CENTRO UNIVERSITÁRIO DE BRASÍLIA – UNICEUB
FACULDADE DE CIÊNCIAS EXATAS E DE TECNOLOGIA
CURSO ENGENHARIA DE COMPUTAÇÃO

AUTENTICAÇÃO EM REDES WIRELESS COM CERTIFICAÇÃO DIGITAL EVITANDO "EVIL TWIN"

ALYSSON NISHIYAMA DE OLIVEIRA

RA: 9965560

Brasília – DF
2007

ALYSSON NISHIYAMA DE OLIVEIRA

AUTENTICAÇÃO EM REDES WIRELESS COM CERTIFICAÇÃO DIGITAL EVITANDO "EVIL TWIN"

Trabalho apresentado à Banca Examinadora da Faculdade de Ciências Exatas e de Tecnologia – UniCeub, para a conclusão do Curso Engenharia de Computação.

Orientador: Marco Antônio.

Brasília – DF
2007

ALYSSON NISHIYAMA DE OLIVEIRA

AUTENTICAÇÃO EM REDES WIRELESS COM CERTIFICAÇÃO DIGITAL EVITANDO "EVIL TWIN"

Trabalho apresentado à Banca Examinadora da Faculdade de Ciências Exatas e de Tecnologia – UniCeub, para a conclusão do Curso Engenharia de Computação.

Orientador: Marco Antônio.

Banca Examinadora

_____
Prof. Marco Antônio
Orientador

_____
Prof.
Examinador

_____
Prof.
Examinador

Brasília – DF
Novembro / 2007

DEDICATÓRIA

Ao professor Marco Antônio pela orientação firme e segura. Pela paciência e amizade e, sobretudo, pelo tempo de dedicação em ouvir, ponderar e conduzir.

AGRADECIMENTOS

À minha família e namorada, razões de toda a minha luta. Aos familiares, que mesmo distantes acreditaram na força de concretização do sonho que idealizamos juntos. Aos amigos que foram sem dúvida, a fonte de descontração e crescimento pessoal no decorrer de minha vida acadêmica.

Maravilhas nunca faltam no mundo.
O que falta é a capacidade de
sentí-las e admirá-las.

Johannes Peter Schimin

RESUMO

A tecnologia 802.11 para redes sem fio tem sido amplamente utilizada por instituições e empresas com a finalidade de economia em infra-estrutura de cabeamento, além de prover interligação, maior mobilidade e flexibilidade para redes locais. Em contrapartida, existem algumas preocupações adicionais em segurança que são inerentes a um meio de comunicação sem fio. Visando o aumento no conhecimento dos pesquisadores, analistas e técnicos, este trabalho analisa as vulnerabilidades e os ataques em redes sem fio conhecidos na atualidade, propondo uma possível solução viável para este problema. As redes sem fio – ou wireless – também conhecidas como IEEE 802.11x, Wi-Fi ou WLANs, são redes que utilizam sinais de rádio para a sua comunicação. O princípio de funcionamento das redes sem fio baseia-se na transmissão de dados através da camada atmosférica utilizando a propagação das ondas de rádio eletromagnéticas, entretanto o wireless engloba o uso de raios de luz infravermelha, apesar das ondas de rádio ser o meio mais difundido. A mobilidade é uma característica que atualmente tem se tornado um requisito fundamental na computação. Novas tecnologias de comunicação sem fio surgem a cada dia, e junto a isso cresce também a necessidade de segurança. Sem dúvida, as redes domésticas são as mais desprovidas de segurança e, portanto, as mais vulneráveis a ataques. O presente trabalho propõe uma solução que realiza autenticação mútua por meio de certificação digital em redes WLAN 802.11 no objetivo de reduzir os problemas de segurança associados a comunicação sem fio. Uma solução prática de simples implementação voltada para as redes domesticas visando garantir a autenticidade das redes para seus usuários.

Palavras-chave: Wireless, WLAN 802.11, Autenticação Mútua, Criptografia, Certificação Digital, Ataque, Controle de Acesso.

LISTA DE GRÁFICOS, TABELAS E FIGURAS

GRÁFICOS

TABELAS

FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

AP Access Point ARP Address Resolution Protocol

AES Advanced Encryption Standard

BSS Basic Service Set

CRC-32 Cyclic Redundancy Check

D.o.S Denail Of Service

DHCP Dynamic Host Configuration Protocol

EFS Encripted File Sistem

ESS Estended Service Set

EAP Extensible Authentication Protocol

EAP-LEAP Extensible Authentication Protocol - Lightweight Extensible Authentication Protocol

EAP-TLS Extensible Authentication Protocol - Transport Layer Security

EAP-TTLS Extensible Authentication Protocol - Tunneled Transport Layer Security

FTP File Transfer Protocol

GHz Gigahertz

IAS Internal Authentication Server ICP Infra-estrutura de chaves públicas

IEEE Institute of Eletrical and Eletronic Engineers

ICV Integrity Check Value

IPSec Internet Protocol Security

ISM Industrial, Scientific and Medical

LAN Local Area Network

MAC Media Access Control

Mbps Megabits per second

MIC Message Integrity Code

MSCHAPv2 Microsoft Challenge-Handshake Authentication Protocol v. 2

NetBEUI NetBIOS Extended User Interface

NetBIOS Network Basic Input/Output System

OSI Open Systems Interconection

PCI Peripheral Component Interconnect

PCMCIA Personal Computer Memory Card International Association

PEAP Protected Extensible Authentication Protocol

RADIUS Remote Authentication Dial-In User Server

RC4 Route Coloniale 4

SSID Service Set Identifier

STA Stations

TKIP Temporal Key Integrity Protocol

USB Universal Serial Bus

VPN Virtual Private Network

WPA Wi-Fi Protected Access

WPA-PSK Wi-Fi Protected Access - Pre-Shared Key

WPA-PSK.TKIP Wi-Fi Protected Access - Pre-Shared Key.Temporal Key Integrity Protocol

WEP Wired Encriptation Protocol

WLAN Wireless Local Area Network

WLL Wireless Local Loop

WMAN Wireless Metropolitan Area Network

WPAN Wireless Personal Area Network).

WWAN Wireless Wide Area Network

Wi-Fi Wireless-Fidelity

WiMAX Worldwide Interoperability for Microwave Access

SUMÁRIO

CAPÍTULO 1

INTRODUÇÃO

1.1 MOTIVAÇÃO

O avanço da tecnologia nos últimos anos tem permitido o rápido crescimento do seguimento das telecomunicações. A comodidade e mobilidade da computação móvel vêm se tornando um requisito na era em que vivemos, a da informação.

As redes sem fio têm criado novas práticas e novos usos do espaço urbano que vão, pouco a pouco, constituindo-se em lugares centrais da era da conexão. Várias cidades no mundo estão oferecendo este tipo de rede aos seus cidadãos constituindo uma verdadeira “cidade desplugada” (Townsend, 2003).

Várias empresas já oferecem serviços de conexão sem fio aos seus clientes em vários pontos, ou hotspots como são conhecidos, tais como aeroportos, hotéis, restaurantes, cafés e universidades.

O desafio no mundo das redes sem fio é a segurança, que é a principal motivação deste projeto. Aspectos de segurança, especialmente autenticação, controle de acesso e confidencialidade devem ser cuidadosamente estudados afim de se implementar uma estrutura wireless segura e confiável.

Se conectar à uma rede Wireless, sem a certeza de esta ser uma rede autêntica, é um enorme problema, pois um cliente móvel pode se conectar à uma rede falsa criada por um *hacker* pensando estar se conectando à uma rede legitima, e isso pode causar enormes conseqüências, tais como invasão de privacidade, roubo de informações confidenciais, roubo de senhas de banco, etc.

Para as redes corporativas, este problema pode ser contornado criando-se uma estrutura de autenticação ou integrando-se a uma já existente. Entretanto, para as redes domésticas, isso torna-se uma real preocupação devido ao fato da complexidade, e logo a inviabilidade, da solução.

1.2 OBJETIVOS

Este projeto tem como objetivo viabilizar um modelo prático para que usuários domésticos possam assegurar a autenticidade de suas redes Wireless (WLAN).

Desenvolvendo um Linux Live USB – um Kernel do Linux customizado compilado inteiramente da fonte com o mínimo de recursos necessários podendo ser inicializado pelo um pendrive – que juntamente com uma aplicação em Perl, seja capaz de prover autenticação mútua em redes Wireless através de certificação digital.

Com essa implementação, o cliente ao tentar se conectar à rede wireless e trocar os certificados digitais com o access point, poderá verificar a autenticidade da rede e solicitar autenticação ao access point, o qual repassará ao Servidor Linux Live USB, que este por sua vez autenticará o cliente. Então assim, realizando autenticação mútua, assegurará a autenticidade da identidade de ambos.

1.3 ESTRUTURA

Este trabalho está organizado em seis capítulos. Os primeiros três capítulos fazem a apresentação do tema do projeto fornecendo o embasamento teórico. Nos

capítulos seguintes são analisadas as abordagens e técnicas fundamentadas para a implementação do projeto. O último capítulo encerra com as conclusões e as sugestões de trabalhos futuros.

A seguir a organização detalhada do trabalho:

- Capítulo 1: O capítulo inicial introduz o tema do projeto, especificando a motivação, seus objetivos e a estrutura utilizada.
- Capítulo 2: Nesse capítulo serão abordados os conceitos das redes wireless, especificando as suas variadas tecnologias, e ainda será detalhada a tecnologia de rede sem fio 802.11, foco da implementação do projeto.
- Capítulo 3: Esse capítulo descreve detalhadamente os métodos de autenticação e protocolos de criptografia e integridade utilizados nas redes 802.11. Apresenta as várias vulnerabilidades existentes nas redes 802.11. Faz uma abordagem das técnicas para implementação de autenticação mútua em WLANs, detalhando o protocolo EAP-TLS, o serviço Radius e uma Infra-Estrutura de chaves publicas.
- Capítulo 4: Esse capítulo serão apresentadas as etapas a serem desenvolvidas no desenvolvimento do modelo proposto, demonstrando a utilização dos métodos e técnicas escolhidos na solução.
- Capitulo 5: Esse capítulo será apresentado o ambiente de implementação do projeto, bem como a implementação e a configuração do sistema proposto. Também apresentara após a execução do ambiente, os resultados dos testes de segurança e avaliação global do modelo, apresentando suas limitações e seus resultados mais relevantes.
- Capítulo 6: O capítulo final apresenta as conclusões do projeto e ainda as sugestões de propostas para trabalhos futuros.

CAPÍTULO 2

REDES WIRELESS

Com a crescente evolução da tecnologia, de uma forma geral, o mundo está se tornando cada vez mais móvel. A necessidade humana de se eliminar fios e ter mobilidade – tanto nas mais variadas formas de comunicação, quanto nas de acesso à informação – é hoje uma realidade que norteia o rumo da tecnologia da informação.

“Tecnologias de rede sem fio, no sentido mais simples, permitem um ou mais dispositivos se comunicarem sem qualquer tipo de conexão física - não requerem cabeamento” (KARYGIANNIS; OWENS; s.d., p. 16, tradução nossa). Igualmente a todos outros tipos de redes de comunicação, as redes wireless enviam e recebem dados através de um meio de transmissão. Neste caso, o meio é uma forma de radiação eletromagnética. Ondas de rádio são geradas de um transmissor RF e enviadas para um receptor em outra localidade.

Por permitirem a comunicação sem conexão física, a maior vantagem das redes *wireless* é sem dúvida a mobilidade. A habilidade de utilizar um dispositivo, e ser capaz de se locomover e permanecer conectado à uma rede, fazem dessa tecnologia uma real promessa para o futuro das comunicações.

A telefonia celular é a maior prova de sucesso das redes sem fio. A possibilidade de um usuário poder falar com outro independentemente de sua localização, alavancou as pesquisas e investimentos nessa nova área, que desde então, vêm sendo aperfeiçoada e melhorada.

Outro aspecto importante é a flexibilidade, a qual se traduz em um rápido crescimento da infra-estrutura de comunicação. Enquanto a escalabilidade das redes cabeadas é tradicionalmente complicada e de alto impacto - devido a necessidade de passar cabos, conectar *plugs* em dispositivos, e em muitas vezes de se mexer em tubulações, para conectar um usuário à rede – a infra-estrutura de uma rede *wireless* é qualitativamente a mesma, seja para conectar um ou vários usuários. Uma vez estabelecida a infra-estrutura, adicionar usuários à rede *wireless* é uma mera questão de autorização.

A conveniência permitida pelas tecnologias de rede sem fio, as fazem estar presentes em múltiplas formas, variando desde sistemas complexos, tais como WLANs e celular, até simples dispositivos *wireless* tais como fones de ouvido, microfone, teclados, *mouses*, etc.

Baseado nestas vantagens é fácil perceber que

> “essa tecnologia se encontrará em quase todas as coisas do dia a dia, em nossas casas e em qualquer lugar, para conectar desde diferentes dispositivos de entretenimento, até conectar a internet em locais públicos, além dos diferentes modos de interagir em nosso trabalho. No futuro, as redes móveis serão a regra, e as redes cabeadas serão usadas somente em casos especiais, ao invés da atual situação em que o contrario é verdadeiro” (SANKAR et al., 2006, p. 3, tradução nossa).

2.1 TECNOLOGIAS DE REDES SEM FIO

De uma forma geral, as existentes tecnologias de redes *wireless* se agrupam entre quatro topologias de redes sem fio. A WWAN, WMAN, WLAN e WPAN, compõem essas categorias de tecnologia que se diferenciam basicamente pela a área de cobertura e a taxa de transmissão.

2.1.1 WWAN

Wide Area Network (WAN), também conhecida como rede geograficamente distribuída, é uma rede de computadores de longa distância que abrange uma grande área geográfica. Uma WAN pode abranger desde um estado, região, pais ou mundo.

Uma *Wireless Wide Area Network* (WWAN) também abrange uma grande área geográfica, porém utilizando um meio de transmissão sem fio, ao invés de um meio com fio. WWAN's possuem tecnologias de telefonia celular tais como GPRS, CDMA, TDMA e GSM, para transmissão de dados. Vale ressaltar que, as taxas de transmissão e larguras de banda utilizando essas tecnologias são relativamente baixas, quando comparado com outras tecnologias *wireless*, tais como o padrão IEEE 802.11.

2.1.2 WMAN

Uma *Wireless Metropolitan Area Network* (WWAN) abrange uma área metropolitana, como uma cidade e seu entorno.

O padrão IEEE 802.16 foi criado para especificar uma tecnologia sem fio para redes metropolitanas. Foi atribuído a este padrão, o nome WiMAX.

> "Wi-Max, também chamado de Wi-MAX ou WiMAX, é um acrônimo para Worldwide Interoperability for Microwave Access (Interoperabilidade Mundial para Acesso por Microondas). Trata-se de uma tecnologia de banda larga sem-fio, capaz de atuar como alternativa a tecnologias como cabo e DSL na construção de redes comunitárias e provimento de acesso de última milha." (RNP, 2005).

Em parceria com universidades, instituições e governos, essa tecnologia já esta sendo testada desde 2004, nas cidades de Brasília, Ouro Preto, Mangaratiba e Belo Horizonte. No 2º trimestre de 2007, ela estará comercialmente disponível.

2.1.3 WPAN

Uma *Wireless Metropolitan Area Network* (WWAN) é uma rede wireless usada para a comunicação entre dispositivos que estão à uma pequena distância uns dos outros (COLEMAN; WESTCOTT; 2006, p. 196). Esta topologia opera dentro de um espaço pessoal geralmente para inter-comunicar dispositivos tais como: *laptops, PDA's*, celulares, fones de ouvido, etc. WPANs também são muitos utilizadas como portais de acesso à redes de níveis mais altos, tais como uma LAN ou Internet.

As tecnologias mais comuns deste tipo de topologia são o *Bluetooth*, o *ZigBee* e o Infravermelho. O Infravermelho utiliza a luz como meio de transmissão, ao passo que o *Bluetooth* e *ZigBee* utilizam a rádio freqüência. Tanto o *Bluetooth* quanto o *ZigBee* são baseados no padrão IEEE 802.15.

Com *bluetooth*, terminais moveis distanciados de até 10m podem estabelecer conexões de tráfego tanto síncrono, e.g. voz, quanto assíncrono, e.g. tráfego de dados IP. Neste tipo de rede, cada dispositivo pode se comunicar simultaneamente com até sete outros dispositivos. (CORDEIRO; SADOK; s.d., p. 1).

Já com o *ZigBee*, que foi desenvolvido como uma alternativa relativamente simples para soluções não complexas, e portanto uma tecnologia mais barata; sua comunicação pode chegar a velocidades de até 250 Kbps e a distâncias de até 50

metros em circunstâncias típicas em um ambiente ideal.

2.1.4 WLAN

Uma *Wireless Local Area Network* (WLAN) é uma rede sem fio que permite que usuários móveis se conectem à uma rede local (LAN). WLANs tipicamente fazem o uso de vários *access points* conectados à uma rede cabeada. São comumente utilizadas para prover, a usuários finais, acesso à recursos e serviços da rede, bem como acesso à um gateway para a Internet.

O padrão IEEE 802.11 especifica as tecnologias para *wireless LANs,* incluindo seus protocolos, *data frames*, várias camadas e freqüências.

O projeto a ser desenvolvido será implementado para as redes 802.11 e portanto, o texto agora irá se focar nesse tipo especifico de tecnologia *wireless*.

2.2 REDES 802.11

“O padrão IEEE 802.11 especifica as camadas física e de controle de acesso ao meio (MAC) para redes locais sem fio.” (FREITAG, 2004). No nível físico são tratadas apenas as transmissões com freqüência de rádio (RF) e infravermelho (IR), embora outras formas de transmissão sem fio possam ser usadas, como microondas e laser, por exemplo. No nível de enlace é definido um protocolo de controle de acesso ao meio (protocolo MAC) bastante semelhante ao protocolo usado em redes locais *Ethernet* (CSMA/CD).

O padrão IEEE 802.11 utiliza comunicação *half-duplex*, e especifica uma

arquitetura comum, métodos de transmissão, e outros aspectos de transferência de dados sem fio, permitindo a interoperabilidade entre os diversos produtos WLAN. Atualmente, existe alguns hardwares no mercado vêm utilizando topologias não padronizadas pelo IEEE 802.11 para atender necessidades especificas de redes *wireless*.

2.2.1 Arquitetura 802.11

As redes 802.11 consistem em 4 componentes físicos: estações, *access points*, sistema de distribuição e meio de transmissão.

2.2.1.1 Estações

Estações são dispositivos de computação com interfaces de rede *wireless*. Essas interfaces podem ser usadas em *desktops, laptops, PDAs*, telefones e outros, para se conectarem à uma rede wireless. As estações procuram por um espectro de freqüência disponível para conectividade, e se associam com um *access point* ou qualquer outro cliente *wireless*.

2.2.1.2 Access points

“Um access point é um dispositivo *half-duplex* com inteligência de um sofisticado *switch Ethernet*” (WIRELESS, 2002, p. 72, tradução nossa). Access points são dispositivos *half-duplex* devido ao fato da rádio freqüência utilizar comunicação *half-duplex*, isto é, permite apenas uma interface de rádio transmitir em

um determinado tempo. Tem a inteligência de um switch devido a sua habilidade de endereçar e direcionar diretamente o tráfego wireless.

