de simples implementação e altamente seguro. Por outro lado, GEORGE (a) considera isso *marketing*, o autor afirma que o método apresenta falhas de segurança e sua implementação não é tão simples quanto afirmado.

Já compreendida a estrutura do EAP, conforme definido no padrão IEEE 802.1X nas próximas seções será mostrado o EAPOL e o funcionamento da comunicação destes protocolos.

3.5 EAPOL

EAP over LANs, ou EAPOL, é uma técnica de encapsulamento usada para transmitir pacotes EAP entre a PAE do suplicante e a PAE do autenticador. Os *frames* de EAPOL transmitidos pela PAE não devem ter *tag*² de *vlan*³, mas podem ter uma *tag* de prioridades. Assim, toda a PAE deve ser capaz de receber EAPOL *frames* tanto com *tag* quanto sem *tag* de prioridade.

3.5.1 Formato do Frame e definição dos campos

Na Figura 8 pode ser observado o formato para *Ethernet* de um frame EAPOL.

² Marca em um *frame*.

³ Rede virtual normalmente utilizada em redes *Ethernet*.

	Número do octeto
PAE Ethernet Type	1-2
Protocol Version	3
Packet Type	4
Packet Body Length	5-6
Packet Body	7-N

Figura 8 - Formato do frame EAPOL para 802.3.
Fonte: IEEE 802.1x

- *PAE Ethernet Type* – Especifica o tipo de Ethernet que deve ser utilizado pela PAE.
- *Protocol Version* – Este campo identifica a versão do protocolo EAPOL suportada pelo remetente do frame.
- *Packet Type* – Determina o tipo de pacote transmitido. Possui alguns tipos reservados para uso futuro, que não devem ser utilizados. Os tipos determinados pela norma são:
 - *EAP-Packet*
 - *EAPOL-Start*
 - *EAPOL-Logoff*
 - *EAPOL-Key* – É uma função opcional, devendo ser utilizada apenas quando há suporte para transmitir informação de chave entre o autenticador e o suplicante.
 - *EAPOL-Encapsulated-ASF-Alert* – Tipo utilizado para permitir que alertas sejam encaminhados através de portas não autorizadas.
- *Packet Body length* – Define o tamanho, em octetos, do campo *Packet Body*. Caso o campo *Packet Body* não esteja presente, assume valor 0.
- *Packet Body* – Contém dados referentes ao tipo definido no *Packet Type*.

Este campo apenas está presente se o *Packet Type* do frame possui valor igual a *EAP-Packet*, *EAPOL-Key* ou *EAPOL-Encapsulated-ASF-Alert*.

3.5.2 Validação de frames EAPOL recebidos

Uma PAE deve processar um frame EAPOL recebido apenas se este preenche as seguintes condições:

- O endereço de MAC⁴ do destinatário corresponde ao endereço de MAC do grupo da PAE (conforme definido na norma IEEE 802.1d) ou ao endereço específico da PAE.
- O *PAE Ethernet Type* contém o valor do tipo de Ethernet utilizado pela PAE.
- O campo *Packet Type* contém um dos seguintes valores: *EAP-Packet*, *EAPOL-Start*, *EAPOL-Logoff* ou *EAPOL-Key*.

No caso de *frames* que contenham *Packet Type* com valor igual a *EAPOL-Start* ou *EAPOL-Logoff*, qualquer octeto encontrado na PDU após o campo *Packet Type* dever ser ignorado. Já para *frames* com *Packet Type* igual a *EAP-Packet* ou *EAPOL-Key* devem ser ignorados os octetos encontrados após o campo *Packet Body*.

3.6 Exemplo de comunicação do EAP

Como visto na seção 3.3.1 é de responsabilidade da PAE do autenticador a retransmissão das mensagens entre o suplicante e o servidor de autenticação. Para realizar esta retransmissão é necessário que os pacotes EAPs transmitidos entre o

⁴ Media Access Control

servidor de autenticação e o autenticador sejam encapsulados em pacotes EAPOL antes de serem transmitidos a PAE do suplicante. Além de encapsular os pacotes EAP, a PAE do autenticador deve desencapsular os pacotes EAPOL enviados pelo suplicante, e em seguida retransmiti-los ao servidor de autenticação.

As Figura 9 e Figura 10 ilustram a transmissão de pacotes entre a PAE do suplicante e a PAE do servidor de autenticação. Assim que o suplicante é conectado ao autenticador, a PAE do autenticador requisita sua identificação e passa a retransmitir os pacotes de autenticação entre o suplicante e o servidor de autenticação. Sendo os pacotes EAPOL representados através de linhas cheias e os pacotes EAP por linhas tracejadas. O protocolo OTP⁵ (*One Time Password authentication*) é utilizado de maneira ilustrativa, podendo ser substituído por outros protocolos de autenticação.

A Figura 9 mostra uma situação onde o usuário foi autenticado com sucesso. Logo, a porta em que este está conectado passa ao estado de autorizada e o usuário tem acesso a rede.

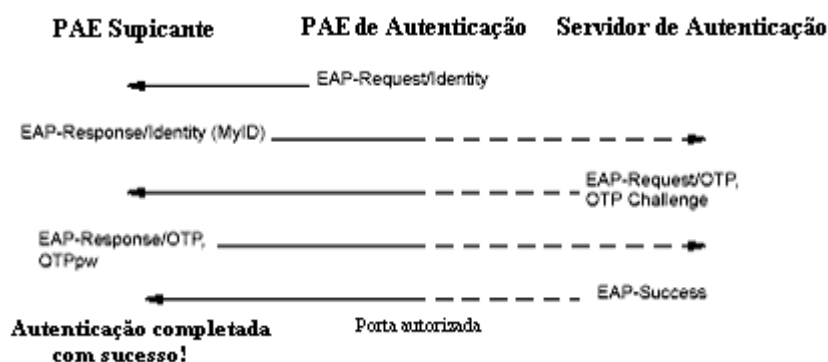


Figura 9 - Autenticação com sucesso.
Fonte: IEEE 802.1x

⁵ Conforme RFC 2289 OTP é um sistema de autenticação onde uma nova senha é gerada (sendo utilizada apenas uma vez) a cada processo de autenticação.

Já a Figura 10 mostra uma situação onde o usuário não está autorizado a acessar a rede. Assim, não há sucesso no processo de autenticação e a porta se mantém no estado de não autorizado.

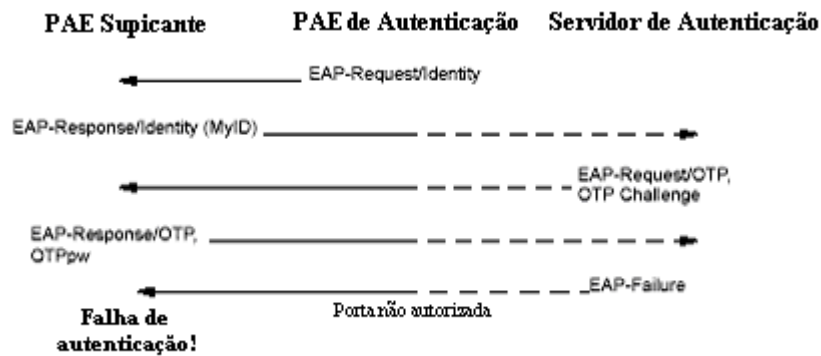


Figura 10 - Autenticação rejeitada.
Fonte: IEEE 802.1x

A norma sugere que a comunicação entre o autenticador e o servidor de autenticação seja feita utilizando-se o Radius. Obedecendo esta indicação, esta tecnologia será utilizada para a implantação do ambiente de teste. Para maior compreensão de seu funcionamento, no próximo capítulo, será apresentada uma definição deste protocolo.

4. Tecnologias utilizadas

No ambiente de teste implementado está sendo utilizado um servidor Radius para a autenticação dos usuários. Como no processo de autenticação as operações de leitura são mais freqüentes que as de escrita o acesso à base de dados deve ser otimizado. Com esse propósito o Radius foi interligado a um servidor LDAP, possibilitando assim, uma leitura de dados rápida e eficiente além das vantagens de um serviço de diretório que pode ser integrado a outras aplicações da instituição. Como por exemplo, o serviço RAS e VPN existentes na UFSC.

4.1 Radius

Segundo a RFC 2138 o crescente uso do serviço de acesso discado gerou a necessidade de um suporte administrativo. Para solucionar essa necessidade foi desenvolvido o protocolo Radius, com intenção de disponibilizar acesso à rede com autenticação, autorização e contabilização. Devido a sua simplicidade, eficiência e facilidade de implementação atualmente o Radius é suportado pela grande maioria dos equipamentos de rede, ou seja, por servidores VPN, *Access Points*, *switches*, roteadores e outros equipamentos de acesso à rede.

Ao receber uma requisição de conexão (oriunda de algum dispositivo de acesso) o servidor Radius autentica o usuário, verificando seu nome e senha. Após a autenticação, o servidor envia ao cliente todas as informações (configurações) necessárias para que o usuário tenha acesso a rede. Estas configurações podem

ser, por exemplo, o tipo do serviço que deverá ser usado pelo cliente (PPP⁶, telnet⁷, ...).

Para garantir a segurança, todos os autenticadores devem ser cadastrados no servidor Radius, sendo definida uma senha para autenticar a comunicação entre estes.

Ao invés do TCP o Radius utiliza o UDP como protocolo de transporte. A utilização do UDP é vantajosa, pois o fato deste protocolo não ser orientado a conexão simplifica a implementação do servidor e melhora sua eficiência.

Para facilitar o acesso aos dados o servidor Radius será integrado com o serviço de diretórios LDAP, este possui uma estrutura de diretórios otimizada nas operações de leitura. Na seção 4.2 é feita uma breve definição do funcionamento do LDAP.

A integração do 802.1x com o Radius possibilita a mobilidade na rede. Os detalhes para a implementação dessa funcionalidade são apresentados na seção 4.1.1.

4.1.1 Mobilidade

A mobilidade é uma funcionalidade de grande utilidade em redes de computadores. Através da mobilidade é possível que um usuário utilize sua política de controle de acesso para obter suas configurações e permissões utilizando qualquer ponto de acesso da rede.

Segundo a RFC 3580 esta funcionalidade é obtida através de alguns

⁶ Point-to-Point Protocol - Protocolo de comunicação usado para suportar comunicação através de uma conexão entre dois pontos.

⁷ Protocolo de comunicação utilizado para abrir uma sessão (terminal) em uma máquina remota.

parâmetros enviados pelo servidor Radius. Baseando-se no resultado da autenticação a porta é alocada a vlan a qual o usuário pertence.

Após autenticar o usuário, o servidor Radius indica ao autenticador a vlan a qual este usuário pertence. Estes parâmetros são informados através de atributos de túneis incluídos na mensagem de permissão de acesso. Os atributos de túneis são:

- Tunnel-Type=VLAN
- Tunnel-Medium-Type=802
- Tunnel-Private-Group-ID=VLANID

O parâmetro Tunnel-Private-Group-ID informa o identificador da sub-rede a qual o usuário pertence, através deste atributo o autenticador aloca a porta utilizada a vlan indicada pelo servidor Radius. Assim em qualquer ponto da rede este usuário obterá um endereço IP da sua sub-rede e utilizará as políticas de acesso definidas para esta.

4.2 LDAP

O *Lightweight Directory Access Protocol* (LDAP) é um protocolo de acesso a diretórios. Segundo a RFC 2251 ele surgiu a partir do serviço de diretórios X.500. O protocolo da camada de aplicação X.500, é conhecido por ser de implementação complexa e de alto custo. Logo o LDAP foi inicialmente desenvolvido como um protocolo leve que serviria como ponte às requisições a um servidor X.500.

Segundo Carter (2003), diretórios são facilmente confundidos com bases de dados, mas ambos diferem em alguns aspectos. Sendo que, a maior diferença

observada entre os dois é que um diretório é desenvolvido para executar muito mais operações de leitura do que escrita, enquanto que em uma base de dados as operações de leitura e escrita ocorrem com a mesma frequência. Uma vez que a maioria de suas operações seria de leitura o LDAP não suporta algumas ferramentas de escrita comuns nas bases de dados. Sendo um protocolo leve as operações de acesso a dados são rápidas e otimizadas.

Os modelos do LDAP representam o serviço provido pelo servidor pela visão do cliente. Conforme a RFC 2251 são definidos dois modelos, o modelo de protocolo e o modelo de dados.

O modelo de protocolo é definido como as operações realizadas entre cliente e servidor, funcionando da seguinte maneira. O cliente faz uma requisição ao servidor, este então fica responsável por executar as operações necessárias para executar o que lhe foi pedido. Ao completar, o servidor retorna os resultados ao cliente.

O modelo de dados do LDAP é estruturado na forma de uma árvore composta de entradas. Cada entrada possui atributos que a qualificam, sendo um deles, conhecido como *Relative Distinguished Name* (RDN), único de cada entrada e responsável pela identificação desta. Na Figura 11 pode ser observado um modelo de árvore do LDAP, sendo destacada a entrada com RDN igual a cn:gerald carter e listados alguns de seus atributos.

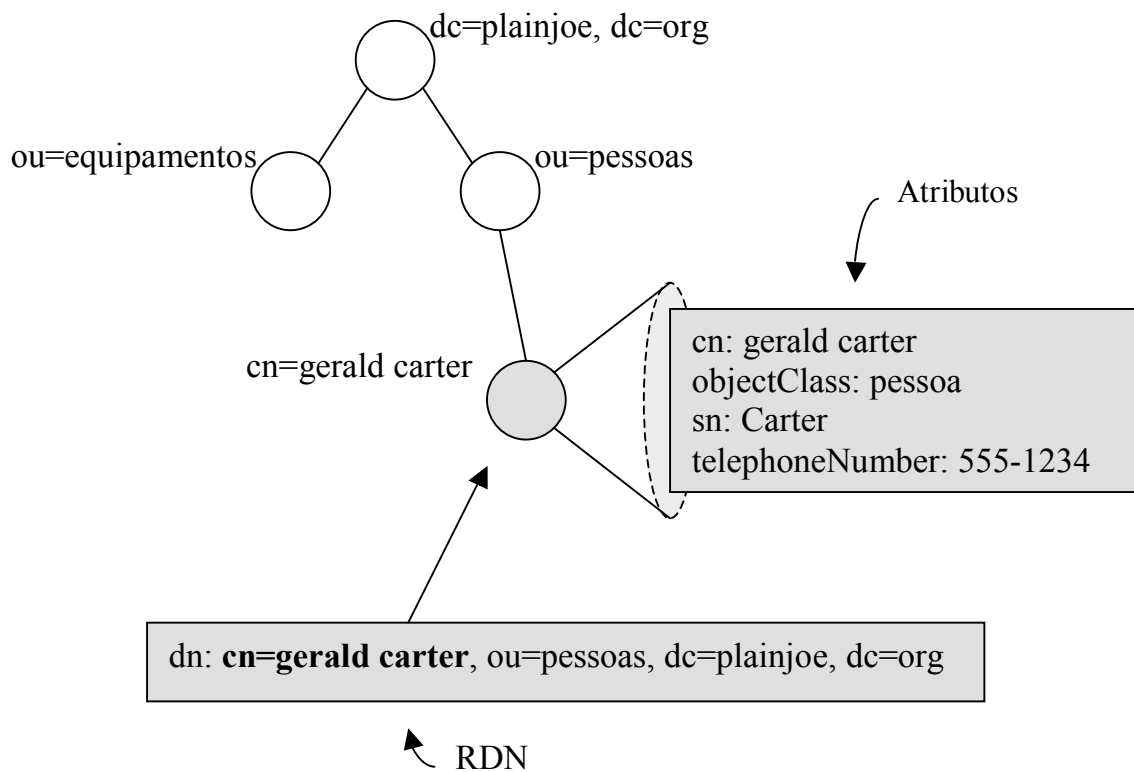


Figura 11 - Modelo de árvore LDAP
Fonte: Carter

Segundo Carter uma entrada pode ser localizada através do atributo DN (Distinguished Name), este corresponde à combinação dos RDN de todas as entradas existentes no caminho entre o nó inicial da árvore e a entrada requisitada. Para a entrada com RDN `cn=gerald carter`, mostrada na Figura 11, o valor do DN é “`cn=gerald carter, ou=peessoas, dc=plainjoe, dc=org`”.