Eles atuam como um ponto de acesso para os clientes à uma rede. Informam sua disponibilidade e, autenticam e associam os clientes moveis à rede wireless. Todo *Access Point* possui uma área de cobertura, a qual é determinada pela força do seu sinal e características de sua antena. Fatores como interferência (de outros dispositivos) e objetos (como paredes) afetam diretamente nesta área de cobertura.

2.2.1.3 Sistema de distribuição

A função de um sistema de distribuição é permitir que *access points* se comuniquem entre si, afim de rastrear os movimentos das estações moveis, e ainda, prover conectividade entre uma WLAN e outras redes, tais como uma LAN corporativa, um provedor de acesso, ou até mesmo a Internet.

2.2.1.4 Meio de Transmissão

O padrão 802.11 definiu várias camadas físicas como formas de transmissão sem fio. Entre elas, duas faixas de freqüência RF (2,5 Ghz e 5 Ghz) e uma camada física infravermelha, apesar das camadas RF serem até agora mais populares.

2.2.2 Topologias de Rede

O padrão 802.11 define 3 tipos de topologias, conhecidos como *service sets,*

que descrevem como essas *interfaces* de rádio devem ser usadas para se comunicarem umas com as outras.

2.2.2.1 BSS - *Basic Service Set*

Esse tipo de rede wireless se caracteriza por um grupo de estações que se comunicam entre si. Essa comunicação ocorre dentro de uma área, chamada de *basic service area* (BSA). No basic service set é possível haver dois tipos de configuração: a independente e a de infra-estrutura.

2.2.2.2 Redes Independente (*Ad hoc*)

Conhecidas como *independent* BSS (IBBS), as estações, nessa configuração, se comunicam diretamente entre si e por isso devem estar dispostas em uma área de comunicação direta (GAST; 2002; p. 24). Aqui não há nenhum terceiro elemento envolvido, apenas as estações.

O objetivo dessa configuração é permitir que um número pequeno de estações se conectem por um objetivo específico e por um período pequeno de tempo. Um ótimo exemplo seria a implementação de uma pequena LAN para a realização de partidas de jogos em rede.

Devido à pequena duração, pequeno tamanho, e objetivo especifico, IBBSs são também freqüentemente chamadas de redes *ad hoc*.

2.2.2.3 Redes de Infra-estrutura

Conhecidas como *infrastructure* BSS, as estações nessa configuração se comunicam diretamente com um *access point*. O *access point* é utilizado para toda comunicação realizada em uma rede de infra-estrutura. Em uma comunicação entre uma estação de uma *infrastructure* BBS e outra qualquer, toda a transmissão dar-se-á através de dois saltos. Os *frames* são transmitidos primeiramente para o *access point*, e então esse os transmitirá para a estação destino. Com isso, a área de cobertura de uma rede de infra-estrutura será definida pela área de cobertura do *access point.*

As redes de infra-estrutura recebem uma identificação, o *Basic Service Set Identifier* (BSSID). O padrão 802.11 definiu o BSSID como o endereço MAC da interface de rede wireless de um *access point*. Esse endereço identifica unicamente cada BSS.

2.2.2.4 ESS – *Extended Service Set*

Uma *extended service set* é composta por uma ou mais BSSs conectadas através de um sistema de distribuição (COLEMAN; WESTCOTT; 2006, p. 196). Cada BSS de uma WLAN, formada por um *access point* e clientes conectados a ele, se agrupam através do sistema de distribuição, e assim permitem que uma estação móvel associada a um *access point* em uma localidade possa se comunicar com outra estação móvel associada a um *access point* em uma outra localidade.

2.2.3 Serviços das Redes 802.11

A arquitetura do padrão IEEE 802.11 consiste em serviços essenciais implementados pelas estações, access points, e o sistema de distribuição. Os serviços implementados por access points e estações são conhecidos como *stations services* (SS), e os serviços implementados pelo sistema de distribuição são conhecidos como *distribution system services* (DSS). Ao todo são nove serviços, onde apenas três deles são usados para enviar dados, os seis restantes são serviços de gerenciamento os quais permitem que a rede wireless mantenha a localização das estações móveis e envie frames de forma correta.

2.2.3.1 *Station Services*

A. Autenticação – Devido a limitação da segurança física de uma rede WLAN em comparação com as redes cabeadas, rotinas adicionais de autenticação são definidas para garantir que somente usuários autorizados tenham acesso à rede. A autenticação é o primeiro dos dois passos necessários para estabelecer conexão à uma rede 802.11 (COLEMAN; WESTCOTT; 2006, p. 234). Uma estação para trafegar dados na rede deve primeiramente, nessa ordem, se autenticar e depois se associar.

B. Desautenticação – A desautenticação termina um relação de confiança previamente autenticada. Se uma estação deseja se desautenticar de um *access point,* ou este deseja desautenticar estações, ambos podem enviar um *frame* avisando o término da autenticação. É importante ressaltar que ela é uma

notificação, e portanto não pode ser recusada.

C. Privacidade – Como em uma rede wireless a transmissão é realizada via *broadcast*, todas as estações e dispositivos dentro da área de cobertura são aptas a escutar todas as mensagens trafegadas na rede. Isso obviamente prejudica o nível de segurança das redes sem fio, que comparada com as redes cabeadas não apresentam este mesmo tipo de problema. De forma a garantir um nível razoável de confidencialidade das informações, o IEEE 802.11 definiu protocolos de privacidade para a criptografia dos frames de dados. Inicialmente com o WEP, depois o WPA, e mais recentemente o WPA2 (GAST; 2002; p. 31). Esses protocolos de privacidade serão detalhados no capítulo 3.

D. Transmissão da MSDU – Este serviço prove entrega confiável de *frames* de dados de uma estação para outra estação Ele é similar àqueles providos por todas outras LANs do padrão IEEE 802.

2.2.3.2 *Distribution System Services*

A. Associação – A associação é o segundo passo necessário para estabelecer conexão à uma rede WLAN. Após a estação ser autenticada, ela precisa então se associar ao access *point*. Somente após a associação é que a estação se torna membro do BSS. O padrão define que cada estação pode associar-se com apenas um único *access point*, apesar do *access point* poder se associar com várias estações (SANKAR et al., 2006, p. 87).

B. Desassociação – A desassociação é um serviço que estações ou *access points* utilizam para terminar uma associação existente, ou seja, quando não desejam mais fazer parte de um BSS (GAST; 2002; p. 30). A desassociação por ser uma notificação e não uma requisição, não pode ser recusada, e para ela ocorrer, basta simplesmente que o dispositivo envie um *frame* de notificação. Uma estação sempre se desassocia ao desligar o sistema operacional, enquanto que o *access point*, por exemplo, pode desassociar todas as estações para a realização de uma eventual manutenção.

C. Reassociação – Esse serviço permite que uma estação se mova entre BSSs dentro de uma ESS. A estação deve emitir um *frame* de Reassociação para mover-se de um *access point* para outro que indique melhores condições de sinal (SANKAR et al., 2006, p. 88). A reassociação nunca deve ser iniciada pelo *access point*, mas somente por estações. Após o processo de reassociação ser completado, o sistema de distribuição atualiza seu registro de localização afim de que a estação móvel possa ser alcançada através de um *access point* diferente.

D. Distribuição – Este é o primeiro serviço usado por uma estação. O serviço de distribuição é utilizado em um ESS para distribuir mensagens através do sistema de distribuição. Após uma estação enviar uma mensagem e esta ao chegar ao *access point*, este utiliza o serviço de distribuição para entregar a mensagem ao seu destino. A forma como as mensagem são distribuídas dentro de um sistema de distribuição não é especificado pelo padrão 802.11, isso fica a cargo de cada implementação.

E. Integração – Permite que frames sejam entregues através de um portal entre o sistema de distribuição e uma rede diferente da 802.11 (GAST, 2002, p. 30). O serviço de integração converte *frames* 802.11 para *frames* de outro tipo de redes, e vice versa.

CAPÍTULO 3

SEGURANÇA EM WLANs

Segurança é um dos tópicos mais discutidos quando falamos em redes *wireless*. Devido ao crescimento do uso desse padrão em redes de computadores – pesquisas realizadas pela In-Stat, empresa de pesquisa de comunicações digitais, mostram que a utilizacão de dispositivos 802.11 em 2006 passaram de 40 milhões de unidades – a preocupação com a segurança se torna um fator crucial para a viabilidade e implementação dessas redes. As empresas devem atentar cuidadosamente para segurança *wireless* antes de poderem oferecer mobilidade para seus usuários.

Esse seguimento nas redes de telecomunicações foi sempre indagado pela lacuna causada quando o assunto é segurança física. Como as redes sem fio possuem meio físico de transmissão compartilhado – comunicação de natureza *broadcast* – todo tráfego que é transmitido ou recebido através da rede pode ser interceptado. Os sinais são trafegados no ar e assim ultrapassam as barreiras físicas de uma Organização. Sua ampla área de cobertura permite que qualquer cliente com uma interface wireless possa interceptar informações sem necessariamente estar no mesmo ambiente físico da rede. Esses fatores tornam a segurança física de redes sem fio consideravelmente inferior comparada com as redes cabeadas.

> As WLANs, devido a sua comunicação de natureza *broadcast*, necessitam de mecanismos adicionais tais como:
>
> - Autenticação para prevenir o acesso não autorizado aos recursos da rede.
> - Criptografia para proteger a integridade e privacidade das informações transmitidas" (CISCO SYSTEMS, 2002, p. 2, tradução nossa).

O objetivo desses mecanismos adicionais de segurança é fazer com que o tráfego *wireless* seja mais seguro, como nas redes cabeadas.

3.1 MÉTODOS DE AUTENTICAÇÃO

O padrão 802.11 especifica dois mecanismos para a autenticação de clientes *wireless*: autenticação aberta e autenticação de chave compartilhada. Outro mecanismo, como autenticação baseada em MAC, também é usualmente utilizado.

3.1.1 Autenticação Aberta

Autenticação aberta é o mais simples dos dois métodos de autenticação especificados pelo padrão 802.11. Esse tipo de autenticação não realiza nenhum tipo de verificação, o *access point* simplesmente garante qualquer requisição de autenticação solicitado por clientes.

A autenticação aberta ocorre com a troca de quatro *frames* entre o cliente e o *access point*:

A. O cliente envia um *frame* de autenticação para o *access point.*

B. O *access point* então responde enviando um ACK.

C. O *access point* envia um *frame* de autenticação para o cliente, confirmando a autenticação.

D. O cliente então responde enviando um ACK (COLEMAN; WESTCOTT, 2006, p. 235).

"A autenticação aberta é usada em diversos cenários, porém há duas

> razões principais para seu uso. Primeiro, ela é considerada o mais seguro dos dois métodos de autenticação disponíveis [...]. Segundo, ela é simples de configurar porque não requer nenhuma configuração. Todos dispositivos wireless 802.11 são configurados para utilizarem autenticação aberta por *default*" (PLANET3 WIRELESS, 2002, p. 175).

3.1.2 Autenticação de Chave Compartilhada

Autenticação de chave compartilhada é um método que envolve criptografia para a autenticação. Este método de autenticação é baseado em um protocolo de desafio-resposta, onde é necessário que uma chave WEP estática seja configurada na estação cliente e no *access point*.

O processo de autenticação utilizando chave compartilhada ocorre como segue:

1. O cliente envia uma solicitação de autenticação para o *access point*.
2. O *access point* responde com a resposta de autenticação contendo o texto de desafio.
3. O cliente utiliza sua chave WEP para criptografar o texto de desafio e retorna o resultado para o access point.
4. Se o *access point* conseguir descriptografar o resultado computado pelo cliente, ele responde com a resposta de autenticação que garante o acesso ao cliente. (CISCO SYSTEMS, 2002, p. 6).

Na autenticação de chave compartilhada, a chave WEP se faz necessária, e depois que o processo de autenticação é completado, todo tráfego que é transmitido é criptografado com a chave WEP estática. Na autenticação aberta, o protocolo

WEP não é utilizado para o processo de autenticação, entretanto o WEP pode ser ainda utilizado para a criptografia dos dados. O problema de segurança é que no método de chave compartilhada, durante a autenticação, o *access point* envia 128 octetos de texto para o cliente, a estação cliente então criptografa esse texto e o retorna para o *access point.* Esse cenário permite que um *hacker* utilizando um *sniffer* veja tanto o texto de desafio quanto o desafio criptografado. Com essas duas informações, o *hacker* pode utilizar um programa para descobrir a chave WEP. Uma vez que a chave WEP é compromissada, então todo o trafego poderá ser descriptografado pelo *hacker*. É por essa razão que a autenticação aberta é considerada mais segura que a autenticação de chave compartilhada.

3.1.3 Autenticação baseada em MAC

A autenticação baseada em MAC não é especificada no padrão 802.11, porém vários fabricantes provêm este tipo de autenticação em seus *access points*.

Toda interface de rede possui um endereço físico conhecido como endereço MAC. A autenticação baseada em MAC pode ser configurada para tanto permitir quanto negar o tráfego de um especifico endereço MAC. O *access point* verifica o endereço MAC do cliente em uma lista de endereços configurados localmente ou em um servidor de autenticação externo, e então autoriza ou não o seu acesso.

Como o endereço MAC pode ser forjado, e assim um *hacker* pode facilmente passar pela autenticação forjando um endereço MAC de quaisquer clientes permitidos, a autenticação baseada em MAC não é considerada segura para redes *wireless* corporativas (COLEMAN; WESTCOTT; 2006, p. 365).

3.2 MÉTODOS DE CRIPTOGRAFIA E INTEGRIDADE

As redes 802.11 operam em bandas de freqüências não-licenciadas e todas as transmissões de dados trafegam abertamente no ar. Como vimos anteriormente, é extremamente importante que as redes 802.11 possuam mecanismos de criptografia para proteger a confidencialidade e integridade das informações transmitidas.

A confidencialidade é fundamental em qualquer sistema de comunicação, sua função é garantir que uma pessoa não possa ter acesso a informações sem autorização. Vários métodos de criptografia são utilizados na tentativa de prevenir a exposição das informações trafegadas na rede.

A integridade tem papel importante na transmissão de informações pois garante que uma mensagem enviada chegue ate seu destinatário de forma correta, inalterada. Algoritmos de *checksum* (soma de controle) são utilizados para que o conteúdo da mensagem transmitida seja protegido e mantido inalterado ao longo da transmissão.

Nesta seção veremos alguns algoritmos de criptografia para privacidade e integridade de dados utilizados nas redes WLAN.

3.2.1 WEP *(Wired Equivalent Privacy)*

O WEP é um protocolo de criptografia especificado pelo padrão 802.11 para confidencialidade, controle de acesso e integridade dos dados. “O WEP é simétrico, uma vez que usa chaves compartilhadas e estas chaves devem ser as mesmas no

cliente e no ponto de acesso (JUNIOR et al., 2004, p. 41)". Ele permite ao administrador definir um conjunto de chaves de acesso a rede sem fio, onde é negado o acesso a quem não possuir a chave necessária.

"Conforme especifica o padrão, o WEP utiliza o algoritmo RC4 com chave de 40 ou 104 bits – também chamada de chave secreta compartilhada – que somado ao vetor inicial de 24 bits temos chaves de 64 e 128 bits (NETO, 2004, pg. 6)". A chave secreta compartilhada é utilizada para criptografar a transmissão de qualquer informação nas redes sem fio com o protocolo WEP.

A chave compartilhada após ser concatenada com o vetor de inicialização (chave IV) é inserida no algoritmo de criptografia RC4, que gera uma seqüência de números pseudo-randômica (PRNG). Não existe especificação para a geração da chave IV, e normalmente, ela é gerada seqüencialmente sendo reinicializada toda vez que a placa de rede é conectada na estação de trabalho.

Para garantir a integridade dos dados, é utilizado o algoritmo CRC-32 para calcular o *checksum* (soma de controle) do texto a ser enviado, e então os dois são concatenados. Em seguida faz-se a operação binária XOR (OU exclusivo) do texto+CRC-32 com o PRNG. Este resultado é concatenado com o vetor de inicialização IV (não criptografado) e enviado pelo emissor.

O receptor, ao receber os dados da transmissão, utiliza a chave IV transmitida e a chave compartilhada como entradas para o algoritmo RC4, obtendo o mesmo PRNG gerado para a transmissão. É feito então um XOR do texto+CRC-32 criptografado com o PRNG para descriptografar o texto cifrado. Por fim, aplica-se o CRC-32 e compara-o com aquele enviado para checar a integridade da mensagem recebida.

O protocolo WEP é considerado altamente vulnerável, conforme será detalhado na seção 3.3.4, e o seu uso deve ser desaconselhado.

3.2.2 WPA (*Wi-Fi Protected Access*)

O WPA, um padrão para interoperabilidade de segurança em redes WLAN, foi criado em 2003 por membros da Wi-Fi Alliance – associação de indústrias de mais de 200 empresas no ramo das redes sem fio – que estavam preocupados em aumentar o nível de segurança das redes sem fio e minimizar os problemas existentes no protocolo WEP, o padrão de segurança que existia até o momento.

Devido ao longo tempo de espera para o lançamento das especificações de segurança do IEEE 802.11i – que apesar do trabalho ter iniciado no início de 2001, o padrão foi ratificado somente em junho de 2004 – a Wi-Fi Alliance lançou o WPA adotando o máximo possível que podia das especificações do 802.11 disponíveis até o momento. Pretendia-se disponibilizar um padrão de segurança para o mercado das redes sem fio, porém mais imediato do que o comitê IEEE podia fornecer e com a preocupação de ter compatibilidade com o 802.11i, pois esse seria lançando futuramente (SANKAR et al., 2006, p. 225-226).

De acordo com Dennis Eaton, presidente da Wi-Fi Alliance, a finalização do IEEE 802.11i estava ainda para ser completado no ano seguinte, como um padrão completo ratificado, e que isso significava muito tempo. Segundo ele, era preciso fazer algo rapidamente (WI-FI PLANET, 2002).

O WPA trouxe para as redes WLAN as seguintes vantagens:

- **Autenticação mútua** - Implementação do padrão de autenticação 802.1X EAP para permitir autenticação mútua.

- **Criptografia mais forte** - Inclusão do protocolo TKIP (*Temporal Key Integrity Protocol)* algoritmo RC4 que o WEP utiliza para reforçar a confidencialidade da mensagens.

- **Integridade mais confiável** - Utilizacão do algoritmo *Michel Message Integrity Check* para garantir a integridade das mensagens.

3.3 VULNERABILIDADES E ATAQUES

As redes sem fio são particularmente vulneráveis a ataques devido a grande dificuldade de prevenir o acesso físico a elas. Qualquer manipulação na rede que diretamente ou potencialmente comprometa a confidencialidade, integridade e disponibilidade será considerada uma ameaça. Este capítulo abordará a ampla variedade de vulnerabilidades e ataques existentes nas redes sem fio 802.11.

É importante ressaltar que alguns desses ataques podem ser minimizados através de soluções existentes. Entretanto, outros não podem ser prevenidos mas apenas detectados (COLEMAN; WESTCOTT; 2006, p. 388).

3.3.1 Taxonomia dos Ataques em Redes

A figura abaixo mostra a taxonomia dos ataques em redes de computadores, como para qualquer tipo de rede, as redes sem fio 802.11 seguem o mesmo modelo.

Conforme a figura abaixo mostra, os ataques são tipicamente divididos em passivo e ativo. Essas duas principais classes são divididas em outros tipos de ataques que são definidos nos sub-tópicos a seguir.

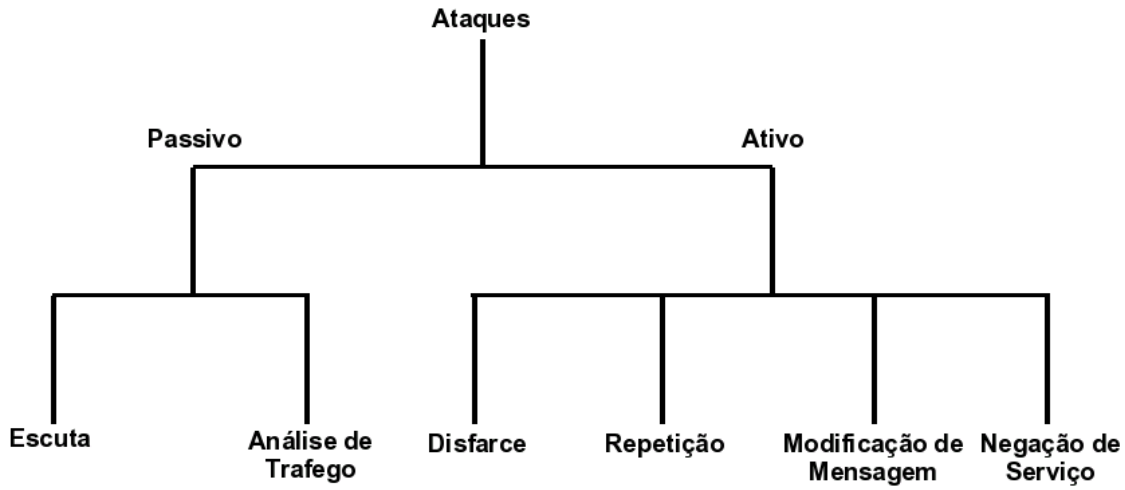

Gráfico 1- Taxonomia dos Ataques em Redes

3.3.1.1 Ataque passivo

Consiste em um ataque em que um acesso não autorizado a rede é realizado, porém nenhum conteúdo é modificado. Esse tipo de ataque pode ser uma simples escuta ou uma completa analise de tráfego.

- **Escuta –** Devido as características de uma comunicação sem fio, o atacante pode facilmente escutar e armazenar todo trafego de uma WLAN. Este tipo de ataque consiste em apenas monitorar a transmissão para a obtenção do conteúdo transmitido. Ataques passivo como a escuta não deixam rastro da presença do *hacker,* pois não é preciso que ele se conectem a um *access*

point para escutar os pacotes transmitidos no segmento *wireless*.

- **Analise de Tráfego –** Este tipo de ataque consiste na análise do tráfego monitorado a procura de padrões de comunicações afim de obter informações valiosas. Quando o tráfego em uma WLAN é criptografado, o *hacker* pode analisar o tráfego para obter informações, como por exemplo, a identificação da rede – ao menos que seja desligado, *access points* enviam através de *broadcast* seus SSIDs – juntamente com um dispositivo GPS é possível que se obtenha a localização física de um *access point* dentro de uma área, e pode também descobrir o tipo de protocolo utilizado na transmissão – baseado no tamanho, tipo e número de pacotes transmitidos em um certo período de tempo.

3.3.1.2 Ataque Ativo

Um ataque que realiza alguma modificação desautorizada na rede, como por exemplo, em uma mensagem, dados, ou arquivo. Ao contrario do ataque passivo, este tipo de ataque requer que o atacante transmita dados na rede.

- **Disfarce –** O atacante assume a identidade de um usuário autorizado e com isso obtêm certos privilégios.

- **Repetição –** O Atacante monitora a transmissão (ataque passivo) e retransmite mensagens como se fosse o usuário legítimo.

- **Modificação de Mensagem –** O atacante altera uma mensagem legítima, apagando, adicionando, editando ou reordenado.

- **Negação de Serviço –** Conhecido como DOS, este é um ataque na disponibilidade de uma rede impossibilitando que usuários legítimos tenham acesso a seus recursos e assim servindo de vários propósitos para um atacante. Sistemas de monitoramento podem detectar e identificar esse tipo de ataque, entretanto não há muito que pode ser feito para prevenir ataques de negação de serviço (COLEMAN; WESTCOTT, 2006, p. 396).

3.3.2 MAC Spoofing

Todas as *interfaces* de redes sem fio 802.11 possuem um endereço físico conhecido como endereço MAC. Este endereço é número hexadecimal de 12 dígitos que se encontra no cabeçalho da camada 2 dos *frames* 802.11. Normalmente, filtros MAC são configurados para aplicarem restrição que permitem o tráfego de apenas estações específicas. Todos as outras estações cujos endereços MAC não se encontrem na lista, não são permitidas a trafegar na rede. Entretanto, o endereço MAC pode ser facilmente forjado. Um *hacker* pode monitorar o tráfego na rede e assim obter uma lista de endereços MAC autorizados para se comunicar e então forjar seu endereço MAC e driblar esse filtro de segurança.

Devido a esse problema de segurança e todo o trabalho administrativo envolvido em configurar todos os endereços MAC permitidos, este tipo de filtro não é considerado um mecanismo de segurança confiável para as redes sem fio corporativas e deve ser implementado apenas como um último recurso.

3.3.3 Ataques de DOS

As Redes WLAN são bastante vulneráveis a ataques de DOS. Um atacante pode possivelmente tornar todo um BSS indisponível, ou impedir a conexão de estações legítimas (HE; MITCHELL, s.d., p. 3). Utilizando as características das redes sem fio, um atacante pode realizar ataques DOS de várias formas. Ataques DOS podem ocorrer nas camadas 1 e 2 do modelo OSI.

Os ataques na camada 1 são conhecidos como *jamming.* Na sua forma intencional, um atacante utiliza algum tipo de gerador de sinais para enviar uma grande quantidade de sinais na mesma freqüência de operação da rede afim de impedir a disponibilidade do serviço para os usuários legítimos. Esse incidente pode ocorrer também de forma não intencional. Interferências de rádio tais como ondas de microondas, telefones celulares, *access points* de redes vizinhas e outros dispositivos, também podem causar negação de serviço. Apesar do DOS não-intencional não ser necessariamente um ataque, ele pode causar tanto prejuízo quanto um ataque de DOS intencional.

Os tipos mais comum de ataque de negação de serviço nas redes 802.11 são na camada 2. A ampla variedade de ataques na camada 2 existentes são o resultado de inundações na rede com *frames* 802.11. A técnica mais utilizada consiste em forjar *frames* de desautenticação e desassociação e transmiti-los repetidamente. Devido ao fato de que estes tipos de *frames* de gerenciamento são *frames* de notificação que não podem ser ignorados, as estações são constantemente impedidas de utilizar o serviço.

3.3.4 Recuperação de Chaves WEP

Como já visto anteriormente, o WEP é um protocolo para a criptografia de pacotes transmitidos em redes 802.11. Em uma rede protegida pelo protocolo WEP, todos os pacotes são criptografados utilizando uma chave comum inserida no algoritmo RC4. Essa chave é compartilhada entre todas as estações móveis. Uma vez comprometida, um atacante obtém acesso total a rede. Apesar de ser conhecido como inseguro e ter sido substituído pelo seu sucessor, o WPA, este protocolo ainda é utilizado amplamente após 6 anos de ataques de recuperação de chaves terem sido descobertos (TEWS et al., s.d, p. 1).

Em 2001, Jesse Walker, funcionário da Intel, publicou um artigo para a comunidade de segurança – *Unsafe at any key size; An Analysis of the WEP encapsulation* – identificando vários problemas no mecanismo de funcionamento do protocolo WEP. Depois, em uma pesquisa feita por Nikita Borisov, foi possível quebrar uma chave WEP de 40 bits em 4 horas utilizando 250 computadores. Hoje existem várias ferramentas gratuitas na Internet que permitem a quebra do protocolo WEP em poucos minutos.

Existem vários métodos utilizados para a quebra de chaves WEP. Entretanto, o mais eficiente deles é o ataque *Fluhrer-Mantin-Shamir* (FMS). Um ataque passivo que constitui na captura de pacotes para analise e quebra de chave posteriores.

Este ataque explora uma falha do algoritmo *key scheduling* (KSA) do protocolo de criptografia do RC4. O KSA é o primeiro passo do RC4, ele transforma a chave em uma matriz da qual o RC4 gera a cifra criptografada. A falha constitui-se

em uma anomalia estatística que uma ampla classe de chaves fracas, da qual uma pequena parte da chave secreta determina um numero grande de bits da permutação inicial. (FLUHRER et al., s.d, p. 1).

O ataque propõe a captura de um numero grande de dados criptografados e a procura por pacotes cuja chave tem uma estrutura faca. Estes são os chamados “pacotes interessantes”.

O *AirSnort* é uma ferramenta famosa por sua simplicidade que implementa o ataque FMS. Ele inicia pela captura dos pacotes que são analisados posteriormente. Durante a captura, o *AirSnort* mostra o andamento do processo, incluindo o número de pacotes “interessantes”. Assim que um número suficiente destes pacotes é capturado, o processo de quebra pode ser iniciado. De acordo com a documentação da ferramenta, são necessários aproximadamente 1500 pacotes “interessantes” para quebrar uma chave WEP de 128 bits. A captura de 1500 pacotes “interessantes”, dependendo do volume do trafego da rede, poder levar várias horas ou até mesmo dias. Entretanto, a quebra do protocolo WEP pode ser acelerado injetando tráfego adicional na rede WLAN sem mesmo estar associada a esta, tornando possível a quebra do WEP em poucos minutos.

3.3.5 Ataque *Evil Twin*

Um ataque *evil twin* é uma situação em que um atacante utiliza um suposto *access point* – conhecido como *rogue access point* – se fazendo passar por um legítimo, para interceptar efetivamente as conexões de estações móveis. Este ataque ocorre geralmente em *hotspots* que oferecem conexão com a Internet e em redes *wireless* domésticas, e fazem com que usuários pensem que estão se

conectando a uma rede legítima, porém estão se conectando a uma rede maliciosa.

Um *rogue access point* pode ser definido como qualquer *access point* estranho, sem controle e sem autorização em uma determinada infra-estrutura de rede. Os *rogue access points* estão relacionados com um dos grandes problemas existentes nas organizações, que é a grande facilidade de criação de redes WLAN 802.11, porém sem conhecimento dos administradores de segurança e, principalmente, sem atender os requisitos e políticas de segurança exigidos. Essa facilidade abre uma grande janela de oportunidades para atacantes que podem explorar essa vulnerabilidade de forma a causar grandes prejuízos para a organização (NAKAMURA; LIMA, s.d., p. 1).

Entretanto, no ataque *evil twin* o *rogue access point* é utilizado pelo atacante para induzir usuários de uma rede WLAN a se conectarem a uma falsa WLAN e assim efetuar outros ataques das mais variadas formas possíveis. Como a vitima se conecta à Internet através do sistema do atacante, ela se torna um alvo falso para o atacante utilizar outros tipos de ataques. Por exemplo, páginas falsas da web podem ser utilizadas para conseguir senhas de bancos ou informações de cartões de crédito dos usuários, ou redirecioná-los para sites que tenham conteúdo malicioso tais como vírus e *trojans*.

Este ataque pode ser executado utilizando um único *laptop* com duas interfaces *wireless*. Sendo uma utilizada como um *software access point* e a outra para realizar ataques DOS. O atacante configura o *rogue access point* com os mesmos parâmetros (SSID e MAC do *access point* legítimo) da rede *wireless* – os quais podem ser facilmente obtidos através de um monitoramento dos *frames* 802.11 – e então fazem com que as estações móveis se reassociem ao falso *access point*. Para que isso ocorra, é preciso primeiramente interromper as atuais conexões

do *access point* legítimo utilizando um ataque de negação de serviço (forjando *frames* de desautenticação e desassociação), ou então o r*ogue access point* deve emitir um sinal mais forte do que o legítimo, forçando assim o *roaming* dos usuários para a falsa rede.

Pelo fato do padrão 802.11 não disponibilizar autenticação do *access point*, a estação móvel se reassocia desapercebidamente com o *rogue access point*. Uma vez ocorrida a reassociação, todo o trafego da conexão é interceptado pelo *hacker.* Pela natureza do ataque, praticamente invisível para o usuário, a quantidade de informações que o atacante pode interceptar nesta situação é limitada somente pelo tempo que ele pode executar sem ser pego (VLADIMIROV et al., 2004, p. 172).

Este tipo de ataque vem crescendo muito devido ao aumento do numero de *hotspots* no mundo. Só no Brasil são 1.897, segundo a JiWire, empresa norte-americana que mantém um diretório sobre *hotspots* desde o ano de 2003. Navegar na Internet através destes locais públicos pode ser altamente perigoso, e infelizmente há muito pouco que usuários podem fazer para se protegerem contra este tipo de ataque. Entretanto, a solução para a minimização deste grave problema de segurança deve partir das empresas que oferecem este tipo de serviço.

O presente projeto propõe uma solução efetiva para este tipo de ataque, utilizando autenticação mútua com certificação digital, a qual será abordada e detalhada no Capitulo 4 deste trabalho.

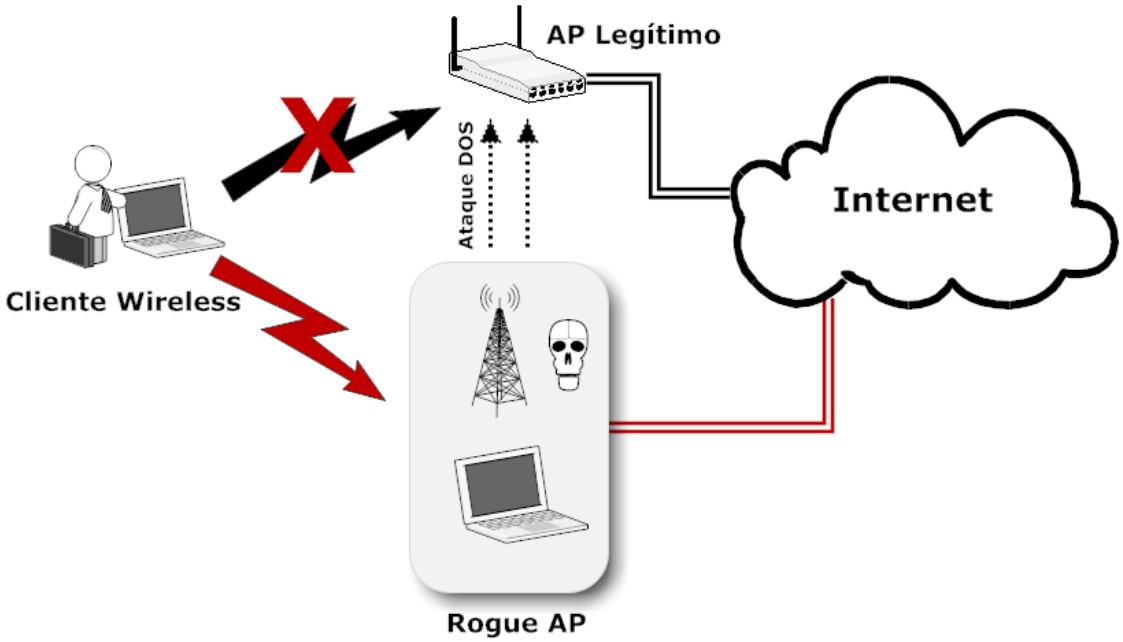

Figura 1 – Ataque *Evil Twin*

3.4 AUTENTICAÇÃO MÚTUA

Autenticação mútua, também chamada de autenticação em duas vias, é um processo ou tecnologia no qual ambas as entidades em um enlace de comunicação autenticam uma ao outra. Em um ambiente de rede, o cliente autentica o servidor e vice-versa. Desta forma, usuários de rede podem assegurar que estão se comunicando exclusivamente com entidades legítimas e servidores podem assegurar que todos os usuários estão tentando obter acesso para propósitos legítimos (SEARCHSECURITY, 2007).

Com a autenticação mútua é possível evitar o ataque *evil twin*, discutido na seção anterior. Com esse processo podemos garantir a identidade da rede na qual usuários se conectam e diminuir os grandes riscos de segurança existentes na utilizacão de redes publicas.

Esta seção abordará os requisitos e as formas para implementação de

autenticação mútua em redes 802.11.

3.4.1 Framework 802.1x/EAP

O padrão 802.1x, controle de aceso a rede baseado em porta, apesar de originalmente ter sido desenvolvido para ser utilizado em *switches Ethernet,* sua utilizacão também é possível nas redes wireless 802.11.

Seu funcionamento controla a conexão de um usuário a uma porta *Ethernet*, bloqueando a conexão e aguardando a verificação da identidade do usuário com o sistema de autenticação.

Três principais componentes constituem o protocolo 802.1x:

Suplicante – O cliente que requisita autenticação e acesso aos recursos da rede.

Autenticador – O dispositivo que bloqueia ou permite o trafego através da porta. Ao contrario, ao trafego de autenticação, todo trafego é bloqueado até que a identidade do suplicante seja verificada.

Servidor Autenticador – O servidor que valida as credenciais do suplicante que requisita acesso, e notifica o autenticador que o suplicante foi autorizado. O servidor autenticador mantém uma base de usuários ou pode requisitar a uma base de usuários externa afim de autenticar as credencias do usuário.

Em uma rede *Ethernet* 802.3, o suplicante seria uma estação de trabalho, o

autenticador seria um *switch* gerenciável e o servidor autenticador seria um servidor RADIUS (*Remote Authentication Dial In User Service*). Em uma rede *wireless* 802.11, o suplicante seria uma estação móvel, o autenticador seria um *access point* e o servidor autenticador seria um servidor RADIUS. Neste caso é considerado que uma associação entre uma estação e um *access point* é uma porta lógica para efeitos de interpretação do protocolo 802.1x (COLEMAN; WESTCOTT; 2006, p. 367).

3.4.1.1 EAP

Apesar do suplicante, autenticador e do servidor autenticador trabalharem para proverem o controle de acesso baseado em porta, um protocolo de autenticação é preciso para realizar o processo de autenticação. O protocolo de autenticação extensível (EAP) é utilizado para prover autenticação de usuário.

O EAP, primeiramente definido para o protocolo ponto a ponto (PPP), é um protocolo para a negociação de um método de autenticação. O EAP é definido na RFC 2284 e especifica as características do método de autenticação incluindo as credenciais do usuário exigidas (senha, certificado, etc.), o protocolo a ser utilizado (MD5, TLS, GSM, OTP, etc), suporte para geração de chave, e suporte para autenticação mútua. O suplicante e o servidor autenticador se comunicam utilizando o protocolo EAP (PLANET3 WIRELESS, 2002, p. 278).