Já apresentados tanto o LDAP, quanto o Radius e o 802.1x, estas tecnologias serão utilizadas para a implementação do ambiente de teste descrito no próximo capítulo.

5. Ambiente de Teste

Para analisar experimentalmente o funcionamento do padrão 802.1x foi montado um ambiente de teste. Este capítulo se dedica a descrever o funcionamento e os equipamentos envolvidos na montagem deste.

5.1 Funcionamento do ambiente

Para montagem desse ambiente foram utilizados alguns computadores, comutadores ethernet e as tecnologias Radius e LDAP. A Figura 12 mostra como foram interligados estes equipamentos.

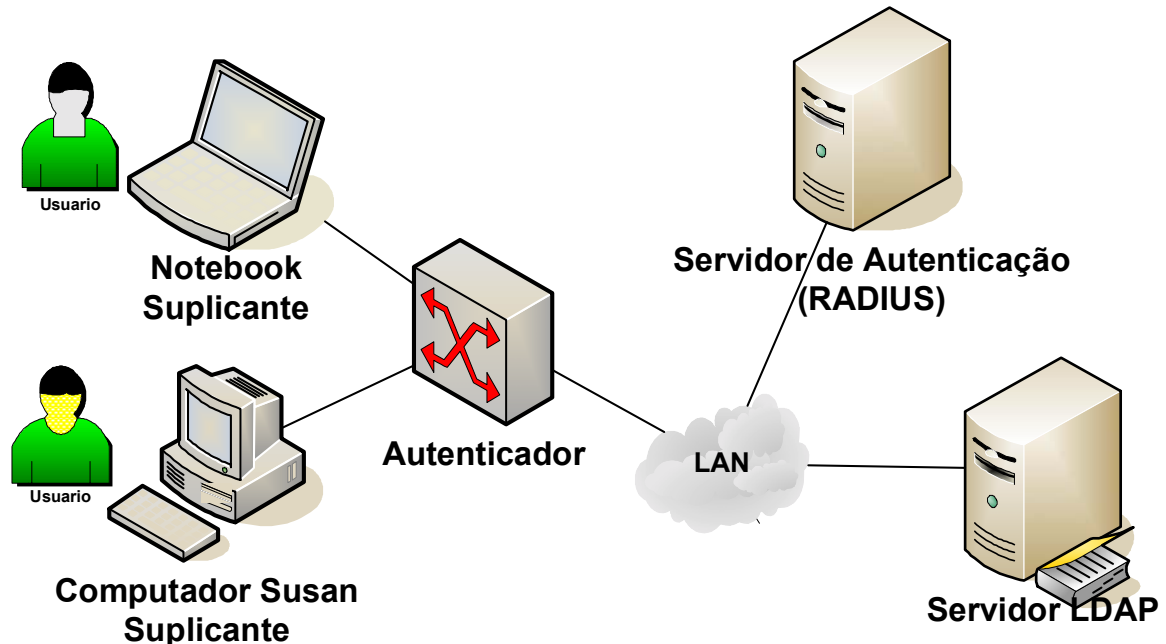


Figura 12 - Modelo do ambiente de teste.

Os suplicantes são conectados a um *switch* com autenticação 802.1x habilitada em suas portas. Como o *switch* exige a autenticação de seus usuários, o

suplicante só terá acesso a rede se após o termino do processo de autenticação ele for considerado um usuário autorizado.

Ao perceber que há um equipamento conectado em uma de suas portas, o *switch* envia um pacote EAPOL requisitando que o suplicante se identifique. Começando assim o processo de autenticação.

O suplicante envia seu identificador, e outras informações que forem necessárias, ao autenticador através de pacotes EAPOL. O *switch* transforma estes pacotes EAPOL em pacotes EAP e encaminha-os ao servidor Radius.

O Radius ao receber a identificação do usuário, requisita outros dados necessários à autenticação, como uma senha por exemplo, ao autenticador. Já de posse das informações do suplicante o Radius se conecta ao servidor LDAP, que busca as configurações deste usuário em seus diretórios. Caso este seja um usuário autorizado o Radius envia um pacote EAP ao autenticador, e assim o *switch* libera a porta para que o suplicante tenha acesso a rede.

A comunicação entre os equipamentos do sistema é realizada através do protocolo EAP. Um fator importante para a implementação do sistema é o tipo de EAP que será utilizado. Essa importância se deve ao fato do tipo de EAP determinar a forma de autenticação do usuário.

Dentre os possíveis tipos de EAP o EAP-MD5 foi escolhido para a implementação do sistema. Esta escolha se justifica pela sua simplicidade de implementação e pelo grande numero de equipamentos que suportam este padrão. Abaixo será apresentado o EAP-MD5 seguido no restante do capítulo com mais informações sobre os equipamentos utilizados no ambiente.

5.2 EAP – MD5

Conforme INTEROP (a), o MD5 é o método de autenticação de mais simples implementação. Seu processo de autenticação funciona da seguinte maneira:

- Autenticador envia ao suplicante uma *string* + um numero serial;
- Suplicante anexa à *string* a sua senha então calcula o *hash*;
- Suplicante envia ao autenticador o *hash*, provando assim que conhece a senha.

Este processo é conhecido como *Challenge-based authentication*.

No MD-5 a senha não é transmitida pela rede, o suplicante mostra que conhece a senha enviando o seu *hash*. Segundo Allen o nome do usuário é transmitido em texto em claro, mas sua senha é codificada utilizando-se *hash* MD-5. Para verificar se o *hash* enviado pelo suplicante corresponde ao *hash* da senha, o autenticador deve ter acesso à senha em texto em claro (sem criptografia) ou com criptografia reversível. Deste modo existe o perigo de obtenção de todas as senhas do sistema através de um acesso não autorizado ao arquivo de senhas no servidor.

Outra vulnerabilidade desse método é que ele autentica apenas o suplicante. Isso é um grande problema no caso de redes sem fio, fazendo com que muitos equipamentos para rede sem fio não suportem o MD-5.

Como visto o MD-5 não é um método de autenticação extremamente seguro, mas tem a grande vantagem de não apresentar a complexidade de implementação encontrada em outros métodos.

5.3 Recursos de Hardware

Para montagem do sistema foram utilizados alguns computadores, que atuaram como suplicantes ou servidores, e alguns modelos de switches que fazem o papel de autenticador. Abaixo pode ser observada a descrição desses equipamentos.

5.3.1 Computadores

Na implementação do sistema foram utilizados quatro computadores, sendo: um para o servidor Radius, outro para o LDAP e outros dois exerceram o papel de suplicante. Detalhes sobre os equipamentos são listados abaixo:

- Computador Susan
 - Pentium II, MMX, 350MHz
 - Memória – 128Mb de RAM
 - HD – 6 GB
 - SO – Microsoft Windows 2000 – Service Pack 4

- Servidor Radius
 - AMD Atlon XP 1900
 - Memória – 256Mb
 - HD – 40 GB
 - SO – Microsoft Windows 2000 Server – Service Pack 4

- Servidor LDAP
 - Pentium 3, 866MHz
 - Memória – 1Gb
 - HD – 40Gb
 - SO – Linux distribuição Redhat 9.0

- Notebook
 - Pentium 4 1.80GHz
 - Memória – 512MB
 - HD – 17Gb
 - SO – Windows XP Professional

5.3.2 Switches

Diversos tipos de switches foram analisados para serem testados no ambiente, porém como se trata de uma tecnologia relativamente nova, muitos equipamentos ainda não suportam este padrão. A descrição dos switches testados pode ser observada abaixo:

- Enterasys Matrix V-Series
 - Modelo: V2V124-24
 - Portas: 24 Ethernet
 - Versão de software: 2.3.2.1

- Enterasys Matrix E-Series
 - Modelo: 1H582-25
 - Portas: 24 Ethernet
2 Gbic
 - Versão de software: 03.02.11

- D-Link
 - Modelo: DES-3526
 - Portas: 24 Ethernet
2 Gbic
 - Versão de software: 2.00-B15

- Catalyst
 - Modelo: 3750 Series
 - Portas: 24 Ethernet
4 Gbic
 - Versão de software: IOS 12.2 20(SE)

5.4 Suplicante

O ambiente de teste contou com dois suplicantes. O primeiro deles foi a máquina chamada de computador Susan com sistema operacional Windows 2000, o segundo um *notebook* cujo sistema operacional é Windows XP. Estes sistemas foram selecionados para os testes devido a grande quantidade de usuários que os utiliza, e principalmente, pelo fato destes sistemas operacionais suportarem a

autenticação sem a necessidade da instalação de um software cliente 802.1x.

No caso de outros sistemas operacionais, deve ser instalado na máquina um cliente 802.1x. Existem diversas opções de *software*, sendo algumas ferramentas pagas e outros *softwares* livres.

5.4.1 MS Windows XP

Para utilizar autenticação 802.1x no Windows XP é necessário apenas configurá-la. A configuração é bastante simples, exigindo apenas habilitar o 802.1x e definir o tipo de EAP que será utilizado. Este sistema operacional permite o EAP-MD5, PEAP e EAP-TLS. A Figura 13 mostra a interface de configuração do 802.1x no Windows XP.

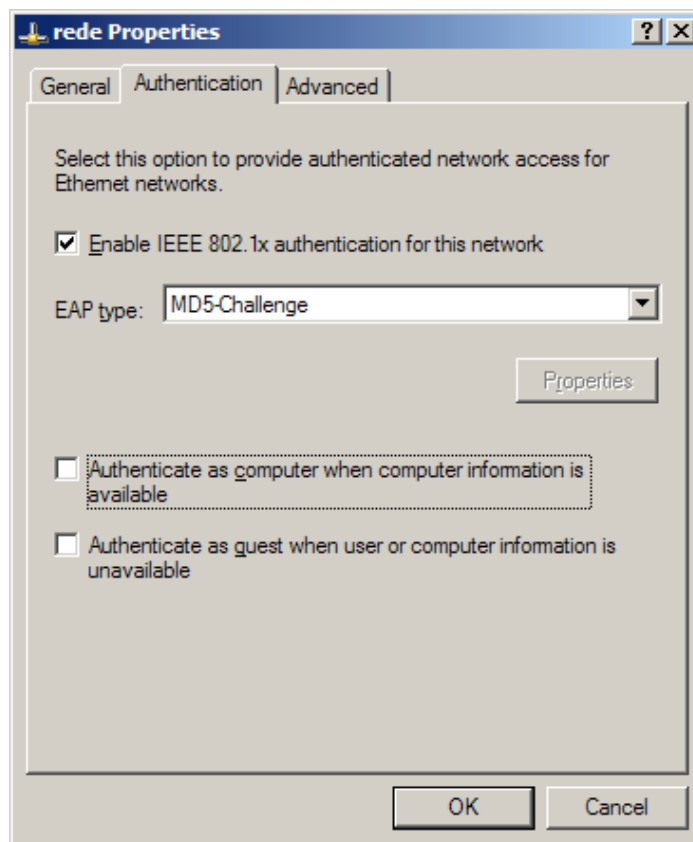


Figura 13 - Interface de configuração do 802.1x, Windows XP

5.4.2 MS Windows 2000

Segundo o Microsoft o Windows 2000 por padrão não suporta a autenticação 802.1x. Porém é possível obter no *site* da Microsoft uma atualização gratuita que possibilita a autenticação. A configuração é bastante similar a do XP, com opções para EAP-MD5, PEAP e EAP-TLS.

5.4.3 Outros clientes

Existem diversas opções de software de clientes 802.1x para diferentes tipos de sistemas operacionais. A maioria deles pode ser facilmente obtido na Internet, sendo algumas ferramentas gratuitas e outras pagas.

A Tabela 2 é resultado do cruzamento de informações obtidas em INTEROP (b) e HESSING. Nela estão relacionados alguns softwares de suplicantes aos respectivos sistemas operacionais que estes são compatíveis. Os campos marcados com um X indicam que há compatibilidade entre o cliente e o sistema operacional relacionado.

	BSD	LINUX	MAC	POCKET PC	WIN 95	WIN 98	WIN ME	WIN NT	WIN 2K	WIN XP
Alfa+Ariss SecureW2				X					X	X
Cisco		X	X		X	X		X	X	X
Free1X.ORG	X	X								
Funk Odyssey				X		X	X		X	X
Meetinghouse		X	X	X		X	X	X	X	X
Microsoft									X	X
Native			X	X						X
Open1x		X								
Weap									X	X
Wire1x						X	X		X	X

Tabela 2 - Relação entre sistemas operacionais e clientes 802.1x compatíveis.

5.5 Autenticador

Devido ao padrão 802.1x se tratar de uma tecnologia nova muitos equipamentos ainda não suportam esse tipo de autenticação, outros suportam a autenticação baseada em portas, mas não possuem todas as funcionalidades do padrão implementadas.

Dentre os autenticadores testados os que apresentaram suporte ao padrão foram:

- Enterasys Matrix V-Series
- Enterasys Matrix E-Series
- D-link DES3526
- Cisco Catalyst 3750

5.5.1 Configuração básica

Para habilitar o autenticador a autenticar seus usuários via 802.1x são necessárias algumas configurações:

- Cadastro de pelo menos 1 servidor Radius informando seu número IP e senha secreta utilizada para comunicação entre servidor e autenticador.
- Habilitar autenticação 802.1x no equipamento.

Com esses passos o autenticador já está apto a utilizar o 802.1x para autenticação, porém alguns equipamentos suportam ainda outras funcionalidades. Algumas funcionalidades mais importantes serão apresentadas na seção 5.5.2. Para mais informações sobre a configuração dos autenticadores consulte o ANEXO C, onde são mostrados detalhes do equipamento Enterasys Matrix E-Series.

5.5.2 Funcionalidades adicionais

Nesta seção serão apresentadas outras funcionalidades, além da autenticação do usuário, consideradas importantes e úteis para o funcionamento e administração de uma rede de computadores. As funcionalidades aqui listadas são suportadas por pelo menos um dos equipamentos testados, porém nem sempre os equipamentos com suporte a 802.1x apresentam todas as funcionalidades possíveis.

- **Mapeamento de Vlans** – Através do mapeamento de *vlans* é possível alocar uma *vlan* a uma porta do equipamento baseando-se nas configurações do usuário conectado nesta. Inicialmente a porta não é alocada a nenhuma *vlan*, após a autenticação do usuário o servidor Radius informa ao autenticador a *vlan* a qual pertence o usuário. Assim o equipamento aloca esta porta a *vlan* indicada e o usuário pode acessar a rede, em qualquer ponto de acesso, utilizando suas políticas de controle de acesso. Esta é uma funcionalidade de grande utilidade pois proporciona mobilidade ao usuário da rede, porém um grande número de equipamentos ainda não suporta esta funcionalidade. Para maiores detalhes sobre o funcionamento do mapeamento de *vlan* consulte o Anexo B.
- **Autenticação *Multihost*** – Esta funcionalidade permite a autenticação de mais de um usuário através de uma porta do autenticador. Isto é de grande utilidade para ambientes onde os usuários estão conectados a um equipamento que não suporta 802.1x, mas este está conectado a um autenticador. A Figura 14 ilustra um ambiente de autenticação 802.1x utilizando autenticação *multi-host*.