O modelo de autenticação do cliente 802.1x/EAP funciona da seguinte forma:

A. O cliente requisita associação com o *access point*

B. O *access point* responde com uma requisição de identidade EAP

C. O cliente envia uma identidade EAP para o *access point*

D. A identidade EAP do cliente é repassada para o servidor de autenticação

E. O servidor de autenticação envia uma requisição de autorização para o *access point*

F. O *access point* repassa a requisição de autorização para o cliente

G. O cliente envia a autorização EAP para o *access point*

H. O *access point* repassa a autorização EAP para o servidor de autenticação

I. O servidor de autenticação envia uma mensagem de sucesso EAP para o *access point*

J. O *access point* repassa a mensagem de sucesso EAP para o cliente e o autoriza para trafegar na rede.

O *framework* 802.1x/EAP, quando utilizado em redes wireless, provê um ambiente bastante seguro e flexível baseado em vários esquemas de autenticação utilizados hoje.

3.4.1.2 EAP-TLS

O protocolo EAP, conforme já comentado, é bastante flexível e existe em vários tipos de autenticação. Alguns como o Cisco LEAP são proprietários, enquanto outros como o PEAP são considerados protocolos padrões. Podemos citar alguns tipos do protocolo EAP tais como EAP-MD5, EAP-OTP, EAP-GTC, EAP-TLS, EAP-

TTLS, CISCO LEAP, PEAP e EAP-FAST.

O EAP-TLS, baseado no protocolo TLS definido na RFC 2246, permite autenticação mútua, o qual além de prover uma camada de transporte criptografada e a capacidade de mudar as chaves dinamicamente, ainda previne a existência de *rogue access points* em redes WLAN (SANKAR et al., 2006, p. 171).

A sua implementação requer o suporte EAP-TLS no servidor RADIUS, e ainda, por ser baseado em certificados digitais, requer uma infra-estrutura de chaves publicas (ICP) para gerenciar os certificados para os usuários da rede *wireless*.

Este é o tipo de autenticação EAP que será utilizado na implementação do presente projeto.

3.4.2 Certificação Digital

A solução para problemas de identificação, autenticação e privacidade em sistemas de computação definitivamente se encontra no campo da criptografia. Devido a natureza não-física do meio computacional, métodos tradicionais de realizar fisicamente uma marca única no meio, como um selo ou assinatura, são inúteis. Ao invés disto, algumas marcas devem ser introduzidas dentro da própria informação de forma a identificar a fonte, autenticar o conteúdo e prover privacidade.

3.4.2.1 Chave Pública

Chave pública se refere a um mecanismo de criptografia. É chamado de chave pública para se diferenciar do tradicional e mais intuitivo mecanismo de

criptografia conhecido como chave simétrica, segredo compartilhado, chave secreta e também chave privada (CGIGROUP, 2004, pg. 3).

Criptografia de chave simétrica é um mecanismo em que a mesma chave é utilizada para tanto criptografar quanto para descriptografar. Ela é mais intuitiva devido a sua similaridade com o que é esperado utilizar para abrir e fechar uma porta: a mesma chave. Essa característica requer mecanismos sofisticados para distribuir seguramente a chave secreta entre ambas as partes. A chave pública, por outro lado, introduz outro conceito envolvendo pares de chaves – uma para criptografar e outra para descriptografar. Este mecanismo é mais eficiente e atrativo, além de prover várias vantagens sobre a chave simétrica tais como distribuição de chave simplificada e assinatura digital.

A chave pública é comumente usada para identificar o método de criptografia que utiliza um par de chaves assimétricas, sendo uma pública e uma privada. Este par de chaves é utilizado para criptografar e descriptografar. Enquanto a chave pública é distribuída livremente, a chave privada nunca é distribuída e deve ser mantida em segredo. Dado um para de chaves, dados criptografados com a chave pública somente podem ser descriptografados com a chave privada; de maneira análoga, dados criptografados com a chave privada somente podem ser descriptografados com a chave pública. Esta característica é utilizada para implementar criptografia e assinatura digital.

3.4.2.1.1 Criptografia

Considere a seguinte situação. Para A enviar uma mensagem privada para B, é necessário que primeiramente A tenha a chave pública de B; devido a esta

chave ser pública, B pode enviá-la através da rede sem nenhuma preocupação. Um vez que A tenha a chave pública de B, ele criptografa a mensagem utilizando a chave pública de B e a envia. Ao receber a mensagem, B descriptografa a mensagem utilizando sua chave privada.

3.4.2.1.2 Assinatura Digital

Assinatura digital é um mecanismo pelo qual uma mensagem é autenticada, isto é, uma garantia que a mensagem provém de um remetente legítimo. No processo de assinar digitalmente uma mensagem, um *hash* (resumo criptográfico) é gerado para a mensagem e depois este *hash* é criptografado com a chave privada do remetente, e então a mensagem é enviada. Ao chegar ao destinatário, é calculado um novo *hash* para a mensagem e este é comparado com o *hash* original obtido após descriptografá-lo utilizando a chave pública do remetente. Caso o resultado seja igual, a assinatura digital é validada, ou seja, aquela mensagem foi realmente enviada pelo remetente que diz ser. Caso não seja, devido a alguma alteração da mensagem no caminho o qual resultaria em um *hash* diferente, ou se o *hash* fosse criptografado utilizando uma chave diferente da chave privada do remetente legítimo, a assinatura digital não é validada, ou seja, aquela mensagem não foi enviada pelo remetente que diz ser.

3.4.2.2 Certificado Digital

Conforme explicado anteriormente, a utilizacão de chaves assimétricas, para criptografia e assinatura digital, garantem a privacidade e autenticidade de uma

mensagem. Mas como é possível garantir que uma chave pública pertence realmente a pessoa que diz ser? Esta questão é resolvida através do uso de certificados digitais.

Um certificado digital é um documento eletrônico que prova a identidade do dono de uma chave pública. Os certificados digitais são assinados e emitidos seguramente por uma entidade confiável chamada Autoridade Certificadora (AC). Contanto que uma pessoa confie na AC, é possível garantir que aquela chave pertence a pessoa que diz ser (SAFESCRYPT, s.d., pg. 7).

Um certificado digital normalmente contem as seguintes informações:

- A identidade da AC
- A identidade do pessoa ou entidade
- A chave pública
- O período de validade do certificado
- A assinatura da AC que assinou o certificado
- O numero de série

Com o certificado digital, ao invés de apenas a chave pública, o destinatário pode verificar algumas questões a respeito do remetente afim de garantir a sua identidade:

- Comparar a identidade da pessoa
- Verificar se o certificado ainda é válido
- Verificar se o certificado foi assinado por uma AC confiável
- Verificar a assinatura do certificado certificando que esta não foi alterada

3.4.2.3 Infra-Estrutura de Chaves Públicas

Os certificados digitais são uma parte de uma série de componentes que compõem uma infra-estrutura de chaves públicas (ICP). Uma ICP inclui entidades chamadas autoridades certificadoras (ACs) que emitem, gerenciam e revogam certificados digitais; partes que utilizam os certificados como indicadores de autenticação, e clientes que requisitam e utilizam os certificados digitais. Uma AC pode ainda criar uma autoridade de registro (AR) para delegar a tarefa de identificar indivíduos que requisitam certificados (BOETTCHER; POWELL, 2002, pg. 6).

CAPÍTULO 4

PROPOSTA DE SOLUÇÃO E MODELO

4.1 APRESENTAÇÃO GERAL DO MODELO PROPOSTO

O presente projeto propõe uma solução que implementa autenticação mútua através de certificados digitais para as redes WLAN 802.11 no objetivo de minimizar o problema do ataque *Evil Twin*, detalhado no capitulo 3.

4.1.1 Autenticação Mútua

Como forma de garantir que usuários de uma rede WLAN possam assegurar que sempre se conectarão a uma rede legítima, a solução proposta utiliza-se do princípio de autenticação mútua para garantir a autenticidade tanto dos clientes quanto da rede WLAN.

Figura 2 - Esquema de Autenticação Mútua

Conforme mostra a figura, uma Autoridade Certificadora AC emite certificados digitais para o cliente e a rede WLAN, assim uma relação de confiança é estabelecida. Desta forma, o cliente e a rede WLAN podem confiar em qualquer elemento que tenha um certificado digital assinado por esta Autoridade Certificadora.

No momento do estabelecimento de uma comunicação, o cliente e a rede WLAN trocam seus respectivos certificados digitais e assim validam a autenticidade um do outro, e então iniciam uma comunicação segura. Em caso do certificado digital recebido pelo cliente não ser assinado por uma Autoridade Certificadora em que este confie - um certificado digital não válido - este terá a certeza de que a rede WLAN que ele está tentando se conectar não é uma rede legítima e assim não estabelecerá uma conexão. O mesmo vale para a rede WLAN caso esta receba um certificado digital do cliente que não seja assinado por uma AC que confie.

Desta forma, no ataque *Evil Twin* o *hacker* ao implementar um Rogue AP poderá forjar o SSID e MAC do *access point* legítimo, mas nunca um certificado digital válido, e isso o impedirá de obter sucesso no ataque.

4.1.2 O Modelo

A idéia do projeto é proporcionar uma estrutura de autenticação segura para as redes 802.11, utilizando protocolos existentes e totalmente baseado em *software* livre, que pudesse ser facilmente implementada por um usuário comum, desprovido de maiores conhecimentos técnicos.

As redes WLAN domésticas, maiores utilizadas por usuários comuns, são as mais desprovidas de segurança. Isso devido a falta de conhecimentos técnicos necessários e a dificuldade em ajustar vários parâmetros na configuração de um

equipamento de rede, por exemplo um modem ou roteador *wireless*, e até mesmo de seu computador.

Então facilidade e praticidade são os objetivos do modelo proposto que implementa um servidor autenticador com todos recursos necessários para prover o controle de acesso e autenticação para uma rede WLAN de acordo com a topologia mostrada abaixo.

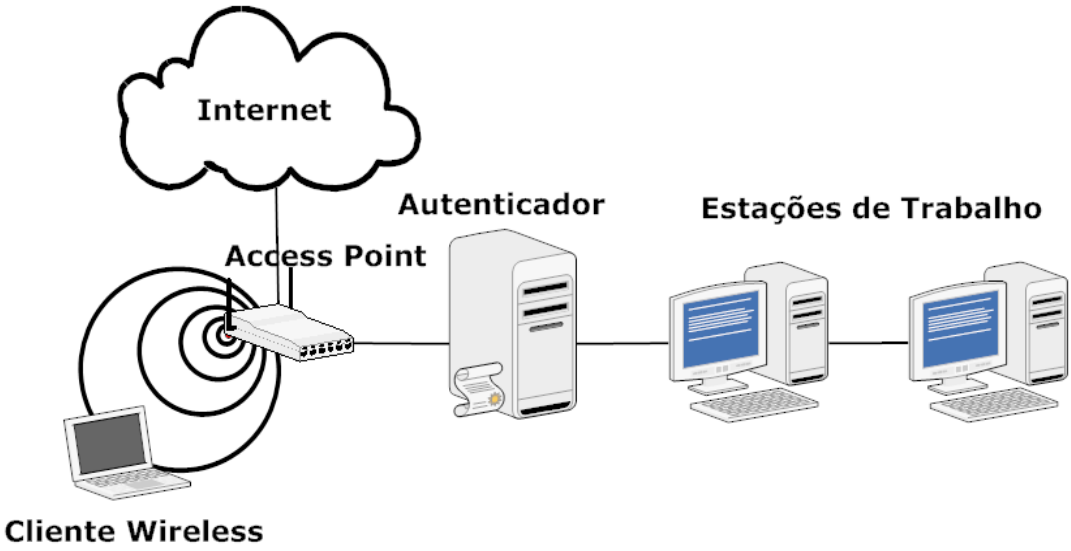

Figura 3 – Topologia do Modelo

Para o cliente se conectar à rede WLAN através do *access point*, é realizada autenticação mútua através do servidor autenticador que desta forma permite a garantia de autenticidade de ambos.

A modelo proposto permite que um computador com recursos mínimos necessários possa facilmente tornasse o servidor autenticador. Através de um *pendrive*, o servidor autenticador pode ser inicializado e assim estar preparado para autenticar e controlar todos os acessos a rede WLAN.

Uma aplicação de gerenciamento proporciona uma forma simples de

gerenciar o servidor autenticador. O controle dos certificados digitais (emitir, revogar, etc) e o gerenciamento do serviço que controla o acesso à rede, são realizados pela aplicação de gerenciamento.

4.1.3 Fases da implementação

A implementação do projeto foi dividida em 3 fases:

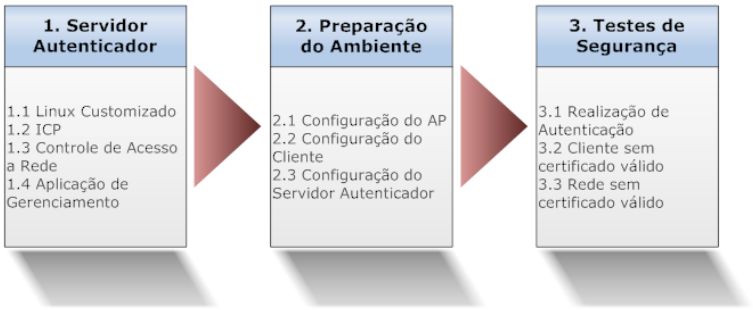

Figura 4 – Fases da Implementação

1. **Servidor Autenticador -** Esta primeira fase contempla o desenvolvimento do modelo proposto que é dividido em 4 etapas:

   a) Linux Customizado – Implementação de um Linux altamente customizado com recursos mínimos necessários compilado diretamente da fonte.

   b) ICP – Implementação de uma Infra-Estrutura de Chaves Públicas de modo a emitir certificados digitais utilizados na autenticação.

   c) Controle de Acesso à Rede – Implementação de um serviço de rede

capaz de controlar qualquer acesso à rede WLAN utilizando certificados digitais.

d) Aplicação de Gerenciamento – Desenvolvimento de uma aplicação de simples utilização capaz de gerenciar a infra-estrutura de chaves públicas bem como o serviço de controle de acesso à rede.

2. **Preparação do Ambiente** - Após o modelo ser desenvolvido, esta fase realiza a preparação do ambiente realizando configurações de todos elementos envolvidos para o devido funcionamento da solução proposta. Esta fase será detalhada no capítulo 5.

3. **Testes de Segurança** – Nesta fase testes são realizados afim de avaliar o modelo proposto. Testes de funcionamento e de segurança são aplicados na tentativa de validar os objetivos propostos e de identificar possíveis falhas. Os detalhes dos testes bem como seus resultados serão descritos no capítulo 5.

4.2 DESCRIÇÃO DAS ETAPAS DO MODELO

Nesta seção será detalhada cada etapa de desenvolvimento do modelo, demonstrando a utilização adequada dos métodos e técnicas escolhidos na solução.

No modelo proposto toda autenticação, ocorrida na rede WLAN, entre o *access point* e os clientes é realizada mutuamente através do Servidor Autenticador que é apresentado em módulos distribuídos em camadas conforme mostra a figura abaixo.

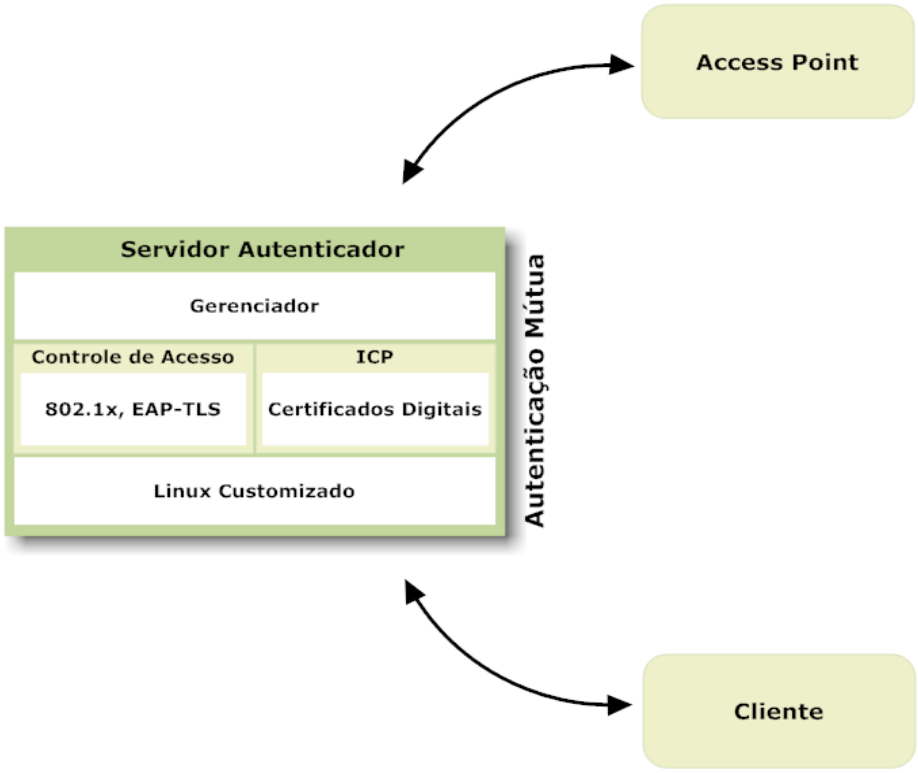

Figura 5 – Módulos Distribuídos em Camadas

4.2.1 Linux Customizado

O desenvolvimento de um Linux altamente customizado foi implementado a partir do projeto Linux *From Scratch* (LFS) na sua versão 6.2. O LFS é um projeto que prove instruções para construção de um sistema Linux completamente da fonte.

Dessa maneira foi possível produzir um sistema Linux bastante compacto, compilando apenas os programas estritamente necessários para o funcionamento do modelo. Outra vantagem, conforme todo sistema é compilado diretamente da fonte, foi a possibilidade de auditar todo o processo de compilação, aplicando todos os *patches* de segurança disponíveis para cada pacote.

Para a construção do Linux customizado foi utilizado o LFS *LiveCD* como sistema anfitrião. O LFS *LiveCD* é um CD capaz de inicializar um sistema

operacional completo, totalmente independente de um disco rígido. O LFS LiveCD é designado especificamente para prover um sistema confiável no objetivo de construir um Linux customizado.

Este sistema anfitrião é usado como ponto de partida o qual fornece os programas de desenvolvimento necessários, incluindo um compilador, um editor de vínculos (*linker*), e um *shell*, para montar o novo sistema customizado.

Abaixo seguem as etapas de construção do Linux customizado.

4.2.1.1 Preparação de uma Nova Partição

Duas novas partições necessárias para o construção do sistema foram criadas. Uma partição *swap* (dev/sda1), para devida alocação de memória virtual destinada para os processos de compilação, e outra ext3 para o sistema de arquivos nativo, onde o novo Linux customizado será compilado e instalado.

Após o sistema de arquivos ser criado, a partição foi montada em /mnt/lfs.

4.2.1.2 Configuração do Ambiente de Trabalho

Nesta etapa é realizada a instalação e configuração de um ambiente de trabalho próprio ao processo de montagem do Linux customizado.

Os pacotes e *patches* que compõem o novo sistema foram baixados no diretório criado como $LFS/sources. Neste diretório são extraídos e compilados os fontes.

Uma variável de ambiente $LFS foi criada e apontada para o ponto de

montagem do novo sistema de arquivos /mnt/lfs. Isso devido o fato de facilitar a referencia do local de montagem da partição onde será instalado o Linux customizado.

Todos os programas compilados que servirão como ferramentas temporárias, e não serão parte do novo sistema, serão instaladas no diretório criado como $LFS/tools. Isso devido ao fato de mantê-los separados daqueles programas compilados que constituirão o novo sistema. Mantendo esses programas em um diretório separado facilita no momento de descartá-los após serem utilizados.

Um usuário, lfs, foi criado para a compilação dos pacotes. Assim, conforme este usuário não tem privilégios, não haverá a possibilidade de comprometer o sistema durante o processo de compilação.

4.2.1.3 Construindo um Sistema Temporário

Nesta etapa serão instalados alguns pacotes que dão forma ao conjunto básico de desenvolvimento (ou "*toolchain*") que será usado para construir o Linux customizado. Alguns destes pacotes são necessários para resolver dependências cruzadas ou circulares; por exemplo, para compilar um compilador, é necessário ter um compilador.

Uma primeira versão do jogo de ferramentas (*toolchain*), incluindo o Binutils e o GCC será instalada. A etapa seguinte é configurar a Glibc, a biblioteca C. A Glibc será compilada pelos programas do *toolchain* construídos em primeira versão. Então, uma segunda versão do conjunto de ferramentas será configurada. Desta vez, as ferramentas serão vinculadas dinamicamente ao Glibc recém-configurado.

Os pacotes restantes são configurados usando este segundo conjunto de ferramentas. Quando isto é feito, o processo da instalação do Linux customizado não mais depende do sistema anfitrião, com exceção do *kernel*.

4.2.1.4 Construindo o Linux Customizado

Nesta etapa, os programas que compõem o Linux customizado são construídos. O programa *chroot* (*change root*) é usado para entrar em um ambiente virtual e inicializar um novo *shell* cujo o diretório de raiz seja definido na partição do LFS. A vantagem principal desse "*chrooting*" é permitir o uso do sistema anfitrião enquanto o LFS estiver sendo configurado.

Aqui são compilados todos os programas que farão parte do novo sistema real.

4.2.1.5 Configurando os *Scripts* de Inicialização e Compilando o *Kernel*

Nesta etapa, o pacote *LFS-Bootscripts* é instalado. Este pacote contém uma série de *scripts* para iniciar e parar o Linux customizado na sua inicialização e desligamento.

Agora o *kernel* 2.6.16.27 é configurado e compilado.

Agora com o novo sistema quase completo é o momento de configurá-lo para que o mesmo possa ser inicializado. O GRUB, gerenciador de inicialização, é configurado para que o *kernel* seja inicializado.

4.2.2 ICP – Infra-Estrutura de Chaves Públicas

Este módulo prevê a implementação de uma Infra-Estrutura de Chaves Públicas (ICP), uma estrutura necessária para o funcionamento da certificação digital.

Essa ICP permitirá a criação de uma Autoridade Certificadora (AC) que irá emitir e assinar os certificados digitais, e assim estabelecer uma relação de confiança entre os pares de comunicação sem fio.

Para a implementação da ICP foi compilado e instalado o OpenSSL 0.9.8e, um projeto *Open Source* que implementa os protocolos SSL (*Secure Sockets Layer*) e TLS (*Transport Layer Security*).

O OpenSSL criará certificados digitais X.509 e padrões de assinatura PKCS#12. Esse padrão de assinatura especifica um formato preferido por vários gerenciadores de operações de certificados e é suportado por vários navegadores e recentes versões de sistemas operacionais da família *Windows*. Sua vantagem é a capacidade de armazenar o certificado e a chave correspondente, o certificado raiz, e qualquer outro certificado da cadeia em um único arquivo (CITRIX, 2007).

4.2.3 Controle de Acesso

Este módulo prevê a implementação do *framework* 802.1x/EAP-TLS que permitirá o controle de acesso a rede WLAN utilizando certificados digitais.

Este *framework* utiliza um serviço RADIUS para validar as credenciais do suplicante que requisita o acesso. O RADIUS (*Remote Authentication Dial-In User Service*) é um protocolo cliente/servidor que permite servidores de acesso remoto a se comunicarem com um servidor central afim de autenticarem usuários e autorizarem seus acessos ao serviços requisitados (SEARCHSECURITY, 2007).

No modelo proposto o *access point* é configurado para autenticar os clientes moveis através do serviço RADIUS implementado no Servidor Autenticador.

Para implementar o serviço RADIUS foi compilado e instalado o software livre FreeRADIUS 1.1.7.

4.2.4 Gerenciador

O modelo desenvolvido até aqui estaria completo para realizar autenticação mútua através de certificados digitais em uma rede WLAN. Entretanto, caberia uma tarefa árdua para o usuário poder configurar o Servidor Autenticador afim de implementar o modelo.

Por isso, este módulo gerenciador consiste em uma aplicação desenvolvida em Perl afim de implementar uma interface de simples utilização capaz de gerenciar a Infra-Estrutura de Chaves Públicas bem como o serviço de controle de acesso à

rede.

Suas funcionalidades consistem em: criar Autoridades Certificadoras; emitir, revogar e assinar certificados digitais; configurar, iniciar e parar o serviço RADIUS.

CAPÍTULO 5

APLICAÇÃO DA SOLUÇÃO COM RESULTADOS

5.1 APRESENTAÇÃO DO AMBIENTE DE SIMULAÇÃO E IMPLEMENTAÇÃO DO MODELO PROPOSTO

Um ambiente simples será utilizado para a implementação do modelo proposto. Uma pequena rede 802.11 composta de um access point, um cliente wireless e o Servidor Autenticador compõem a estrutura da demonstração.

A tabela abaixo detalha os elementos do ambiente de implementação.

Elemento	Descrição
Cliente Wireless	*Notebook* Sony Vaio VGN-FT53DB, Intel Core2 T5500 1.66 Ghz, 1GB RAM, 1 *interface ethernet* 10/100 Mbps, 1 *interface wireless* 802.11, rodando Linux Mandriva 2007.1 64 bits
Access Point	Roteador *Wireless* D-Link DI-624 com suporte a segurança *Wi-Fi Protected Access Version* 2 (WPA2) e 802.1x/EAP
Servidor Autenticador	Máquina virtual rodando no *Notebook* cliente utilizando o *VMware Server 1.0.4 build-56528*

Tabela 1 – Elementos do Ambiente de Implementação

Por motivos de consolidação da infra-estrutura, a fim de facilitar a

demonstração do modelo, a implementação é realizada utilizando tecnologia de virtualização. Por isso, tanto o cliente *wireless* quanto o Servidor Autenticador operam fisicamente em um mesmo computador, o *notebook* Vaio. Apesar disto, eles possuem funcionamento independente rodando em ambientes totalmente isolados, sendo que o Servidor Autenticador opera dentro de uma maquina virtual instalada no cliente *wireless*.

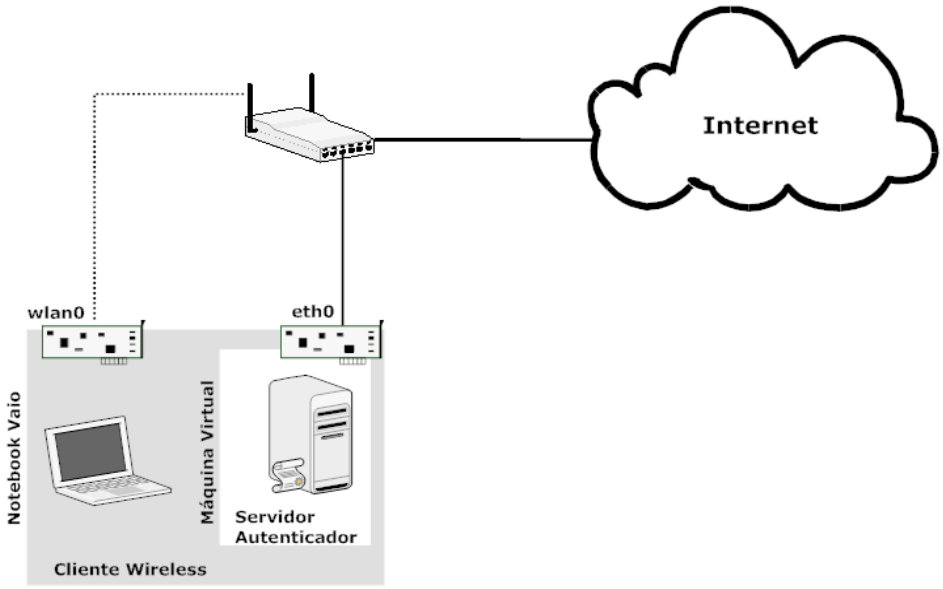

Figura 6 – Ambiente de Implementação

Conforme a topologia de implementação apresentada acima, o *notebook* Vaio possui uma interface *ethernet* (eth0) que é utilizada pelo Servidor Autenticador, rodando em uma máquina virtual, para realizar conexão *ethernet* com o *access point*, e outra *interface wireless* (wlan0) que é utilizada pelo cliente *wireless* para realizar conexão sem fio com a rede WLAN. O *access point* possui uma conexão com Internet banda larga a cabo.

Para a demonstração do funcionamento do modelo proposto prepararemos o

ambiente e realizaremos testes de segurança.

Primeiramente é necessário preparar o ambiente afim de obter o modelo pronto para oferecer um mecanismo de autenticação seguro a uma rede WLAN. São executadas as tarefas de criação de uma Autoridade Certificada, emissão e distribuição dos certificados digitais e a configuração dos elementos envolvidos na autenticação (*access point*, cliente *wireless* e Servidor Autenticador).

Após preparado o ambiente, serão realizados os testes de segurança como forma de validação dos objetivos propostos. Os testes serão aplicados em tentativas de autenticação na rede WLAN em 3 cenários distintos.

Cenário	Descrição
Autenticação válida	Situação de sucesso. Ambos cliente e rede com certificados válidos
Cliente sem certificado válido	Tentativa de um cliente não autorizado a se conectar a rede
Rede sem certificado válido	Este cenário simulará um ataque *Evil Twin*. Cliente tentando se conectar a uma rede não-legítima

Tabela 2 – Cenários dos Testes de Segurança

5.2 DESCRIÇÃO DA APLICAÇÃO DA SOLUÇÃO

5.2.1 Preparação do Ambiente

Na preparação do ambiente serão executadas as tarefas de preparação para

a implementação do modelo proposto.

5.2.1.1 Criando a Autoridade Certificadora

Através do gerenciador awcd, o comando “criaac” é utilizado para criar uma nova Autoridade Certificadora.

Alguns parâmetros são pedidos na criação da nova AC. O nome da AC (CEUB), a senha da chave privada (senhasecreta), e as informações do certificado digital da AC tais como país (BR), estado (DF), cidade (Brasília), nome da Organização (UniCEUB), nome do Departamento (Engenharia) e o nome do certificado (Autoridade Certificadora).

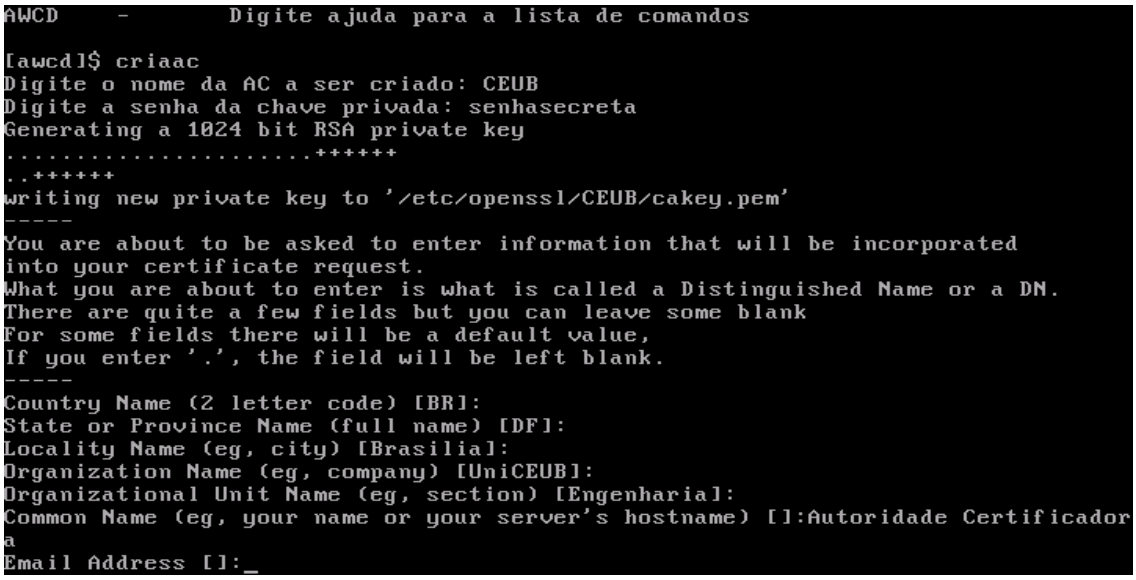

Figura 7 – Criação da AC

O diretório /etc/openssl/CEUB é criado com toda a estrutura e arquivos necessários para a emissão de novos certificados, incluindo o certificado e a chave privada da Autoridade Certificadora.

5.2.1.2 Emitindo os Certificados Digitais

a) Cliente Wireless

Através do gerenciador awcd, o comando “emite” é utilizado para a criação do certificado digital para o cliente *wireless*.

Alguns parâmetros são pedidos na criação do certificado digital. O nome da AC que emitirá o certificado (CEUB), a senha da chave privada da AC (ela servirá no momento em que o certificado digital do cliente for assinado pela AC), o nome do certificado a ser criado (cliente), e as informações do certificado digital tais como país (BR), estado (DF), cidade (Brasília), nome da Organização (UniCEUB), nome

do Departamento (Engenharia) e o nome do certificado (Cliente *Wireless*).

```
[awcd]$ emite
Digite o nome da AC: CEUB
Digite a senha da chave privada da AC: senhasecreta
Digite o nome do certificado a ser criado: cliente
Digite a senha da chave privada: senhasecreta
Generating a 1024 bit RSA private key
.............................++++++
.......++++++
writing new private key to 'clientekey.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [BR]:
State or Province Name (full name) [DF]:
Locality Name (eg, city) [Brasilia]:
Organization Name (eg, company) [UniCEUB]:
Organizational Unit Name (eg, section) [Engenharia]:
Common Name (eg, your name or your server's hostname) []:Cliente Wireless
Email Address []:_
```

Figura 8 – Emissão do Certificado Digital para o Cliente

Após serem fornecidos os parâmetros, serão mostradas as informações e a confirmação para assinar o certificado a ser criado.

```
        Serial Number:
            8f:5a:ef:0d:46:14:30:6a
        Validity
            Not Before: Nov 10 21:06:28 2007 GMT
            Not After : Nov  9 21:06:28 2008 GMT
        Subject:
            countryName               = BR
            stateOrProvinceName       = DF
            localityName              = Brasilia
            organizationName          = UniCEUB
            organizationalUnitName    = Engenharia
            commonName                = Cliente Wireless
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                65:7A:64:41:05:BB:8B:34:75:F2:AA:CB:4A:FD:A1:24:7D:4F:6B:E6
            X509v3 Authority Key Identifier:
                keyid:83:EE:F7:17:8A:B9:39:16:F7:F6:96:C8:E2:B7:89:35:E3:31:52:4
A

Certificate is to be certified until Nov  9 21:06:28 2008 GMT (365 days)
Sign the certificate? [y/n]:_
```

Figura 9 – Assinatura do Certificado Digital para o Cliente

São criados dois arquivos clientecert.pem e clientekey.pem, respectivamente

o certificado digital e a chave privada do cliente. Estes arquivos são gerados no mesmo diretório de onde o gerenciador foi executado.

b) Servidor Autenticador

Novamente o comando “emite” é utilizado. Agora para criar o certificado digital para o Servidor Autenticador. Os mesmos parâmetros são pedidos na criação do certificado digital o qual é criado com o nome de servidor.

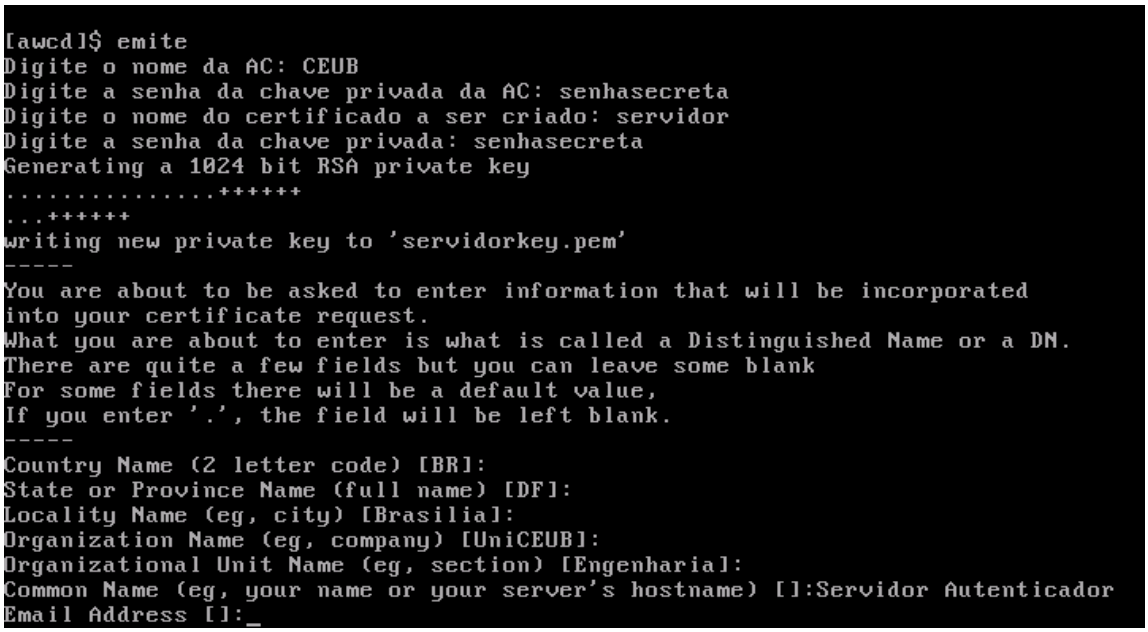

Figura 10 – Emissão do Certificado Digital para o Servidor Autenticador

Após serem fornecidos os parâmetros, serão mostradas as informações e a confirmação para assinar o certificado a ser criado.

Figura 11 – Assinatura do Certificado Digital do Servidor Autenticador

São criados dois arquivos servidorcert.pem e servidorkey.pem, respectivamente o certificado digital e a chave privada do Servidor Autenticador. Estes arquivos são gerados no mesmo diretório de onde o gerenciador foi executado.

5.2.1.3 Configurando o Servidor Autenticador

Na configuração do Servidor Autenticador, através do gerenciador awcd o comando “configuraip” é utilizado para configurar o endereço IP do servidor (192.168.0.3), o comando “configurasenha” para configurar a senha do serviço RADIUS, o comando “importacert” para importar o certificado da AC, o certificado e a chave privada do Servidor Autenticador para o serviço RADIUS, e finalmente o comando “iniciaradius” para iniciar o serviço RADIUS.