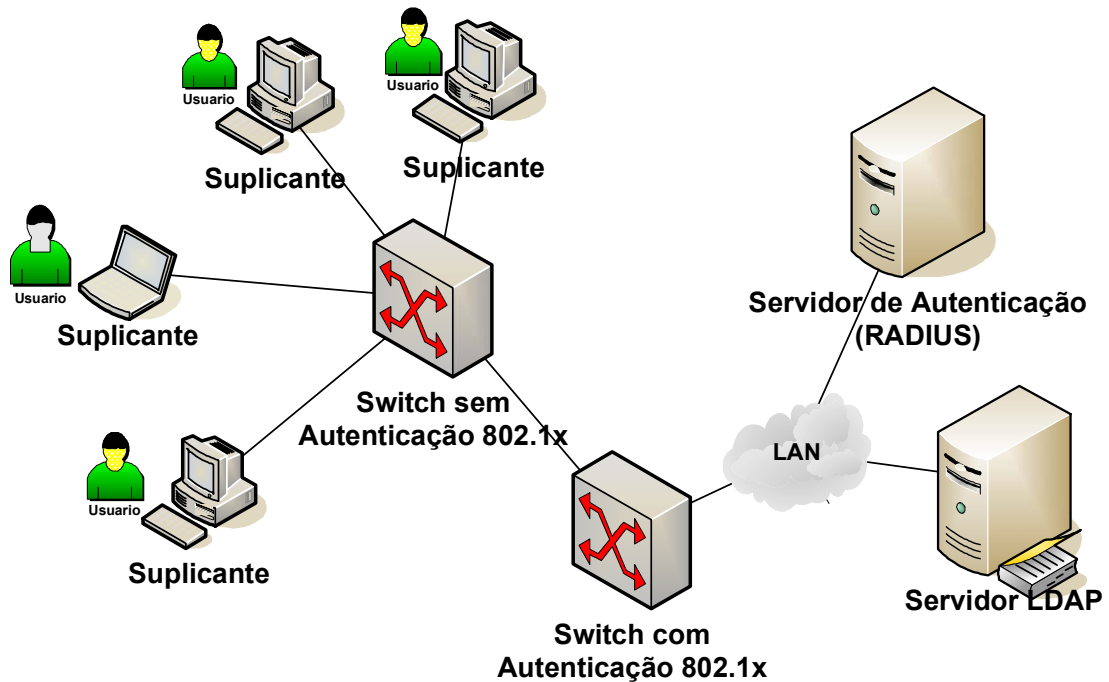


Figura 14 - Ambiente com autenticação *multi-host*

No modelo apresentado na Figura 14 o autenticador autentica os diversos usuários conectados ao switch sem autenticação, através da porta em que este está conectado. Para autenticar vários usuários através de uma mesma porta é necessário a criação de “portas virtuais” para cada equipamento que desejar acessar a rede. O autenticador reconhece o endereço de MAC de cada equipamento conectado e cria uma porta virtual para que este possa se autenticar e acessar a rede.

Nesta situação, caso o usuário não seja autorizado ele não poderá acessar a rede, mas ainda sim terá acesso aos outros equipamentos conectados ao switch sem autenticação. Ou seja, independente da autenticação ou não do usuário através da porta do autenticador, este terá acesso sem restrição ao segmento do switch a qual está conectado, pois este não exige a autenticação de seus usuários.

- **Port Control** – Permite marcar cada porta do equipamento como sendo autorizada, desautorizada ou auto. No caso da porta estar marcada como auto ela passará ao estado de autorizada ou não dependendo do resultado da autenticação.
- **Outras configurações** – Existem diversos parâmetros operacionais que podem ser configurados nos autenticadores, através deles é possível definir alguns valores como:
 - Tempo de espera para uma nova tentativa de autenticação no caso de falha;
 - Número máximo de pedidos de autenticação enviados;
 - Habilitar ou não o pedido de reautenticação de um usuário;
 - Período de espera entre pedidos de reautenticação;
 - Período de detecção de *timeout* do Servidor e Suplicante.
- **Estatísticas** – Os equipamentos possibilitam a visualização de dados sobre o processo de autenticação, estes podem ser gerais ou específicos de cada porta do autenticador. A visualização destes dados é de grande utilidade para a gerência da rede, alguns valores encontrados são:
 - Valores dos parâmetros operacionais (como os listado acima), para cada porta;
 - Estado de cada porta no processo de autenticação;
 - Usuário conectado em uma determinada porta;
 - Motivo de *logoff* do usuário;
 - Estatísticas sobre os pacotes EAPOL transmitidos.

5.6 Servidor de Autenticação

Seguindo a sugestão do padrão IEEE 802.1X foi utilizado um servidor Radius para a autenticação.

São necessárias algumas configurações no servidor para que este suporte a autenticação 802.1x. Os processos mais importantes são:

- Cadastro de Clientes – Devem ser cadastrados os autenticadores, switches por exemplo, informando o numero IP e a senha compartilhada entre switch e servidor.
- Cadastro de Usuários – Devem ser informados os dados do usuário ou a forma de obtenção desses dados.
- Formas de autenticação – É necessário a configuração das formas de autenticação permitidas, definindo o tipo de EAP utilizado.

Existem diversas formas de implementação do protocolo Radius, algumas soluções pagas e outras gratuitas. Pela maior simplicidade na utilização inicialmente foi utilizado um servidor para sistema Windows, posteriormente o estudo desenvolveu-se utilizando um software gratuito para Unix.

5.6.1 MS Radius

Devido à facilidade de instalação e configuração, primeiramente foi utilizado o *Internet Authentication Service* (IAS) como servidor Radius. O IAS é um componente do Windows 2000 Server. Sua vantagem é a interface amigável e a fácil utilização, porém apresenta a limitação de ser uma ferramenta paga.

Na Figura 15 pode-se observar a interface do IAS, mostrando a lista de clientes (autenticadores) cadastrados.

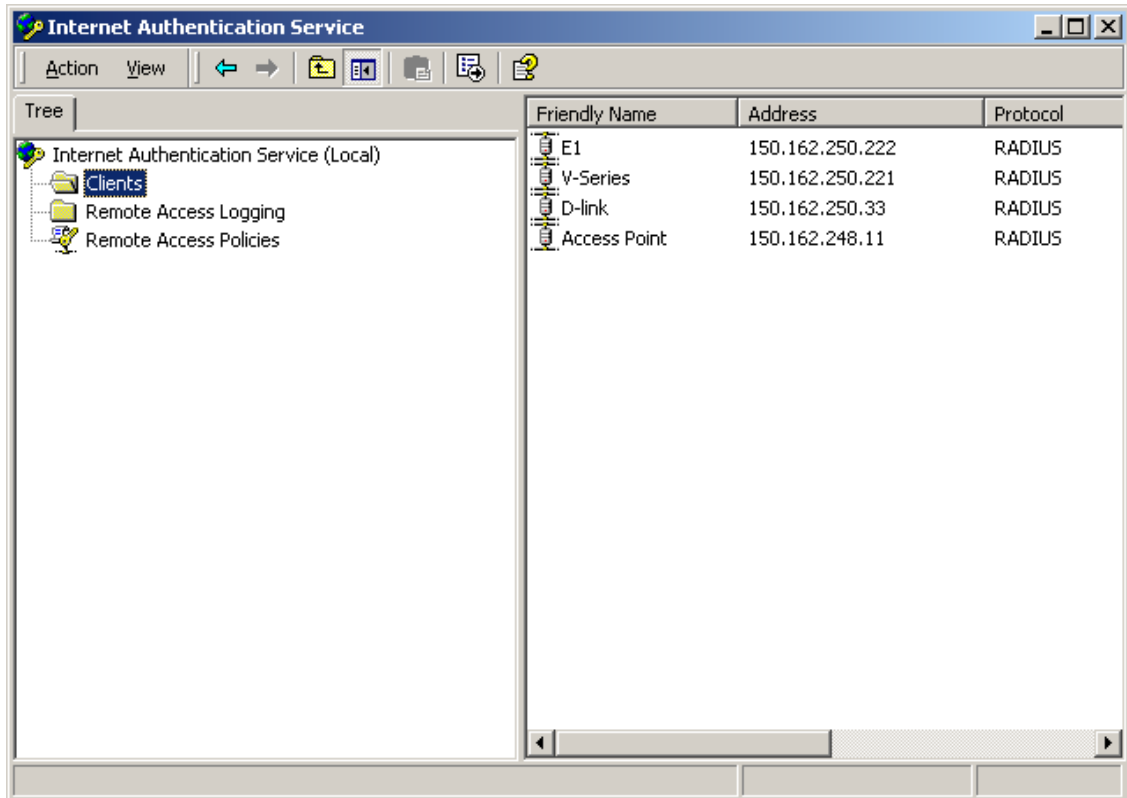


Figura 15 - Interface do IAS no Windows 2000 Server

Após o teste inicial ocorrido com sucesso utilizando um servidor em plataforma Windows, outro servidor foi então utilizado o FreeRADIUS.

5.6.2 FreeRADIUS

Segundo FreeRADIUS, o FreeRADIUS Server foi criado por um grupo conhecido como “the FreeRADIUS project”. É uma ferramenta para sistemas operacionais Unix que implementa um Servidor de protocolos Radius.

O FreeRADIUS foi escolhido pois é um produto gratuito, facilmente obtido na Internet e de código aberto. Por ter código livre o servidor possui muitos benefícios, apresentando ainda algumas ferramentas adicionais às fornecidas por outros servidores.

O servidor foi instalado em uma máquina com sistema operacional Linux – Kurumin 3.2 (Kernel 2.4.25-klg), sendo utilizada a versão 1.0.0 do FreeRADIUS.

São mostrados abaixo alguns parâmetros de configuração do FreeRADIUS, dados de configuração mais completos podem ser observados através do Anexo A.

- **Cadastro de clientes** – Mostra um exemplo do cadastro de um cliente no sistema.

```
o client 150.162.xxx.xxx/32 {
    secret = radius
    shortname = E1
}
```

- **Cadastro de usuários** – No ambiente implementado os dados dos usuários são armazenados em um servidor de diretórios LDAP. Para possibilitar esta integração deve ser informado, na configuração do servidor de autenticação, onde as informações dos usuários serão obtidas.

```
o ldap {
    server = "200.135.xxx.xxx"
    basedn = "ou=People,dc=popsc,dc=rnp,dc=br"
    filter = "(uid=%{Stripped-User-Name:-%{User-Name}})"
    access_attr = "dialupAccess"
}
```

- **Formas de autenticação** – O ambiente de teste utiliza o método EAP-MD5 como forma de autenticação, logo este método deve estar habilitado no servidor FreeRADIUS.

```
o eap {
    default_eap_type = md5
}
```

5.7 OpenLDAP

O Servidor de diretórios foi implementado utilizando-se um *software* gratuito o OpenLDAP, sua seleção foi feita baseando-se no fato de possuir código livre e de já estar sendo utilizado em outras aplicações do ambiente utilizado para testes. O servidor foi instalado na máquina chamada de Servidor LDAP e a versão de software utilizada foi 2.0.27.

O OpenLDAP é desenvolvido e gerenciado pelo OpenLDAP Project. Segundo OpenLDAP, este projeto é composto por pessoas voluntárias que se comunicam através da Web.

Para facilitar a visualização e gerenciamento dos usuários foi utilizado o LDAP Browser. Esta ferramenta permite a configuração de usuários no LDAP através de uma interface gráfica. A Figura 16 mostra a interface do LDAP Browser mostrando os atributos do usuário Susan, sendo alguns destes atributos específicos para a comunicação entre o servidor Radius e o LDAP. As classes e atributos utilizados na integração do OpenLDAP com o FreeRadius podem ser observados no Anexo D.

Demonstrando a possibilidade de integração das estruturas de diretórios entre diversas aplicações, este usuário foi criado utilizando-se um diretório já existente, criado para a utilização de outras aplicações, como o serviço de acesso discado da UFSC (RAS).

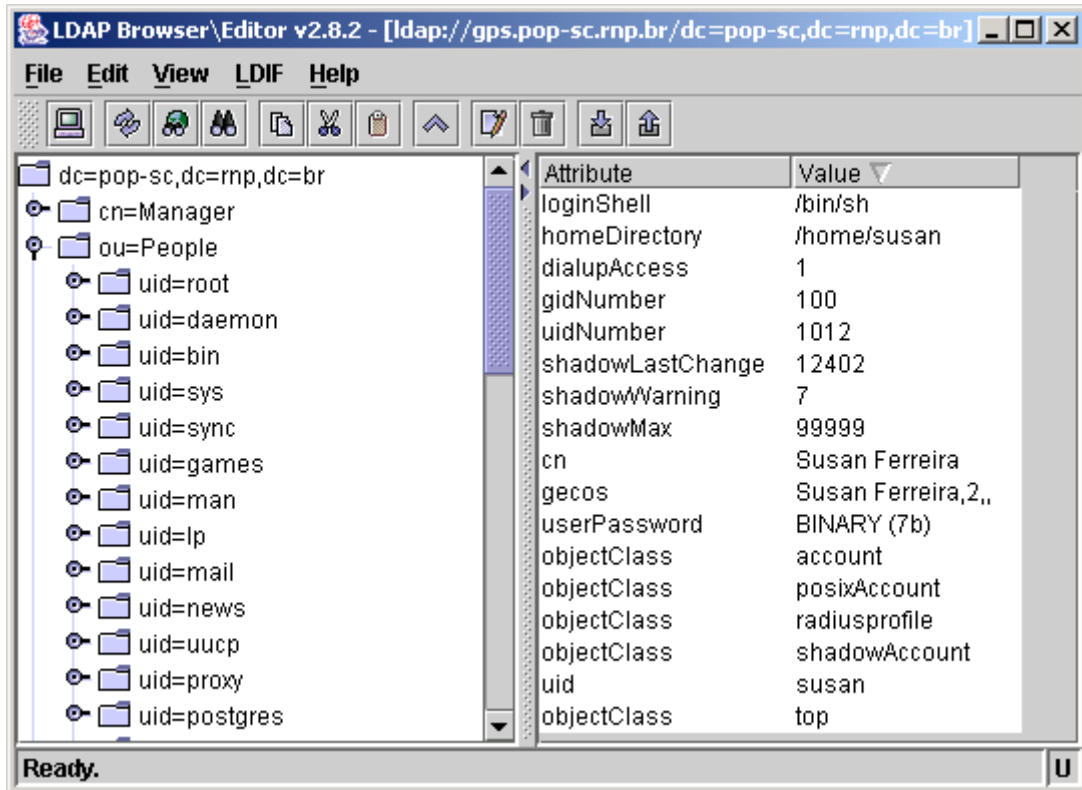


Figura 16 - Interface LDAP Browser

A Figura 17 representa um trecho da árvore do servidor de diretórios utilizado na implementação do ambiente, em destaque é mostrada a entrada correspondente a um usuário da rede listando alguns de seus atributos.