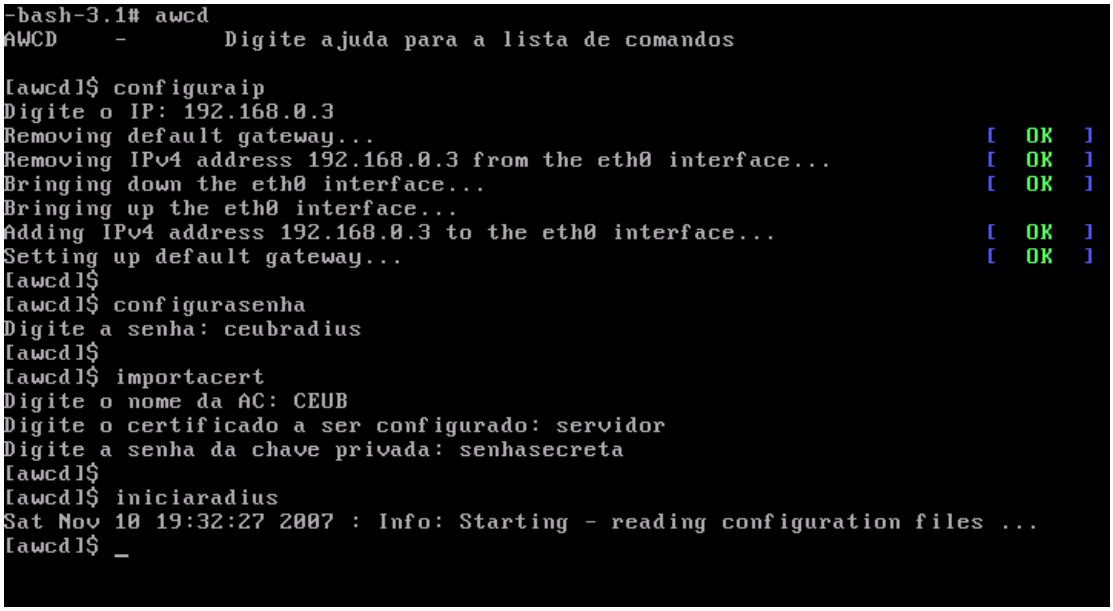

Figura 12 – Configuração do Servidor Autenticador

5.2.1.4 Configurando o *Access Point*

Através da interface de gerenciamento Web do *access point* D-Link DI-624, são configurados os parâmetros SSID (UniCEUB), o protocolo de autenticação (EAP), o endereço IP do servidor RADIUS (192.168.0.3) e a senha do servidor RADIUS (ceubraid).

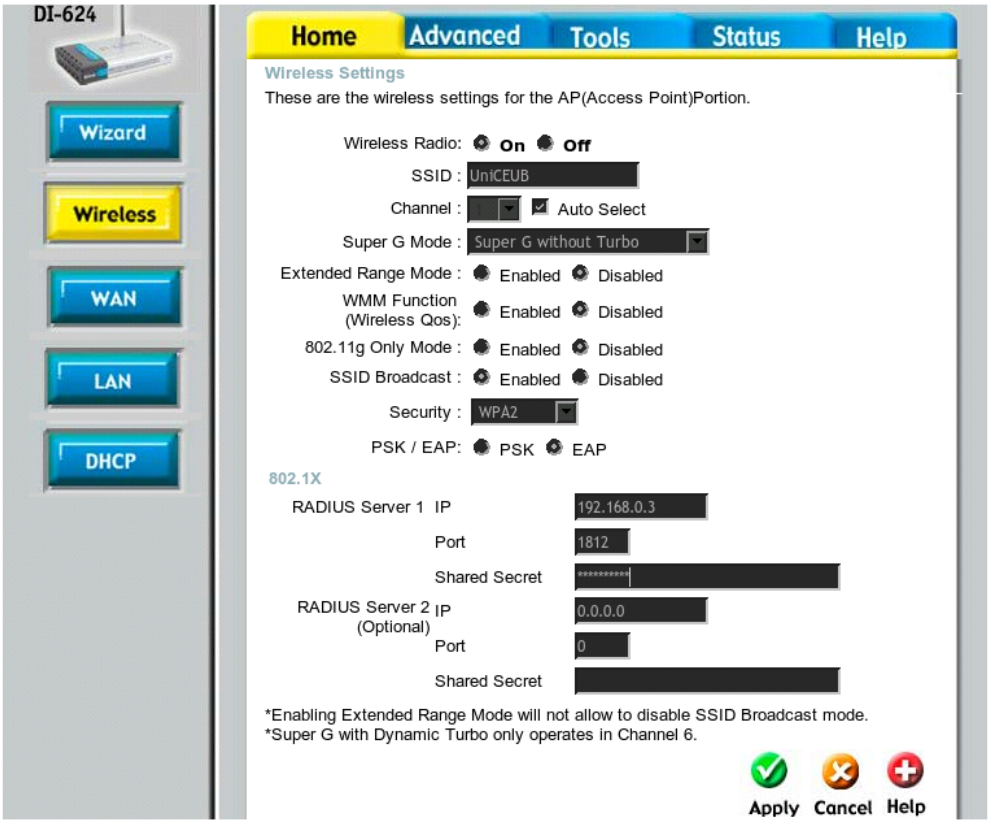

Figura 13 – Configuração do Access Point

5.2.1.5 Configurando o Cliente *Wireless*

Através do gerenciador awcd, o comando “exporta” é utilizado para copiar o certificado digital da AC para o diretório de onde o gerenciador foi executado. Agora é preciso copiar para o cliente wireless o certificado digital da AC além do certificado digital e a chave privada do cliente criados anteriormente.

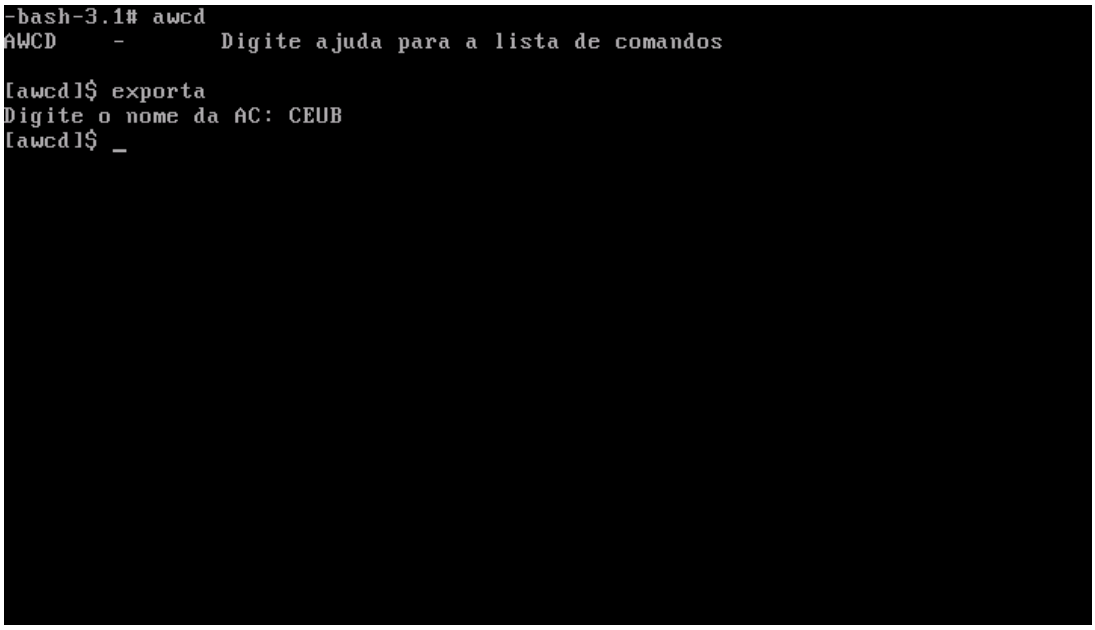

Figura 14 – Exportação do Certificado Digital da AC

O programa wpa_supplicant é utilizado no cliente para gerenciar a conexão *wireless*. É preciso configurar previamente os parâmetros tais como o SSID da rede, o protocolo de autenticação EAP-TLS e a localização dos certificados digitais do cliente e da AC, e a chave privada da AC.

5.2.2 Testes de Segurança

Após preparado o ambiente, os testes de segurança são realizados em 3 cenários distintos. Os testes e seus resultados são detalhados abaixo.

5.2.2.1 Autenticação Válida

Este cenário indica uma situação de sucesso, onde ambos cliente e rede possuem certificados válidos, ou seja certificados digitais assinados por uma AC em que ambos confie.

Uma vez configurado o cliente inicia o wpa_gui, uma interface gráfica para o programa wpa_supplicant, no objetivo de iniciar a conexão *wireless*.

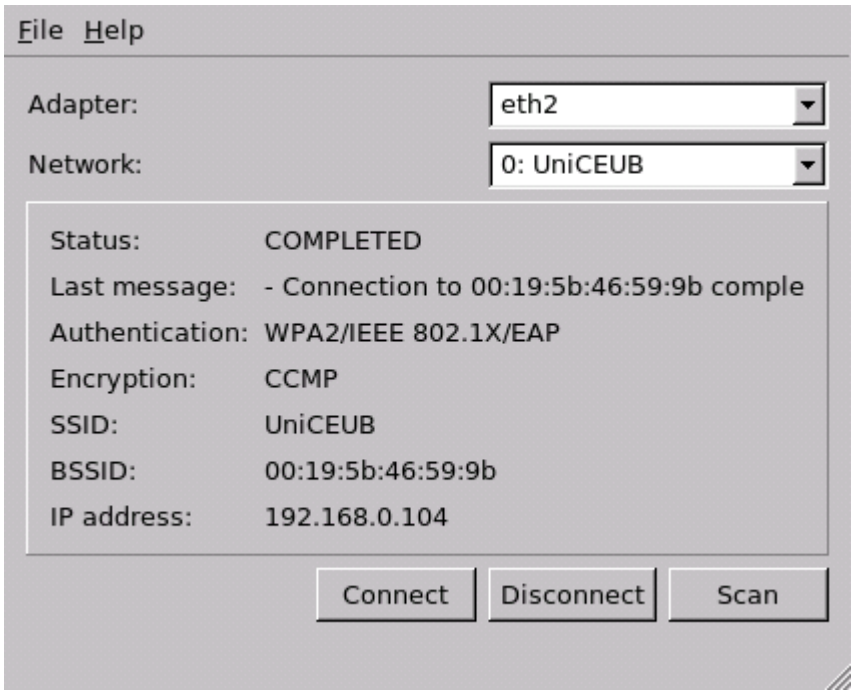

Figura 15 – Teste de Autenticação Válida (A)

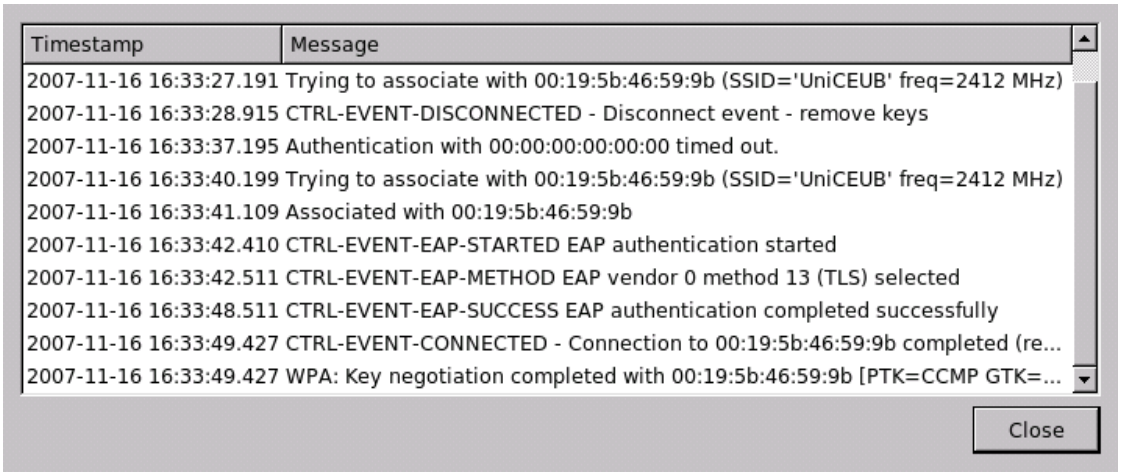

Figura 16 – Teste de Autenticação Válida (B)

Como pode ser observado na figura acima que mostra o log da conexão realizada, após a associação com o *access point* é iniciada a autenticação EAP que neste caso é completada com sucesso.

Após ser completada a autenticação, um teste de conectividade a Internet é realizado com o *host* www.uol.com.br.

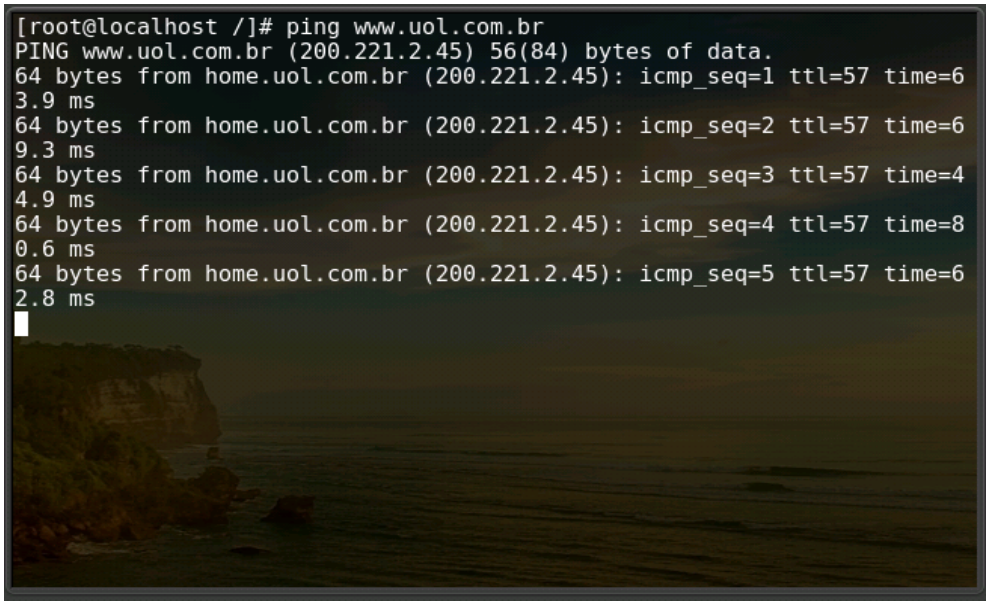

Figura 17 – Teste de conectividade com o host www.uol.com.br

5.2.2.2 Cliente sem Certificado Válido

Este cenário indica uma situação onde um cliente não autorizado tenta se conectar a rede. Tecnicamente isso significa em ele não possuir um certificado digital assinado por uma AC em que a rede confie.

Para a demonstração deste cenário, foi emitido um certificado digital para o cliente assinado por uma AC diferente daquela criada na preparação do ambiente.

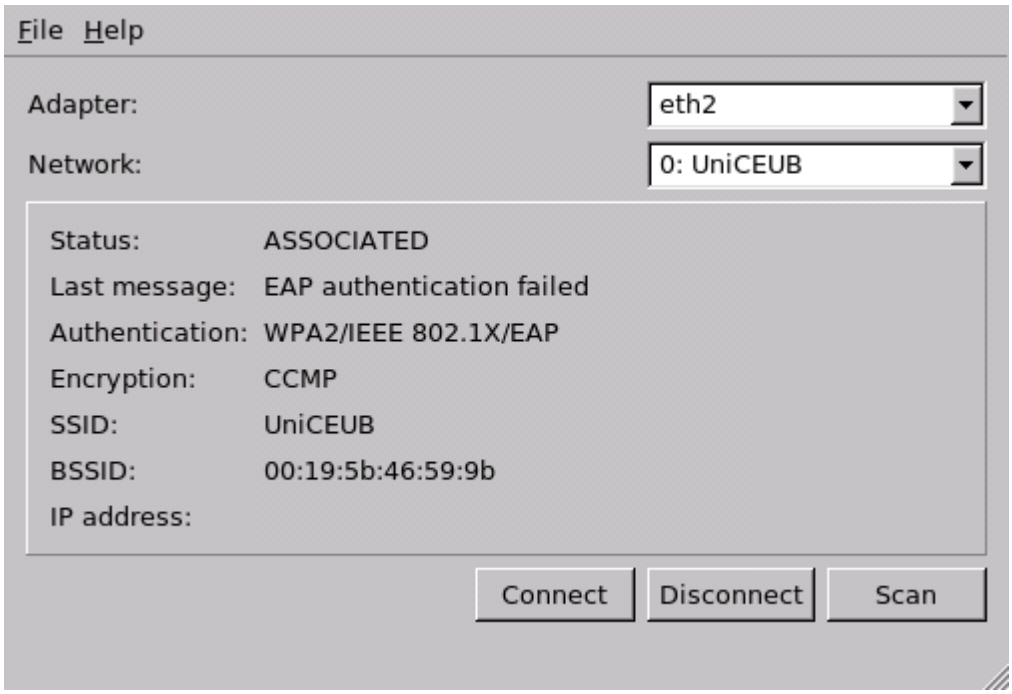

Figura 18 – Teste de Autenticação – Cliente sem Certificado Válido (A)

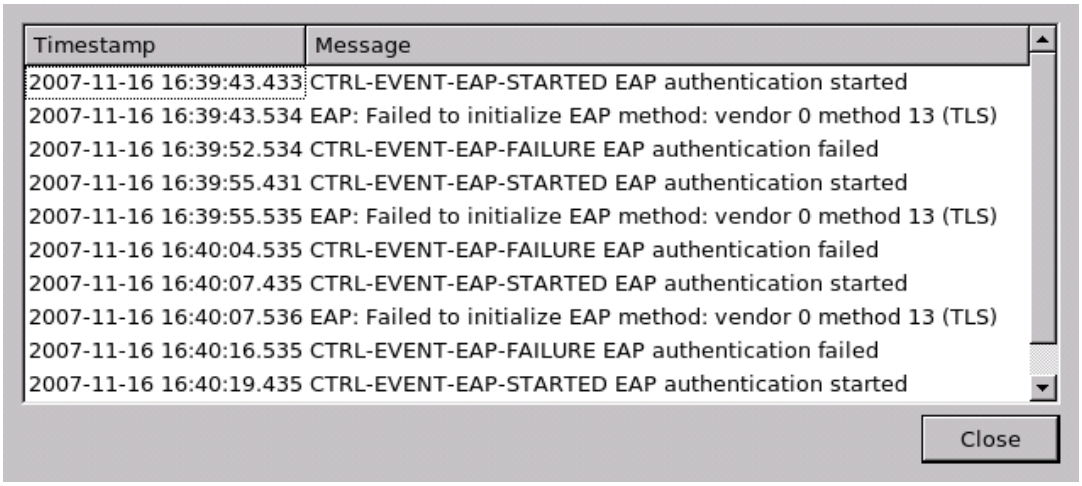

Figura 19 – Teste de Autenticação – Cliente sem Certificado Válido (B)

Como pode ser observado na figura acima que mostra o log da conexão, ocorre uma falha no momento da autenticação EAP, isso devido ao cliente não possuir um certificado válido.