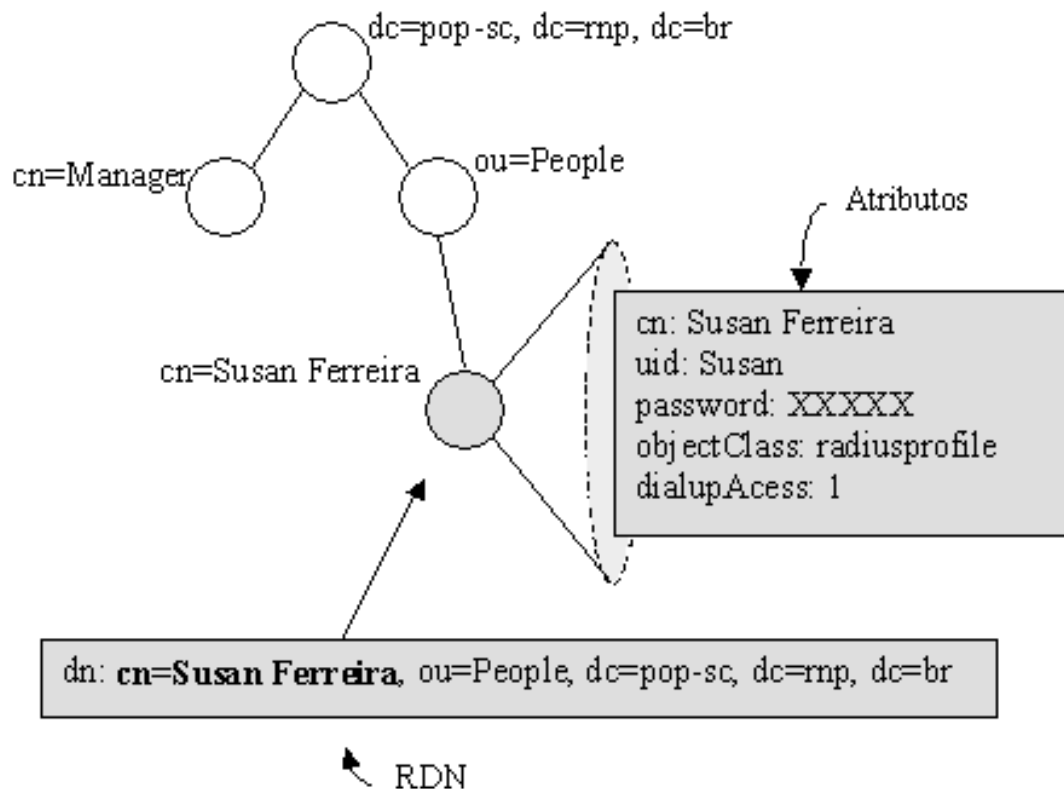


Figura 17 - Árvore do OpenLDAP utilizada na implementação do sistema.

5.8 Resultados obtidos

Analisando-se o ambiente implementado pode ser concluído que a autenticação baseada em portas definida pelo padrão 802.1x funcionou perfeitamente.

Uma grande deficiência, percebida nos testes, é a falta de equipamentos adaptados a suportar todas as suas funcionalidades oferecidas pelo padrão 802.1x. Porém como se trata de uma tecnologia nova, é esperado que a quantidade de equipamentos compatíveis ao padrão aumente rapidamente.

Apesar de não suportada pela maioria dos equipamentos, a mobilidade pode ser testada utilizando-se como autenticador o *switch* Catalyst 3750. Esta funcionalidade funcionou conforme o esperado, pois após a autenticação do usuário,

o autenticador associou corretamente a vlan a porta, de acordo com o perfil do usuário recebido do servidor de autenticação.

Outro ponto positivo percebido nos testes com o 802.1x foi à possibilidade de autenticação de mais de um usuário em uma mesma porta. Esta funcionalidade conhecida como *Multi-host* é possível através da criação de portas virtuais para cada equipamento que deseja se conectar a rede. Esta funcionalidade permite o uso pleno do padrão 802.1x sem a necessidade de descartar o parque de equipamentos existente e, portanto, permite preservar os investimentos.

Através do ambiente de teste, foi concebida uma sub-rede que possibilita maior segurança e mobilidade aos seus usuários. Logo os resultados do testes foram positivos, atestando que o padrão 802.1x atingiu os resultados esperados no trabalho.

6. Conclusões e Trabalhos Futuros

Este capítulo se dedica a conclusão dos estudos sobre o padrão 802.1x e sugerir tópicos relacionados a este que ainda devem ser explorados.

6.1 Conclusão

Após uma pesquisa teórica sobre o padrão 802.1x e algumas tecnologias relacionadas a ele, o padrão pôde ser analisado na prática através de um ambiente de teste. Com base nestes estudos e testes foi atingido o objetivo pretendido com o trabalho. O maior estímulo para a sua realização foi à intenção, de através deste, obter maior segurança e mobilidade para redes de computadores.

Pôde ser observado que o padrão realmente é capaz de adicionar maior segurança às redes de computadores. Esta segurança é obtida através da autenticação baseada em portas, permitindo apenas que usuários autorizados tenham acesso à rede. Diversos métodos de autenticação podem ser utilizados, podendo assim o administrador da rede decidir o nível de segurança desejado.

A mobilidade em uma rede de computadores pôde ser também obtida através deste estudo. Esta é uma funcionalidade de grande utilidade, possibilitando que um usuário utilize suas próprias políticas de acesso conectado em qualquer ponto da rede.

Portando os objetivos pretendidos com o estudo foram atingidos. Resultando em um material de grande conteúdo teórico e prático que possibilita a aplicação de uma nova técnica para obtenção de maior segurança e mobilidade em redes de computadores.

6.2 *Trabalhos Futuros*

Como se trata de uma tecnologia nova, existem muitos aspectos relacionados ao 802.1x que ainda devem ser explorados. Devido à limitação de tempo algumas tecnologias e ferramentas não foram severamente analisadas e testadas, neste estudo, abrindo assim a possibilidade de novos estudos com o intuito de aprofundar mais os conhecimentos sobre o padrão 802.1x. Seguem abaixo algumas sugestões para trabalhos futuros relacionados aos resultados obtidos neste trabalho:

- Aprofundamento teórico e realização de testes utilizando outros tipos de EAP além do EAP-MD5, como por exemplo o PEAP, EAP-TLS ou EAP-FAST;
- Testes com suplicantes em sistemas operacionais diferentes dos analisados neste estudo (Windows XP e 2000) e utilizando diferentes opções de *softwares* clientes 802.1x;
- Estudos sobre a MIB e formas de gerenciamento de sistemas com autenticação 802.1x;
- Exploração dos logs de autenticação para prover:
 - Contabilização de tempo conectado a redes por usuário/grupo;
 - Perfil de uso dos usuários;
 - Detecção de evidencias de acesso indevido com base no perfil, histórico;
- Verificação de segurança nos equipamentos portáteis antes que os mesmo ganhem acesso completo a rede;

- Criação de rede para visitantes com restrições diferentes;
- Montar um ambiente piloto para testar a tecnologia na prática, verificando seus eventuais problemas.

7. REFERÊNCIAS BIBLIOGRÁFICAS

ALLEN, Jon. WILSON, Jeff. **Securing a wireless network.** 30th annual ACM SIGUCCS conference on User services. Novembro de 2002.

CARTER, Gerald. **LDAP System Administration.** Primeira Edição. Editora O`Reilly Media, Inc. Março de 2003. 294p.

HESSING, Chris. PETRONI, Nick, BRYAN, Payne. SIMONS, Terry. **Open1x User's Guide.** Disponível em: <http://sourceforge.net/docman/display_doc.php?docid=24379&group_id=60236#ch1>. Acesso em 4 de Novembro de 2004.

IEEE Draft Standard, **“Standards for Local and Metropolitan Area Networks: Standard for Port Based Network Access Control”**, IEEE Draft P802.1X/D11. Junho de 2001.

INTEROP (a). LAN AccessSecurity Interoperability Lab. **What are EAP Authentication types?** Disponível em: <<http://www.opus1.com/www/whitepapers/8021x-eap-auth-types.pdf>>. Acesso em 4 de Novembro de 2004.

INTEROP (b). LAN AccessSecurity Interoperability Lab. **Cooking up 802.1X Successfully.** Disponível em: <<http://www.opus1.com/www/whitepapers/cookbookfor8021x.pdf>>. Acesso em 4 de Novembro de 2004.

ISO. International Organization for Standardization / International Electrotechnical Committee. “Information Processing Systems – Open Systems Interconnection – Basic Reference Model – Part 2: Security Architecture”. International Standard 7498-2, 1989.

FERNANDES, W. Débora. **Segurança na Internet?** 2002. Dissertação (Mestrado em Engenharia de Produção). Pós-Graduação em Engenharia de Produção. Universidade Federal de Santa Catarina – UFSC. Florianópolis – SC

FILHO, C. Helio. **Controle de Acesso para Gerência de Segurança em Redes Virtuais Emuladas.** 2000. Dissertação (Mestrado em Ciência da Computação). Pós-Graduação em Ciências da Computação. Universidade Federal de Santa Catarina – UFSC. Florianópolis – SC

FreeRADIUS. **FreeRADIUS Frequently Asked Questions & Answers.** Disponível em: <<http://www.freeradius.org/faq/>>. Acesso em 1 de Agosto de 2004.

GAST, Matthew. **A Technical Comparison of TTLS and PEAP.** O`REILLY Articles. Disponível em: <<http://www.oreillynet.com/pub/a/wireless/2002/10/17/peap.html>>. Acesso em 4 de Novembro de 2004.

GEORGE (a), C. Ou. **EAP-FAST: The LEAP and PEAP killer?** Disponível em: <<http://www.lanarchitect.net/Articles/Wireless/LEAP/>> acesso em 28 Outubro 2004.

GEORGE (b), C. Ou. **LEAP: A looming disaster in Enterprise Wireless LANs.** Disponível em: < <http://www.lanarchitect.net/Articles/Wireless/EAP-FAST/index.htm>> acesso em 28 de Outubro 2004.

KWAN, Philip. **White Paper: 802.1x Port Authentication With LDAP.** Foundry Networks. 2003. Disponível em: <[http://www.foundrynet.com/solutions/appNotes/PDFs/802.1xAuthenticationWithLDAP.P.pdf](http://www.foundrynet.com/solutions/appNotes/PDFs/802.1xAuthenticationWithLDAP.pdf) > acesso em 28 de maio 2004.

LDAP BROWSER. Disponível em: <<http://www.ldapadministrator.com/download/index.php?PHPSESSID=c0dfa62987a99959a534d89684100072>> acesso em 15 de novembro 2004.

MATOS, V. Alexandre. **Gerência de Segurança em Aplicações de Bancos de Dados na Web.** 1999. Dissertação (Mestrado em Ciência da Computação). Pós-Graduação em Ciências da Computação. Universidade Federal de Santa Catarina – UFSC. Florianópolis – SC.

MICROSOFT. **Using 802.1x Authentication on Computers Running Windows 2000.** Disponível em: <<http://support.microsoft.com/default.aspx?scid=kb;en-us;313664>>. Acesso em 4 de Outubro de 2004.

OBELHEIRO, R. Rafael. **Modelos de Segurança Baseados em Papéis para Sistemas de Larga Escala: A Proposta RBAC-JaCoWeb.** 2001. Dissertação (Mestrado em Engenharia Elétrica). Pós-Graduação em Engenharia Elétrica. Universidade Federal de Santa Catarina – UFSC. Florianópolis – SC

OpenLDAP. **OpenLDAP 2.2 Administrator's Guide.** Disponível em: <<http://www.openldap.org/doc/admin22/>>. Acesso em 2 de Agosto de 2004.

RAS. **Sistema de Acesso Remoto da redeUFSC.** Disponível em: < <http://www.ras.ufsc.br/>> acesso em 11 de novembro 2004.

RFC2138, C. Rigney. A. Rubens. W. Simpson. S. Willens. **Remote Authentication Dial In User Service (RADIUS).** Abril 1997.

RFC2251, M. Wahl. T. Howes. S. Kille. **Lightweight Directory Access Protocol (v3).** Dezembro 1997.

RFC2284, L. Blunk. J.Vollbrecht. **PPP Extensible Authentication Protocol (EAP).** Março 1998.

RFC2289, N. Haller. C. Metz. P. Nesser. M. Straw. **A One-Time Password System.** Fevereiro 1998.

RFC3580, P. Congdon. B. Aboba. A.Smith. G. Zorn. J. Roes. **IEEE 802.1X Remote Authentication Dial In User Service (RADIUS) Usage Guidelines.** Setembro 2003.

RFC3748, B. Aboba. L. Blunk. J.Vollbrecht. J.Carlson. H. Levkowetz. **Extensible Authentication Protocol (EAP)**. Junho 2004.

RHODEN, E. Guilherme. **Detecção de Intrusões em Backbones de Redes de Computadores Através da Análise de Comportamento com SNMP**. 2002. Dissertação (Mestrado em Ciência da Computação). Pós-Graduação em Ciências da Computação. Universidade Federal de Santa Catarina – UFSC. Florianópolis – SC

SILVA, M. Paulo. **Políticas de Segurança da Informação nas Organizações**. 2003. Dissertação (Mestrado em Ciência da Computação). Pós-Graduação em Ciências da Computação. Universidade Federal de Santa Catarina – UFSC. Florianópolis – SC

SOARES, F. G. Luiz. LEMOS, Guido. COLCHER, Sérgio. **Redes de computadores: da LANs, MANs e WANs às rede ATM**. Segunda Edição. Editora Campos. 1995. 705p.

STALLINGS, William. **Cryptography and Network Security: Principles and Practice**. Second Edition. Prentice Hall. 1998. 569p.

8. ANEXOS

ANEXO A – Arquivos de Configuração do FreeRADIUS

Neste anexo podem ser observados trechos de alguns arquivos de configuração do FreeRADIUS considerados relevantes para a implementação do sistema.

- **radiusd.conf – LDAP**

A texto a seguir corresponde ao arquivo radiusd.conf mostrando algumas configurações do FreeRADIUS, inclusive o trecho que possibilita a integração deste ao servidor LDAP.

```
prefix = /usr
exec_prefix = /usr
sysconfdir = /etc
localstatedir = /var
sbindir = ${exec_prefix}/sbin
logdir = /var/log/freeradius
raddbdir = /etc/freeradius
radacctdir = ${logdir}/radacct

confdir = ${raddbdir}
run_dir = ${localstatedir}/run/freeradius

log_file = ${logdir}/radius.log

libdir = /usr/lib/freeradius

pidfile = ${run_dir}/freeradius.pid

user = freerad
group = freerad

max_request_time = 30

delete_blocked_requests = no
```

```
cleanup_delay = 5

max_requests = 1024

bind_address = *

port = 0

hostname_lookups = no
allow_core_dumps = no

regular_expressions      = yes
extended_expressions     = yes

log_stripped_names = no

log_auth = no

log_auth_badpass = no
log_auth_goodpass = no

usercollide = no

lower_user = no
lower_pass = no

nospace_user = no
nospace_pass = no

checkrad = ${sbindir}/checkrad

security {
    max_attributes = 200
    reject_delay = 1
    status_server = no
}

proxy_requests = yes
$INCLUDE ${confdir}/proxy.conf

$INCLUDE ${confdir}/clients.conf

snmp = no
$INCLUDE ${confdir}/snmp.conf

thread pool {
    start_servers = 5
    max_servers = 32
    min_spare_servers = 3
    max_spare_servers = 10
    max_requests_per_server = 0
}

modules {
    pap {
        encryption_scheme = crypt
    }

    chap {
        authtype = CHAP
    }
}
```

```

pam {
    pam_auth = radiusd
}

unix {
    cache = no

    cache_reload = 600

    passwd = /etc/passwd
    shadow = /etc/shadow
    group = /etc/group

    radwtmp = ${logdir}/radwtmp
}

$INCLUDE ${confdir}/eap.conf

mschap {
    authtype = MS-CHAP
}

ldap {
    server = "xxx.xxx.xxx.xxx"
    basedn = "ou=People,dc=pop-sc,dc=rnp,dc=br"
    filter = "(uid=%{Stripped-User-Name:-%{User-Name}})"
    start_tls = no
    access_attr = "dialupAccess"
    dictionary_mapping = ${raddbdir}/ldap.attrmap
    ldap_connections_number = 5
    password_attribute = userPassword
    timeout = 4
    timelimit = 3
    net_timeout = 1
}

realm IPASS {
    format = prefix
    delimiter = "/"
    ignore_default = no
    ignore_null = no
}

realm suffix {
    format = suffix
    delimiter = "@"
    ignore_default = no
    ignore_null = no
}

realm realmpercent {
    format = suffix
    delimiter = "%"
    ignore_default = no
    ignore_null = no
}

realm ntdomain {