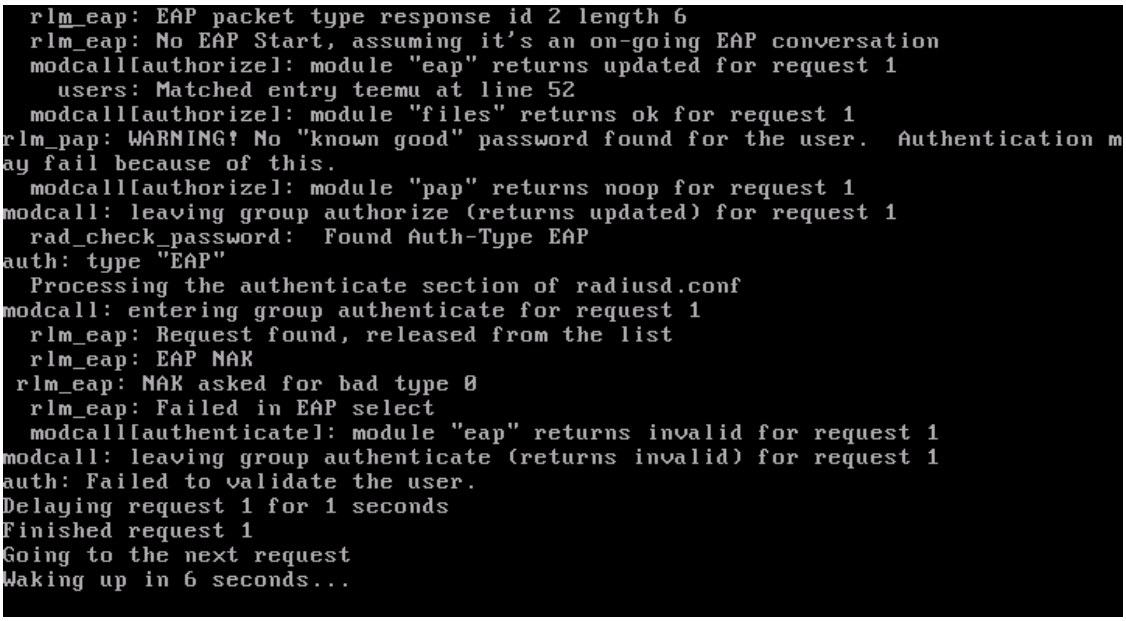

Figura 20 – Teste de Autenticação – Log do Servidor Autenticador

A figura acima mostra o log do Servidor Autenticador que indica o momento

em que ocorre a falha na validação do cliente.

5.2.2.3 Rede sem Certificado Válido

Este cenário simula um ataque *Evil Twin*, onde o cliente tenta se conectar a uma rede não-legitima, ou seja, sem certificado digital assinado por uma AC em que ele confie.

Para a demonstração deste cenário, foi emitido um certificado digital para o Servidor Autenticador assinado por uma AC diferente daquela criada na preparação do ambiente.

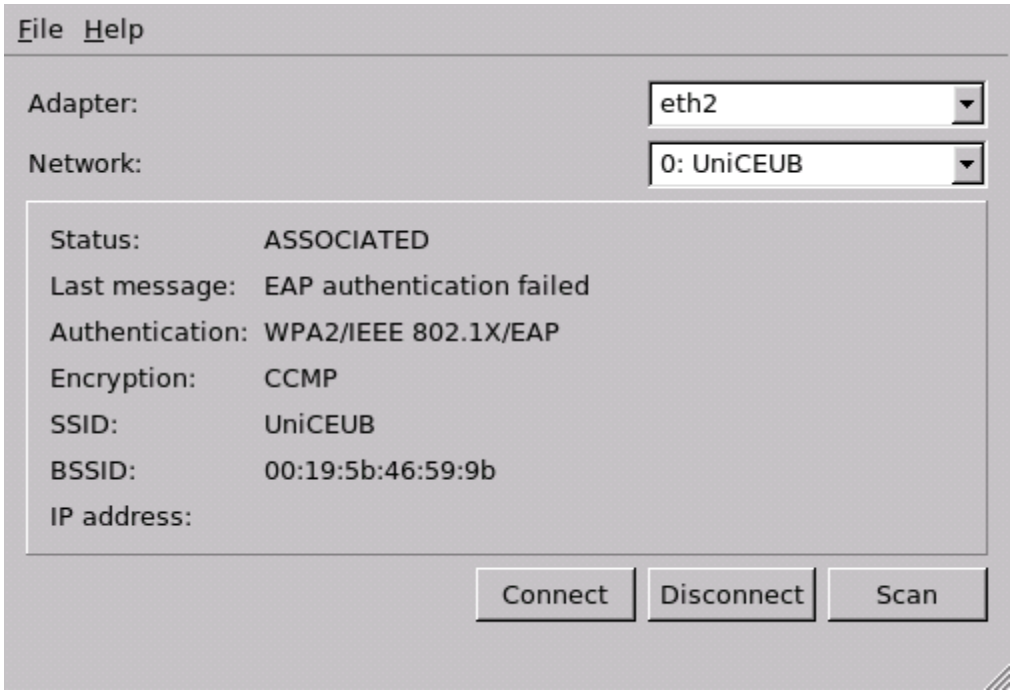

Figura 21 – Teste de Autenticação – Rede sem Certificado Válido (A)

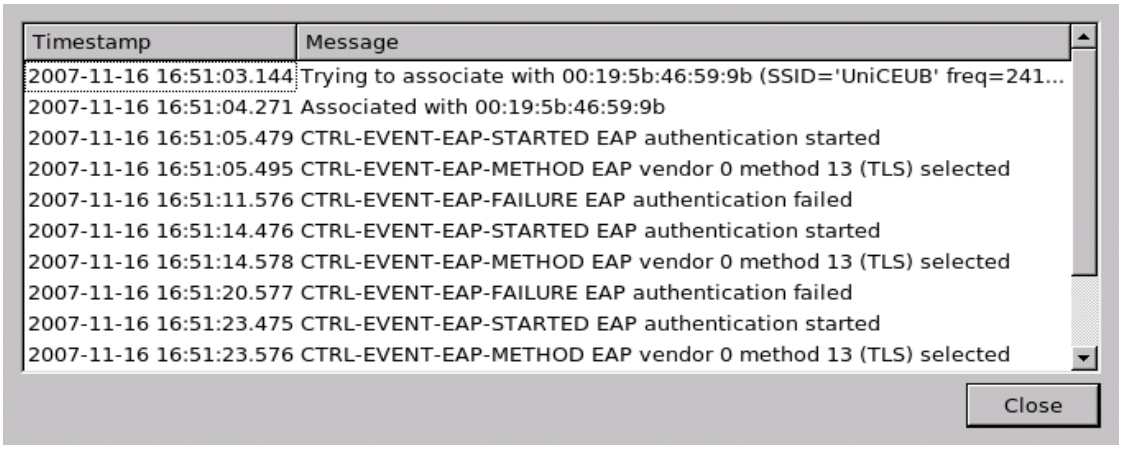

Figura 22 – Teste de Autenticação – Rede sem Certificado Válido (B)

Como pode ser observado na figura acima que mostra o log da conexão, após a associação com o access point, devido a falha da autenticação EAP, o cliente não estabelece uma conexão com a rede sem o certificado válido.

5.3 AVALIAÇÃO GLOBAL DO MODELO DE SOLUÇÃO PROPOSTO

5.3.1 Análise dos Resultados

Na análise dos resultados dos testes realizados é constatado que o modelo proposto conseguiu atingir os objetivos almejados. Disponibilizar uma solução segura que permitisse a realização de autenticação mútua para uma rede WLAN.

Nos diversos cenários testados o modelo reagiu de forma esperada, permitindo garantir a autenticidade de ambas entidades no momento da realização da conexão. Onde nos casos em que uma entidade não possui um certificado válido

a conexão não é estabelecida. Evitando assim que um cliente se conecte a uma rede não-autentica.

5.3.2 Dificuldades e Recomendações

O modelo proposto inicialmente foi planejado para ser um Linux Live USB, isto é um sistema Linux capaz de ser inicializado por um pendrive. Apesar de ter sido implementado desta forma, no momento da preparação do ambiente foi detectado que a maquina virtual *VMware Server 1.0.4 build-56528*, utilizada para rodar o Servidor Autenticador, não podia ser inicializada por um dispositivo USB apenas por um disco rígido, disquete ou CDROM. Isso inviabilizou a intenção do modelo possuir certa mobilidade.

Como recomendações a serem desenvolvidas ao modelo inicial, é sugerido a implementação do modelo proposto em um ambiente que seja possível inicializá-lo através de um pendrive para que a proposta de mobilidade possa ser comprovada.

Outra sugestão ao modelo inicial seria melhorias no código fonte da aplicação de gerenciamento (awcd). Uma possível redução do código e implementação de mecanismos de segurança no fornecimento de senhas de chaves privadas.

5.3.3 Avaliação Global

Na avaliação global do Modelo, o mesmo mostrou-se estável e confirmou a sua facilidade de implementação e gerenciamento, objetivos iniciais propostos, assim permitindo a sua implementação por um usuário desprovido de maiores conhecimentos técnicos.

Como pontos fracos podemos apontar o fato de que a Infra-Estrutura de Chaves Públicas funcione na mesma maquina do servidor RADIUS, ambos se encontram no Servidor Autenticador. Isso pode indicar uma possível falha de segurança onde um *hacker* possa explorar alguma possível vulnerabilidade no servidor RADIUS, e com isso comprometer o Servidor Autenticador e portanto a sua estrutura de certificação digital o que resultaria em uma modelo de autenticação não confiável.

Um dos pontos fortes do modelo é a sua funcionalidade em uma rede WLAN de maior estrutura. Em uma rede WLAN que possui vários *access points*, o Servidor Autenticador ainda é capaz de prover autenticação mútua para toda rede. Devido a sua estrutura de certificação digital centralizada, ele é capaz de emitir e gerenciar os certificados digitais de todos os clientes e *access points* existentes. Outro vantagem do modelo é a possibilidade de ser implementado de forma rápida e pratica, provendo assim maior segurança para uma rede WLAN.

CAPÍTULO 6 – CONSIDERAÇÕES FINAIS

6.1 CONCLUSÃO

A incerteza de um usuário ao se conectar a sua rede sem fio, para realização de tarefas simples como navegar na Internet, torna uma rede exponencialmente vulnerável a ataques de segurança.

A utilização de autenticação mútua é fundamental para certos tipos de comunicação, em especial as redes IEEE 802.11, no intuito de garantir a autenticidade da rede para os clientes e assim prevenir uma vasta gama de problemas de segurança.

Com a utilização de protocolos e software livres existentes o modelo proposto propôs uma solução que através da utilização de certificados digitais permitisse realizar autenticação mútua em uma rede WLAN 802.11.

Os resultados dos testes executados demonstraram que o modelo atingiu seus objetivos de garantir que os pares assegurassem a identidade um do outro, e desta forma neutralizar os problemas gerados pelo ataque Evil Twin.

A proposta de utilizar de recursos e ferramentas disponíveis, de forma prática e viável, para agregar segurança para as redes domésticas mostrou-se bastante eficaz e assim valida que mais novas abordagens como esta possam surgir futuramente.

Concluindo, com o grande avanço tecnológico da comunicação sem fio surge também à necessidade de segurança *wireless*. A lacuna da segurança física das redes sem fio em comparação com as cabeadas torna este tipo especial de rede mais vulnerável, e este projeto contribuiu com uma forma de autenticação mais

segura para as redes WLAN domésticas.

6.2 TRABALHOS FUTUROS

O tema e a solução proposta abordados no presente trabalho podem ser objetos de novas pesquisas no sentido de proporem melhorias para o modelo bem como no desenvolvimento de um novo.

- Desenvolver um hardware específico que possa rodar seguramente o software do modelo proposto ou de um novo desenvolvido.

- Desenvolver um framework de rede que agregue várias funções necessárias, incluindo a do modelo proposto, para uma rede WLAN 802.11 domestica. Funções de *firewall*, roteador, *access point, proxy*.

- Implementar uma solução que realize autenticação mútua em outros tipos de redes sem fio. O *bluetooth* seria um bom exemplo de rede sem fio tipicamente utilizada por usuários domésticos que poderia ser implementado.

REFERÊNCIAS BIBLIOGRÁFICAS

[BEAVER, 2005] BEAVER, Kevin; DAVIDS, Peter T. Hacking Wireless Networks for Dummies. Wiley Publishing, 2001.

[CISCO SYSTEMS, 2002] CISCO SYSTEMS. A Comprehensive Review of 802.11 Wireless LAN Security and the Cisco Wireless Security Suite. Cisco Systems, 2002.

[CITRIX, 2007] CITRIX. How to Use OpenSSL to Create PKCS#12 Certificate Files. Artigo publicado no site Citrix. Disponível em: http://support.citrix.com/ article/CTX106630 – Visitado em Novembro/2007

[COLEMAN, 2006] COLEMAN, David D.; WESTCOTT, David A. CWNA Certified Wireless Network Administrator. Wiley Publishing, 2006.

[DUARTE, 2003] DUARTE, Luiz Otavio. 2003. Monografia (apresentada ao final do curso de Bacharel em Ciência da Computação – Departamento de Ciências de Computação e Estatística do Instituto de Biociências, Letras e Ciências Exatas (IBILCE), Universidade Estadual Paulista Julio de Mesquita Filho, São Jose do Rio Preto.

[GAST, 2002] GAST, Matthew. 802.11 Wireless Networks: The Definitive Guide. O'Reilly, 2002.

[HE] HE, Changhua; MITCHELL, John C. Electrical Engineering and Computer Science Departments Stanford University, Stanford CA.

[INTELLIGRAPHICS, 2007] INTELLIGRAPHICS. Introduction to IEEE 802.11. Artigo publicado no site IntelliGraphics. Disponível em: http://www.intelligraphics.com/articles/80211_article.html – Visitado em Julho/2007

[JIWIRE, 2007] JIWIRE. Consulta realizada sobre hotspots no site JiWire. Disponível

em: http://www.jiwire.com – Visitado em Outubro/2007

[JUNIOR, 2004] JUNIOR, Carlos Alberto de Carvalho; BRABO, Gustavo da Silva; AMORAS, Rômulo Augusto de Sales. 2004. Monografia (apresentada ao final do curso de Bacharel em Ciência da Computação – Centro de Ciências Exatas e de Tecnologia, Universidade da Amazônia, Belém.

[JUNIOR] JUNIOR, Paulo Ditarso Maciel et. al. Avaliando a Sobrecarga Introduzida nas Redes 802.11 pelos Mecanismos de Segurança WEP e VPN/IPSec. Laboratório de Redes de Alta Velocidade, Programa de Engenharia de Sistemas e Computação – COPPE/UFRJ, Rio de Janeiro.

[KARYGIANNIS] KARYGIANNIS, Tom; OWENS, Les. Wireless Network Security. 802.11, Bluetooth and Handheld Devices.

[NETO, 2004] NETO, Roberto Miyano. 2004. Monografia (apresentada ao final do curso de Bacharel em Engenharia de Computação – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro

[OLEXA, 2005] OLEXA, Ron. Implementing 802.11, 802.16, and 802.20 Wireless Networks. Planning, Troubleshooting and Operations. Newnes, 2005.

[PINHEIRO, 2006] PINHEIRO, José Mauricio Santos. As Redes com ZigBee. Artigo publicado no site Mundo Wi-Fi. Agosto 2006, Disponível em: http://www.mundowifi.com.br/artigos/177-as-redes-com-zigbee.html – Visitado em Junho/2007

[PLANET3WIRELESS, 2002] PLANET3WIRELESS. Certified Wireless Network Administrator: Official Study Guide. Planet3 Wireless, 2002.

[RNP, 2005] Rede Nacional de Ensino e Pesquisa. O que é Wi-Max. Noticia publicada no site da RNP, set. 2005. Disponível em acesso em:

http://www.rnp.br/noticias/2005/not-050927-coord.html – Visitado em maio/2007.

[SAMPUBLISHING, 2007] Sam Publishing. Artigo publicado no site SamPublishing. Disponível em: http://www.samspublishing.com/articles/ article.asp?p= 24411&seqNum = 7&rl=1 – Visitando em Junho/2007

[SEARCHMOBILECOMPUTING, 2007] Search Mobile Computing. “ZigBee” - Pesquisa no site Search Mobile Computing utilizando o termo “ZigBee” - Disponível em: http://searchmobilecomputing.techtarget.com/sDefinition/ 0,290660,sid40_gci1127402,00.html?bucket=DEF&topic=299724 – Visitado em Junho/2007

[SEARCHSECURITY, 2007] Search Security – Definição do site Search Security para o termo “Mutual Authentication” - Disponível em: http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci1255857,00.html – Visitado em Outubro/2007

[SEARCHSECURITY, 2007] Search Security – Definição do site Search Security para o termo “RADIUS” - Disponível em: http://searchsecurity.techtarget.com/ sDefinition/0,,sid14_gci214249,00.html – Visitado em Novembro/2007

[TEWS] TEWS, Erik. Breakin Wireless in Less Than 60 Seconds

[TOWNSEND] TOWNSEND, Anthony. To, Wired / Unwired: The Urban Geography of Digital Networks, PhD dissertation, MIT, September 2003.

[WI-FI PLANET, 2002] WI-FI PLANET. WPA: New Protection for 802.11. Noticia publicada no site Wi-Fi Planet. Outubro 2002, Disponível em: http://www.wi-fiplanet.com/news/article.php/1491771 – Visitado em Agosto/2007.