```

```

        format = prefix
        delimiter = "\\\"
        ignore_default = no
        ignore_null = no
    }

checkval {
    item-name = Calling-Station-Id

    check-name = Calling-Station-Id

    data-type = string
}

preprocess {
    huntgroups = ${confdir}/huntgroups
    hints = ${confdir}/hints

    with_ascend_hack = no
    ascend_channels_per_line = 23

    with_ntdomain_hack = no

    with_specialix_jetstream_hack = no

    with_cisco_vsa_hack = no
}

files {
    usersfile = ${confdir}/users
    acctusersfile = ${confdir}/acct_users
    compat = no
}

detail {
    detailfile = ${radacctdir}/%{Client-IP-Address}/detail-%Y%m%d
    detailperm = 0600
}

acct_unique {
    key = "User-Name, Acct-Session-Id, NAS-IP-Address, Client-IP-
Address, NAS-Port"
}

$INCLUDE ${confdir}/sql.conf

radutmp {
    filename = ${logdir}/radutmp

    username = %{User-Name}

    case_sensitive = yes

    check_with_nas = yes

    perm = 0600

    callerid = "yes"
}

```



```
}

radutmp sradutmp {
    filename = ${logdir}/sradutmp
    perm = 0644
    callerid = "no"
}

attr_filter {
    attrsfile = ${confdir}/attrs
}

counter daily {
    filename = ${raddbdir}/db.daily
    key = User-Name
    count-attribute = Acct-Session-Time
    reset = daily
    counter-name = Daily-Session-Time
    check-name = Max-Daily-Session
    allowed-servicetype = Framed-User
    cache-size = 5000
}

always fail {
    rcode = fail
}
always reject {
    rcode = reject
}
always ok {
    rcode = ok
    simulcount = 0
    mpp = no
}

expr {
}

digest {
}

exec {
    wait = yes
    input_pairs = request
}

exec echo {
    wait = yes

    program = "/bin/echo %{User-Name}"

    input_pairs = request

    output_pairs = reply
}
```

```
}

ippool main_pool {

    range-start = 192.168.1.1
    range-stop = 192.168.3.254

    netmask = 255.255.255.0

    cache-size = 800

    session-db = ${raddbdir}/db.ippool

    ip-index = ${raddbdir}/db.ipindex

    override = no

    maximum-timeout = 0
}

}

instantiate {
    exec

    expr
}

authorize {
    preprocess

    chap

    mschap

    eap

    ldap
}

authenticate {
    Auth-Type PAP {
        pap
    }

    Auth-Type CHAP {
        chap
    }

    Auth-Type MS-CHAP {
        mschap
    }

    unix
}
```

```

    eap
}

preacct {
    preprocess
    acct_unique
    suffix
    files
}

accounting {
    detail
    unix
    radutmp
}

session {
    radutmp
}

post-auth {
}

pre-proxy {
}

post-proxy {
    eap
}

```

- **eap.conf – EAPs.**

Arquivo de configuração dos métodos de autenticação utilizados.

```

#
# Whatever you do, do NOT set 'Auth-Type := EAP'. The server
# is smart enough to figure this out on its own. The most
# common side effect of setting 'Auth-Type := EAP' is that the
# users then cannot use ANY other authentication method.
#
# $Id: eap.conf,v 1.4 2004/04/15 18:34:41 aland Exp $
#
eap {
    # Invoke the default supported EAP type when
    # EAP-Identity response is received.
    #
    # The incoming EAP messages DO NOT specify which EAP

```

```
# type they will be using, so it MUST be set here.
#
# For now, only one default EAP type may be used at a time.
#
# If the EAP-Type attribute is set by another module,
# then that EAP type takes precedence over the
# default type configured here.
#
default_eap_type = md5

# A list is maintained to correlate EAP-Response
# packets with EAP-Request packets. After a
# configurable length of time, entries in the list
# expire, and are deleted.
#
timer_expire      = 60

# There are many EAP types, but the server has support
# for only a limited subset. If the server receives
# a request for an EAP type it does not support, then
# it normally rejects the request. By setting this
# configuration to "yes", you can tell the server to
# instead keep processing the request. Another module
# MUST then be configured to proxy the request to
# another RADIUS server which supports that EAP type.
#
# If another module is NOT configured to handle the
# request, then the request will still end up being
# rejected.
ignore_unknown_eap_types = no

# Cisco AP1230B firmware 12.2(13)JA1 has a bug. When given
# a User-Name attribute in an Access-Accept, it copies one
# more byte than it should.
#
# We can work around it by configurably adding an extra
# zero byte.
cisco_accounting_username_bug = no

# Supported EAP-types

#
# We do NOT recommend using EAP-MD5 authentication
# for wireless connections. It is insecure, and does
# not provide for dynamic WEP keys.
#
md5 {
}

# Cisco LEAP
#
# We do not recommend using LEAP in new deployments. See:
# http://www.securiteam.com/tools/5TP012ACKE.html
#
# Cisco LEAP uses the MS-CHAP algorithm (but not
# the MS-CHAP attributes) to perform it's authentication.
#
# As a result, LEAP *requires* access to the plain-text
# User-Password, or the NT-Password attributes.
# 'System' authentication is impossible with LEAP.
#
```

```

#leap {
#}

# Generic Token Card.
#
# Currently, this is only permitted inside of EAP-TTLS,
# or EAP-PEAP. The module "challenges" the user with
# text, and the response from the user is taken to be
# the User-Password.
#
# Proxying the tunneled EAP-GTC session is a bad idea,
# the users password will go over the wire in plain-text,
# for anyone to see.
#
#gtc {
# The default challenge, which many clients
# ignore..
#challenge = "Password: "

# The plain-text response which comes back
# is put into a User-Password attribute,
# and passed to another module for
# authentication. This allows the EAP-GTC
# response to be checked against plain-text,
# or crypt'd passwords.
#
# If you say "Local" instead of "PAP", then
# the module will look for a User-Password
# configured for the request, and do the
# authentication itself.
#
# auth_type = PAP
#}

## EAP-TLS
#
# To generate ctest certificates, run the script
#
# ../scripts/certs.sh
#
# The documents on http://www.freeradius.org/doc
# are old, but may be helpful.
#
# See also:
#
# http://www.dslreports.com/forum/remark,9286052~mode=flat
#
#tls {
private_key_password = whatever
private_key_file = ${raddbdir}/certs/cert-srv.pem

# If Private key & Certificate are located in
# the same file, then private_key_file &
# certificate_file must contain the same file
# name.
#
# certificate_file = ${raddbdir}/certs/cert-srv.pem

# Trusted Root CA list
#
# CA_file = ${raddbdir}/certs/demoCA/cacert.pem

#
# dh_file = ${raddbdir}/certs/dh

```

```

# random_file = ${raddbdir}/certs/random

#
# This can never exceed the size of a RADIUS
# packet (4096 bytes), and is preferably half
# that, to accomodate other attributes in
# RADIUS packet. On most APs the MAX packet
# length is configured between 1500 - 1600
# In these cases, fragment size should be
# 1024 or less.
#
# fragment_size = 1024

# include_length is a flag which is
# by default set to yes If set to
# yes, Total Length of the message is
# included in EVERY packet we send.
# If set to no, Total Length of the
# message is included ONLY in the
# First packet of a fragment series.
#
# include_length = yes

# Check the Certificate Revocation List
#
# 1) Copy CA certificates and CRLs to same directory.
# 2) Execute 'c_rehash <CA certs&CRLs Directory>'.
# 'c_rehash' is OpenSSL's command.
# 3) Add 'CA_path=<CA certs&CRLs directory>'
# to radiusd.conf's tls section.
# 4) uncomment the line below.
# 5) Restart radiusd
# check_crl = yes

#
# If check_cert_cn is set, the value will
# be xlat'ed and checked against the CN
# in the client certificate. If the values
# do not match, the certificate verification
# will fail rejecting the user.
#
# check_cert_cn = %{User-Name}
# }

# The TTLS module implements the EAP-TTLS protocol,
# which can be described as EAP inside of Diameter,
# inside of TLS, inside of EAP, inside of RADIUS...
#
# Surprisingly, it works quite well.
#
# The TTLS module needs the TLS module to be installed
# and configured, in order to use the TLS tunnel
# inside of the EAP packet. You will still need to
# configure the TLS module, even if you do not want
# to deploy EAP-TLS in your network. Users will not
# be able to request EAP-TLS, as it requires them to
# have a client certificate. EAP-TTLS does not
# require a client certificate.
#
#ttls {
# The tunneled EAP session needs a default

```

```

# EAP type which is separate from the one for
# the non-tunneled EAP module. Inside of the
# TLS tunnel, we recommend using EAP-MD5.
# If the request does not contain an EAP
# conversation, then this configuration entry
# is ignored.
# default_eap_type = md5

# The tunneled authentication request does
# not usually contain useful attributes
# like 'Calling-Station-Id', etc. These
# attributes are outside of the tunnel,
# and normally unavailable to the tunneled
# authentication request.
#
# By setting this configuration entry to
# 'yes', any attribute which NOT in the
# tunneled authentication request, but
# which IS available outside of the tunnel,
# is copied to the tunneled request.
#
# allowed values: {no, yes}
# copy_request_to_tunnel = no

# The reply attributes sent to the NAS are
# usually based on the name of the user
# 'outside' of the tunnel (usually
# 'anonymous'). If you want to send the
# reply attributes based on the user name
# inside of the tunnel, then set this
# configuration entry to 'yes', and the reply
# to the NAS will be taken from the reply to
# the tunneled request.
#
# allowed values: {no, yes}
# use_tunneled_reply = no

#}

#
# The tunneled EAP session needs a default EAP type
# which is separate from the one for the non-tunneled
# EAP module. Inside of the TLS/PEAP tunnel, we
# recommend using EAP-MS-CHAPv2.
#
# The PEAP module needs the TLS module to be installed
# and configured, in order to use the TLS tunnel
# inside of the EAP packet. You will still need to
# configure the TLS module, even if you do not want
# to deploy EAP-TLS in your network. Users will not
# be able to request EAP-TLS, as it requires them to
# have a client certificate. EAP-PEAP does not
# require a client certificate.
#
#peap {
# The tunneled EAP session needs a default
# EAP type which is separate from the one for
# the non-tunneled EAP module. Inside of the
# PEAP tunnel, we recommend using MS-CHAPv2,
# as that is the default type supported by
# Windows clients.

```

```

#     default_eap_type = mschapv2
#}

#
# This takes no configuration.
#
# Note that it is the EAP MS-CHAPv2 sub-module, not
# the main 'mschap' module.
#
# Note also that in order for this sub-module to work,
# the main 'mschap' module MUST ALSO be configured.
#
# This module is the *Microsoft* implementation of MS-CHAPv2
# in EAP. There is another (incompatible) implementation
# of MS-CHAPv2 in EAP by Cisco, which FreeRADIUS does not
# currently support.
#
mschapv2 {
}

```

- **clients.conf – Clientes**

No arquivo `clients.conf` são feitas definições dos clientes (autenticadores) do servidor de autenticação, o texto abaixo corresponde a um trecho deste arquivo mostrando alguns clientes cadastrados no sistema.

```

client 150.162.xxx.xxx/32 {
    secret          = radius
    shortname       = E1
}

client 150.162.xxx.xxx/32 {
    secret          = radius
    shortname       = VSeries
}

client 150.162.xxx.xxx/32 {
    secret          = radius
    shortname       = DLink
}

```


ANEXO B – Log do switch Cisco Catalyst 3750 utilizando mapeamento de vlans.

O texto abaixo mostra trechos do arquivo de log obtido no switch Cisco Catalyst 3750, mostrando diversas ocorrências de processos de autenticação na porta Gigabit Ethernet 1/0/6. Este equipamento possibilita o mapeamento de vlans as portas após a autenticação do usuário, no caso mostrado nos logs à porta Gigabit Ethernet 1/0/6 e alocada a vlan 1248.

```

1d: dot1x_auth Gi1/0/6: during state auth_initialize, got event 0(cfg_auto

1d: @@@ dot1x_auth Gi1/0/6: auth_initialize -> auth_disconnected
1d: dot1x-sm:Gi1/0/6:0000.0000.0000:auth_disconnected_enter_action called
1d: dot1x-sm:
t1x_update_port_status called with port_status = DOT1X_PORT_STATUS_UNAUTHORIZE

1d: %LINK-3-UPDOWN: Interface GigabitEthernet1/0/6, changed state to up
1d: dot1x-sm:Gi1/0/6:0000.0000.0000:dot1x_process_txWhen_expire called
1d: dot1x_auth Gi1/0/6: during state auth_connecting, got event 18(txWhen_
pire)
1d: @@@ dot1x_auth Gi1/0/6: auth_connecting -> auth_connecting
1d: dot1x-sm:Gi1/0/6:0000.0000.0000:auth_connecting_connecting_action called
1d: dot1x-ev:dot1x_post_message_to_auth_sm: Tx for req_id for supplicant 0000.
00.0000

1d: dot1x-packet:Transmitting EAP-Request-ID packet to port GigabitEthernet1/0

1d: dot1x-ev:dot1x_tx_eap: EAP Ptk
1d: dot1x-ev:EAP-code=REQUEST
1d: dot1x-ev:EAP Type= IDENTITY
1d: dot1x-ev:ID=1

1d: dot1x-sm:Gi1/0/6:0000.0000.0000:dot1x_process_txWhen_expire called
1d: dot1x_auth Gi1/0/6: during state auth_connecting, got event 18(txWhen_
pire)
1d: @@@ dot1x_auth Gi1/0/6: auth_connecting -> auth_connecting
1d: dot1x-sm:Gi1/0/6:0000.0000.0000:auth_connecting_connecting_action called
1d: dot1x-ev:dot1x_post_message_to_auth_sm: Tx for req_id for supplicant 0000.
00.0000

1d: dot1x-packet:Transmitting EAP-Request-ID packet to port GigabitEthernet1/0

1d: dot1x-ev:dot1x_tx_eap: EAP Ptk
1d: dot1x-ev:EAP-code=REQUEST
1d: dot1x-ev:EAP Type= IDENTITY

```

1d: dot1x-ev:ID=1

1d: dot1x-sm:Gi1/0/6:0000.0000.0000:dot1x_process_txWhen_expire called
 1d: dot1x_auth Gi1/0/6: during state auth_connecting, got event 18(txWhen_
 pire)
 1d: @@@ dot1x_auth Gi1/0/6: auth_connecting -> auth_connecting
 1d: dot1x-sm:Gi1/0/6:0000.0000.0000:auth_connecting_connecting_action called
 1d: dot1x-sm:dot1x_auth_connecting_action:0000.0000.0000 auth_count=4 exceeded
 ax auth count=3

1d: dot1x-ev:Default and only instance. evaluation for guest vlan move

1d: dot1x_auth Gi1/0/6: during state auth_connecting, got event 13(reAuthM
 _exceeded)
 1d: @@@ dot1x_auth Gi1/0/6: auth_connecting -> auth_disconnected
 1d: dot1x-sm:Gi1/0/6:0000.0000.0000:auth_connecting_exit alled
 1d: dot1x-sm:Gi1/0/6:0000.0000.0000:auth_disconnected_enter_action called
 1d: dot1x-packet:Received an EAPOL frame on interface GigabitEthernet1/0/6
 1d: dot1x-ev:Received pkt saddr =0000.3992.a4d6 , daddr = 0180.c200.0003,pa-e
 er-type = 34958
 1d: dot1x-ev:Couldn't find a supplicant block for mac 0000.3992.a4d6

1d: dot1x-ev:Found a supplicant block for mac 0000.0000.0000 2B92A08

1d: dot1x-ev:Found a supplicant block for mac 0000.3992.a4d6 28800C0

1d: dot1x-ev:Found a supplicant block for mac 0000.3992.a4d6 28800C0

1d: dot1x_auth Gi1/0/6: initial state auth_initialize has enter
 1d: dot1x-sm:Gi1/0/6:0000.3992.a4d6:auth_initialize_enter called
 1d: dot1x-ev:auth_initialize_enter:0000.3992.a4d6: Current ID=0