ANEXO I – PACOTES DO LINUX CUSTOMIZADO

Autoconf (2.59) - 904 KB:

Download: http://ftp.gnu.org/gnu/autoconf/autoconf-2.59.tar.bz2

Automake (1.9.6) - 748 KB:

Download: http://ftp.gnu.org/gnu/automake/automake-1.9.6.tar.bz2

Bash (3.1) - 2,475 KB:

Download: http://ftp.gnu.org/gnu/bash/bash-3.1.tar.gz

Bash Documentation (3.1) - 2,013 KB:

Download: http://ftp.gnu.org/gnu/bash/bash-doc-3.1.tar.gz

Berkeley DB (4.4.20) - 7,767 KB:

Download: http://downloads.sleepycat.com/db-4.4.20.tar.gz

Binutils (2.16.1) - 12,256 KB:

Download: http://ftp.gnu.org/gnu/binutils/binutils-2.16.1.tar.bz2

Bison (2.2) - 1,052 KB:

Download: http://ftp.gnu.org/gnu/bison/bison-2.2.tar.bz2

Bzip2 (1.0.3) - 654 KB:

Download: http://www.bzip.org/1.0.3/bzip2-1.0.3.tar.gz

Coreutils (5.96) - 4,948 KB:

Download: http://ftp.gnu.org/gnu/coreutils/coreutils-5.96.tar.bz2

DejaGNU (1.4.4) - 1,056 KB:

Download: http://ftp.gnu.org/gnu/dejagnu/dejagnu-1.4.4.tar.gz

Diffutils (2.8.1) - 762 KB:

Download: http://ftp.gnu.org/gnu/diffutils/diffutils-2.8.1.tar.gz

E2fsprogs (1.39) - 3,616 KB:

Download:http://prdownloads.sourceforge.net/e2fsprogs/e2fsprogs-1.39.tar.gz?download

Expect (5.43.0) - 514 KB:

Download: http://expect.nist.gov/src/expect-5.43.0.tar.gz

File (4.17) - 544 KB:

Download: ftp://ftp.gw.com/mirrors/pub/unix/file/file-4.17.tar.gz

Findutils (4.2.27) - 1,097 KB:

Download: http://ftp.gnu.org/gnu/findutils/findutils-4.2.27.tar.gz

Flex (2.5.33) - 680 KB:

Download: http://prdownloads.sourceforge.net/flex/flex-2.5.33.tar.bz2?download

Gawk (3.1.5) - 1,716 KB:

Download: http://ftp.gnu.org/gnu/gawk/gawk-3.1.5.tar.bz2

GCC (4.0.3) - 32,208 KB:

Download: http://ftp.gnu.org/gnu/gcc/gcc-4.0.3/gcc-4.0.3.tar.bz2

Gettext (0.14.5) - 6,940 KB:

Download: http://ftp.gnu.org/gnu/gettext/gettext-0.14.5.tar.gz

Glibc (2.3.6) - 13,687 KB:

Download: http://ftp.gnu.org/gnu/glibc/glibc-2.3.6.tar.bz2

Glibc LibIDN add-on (2.3.6) - 99 KB:

Download: http://ftp.gnu.org/gnu/glibc/glibc-libidn-2.3.6.tar.bz2

Grep (2.5.1a) - 516 KB:

Download: http://ftp.gnu.org/gnu/grep/grep-2.5.1a.tar.bz2

Groff (1.18.1.1) - 2,208 KB:

Download: http://ftp.gnu.org/gnu/groff/groff-1.18.1.1.tar.gz

GRUB (0.97) - 950 KB:

Download: ftp://alpha.gnu.org/gnu/grub/grub-0.97.tar.gz

Gzip (1.3.5) - 324 KB:

Download: ftp://alpha.gnu.org/gnu/gzip/gzip-1.3.5.tar.gz

Iana-Etc (2.10) - 184 KB:

Download: http://www.sethwklein.net/projects/iana-etc/downloads/iana-etc-2.10.tar.bz2

Inetutils (1.4.2) - 1,019 KB:

Download: http://ftp.gnu.org/gnu/inetutils/inetutils-1.4.2.tar.gz

IPRoute2 (2.6.16-060323) - 378 KB:

Download: http://developer.osdl.org/dev/iproute2/download/iproute2-2.6.16-060323.tar.gz

Kbd (1.12) - 618 KB:

Download: http://www.kernel.org/pub/linux/utils/kbd/kbd-1.12.tar.bz2

Less (394) - 286 KB:

Download: http://www.greenwoodsoftware.com/less/less-394.tar.gz

LFS-Bootscripts (6.2) - 24 KB:

Download: http://www.linuxfromscratch.org/lfs/downloads/6.2/lfs-bootscripts-6.2.tar.bz2

Libtool (1.5.22) - 2,856 KB:

Download: http://ftp.gnu.org/gnu/libtool/libtool-1.5.22.tar.gz

Linux (2.6.16.27) - 39,886 KB:

Download: http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.16.27.tar.bz2

Linux-Libc-Headers (2.6.12.0) - 2,481 KB:

Download: http://ep09.pld-linux.org/~mmazur/linux-libc-headers/linux-libc-headers-2.6.12.0.tar.bz2

M4 (1.4.4) - 376 KB:

Download: http://ftp.gnu.org/gnu/m4/m4-1.4.4.tar.gz

Make (3.80) - 900 KB:

Download: http://ftp.gnu.org/gnu/make/make-3.80.tar.bz2

Man-DB (2.4.3) - 798 KB:

Download: http://savannah.nongnu.org/download/man-db/man-db-2.4.3.tar.gz

Man-pages (2.34) - 1,760 KB:

Download: http://www.kernel.org/pub/linux/docs/manpages/man-pages-2.34.tar.bz2

Mktemp (1.5) - 69 KB:

Download: ftp://ftp.mktemp.org/pub/mktemp/mktemp-1.5.tar.gz

Module-Init-Tools (3.2.2) - 166 KB:

Download: http://www.kerneltools.org/pub/downloads/module-init-tools/module-init-tools-3.2.2.tar.bz2

Ncurses (5.5) - 2,260 KB:

Download: ftp://invisible-island.net/ncurses/ncurses-5.5.tar.gz

Patch (2.5.4) - 183 KB:

Download: http://ftp.gnu.org/gnu/patch/patch-2.5.4.tar.gz

Perl (5.8.8) - 9,887 KB:

Download: http://ftp.funet.fi/pub/CPAN/src/perl-5.8.8.tar.bz2

Procps (3.2.6) - 273 KB:

Download: http://procps.sourceforge.net/procps-3.2.6.tar.gz

Psmisc (22.2) - 239 KB:

Download: http://prdownloads.sourceforge.net/psmisc/psmisc-22.2.tar.gz?download

Readline (5.1) - 1,983 KB:

Download: http://ftp.gnu.org/gnu/readline/readline-5.1.tar.gz

Sed (4.1.5) - 781 KB:

Download: http://ftp.gnu.org/gnu/sed/sed-4.1.5.tar.gz

Shadow (4.0.15) - 1,265 KB:

Download: ftp://ftp.pld.org.pl/software/shadow/shadow-4.0.15.tar.bz2

Sysklogd (1.4.1) - 80 KB:

Download: http://www.infodrom.org/projects/sysklogd/download/sysklogd-1.4.1.tar.gz

Sysvinit (2.86) - 97 KB:

Download: ftp://ftp.cistron.nl/pub/people/miquels/sysvinit/sysvinit-2.86.tar.gz

Tar (1.15.1) - 1,574 KB:

Download: http://ftp.gnu.org/gnu/tar/tar-1.15.1.tar.bz2

Tcl (8.4.13) - 3,432 KB:

Download: http://prdownloads.sourceforge.net/tcl/tcl8.4.13-src.tar.gz?download

Texinfo (4.8) - 1,487 KB:

Download: http://ftp.gnu.org/gnu/texinfo/texinfo-4.8.tar.bz2

Udev (096) - 190 KB:

Download: http://www.kernel.org/pub/linux/utils/kernel/hotplug/udev-096.tar.bz2

Udev Configuration Tarball - 4 KB:

Download: http://www.linuxfromscratch.org/lfs/downloads/6.2/udev-config-6.2.tar.bz2

Util-linux (2.12r) - 1,339 KB:

Download: http://www.kernel.org/pub/linux/utils/util-linux/util-linux-2.12r.tar.bz2

Vim (7.0) - 6,152 KB:

Download: ftp://ftp.vim.org/pub/vim/unix/vim-7.0.tar.bz2

Vim (7.0) language files (optional) - 1,228 KB:

Download: ftp://ftp.vim.org/pub/vim/extra/vim-7.0-lang.tar.gz

Zlib (1.2.3) - 485 KB:

Download: http://www.zlib.net/zlib-1.2.3.tar.gz

ANEXO II – *SCRIPTS* DO *LFS-BOOTSCRIPTS*

checkfs - Verifica a integridade dos sistemas de arquivos antes que sejam montados (exceto os sistemas de arquivos baseados em journal e rede)

cleanfs - Remove os arquivos que não devem ser preservados entre as reinicializações, tais como aqueles em /var/run/ e /var/lock/; recria o /var/run/utmp e remove os arquivos /etc/nologin, /fastboot e /forcefsck, quando presentes

console - Carrega a tabela correta do keymap para o layout de teclado desejado; ajusta também a fonte de tela

functions - Contém as funções comuns, como as de verificação de erro e status, que são usadas por diversos scripts

halt - Desliga o sistema

hotplug - Carrega módulos para dispositivos do sistema

ifdown - Auxiliar ao script de rede para a finalização dos dispositivos de rede

ifup - Auxiliar ao script de rede para a inicialização dos dispositivos de rede

localnet - Define os dispositivos hostname e local loopback do sistema

mountfs - Monta todos os sistemas de arquivos, exceto os que estão definidos como noauto e os baseados em rede

mountkernfs - Monta os sistemas de arquivo virtuais do kernel, como o proc

network - Configura as interfaces de rede, como a placa de rede, e define o gateway padrão (onde aplicável).

rc - O script mestre de controle dos níveis de execução (run-level); é responsável pela execução de todos os demais bootscripts, um a um, em uma seqüência determinada pelo nome das ligações simbólicas que estão sendo processadas

reboot - Reinicializa o sistema

sendsignals - Certifica-se que cada processo está terminado antes que o sistema reinicialize ou desligue

setclock - Ajusta o relógio do kernel para a hora local quando o relógio do hardware não está ajustado com a hora UTC

static - Fornece a funcionalidade necessária para atribuir um endereço IP (Internet Protocol) estático para uma interface de rede
swap - Habilita e desabilita os arquivos e a partição de troca (swap).

sysklogd - Inicia e finaliza os log daemons do sistema e do kernel

template - Um modelo para criar scripts de inicialização padronizados para outros serviços do sistema

udev - Prepara o diretório /dev e inicializa o Udev

APÊNDICE I – ARQUIVO DE CONFIGURAÇÃO DO GRUB

```
# Begin /boot/grub/menu.lst

# By default boot the first menu entry.
default 0

# Allow 30 seconds before booting the default.
timeout 30

# Use prettier colors.
color green/black light-green/black

# The first entry is for LFS.
title LFS 6.2
root (hd0,1)
kernel /boot/lfskernel-2.6.16.27 root=/dev/sda2

title LFS 6.2-custom
root (hd0,1)
kernel /boot/lfskernel-2.6.16.27-custom root=/dev/sda2

title LFS 6.2-kernerl-2.18.8
root (hd0,1)
kernel /boot/kernel-2.6.18.8 root=/dev/sda2
```

APÊNDICE II – CÓDIGO FONTE DA APLICAÇÃO

AWCD

```
#!/usr/bin/perl -X
#
# AWCD - Ferramenta para regerenciamento de certificados digitais e
#        do servico Radius - parte da implementacao do Projeto Final
#        - Autenticacao em Redes Wireless com Certificacao Digital
#
# Versao: 1.0
#
# Autor: Alysson Nishiyama <castor_troynz@hotmail.com>
#

use Switch;

my $p = 1;
my $f = 1;
$| = 1;
my $prompt = "[awcd]".'$ ';

$openssl_dir="/etc/openssl/";
$openssl_conf=$openssl_dir."openssl.conf";

######## Modulo Certificado ##############
#
#

sub cria_ac{

print "Digite o nome da AC a ser criado: ";
my $ac=<STDIN>;
chop($ac);

print "Digite a senha da chave privada: ";
my $senha=<STDIN>;
chop($senha);

my $acdir=$openssl_dir.$ac."/";
my $acnewcertsdir=$acdir."newcerts";
my $acconf=$acdir."openssl.conf";
my $acindex=$acdir."index.txt";
my $ackey= $acdir."cakey.pem";
my $accert= $acdir."cacert.pem";
my $acserial= $acdir."serial";

my $emite_cmd1="mkdir ".$acdir;
my $emite_cmd2="mkdir ".$acnewcertsdir;
my $emite_cmd3="touch $acindex";
my $emite_cmd4="sed -e s/CA_UniCEUB/".$ac."/g $openssl_conf > $acconf";
my $emite_cmd5="openssl req -config $acconf -new -x509 -keyout $ackey -out $accert -days 730 -passout
pass:$senha";
my $emite_cmd6="openssl x509 -in $accert -noout -next_serial -out $acserial";

system("$emite_cmd1");
system("$emite_cmd2");
system("$emite_cmd3");
system("$emite_cmd4");
system("$emite_cmd5");
system("$emite_cmd6");
}
```

```
sub emite_certificado{

print "Digite o nome da AC: ";
my $ac=<STDIN>;
chop($ac);

print "Digite a senha da chave privada da AC: ";
my $acsenha=<STDIN>;
chop($acsenha);

print "Digite o nome do certificado a ser criado: ";
my $certificado=<STDIN>;
chop($certificado);

print "Digite a senha da chave privada: ";
my $senha=<STDIN>;
chop($senha);

my $acdir=$openssl_dir.$ac."/";
my $acconf=$acdir."openssl.conf";
my $clientkey= $certificado."key.pem";
my $clientcert= $certificado."cert.pem";
my $clientp12= $certificado."cert.p12";

my $emite_cmd1="openssl req -config $acconf -new -keyout $clientkey -out newreq.pem -days 730 -passout
pass:$senha";
my $emite_cmd2="openssl ca -config $acconf -policy policy_anything -key $acsenha -out $clientcert -passin
pass:$senha -infiles newreq.pem";
my $emite_cmd3="rm -f newreq.pem";
my $emite_cmd4="openssl pkcs12 -export -in $clientcert -inkey $clientkey -out $clientp12 -clcerts -passin
pass:$senha -passout pass:$senha";
my $emite_cmd5="rm -f $clientcert";
my $emite_cmd6="openssl pkcs12 -in $clientp12 -out $clientcert -passin pass:$senha -passout pass:$senha";
my $emite_cmd7="rm -f $clientp12";

system("$emite_cmd1");
system("$emite_cmd2");
system("$emite_cmd3");
system("$emite_cmd4");
system("$emite_cmd5");
system("$emite_cmd6");
system("$emite_cmd7");
}

sub revoga_certificado{

print "Digite o nome da AC: ";
my $ac=<STDIN>;
chop($ac);

print "Digite o certificado a ser revogado: ";
my $certificado=<STDIN>;
chop($certificado);

my $acdir=$openssl_dir.$ac."/";
my $acconf=$acdir."openssl.conf";
my $revoga_cmd1="openssl ca -config $acconf -revoke $certificado";

system("$revoga_cmd1");
}

sub exporta_certificadoca{

print "Digite o nome da AC: ";
my $ac=<STDIN>;
chop($ac);
```

```
my $exporta_cmd1="cp /etc/openssl/".$ac."/cacert.pem .";

system("$exporta_cmd1");
}

######## Modulo Radius ##############
#

sub configuraip {

print "Digite o IP: ";
my $ip=<STDIN>;
chop($ip);

my $ipfile="/etc/sysconfig/network-devices/ifconfig.eth0/ipv4";
my $ipfiletmp="/etc/sysconfig/network-devices/ifconfig.eth0/ipv4.new";

open (FD, "$ipfile");
open (FDN, ">$ipfiletmp");

foreach (<FD>)
{
if (/IP/)
{
print FDN "IPADDR=".$ip."\n";
}
else
{
print FDN "$_";
}
}

close (FDN);
close (FD);

system("mv $ipfiletmp $ipfile");
system("sh /etc/rc.d/init.d/network restart");

}

sub configurasenha {

print "Digite a senha: ";
my $senha=<STDIN>;
chop($senha);

my $file="/usr/etc/raddb/clients.conf";
my $filetmp="/usr/etc/raddb/clients.conf.new";

open (FD, "$file");
open (FDN, ">$filetmp");

foreach (<FD>)
{
if (/secret\t/)
{
print FDN "\tsecret\t= ".$senha."\n";
}
else
{
print FDN "$_";
}
}

close (FDN);
close (FD);
```

```
system("mv $filetmp $file");

}

sub importacert{

print "Digite o nome da AC: ";
my $ac=<STDIN>;
chop($ac);

print "Digite o certificado a ser configurado: ";
my $certificado=<STDIN>;
chop($certificado);

print "Digite a senha da chave privada: ";
my $senha=<STDIN>;
chop($senha);

my $acdir=$openssl_dir.$ac."/";
my $radiusdir="/usr/etc/raddb/certs/";

my $serverkey= $certificado."key.pem";
my $servercert= $certificado."cert.pem";

my $configura_cmd1="cp $serverkey ".$radiusdir."serverkey.pem";
my $configura_cmd2="cp $servercert ".$radiusdir."servercert.pem";
my $configura_cmd3="cp $acdir/cacert.pem $radiusdir";

my $file="/usr/etc/raddb/eap.conf";
my $filetmp="/usr/etc/raddb/eap.conf.new";

open (FD, "$file");
open (FDN, ">$filetmp");

foreach (<FD>)
{
if (/private_key_password/)
{
print FDN "\t\t\tprivate_key_password = ".$senha."\n";
}
else
{
print FDN "$_";
}
}

close (FDN);
close (FD);

system("mv $filetmp $file");

system("$configura_cmd1");
system("$configura_cmd2");
system("$configura_cmd3");
}

sub iniciaradius {

system("radiusd");

}

sub pararadius {

system("pkill radiusd");
```

```
}

sub verlog {

my $logfile="/usr/var/log/radius/radius.log";
system("tailf $logfile");

}

######## Modulo Shell ##############
#
#

sub shell_exec{
        print "\AWCD\t-\tDigite ajuda para a lista de comandos\n\n";
        volta:{
                print $prompt;
                $cmd2=<STDIN>;
                chop($cmd2);
                @cmd2=split(/ /,$cmd2);
                if ($cmd2 eq "sair") { exit(0) }
                elsif ($cmd2=~/^criaac/) {
                        &cria_ac();
                }
                elsif ($cmd2=~/^emite/) {
                        &emite_certificado()
                }
                elsif ($cmd2=~/^revoga/) {
                        &revoga_certificado()
                }
                elsif ($cmd2=~/^exporta/) {
                        &exporta_certificadoca()
                }
                elsif ($cmd2=~/^configuraip/) {
                        &configuraip()
                }
                elsif ($cmd2=~/^configurasenha/) {
                        &configurasenha()
                }
                elsif ($cmd2=~/^importacert/) {
                        &importacert()
                }
                elsif ($cmd2=~/^iniciaradius/) {
                        &iniciaradius()
                }
                elsif ($cmd2=~/^pararadius/) {
                        &pararadius()
                }
                elsif ($cmd2=~/^reiniciaradius/) {
                        &pararadius()
                        &iniciaradius()
                }
                elsif ($cmd2=~/^verlog/) {
                        &verlog()
                }
                elsif ($cmd2=~/^ajuda/) {
                        print "Lista de comandos:\n\n";
                        print "ajuda\t\t\t\tMostra a lista de comandos.\n";
                        print "sair\t\t\t\tSai do programa.\n\n";
                        print "############## Modulo ICP ###################################\n";
                        print "criaac\t\t\t\tCria AC.\n";
                        print "emite\t\t\t\tEmite certificado.\n";
                        print "revoga\t\t\t\tRevoga certificado.\n";
                        print "exporta\t\t\t\tExporta certificado da AC.\n\n";
                        print "############## Modulo Controle de Acesso #################\n";
                        print "configuraip\t\t\tConfigura IP do servidor.\n";
                        print "configurasenha\t\t\tConfigura senha do Radius.\n";
```

```
                        print "importacert\t\t\tImporta o certificado para o Radius.\n";
                        print "iniciaradius\t\t\tInicia o servico Radius.\n";
                        print "pararadius\t\t\tPara o servico Radius.\n";
                        print "reiniciaradius\t\t\tReinicia o servico Radius.\n";
                        print "verlog\t\t\t\tVisualiza o log do servico Radius.\n\n";

                }
                elsif ($cmd2 eq "") {}
                else { print "comando n□o encontrado\n" }
        }
        goto volta;
}

&shell_exec();
while (1) {};
```

APÊNDICE III - ARQUIVO DE CONFIGURAÇÃO DO WPA_SUPPLICANT

```
network={
        ssid="UniCEUB"
        scan_ssid=0
        mode=0
        proto=RSN
        key_mgmt=WPA-EAP
        auth_alg=OPEN
        eap=TLS
        identity="teemu"
        ca_cert="/etc/wpa_supplicant/cacert.pem"
        client_cert="/etc/wpa_supplicant/clientcert.pem"
        private_key="/etc/wpa_supplicant/clientkey.pem"
        private_key_passwd="ceubclient"
}
```