1d: dot1x_auth Gi1/0/6: during state auth_initialize, got event 0(cfg_auto

1d: dot1x-sm:Gi1/0/6:0000.3992.a4d6:dot1x_process_txWhen_expire called
 1d: dot1x_auth Gi1/0/6: during state auth_connecting, got event 18(txWhen_
 pire)
 1d: @@@ dot1x_auth Gi1/0/6: auth_connecting -> auth_connecting
 1d: dot1x-sm:Gi1/0/6:0000.3992.a4d6:auth_connecting_connecting_action called
 1d: dot1x-ev:dot1x_post_message_to_auth_sm: Tx for req_id for supplicant 0000.
 92.a4d6

1d: dot1x-packet:Transmitting EAP-Request-ID packet to port GigabitEthernet1/0

1d: dot1x-ev:dot1x_tx_eap: EAP Ptk
 1d: dot1x-ev:EAP-code=REQUEST
 1d: dot1x-ev:EAP Type= IDENTITY
 1d: dot1x-ev:ID=0

1d: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/0/6, chan
 d state to up

1d: dot1x-sm:Gi1/0/6:0000.0000.0000:dot1x_process_txWhen_expire called
 1d: dot1x_auth Gi1/0/6: during state auth_connecting, got event 18(txWhen_
 pire)
 1d: @@@ dot1x_auth Gi1/0/6: auth_connecting -> auth_connecting
 1d: dot1x-sm:Gi1/0/6:0000.0000.0000:auth_connecting_connecting_action called
 1d: dot1x-ev:dot1x_post_message_to_auth_sm: Skipping tx for req_id for default
 applicant

1d: dot1x-sm:Gi1/0/6:0000.0000.0000:dot1x_process_txWhen_expire called

```

1d: dot1x_auth Gi1/0/6: during state auth_connecting, got event 18(txWhen_
pire)
1d: @@@ dot1x_auth Gi1/0/6: auth_connecting -> auth_connecting
1d: dot1x-sm:Gi1/0/6:0000.0000.0000:auth_connecting_connecting_action called
1d: dot1x-sm:dot1x_auth_connecting_action:0000.0000.0000 auth_count=4 exceeded
ax auth count=3

```

```

1d: dot1x-registry:** dot1x_vp_statechange:
1d: dot1x-ev:vlan 1248 vp is removed on the interface GigabitEthernet1/0/6
1d: dot1x-ev:Now Processing: 1248 link DOWN for GigabitEthernet1/0/6, accss_v
n = 1248, oper_vlan = 1248
1d: dot1x-registry:dot1x_port_modechange invoked on interface GigabitEthernet1
/6
1d: dot1x-registry:dot1x_port_linkchange invoked on interface GigabitEthernet1
/6
1d: dot1x-err:calling pm_idb_set_port_access_oper_vlanid with vlan=1
1d: dot1x-ev:supp_info=28800C0 txWhen_timer=2880110 quietWhile_timer=28800D0re
thWhen_timer=28800F0 awhile_timer=2880130

```

```

1d: dot1x-ev:destroy supplicant block for 0000.3992.a4d6

```

```

1d: dot1x-ev:Found a supplicant block for mac 0000.3992.a4d6 28800C0

```

```

1d: dot1x-ev:supp_info=2B92A08 txWhen_timer=2B92A58 quietWhile_timer=2B92A18re
thWhen_timer=2B92A38 awhile_timer=2B92A78

```

```

1d: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/0/6, chan
d state to down

```

```

1d: %LINK-3-UPDOWN: Interface GigabitEthernet1/0/6, changed state to down
1d: dot1x-registry:dot1x_port_linkchange invoked on interface GigabitEthernet1
/6

```

```

1d: dot1x-registry:dot1x_port_linkcomingup invoked on interface GigabitEthe
rnet1/0/6

```

```

1d: dot1x-ev:dot1x_port_enable: set dot1x ask handler on interface GigabitEthe
net1/0/6

```

```

1d: dot1x_auth Gi1/0/6: initial state auth_initialize has enter

```

```

1d: dot1x-sm:Gi1/0/6:0000.0000.0000:auth_initialize_enter called

```

```

1d: dot1x-ev:auth_initialize_enter:0000.0000.0000: Current ID=0

```

```

1d: dot1x_auth Gi1/0/6: during state auth_initialize, got event 0(cfg_auto

```

```

1d: @@@ dot1x_auth Gi1/0/6: auth_initialize -> auth_disconnected

```

```

1d: dot1x-sm:Gi1/0/6:0000.0000.0000:auth_disconnected_enter_action called

```

```

1d: dot1x-sm:

```

```

t1x_update_port_status called with port_status = DOT1X_PORT_STATUS_UNAUTHORIZE

```

```

1d: %LINK-3-UPDOWN: Interface GigabitEthernet1/0/6, changed state to up

```

ANEXO C – Configuração e dados do Enterasys Matrix E-Series

São apresentadas abaixo, algumas informações da configuração do switch Enterasys Matrix E-Series, mostrando também o estado de algumas portas do equipamento.

- Autenticação 802.1x habilitada no equipamento, porém o parâmetro port-control de suas portas está configurado para sempre autorizado. Assim não é necessário que o usuário conectado a essas portas se autentique para poder acessar a rede.

```
Matrix>show dot1x
DOT1X is enabled.
```

```
Matrix>show dot1x auth-config
Port
-----
```

```
fe.0.1  Auth-Config:
Auth controlled port control: Forced Authorized
```

```
fe.0.2  Auth-Config:
Auth controlled port control: Forced Authorized
```

```
fe.0.3  Auth-Config:
Auth controlled port control: Forced Authorized
```

```
.....
```

- Configuração do servidor de autenticação primário, mostrando o seu número IP e a porta utilizada para a autenticação.

```
Matrix>show radius
RADIUS status:      Enabled

Server  Server
Index   IP        Auth-Port  Status
-----
1       150.162.xxx.xxx 1812      Primary

RADIUS last-resort-action  Status
-----
Local                       Accept
Matrix>show radius accounting
Accounting status:  Enabled
```

- Abaixo é mostrado os atributos da porta fe.0.16 do equipamento, esta se encontra habilitada a realizar a autenticação 802.1x. Como pode ser visto abaixo o usuário da rede ainda não foi autenticado, logo a porta se encontra no estado de não autorizada.

```
Matrix>show dot1x auth-config fe.0.16

fe.0.16  Auth-Config:
PAE state:          Initialize
Backend auth State: Idle
Admin controlled directions: Both
Oper controlled directions: Both
Auth controlled port status: Unauthorized
Auth controlled port control: Auto
Quiet period:      30
Tx period:         30
Supp Timeout:     30
Server Timeout:   30
Max req:          2
Reauth period:    3600
Reauth enabled:   False
Key tx enabled:   False
```

ANEXO D – Radius.schema

Abaixo é mostrado o arquivo radius.schema utilizado na implementação do servidor de diretórios OpenLDAP, neste são listados as classes de objeto e atributos necessários para a utilização dos serviços do Radius.

```
#####
##### custom radius attributes #####

objectIdentifier myOID 1.1
objectIdentifier mySNMP myOID:1
objectIdentifier myLDAP myOID:2
objectIdentifier myRadiusFlag myLDAP:1
objectIdentifier myObjectClass myLDAP:2

attributetype
( myRadiusFlag:1
  NAME 'radiusAscendRouteIP'
  DESC 'Ascend VSA Route IP'
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
)

attributetype
(myRadiusFlag:2
  NAME 'radiusAscendIdleLimit'
  DESC 'Ascend VSA Idle Limit'
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
)

attributetype
(myRadiusFlag:3
  NAME 'radiusAscendLinkCompression'
  DESC 'Ascend VSA Link Compression'
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
)

attributetype
(myRadiusFlag:4
  NAME 'radiusAscendAssignIPPool'
  DESC 'Ascend VSA AssignIPPool'
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
)
```

```

attributetype
  (myRadiusFlag:5
  NAME 'radiusAscendMetric'
  DESC 'Ascend VSA Metric'
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
  )

#####

attributetype
  ( 1.3.6.1.4.1.3317.4.3.1.1
  NAME 'radiusArapFeatures'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
  )

attributetype
  ( 1.3.6.1.4.1.3317.4.3.1.2
  NAME 'radiusArapSecurity'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
  )

attributetype
  ( 1.3.6.1.4.1.3317.4.3.1.3
  NAME 'radiusArapZoneAccess'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
  )

attributetype
  ( 1.3.6.1.4.1.3317.4.3.1.44
  NAME 'radiusAuthType'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
  )

attributetype
  ( 1.3.6.1.4.1.3317.4.3.1.4
  NAME 'radiusCallbackId'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
  )

attributetype
  ( 1.3.6.1.4.1.3317.4.3.1.5
  NAME 'radiusCallbackNumber'

```

```

DESC ''
EQUALITY caseIgnoreIA5Match
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
SINGLE-VALUE
)

attributetype
( 1.3.6.1.4.1.3317.4.3.1.6
  NAME 'radiusCalledStationId'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
)

attributetype
( 1.3.6.1.4.1.3317.4.3.1.7
  NAME 'radiusCallingStationId'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
)

attributetype
( 1.3.6.1.4.1.3317.4.3.1.8
  NAME 'radiusClass'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
)

attributetype
( 1.3.6.1.4.1.3317.4.3.1.45
  NAME 'radiusClientIPAddress'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
)

attributetype
( 1.3.6.1.4.1.3317.4.3.1.9
  NAME 'radiusFilterId'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
)

attributetype
( 1.3.6.1.4.1.3317.4.3.1.10
  NAME 'radiusFramedAppleTalkLink'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
)

attributetype
( 1.3.6.1.4.1.3317.4.3.1.11

```



```
    NAME 'radiusFramedAppleTalkNetwork'
    DESC ''
    EQUALITY caseIgnoreIA5Match
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
    SINGLE-VALUE
)

attributetype
( 1.3.6.1.4.1.3317.4.3.1.12
  NAME 'radiusFramedAppleTalkZone'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
)

attributetype
( 1.3.6.1.4.1.3317.4.3.1.13
  NAME 'radiusFramedCompression'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
)

attributetype
( 1.3.6.1.4.1.3317.4.3.1.14
  NAME 'radiusFramedIPAddress'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
)

attributetype
( 1.3.6.1.4.1.3317.4.3.1.15
  NAME 'radiusFramedIPNetmask'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
)

attributetype
( 1.3.6.1.4.1.3317.4.3.1.16
  NAME 'radiusFramedIPXNetwork'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
)

attributetype
( 1.3.6.1.4.1.3317.4.3.1.17
  NAME 'radiusFramedMTU'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
)
```

```
attributetype
( 1.3.6.1.4.1.3317.4.3.1.18
  NAME 'radiusFramedProtocol'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
)
```

```
attributetype
( 1.3.6.1.4.1.3317.4.3.1.19
  NAME 'radiusFramedRoute'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
)
```

```
attributetype
( 1.3.6.1.4.1.3317.4.3.1.20
  NAME 'radiusFramedRouting'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
)
```

```
attributetype
( 1.3.6.1.4.1.3317.4.3.1.46
  NAME 'radiusGroupName'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
)
```

```
attributetype
( 1.3.6.1.4.1.3317.4.3.1.47
  NAME 'radiusHint'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
)
```

```
attributetype
( 1.3.6.1.4.1.3317.4.3.1.48
  NAME 'radiusHuntgroupName'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
)
```

```
attributetype
( 1.3.6.1.4.1.3317.4.3.1.21
  NAME 'radiusIdleTimeout'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
)
```

```
attributetype
( 1.3.6.1.4.1.3317.4.3.1.22
  NAME 'radiusLoginIPHost'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
)

attributetype
( 1.3.6.1.4.1.3317.4.3.1.23
  NAME 'radiusLoginLATGroup'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
)

attributetype
( 1.3.6.1.4.1.3317.4.3.1.24
  NAME 'radiusLoginLATNode'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
)

attributetype
( 1.3.6.1.4.1.3317.4.3.1.25
  NAME 'radiusLoginLATPort'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
)

attributetype
( 1.3.6.1.4.1.3317.4.3.1.26
  NAME 'radiusLoginLATService'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
)

attributetype
( 1.3.6.1.4.1.3317.4.3.1.27
  NAME 'radiusLoginService'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
)

attributetype
( 1.3.6.1.4.1.3317.4.3.1.28
  NAME 'radiusLoginTCPPort'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
)
```

```
)  
  
attributetype  
  ( 1.3.6.1.4.1.3317.4.3.1.29  
    NAME 'radiusPasswordRetry'  
    DESC ''  
    EQUALITY caseIgnoreIA5Match  
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.26  
    SINGLE-VALUE  
  )  
  
attributetype  
  ( 1.3.6.1.4.1.3317.4.3.1.30  
    NAME 'radiusPortLimit'  
    DESC ''  
    EQUALITY caseIgnoreIA5Match  
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.26  
    SINGLE-VALUE  
  )  
  
attributetype  
  ( 1.3.6.1.4.1.3317.4.3.1.49  
    NAME 'radiusProfileDn'  
    DESC ''  
    EQUALITY distinguishedNameMatch  
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.12  
    SINGLE-VALUE  
  )  
  
attributetype  
  ( 1.3.6.1.4.1.3317.4.3.1.31  
    NAME 'radiusPrompt'  
    DESC ''  
    EQUALITY caseIgnoreIA5Match  
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.26  
    SINGLE-VALUE  
  )  
  
attributetype  
  ( 1.3.6.1.4.1.3317.4.3.1.50  
    NAME 'radiusProxyToRealm'  
    DESC ''  
    EQUALITY caseIgnoreIA5Match  
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.26  
    SINGLE-VALUE  
  )  
  
attributetype  
  ( 1.3.6.1.4.1.3317.4.3.1.51  
    NAME 'radiusReplicateToRealm'  
    DESC ''  
    EQUALITY caseIgnoreIA5Match  
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.26  
    SINGLE-VALUE  
  )  
  
attributetype  
  ( 1.3.6.1.4.1.3317.4.3.1.52  
    NAME 'radiusRealm'  
    DESC ''  
    EQUALITY caseIgnoreIA5Match
```

```
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
SINGLE-VALUE
)

attributetype
( 1.3.6.1.4.1.3317.4.3.1.32
  NAME 'radiusServiceType'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
)

attributetype
( 1.3.6.1.4.1.3317.4.3.1.33
  NAME 'radiusSessionTimeout'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
)

attributetype
( 1.3.6.1.4.1.3317.4.3.1.34
  NAME 'radiusTerminationAction'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
)

attributetype
( 1.3.6.1.4.1.3317.4.3.1.35
  NAME 'radiusTunnelAssignmentId'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
)

attributetype
( 1.3.6.1.4.1.3317.4.3.1.36
  NAME 'radiusTunnelMediumType'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
)

attributetype
( 1.3.6.1.4.1.3317.4.3.1.37
  NAME 'radiusTunnelPassword'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
)

attributetype
( 1.3.6.1.4.1.3317.4.3.1.38
  NAME 'radiusTunnelPreference'
  DESC ''
  EQUALITY caseIgnoreIA5Match
```

```

        SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
    )

attributetype
( 1.3.6.1.4.1.3317.4.3.1.39
  NAME 'radiusTunnelPrivateGroupId'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
)

attributetype
( 1.3.6.1.4.1.3317.4.3.1.40
  NAME 'radiusTunnelServerEndpoint'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
)

attributetype
( 1.3.6.1.4.1.3317.4.3.1.41
  NAME 'radiusTunnelType'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
)

attributetype
( 1.3.6.1.4.1.3317.4.3.1.42
  NAME 'radiusVSA'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
)

attributetype
( 1.3.6.1.4.1.3317.4.3.1.43
  NAME 'radiusTunnelClientEndpoint'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
)

#need to change asn1.id
attributetype
( 1.3.6.1.4.1.3317.4.3.1.53
  NAME 'radiusSimultaneousUse'
  DESC ''
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27
  SINGLE-VALUE
)

attributetype
( 1.3.6.1.4.1.3317.4.3.1.54
  NAME 'radiusLoginTime'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
)

```

```

attributetype
( 1.3.6.1.4.1.3317.4.3.1.55
  NAME 'radiusUserCategory'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
)

attributetype
( 1.3.6.1.4.1.3317.4.3.1.56
  NAME 'radiusStripUserName'
  DESC ''
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
  SINGLE-VALUE
)

attributetype
( 1.3.6.1.4.1.3317.4.3.1.57
  NAME 'dialupAccess'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
)

attributetype
( 1.3.6.1.4.1.3317.4.3.1.58
  NAME 'radiusExpiration'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
  SINGLE-VALUE
)

attributetype
( 1.3.6.1.4.1.3317.4.3.1.59
  NAME 'radiusCheckItem'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
)

attributetype
( 1.3.6.1.4.1.3317.4.3.1.60
  NAME 'radiusReplyItem'
  DESC ''
  EQUALITY caseIgnoreIA5Match
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26
)

objectclass
( 1.3.6.1.4.1.3317.4.3.2.1
  NAME 'radiusprofile'
  SUP top STRUCTURAL
  DESC ''
  MUST ( uid )
  MAY ( userPassword $

```

```

radiusArapFeatures $ radiusArapSecurity $ radiusArapZoneAccess
$
radiusAuthType $ radiusCallbackId $ radiusCallbackNumber $
radiusCalledStationId $ radiusCallingStationId $ radiusClass $
radiusClientIPAddress $ radiusFilterId $
radiusFramedAppleTalkLink $
radiusFramedAppleTalkNetwork $ radiusFramedAppleTalkZone $
radiusFramedCompression $ radiusFramedIPAddress $
radiusFramedIPNetmask $ radiusFramedIPXNetwork $
radiusFramedMTU $ radiusFramedProtocol $
radiusCheckItem $ radiusReplyItem $
radiusFramedRoute $ radiusFramedRouting $ radiusIdleTimeout $
radiusGroupName $ radiusHint $ radiusHuntgroupName $
radiusLoginIPHost $ radiusLoginLATGroup $ radiusLoginLATNode $
radiusLoginLATPort $ radiusLoginLATService $ radiusLoginService
$
radiusLoginTCPPort $ radiusLoginTime $ radiusPasswordRetry $
radiusPortLimit $ radiusPrompt $ radiusProxyToRealm $
radiusRealm $ radiusReplicateToRealm $ radiusServiceType $
radiusSessionTimeout $ radiusStripUserName $
radiusTerminationAction $ radiusTunnelAssignmentId $
radiusTunnelClientEndpoint $ radiusIdleTimeout $
radiusLoginIPHost $ radiusLoginLATGroup $ radiusLoginLATNode $
radiusLoginLATPort $ radiusLoginLATService $ radiusLoginService
$
radiusLoginTCPPort $ radiusPasswordRetry $ radiusPortLimit $
radiusPrompt $ radiusProfileDn $ radiusServiceType $
radiusSessionTimeout $ radiusSimultaneousUse $
radiusTerminationAction $ radiusTunnelAssignmentId $
radiusTunnelClientEndpoint $ radiusTunnelMediumType $
radiusTunnelPassword $ radiusTunnelPreference $
radiusTunnelPrivateGroupId $ radiusTunnelServerEndpoint $
radiusTunnelType $ radiusUserCategory $ radiusVSA $
radiusExpiration $ dialupAccess $
radiusAscendRouteIP $ radiusAscendIdleLimit $
radiusAscendLinkCompression $
radiusAscendAssignIPPool $ radiusAscendMetric )
)

```


ANEXO E – Artigo

AVALIAÇÃO EXPERIMENTAL DO PROTOCOLO 802.1X PARA PROVER MOBILIDADE E SEGURANÇA EM REDES DE COMPUTADORES

Susan Möller Ferreira

Resumo

Este trabalho tem o intuito de realizar um estudo sobre o padrão 802.1x, objetivando através deste, a obtenção de maior segurança e a possibilidade de mobilidade aos usuários da rede. Inicialmente é realizado um estudo aprofundado sobre o padrão 802.1x. A seguir é descrito o ambiente de testes implementado onde são apresentadas as tecnologias Radius e LDAP. Através do estudo teórico realizado e da implantação do ambiente de teste pôde ser obtido um ambiente que possibilite a mobilidade do usuário na rede e uma melhora na segurança e gerência de redes através da integração entre as tecnologias acima citadas.

Palavras-chave: 802.1x; Segurança em Redes de Computadores; Mobilidade; Radius; LDAP; EAP; EAPOL.

Abstract

This work has the intention to realize a study about the 802.1x standard, objectifying through this, the improvement of security and the mobility to networks users. Initially, it is carried through a deep study on the 802.1x standard. It is followed by the description of the environment of tests implementation and the presentation of the technologies Radius and LDAP. Through the studies and tests realized, it can be concluded that the integration of the technologies cited above makes possible the mobility of the user in the network and an improvement in the security and management of them.

Palavras-chave: 802.1x; Security of computers networks; Mobility; Radius; LDAP; EAP; EAPOL.

1. Introdução

Nos últimos anos as redes de computadores têm se desenvolvido rapidamente. Empresas, lojas, escolas e outras instituições cada vez mais utilizam redes locais em seus

departamentos. Como um grande fluxo de pessoas normalmente transita por essas instituições gerou-se uma preocupação maior com a segurança da rede.

Qualquer pessoa que tenha acesso a um computador da instituição tem livre acesso à rede local. Esta é uma deficiência que pode ser muito prejudicial, pois um usuário mal intencionado pode tentar acessar dados confidenciais, iniciar um ataque ou ainda contaminar com vírus existente em seu laptop todas as estações da rede. Assim criou-se uma necessidade de restringir o acesso à rede apenas a pessoas autorizadas.

Outra limitação observada é a falta de mobilidade ao usuário na rede. A grande vantagem da mobilidade é a possibilidade de um usuário obter o mesmo nível e perfil de acesso à rede independente do local e da estação onde irá efetuar o acesso. Por exemplo, um empregado pode acessar a rede através de um ponto fora de seu departamento utilizando suas próprias configurações. Estas configurações podem incluir seu endereço IP, políticas de *firewall*, permissão de acesso a máquinas restritas a sua rede interna, entre outras. Logo é uma grande necessidade das instituições que os funcionários possam utilizar a rede através de qualquer ponto e obter suas próprias políticas de controle de acesso.

Reconhecendo a necessidade de um novo mecanismo para autenticação a IEEE (*Institute of Electrical and Electronics Engineers*) aprovou o padrão 802.1x em junho de 2001, que tem como promessa aprimorar a segurança em redes de computadores.

Segundo o padrão IEEE 802.1X, o controle de acesso à rede baseado em portas provê autenticação e autorização de equipamentos conectados a porta de uma LAN com características de conexão ponto-a-ponto, prevenindo o acesso a essa porta em casos de falha dos processos de autenticação e autorização.

Vários métodos de autenticação já foram aplicados no controle de acesso à rede, sendo a maioria baseado na autenticação de dispositivos não autenticando seus usuários. Para assegurar que uma LAN está sendo utilizada apenas por usuários autorizados, o padrão IEEE 802.1x define um novo tipo de segurança de acesso, que requisita a todos os usuários uma prévia autenticação antes da disponibilização dos recursos e serviços da rede.

2. Padrão 802.1x

Conforme definido pelo padrão IEEE 802.1x para a implementação de um ambiente de autenticação 802.1x são necessários três componentes: suplicante, autenticador e servidor de autenticação. Estes são representados na Figura 1, onde é mostrada a integração entre alguns equipamentos formando um ambiente de autenticação.

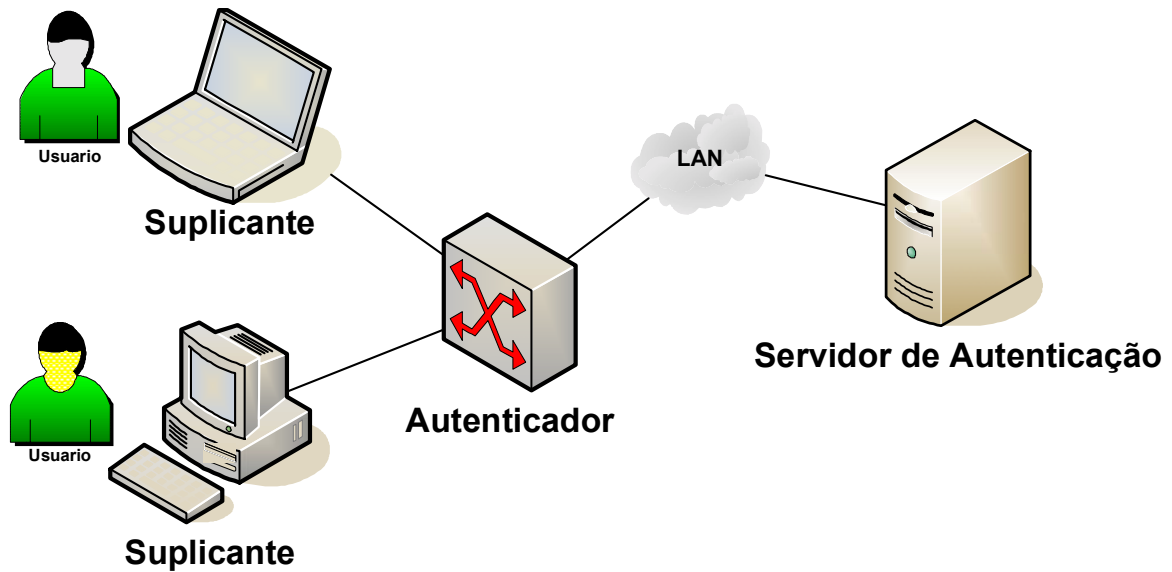


Figura 18 - Cenário do ambiente 802.1x

- **Suplicante** - Dispositivo de rede que necessita ser autenticado na rede.
- **Autenticador** – Equipamento que disponibiliza o acesso aos usuários da rede.
- **Servidor de autenticação** – Entidade que prove o serviço de autenticação.

A operação de controle de acesso baseado em portas cria dois diferentes pontos de acesso para conectar o sistema de autenticação a LAN. Um ponto de acesso, a porta não controlada, permite a troca de informações (protocolos) de autenticação entre o sistema e outros sistemas da LAN, independente da autorização. O outro ponto, a porta controlada, permite a troca de informações somente se o estado da porta for autorizado. O uso de portas controladas e não controladas pode ser observado na Figura 2.

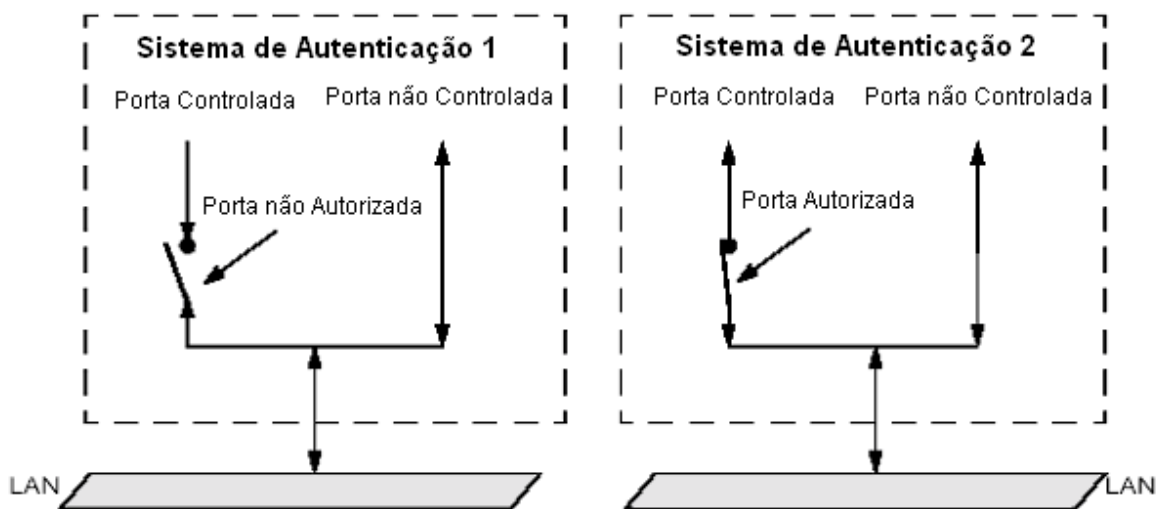


Figura 19 - Efeito do estado de autorização em portas controladas.
Fonte: IEEE 802.1x

A comunicação entre suplicante e autenticador é realizada através do protocolo EAPOL, já a comunicação entre autenticador e servidor de autenticação através do EAP. A seção seguinte apresenta as definições desses protocolos.

2.1 EAP e EAPOL

De acordo com as RFCs 2284 e 3748 o *Point-to-point Extensible Authentication Protocol* (EAP) é um protocolo geral, para autenticação ponto a ponto. O grande benefício do EAP é a sua flexibilidade, podendo suportar múltiplos mecanismos de autenticação. Outra vantagem é que os métodos de autenticação podem ser implementados em um servidor, assim o autenticador não precisa entender cada pacote que chega até ele, podendo apenas retransmiti-los. Isso evita que o autenticador tenha que ser atualizado para suportar cada novo método de autenticação.

EAP over LANs, ou EAPOL, é uma técnica de encapsulamento usada para transmitir pacotes EAP entre o suplicante e o autenticador.

2.2 Comunicação entre os componentes

É de responsabilidade do autenticador a retransmissão das mensagens entre o suplicante e o servidor de autenticação. Para realizar esta retransmissão é necessário que os pacotes EAP transmitidos entre o servidor de autenticação e o autenticador sejam encapsulados em pacotes EAPOL antes de serem transmitidos ao suplicante. Além de encapsular os pacotes EAP, o autenticador deve desencapsular os pacotes EAPOL enviados pelo suplicante, e em seguida retransmiti-los ao servidor de autenticação.

A Figura 93 ilustra a transmissão de pacotes entre o suplicante e o servidor de autenticação. Assim que o suplicante é conectado ao autenticador, o autenticador requisita sua identificação e passa a retransmitir os pacotes de autenticação entre o suplicante e o servidor de autenticação. Sendo os pacotes EAPOL representados através de linhas cheias e os pacotes EAP por linhas tracejadas. O protocolo OTP⁸ (*One Time Password authentication*) é utilizado de maneira ilustrativa, podendo ser substituído por outros protocolos de autenticação. Na situação mostrada na figura o usuário foi autenticado com sucesso. Logo, a porta em que este está conectado passa ao estado de autorizada e o usuário tem acesso à rede.

⁸ Conforme RFC 2289 OTP é um sistema de autenticação onde uma nova senha é gerada (sendo utilizada apenas uma vez) a cada processo de autenticação.

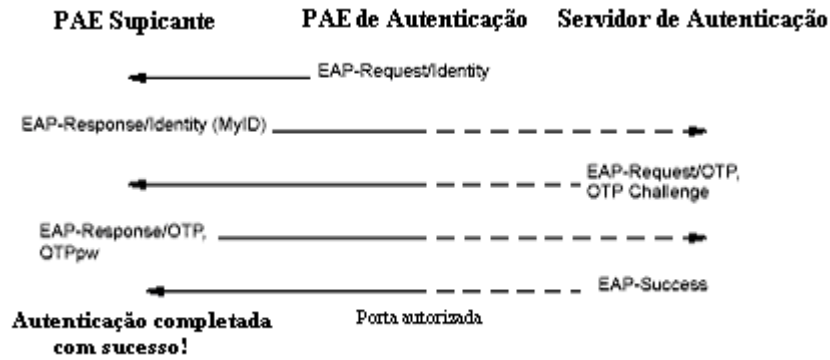


Figura 3 - Autenticação com sucesso.
Fonte: IEEE 802.1x

3. Ambiente de Testes

Para analisar experimentalmente o funcionamento do padrão 802.1x foi montado um ambiente de teste. Para montagem desse ambiente foram utilizados alguns computadores, comutadores ethernet e as tecnologias Radius e LDAP. A Figura 4 mostra como foram interligados estes equipamentos.

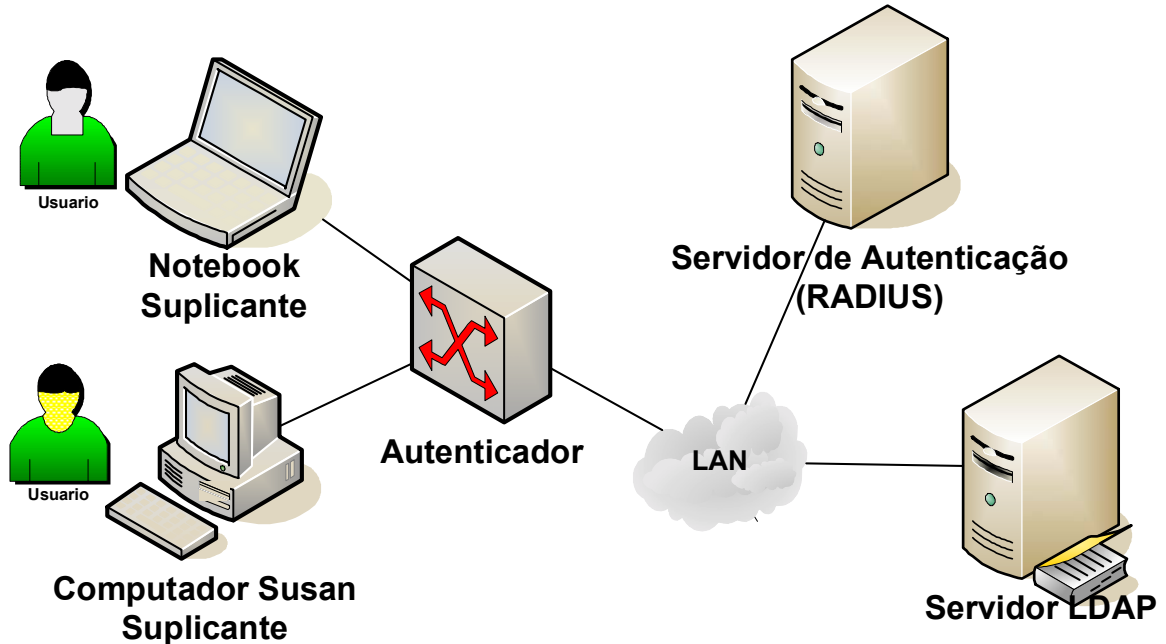


Figura 4 - Modelo do ambiente de teste.

3.1 Tecnologias Utilizadas

No ambiente de teste implementado está sendo utilizado um servidor Radius para a autenticação dos usuários. Como no processo de autenticação as operações de leitura são

mais frequentes que as de escrita o acesso à base de dados deve ser otimizado. Com esse propósito o Radius foi interligado a um servidor LDAP, possibilitando assim, uma leitura de dados rápida e eficiente além das vantagens de um serviço de diretório que pode ser integrado a outras aplicações da instituição. Como por exemplo, o serviço RAS e VPN existentes na UFSC.

3.1.1 Radius

Segundo a RFC 2138, o protocolo Radius foi desenvolvido com intenção de disponibilizar acesso à rede com autenticação, autorização e contabilização. Devido a sua simplicidade, eficiência e facilidade de implementação atualmente o Radius é suportado pela grande maioria dos equipamentos de rede, ou seja, por servidores VPN, *Access Points*, *switches*, roteadores e outros equipamentos de acesso à rede.

Ao receber uma requisição de conexão (oriunda de algum dispositivo de acesso) o servidor Radius autentica o usuário, verificando seu nome e senha. Após a autenticação, o servidor envia ao cliente todas as informações (configurações) necessárias para que o usuário tenha acesso à rede. Estas configurações podem ser, por exemplo, o tipo do serviço que deverá ser usado pelo cliente (PPP⁹, telnet¹⁰, ...).

3.1.2 LDAP

O *Lightweight Directory Access Protocol* (LDAP) é um protocolo de acesso a diretórios. Segundo a RFC 2251 ele surgiu a partir do serviço de diretórios X.500, um protocolo de implementação complexa e de alto custo. O LDAP foi inicialmente desenvolvido como um protocolo leve que serviria como ponte às requisições a um servidor X.500. A vantagem da utilização do LDAP é que este proporciona uma leitura rápida de dados e possibilita a utilização de um serviço de diretórios único compartilhado por diversas aplicações.

3.2 Funcionamento do Ambiente

Os suplicantes são conectados a um autenticador, neste caso um *switch*, com autenticação 802.1x habilitada em suas portas. Como o *autenticador* exige a autenticação de seus usuários, o suplicante só terá acesso à rede se após o término do processo de autenticação ele for considerado um usuário autorizado.

Ao perceber que há um equipamento conectado em uma de suas portas, o autenticador envia um pacote EAPOL requisitando que o suplicante se identifique. Começando assim o processo de autenticação.

O suplicante envia seu identificador, e outras informações que forem necessárias, ao autenticador através de pacotes EAPOL. Então o autenticador transforma estes pacotes EAPOL em pacotes EAP e encaminha-os ao servidor Radius.

O Radius ao receber a identificação do usuário, requisita outros dados necessários à autenticação, como uma senha por exemplo, ao autenticador. Já de posse das informações do

⁹ Point-to-Point Protocol - Protocolo de comunicação usado para suportar comunicação através de uma conexão entre dois pontos.

¹⁰ Protocolo de comunicação utilizado para abrir uma sessão (terminal) em uma máquina remota.

suplicante o Radius se conecta ao servidor LDAP, que busca as configurações deste usuário em seus diretórios. Caso este seja um usuário autorizado o Radius envia um pacote EAP ao autenticador, que então libera a porta para que o suplicante tenha acesso à rede.

A comunicação entre os equipamentos do sistema é realizada através do protocolo EAP. Um fator importante para a implementação do sistema é o tipo de EAP que será utilizado. Essa importância se deve ao fato do tipo de EAP determinar a forma de autenticação do usuário. Dentre os possíveis tipos de EAP o EAP-MD5 foi escolhido para a implementação do sistema. Este é um método de simples de implementação que autentica o usuário através de senha.

3.3 Resultados Obtidos

Analisando-se o ambiente implementado pode ser concluído que a autenticação baseada em portas definida pelo padrão 802.1x funcionou perfeitamente. Foi obtida uma maior segurança às redes de computadores através da autenticação previa dos usuários que requisitarem o acesso à rede. Este processo de autenticação ainda possibilitou a obtenção de dados sobre o perfil de acesso a rede do usuário, estes podem ser utilizados para aprimorar a administração e gerência da rede.

Uma grande deficiência, percebida nos testes, é a falta de equipamentos adaptados a suportar todas as suas funcionalidades oferecidas pelo padrão 802.1x. Porém como se trata de uma tecnologia nova, é esperado que a quantidade de equipamentos compatíveis ao padrão aumente rapidamente.

Através do mapeamento de vlans pôde ser obtida a mobilidade ao usuário na rede, esta é uma funcionalidade ainda não suportada por muitos autenticadores. No mapeamento de vlans após a autenticação do usuário o servidor Radius informa ao autenticador a *vlan* a qual este pertence. Assim o equipamento aloca esta porta a *vlan* indicada e o usuário pode acessar a rede, em qualquer ponto de acesso, utilizando suas próprias políticas de controle de acesso.

Outro ponto positivo percebido nos testes com o 802.1x foi à possibilidade de autenticação de mais de um usuário em uma mesma porta. Esta funcionalidade conhecida como *Multi-host* é possível através da criação de portas virtuais para cada equipamento que deseja se conectar a rede. Esta funcionalidade permite o uso pleno do padrão 802.1x sem a necessidade de descartar o parque de equipamentos existente e, portanto, permite a preservação de investimentos.

Através do ambiente de teste, foi concebida uma sub-rede que possibilita maior segurança e mobilidade aos seus usuários. Logo os resultados do testes foram positivos, atestando que o padrão 802.1x atingiu os resultados esperados no trabalho.

4. Conclusão

Após uma pesquisa teórica sobre o padrão 802.1x e algumas tecnologias relacionadas a ele, o padrão pôde ser analisado na prática através de um ambiente de teste. Com base nestes estudos e testes foi atingido o objetivo pretendido com o trabalho. O maior estímulo para a sua realização foi à intenção, de através deste, obter maior segurança e mobilidade para redes de computadores.

Pôde ser observado que o padrão realmente é capaz de adicionar maior segurança às redes de computadores. Esta segurança é obtida através da autenticação baseada em portas, permitindo apenas que usuários autorizados tenham acesso à rede. Diversos métodos de

autenticação podem ser utilizados, podendo assim o administrador da rede decidir o nível de segurança desejado.

A mobilidade em uma rede de computadores pôde ser também obtida através deste estudo. Esta é uma funcionalidade de grande utilidade, possibilitando que um usuário utilize suas próprias políticas de acesso conectado em qualquer ponto da rede.

Portando os objetivos pretendidos com o estudo foram atingidos. Resultando em um material de grande conteúdo teórico e prático que possibilita a aplicação de uma nova técnica para obtenção de maior segurança e mobilidade em redes de computadores.

5. Referências Bibliográficas

ALLEN, Jon. WILSON, Jeff. **Securing a wireless network**. 30th annual ACM SIGUCCS conference on User services. Novembro de 2002.

CARTER, Gerald. **LDAP System Administration**. Primeira Edição. Editora O'Reilly Media, Inc. Março de 2003. 294p.

HESSING, Chris. PETRONI, Nick, BRYAN, Payne. SIMONS, Terry. **Open1x User's Guide**. Disponível em: <http://sourceforge.net/docman/display_doc.php?docid=24379&group_id=60236#ch1>. Acesso em 4 de Novembro de 2004.

IEEE Draft Standard, “**Standards for Local and Metropolitan Area Networks: Standard for Port Based Network Access Control**”, IEEE Draft P802.1X/D11. Junho de 2001.

INTEROP (a). LAN AccessSecurity Interoperability Lab. **What are EAP Authentication types?** Disponível em: <<http://www.opus1.com/www/whitepapers/8021x-eap-auth-types.pdf>>. Acesso em 4 de Novembro de 2004.

INTEROP (b). LAN AccessSecurity Interoperability Lab. **Cooking up 802.1X Successfully**. Disponível em: <<http://www.opus1.com/www/whitepapers/cookbookfor8021x.pdf>>. Acesso em 4 de Novembro de 2004.

ISO. International Organization for Standardization / International Electrotechnical Committee. “Information Processing Systems – Open Systems Interconnection – Basic Reference Model – Part 2: Security Architecture”. International Standard 7498-2, 1989.

FERNANDES, W. Débora. **Segurança na Internet?** 2002. Dissertação (Mestrado em Engenharia de Produção). Pós-Graduação em Engenharia de Produção. Universidade Federal de Santa Catarina – UFSC. Florianópolis – SC

FILHO, C. Helio. **Controle de Acesso para Gerência de Segurança em Redes Virtuais Emuladas**. 2000. Dissertação (Mestrado em Ciência da Computação). Pós-Graduação em Ciências da Computação. Universidade Federal de Santa Catarina – UFSC. Florianópolis – SC

FreeRADIUS. **FreeRADIUS Frequently Asked Questions & Answers**. Disponível em: <<http://www.freeradius.org/faq/>>. Acesso em 1 de Agosto de 2004.

GAST, Matthew. **A Technical Comparison of TTLS and PEAP.** O'REILLY Articles. Disponível em: <<http://www.oreillynet.com/pub/a/wireless/2002/10/17/peap.html>>. Acesso em 4 de Novembro de 2004.

GEORGE (a), C. Ou. **EAP-FAST: The LEAP and PEAP killer?** Disponível em: <<http://www.lanarchitect.net/Articles/Wireless/LEAP/>> acesso em 28 Outubro 2004.

GEORGE (b), C. Ou. **LEAP: A looming disaster in Enterprise Wireless LANs.** Disponível em: < <http://www.lanarchitect.net/Articles/Wireless/EAP-FAST/index.htm>> acesso em 28 de Outubro 2004.

KWAN, Philip. **White Paper: 802.1x Port Authentication With LDAP.** Foundry Networks. 2003. Disponível em: <<http://www.foundrynet.com/solutions/appNotes/PDFs/802.1xAuthenticationWithLDAP.pdf>> acesso em 28 de maio 2004.

LDAP BROWSER. Disponível em: <<http://www.ldapadministrator.com/download/index.php?PHPSESSID=c0dfa62987a99959a534d89684100072>> acesso em 15 de novembro 2004.

MATOS, V. Alexandre. **Gerência de Segurança em Aplicações de Bancos de Dados na Web.** 1999. Dissertação (Mestrado em Ciência da Computação). Pós-Graduação em Ciências da Computação. Universidade Federal de Santa Catarina – UFSC. Florianópolis – SC.

MICROSOFT. **Using 802.1x Authentication on Computers Running Windows 2000.** Disponível em: <<http://support.microsoft.com/default.aspx?scid=kb;en-us;313664>>. Acesso em 4 de Outubro de 2004.

OBELHEIRO, R. Rafael. **Modelos de Segurança Baseados em Papéis para Sistemas de Larga Escala: A Proposta RBAC-JaCoWeb.** 2001. Dissertação (Mestrado em Engenharia Elétrica). Pós-Graduação em Engenharia Elétrica. Universidade Federal de Santa Catarina – UFSC. Florianópolis – SC

OpenLDAP. **OpenLDAP 2.2 Administrator's Guide.** Disponível em: <<http://www.openldap.org/doc/admin22/>>. Acesso em 2 de Agosto de 2004.

RAS. **Sistema de Acesso Remoto da redeUFSC.** Disponível em: < <http://www.ras.ufsc.br/>> acesso em 11 de novembro 2004.

RFC2138, C. Rigney. A. Rubens. W. Simpson. S. Willens. **Remote Authentication Dial In User Service (RADIUS).** Abril 1997.

RFC2251, M. Wahl. T. Howes. S. Kille. **Lightweight Directory Access Protocol (v3).** Dezembro 1997.

RFC2284, L. Blunk. J.Vollbrecht. **PPP Extensible Authentication Protocol (EAP).** Março 1998.

RFC2289, N. Haller. C. Metz. P. Nesser. M. Straw. **A One-Time Password System.** Fevereiro 1998.

RFC3580, P. Congdon. B. Aboba. A.Smith. G. Zorn. J. Roes. **IEEE 802.1X Remote Authentication Dial In User Service (RADIUS) Usage Guidelines**. Setembro 2003.

RFC3748, B. Aboba. L. Blunk. J.Vollbrecht. J.Carlson. H. Levkowetz. **Extensible Authentication Protocol (EAP)**. Junho 2004.

RHODEN, E. Guilherme. **Detecção de Intrusões em Backbones de Redes de Computadores Através da Análise de Comportamento com SNMP**. 2002. Dissertação (Mestrado em Ciência da Computação). Pós-Graduação em Ciências da Computação. Universidade Federal de Santa Catarina – UFSC. Florianópolis – SC

SILVA, M. Paulo. **Políticas de Segurança da Informação nas Organizações**. 2003. Dissertação (Mestrado em Ciência da Computação). Pós-Graduação em Ciências da Computação. Universidade Federal de Santa Catarina – UFSC. Florianópolis – SC

SOARES, F. G. Luiz. LEMOS, Guido. COLCHER, Sérgio. **Redes de computadores: da LANs, MANs e WANs às rede ATM**. Segunda Edição. Editora Campos. 1995. 705p.

STALLINGS, William. **Cryptography and Network Security: Principles and Practice**. Second Edition. Prentice Hall. 1998. 569p.