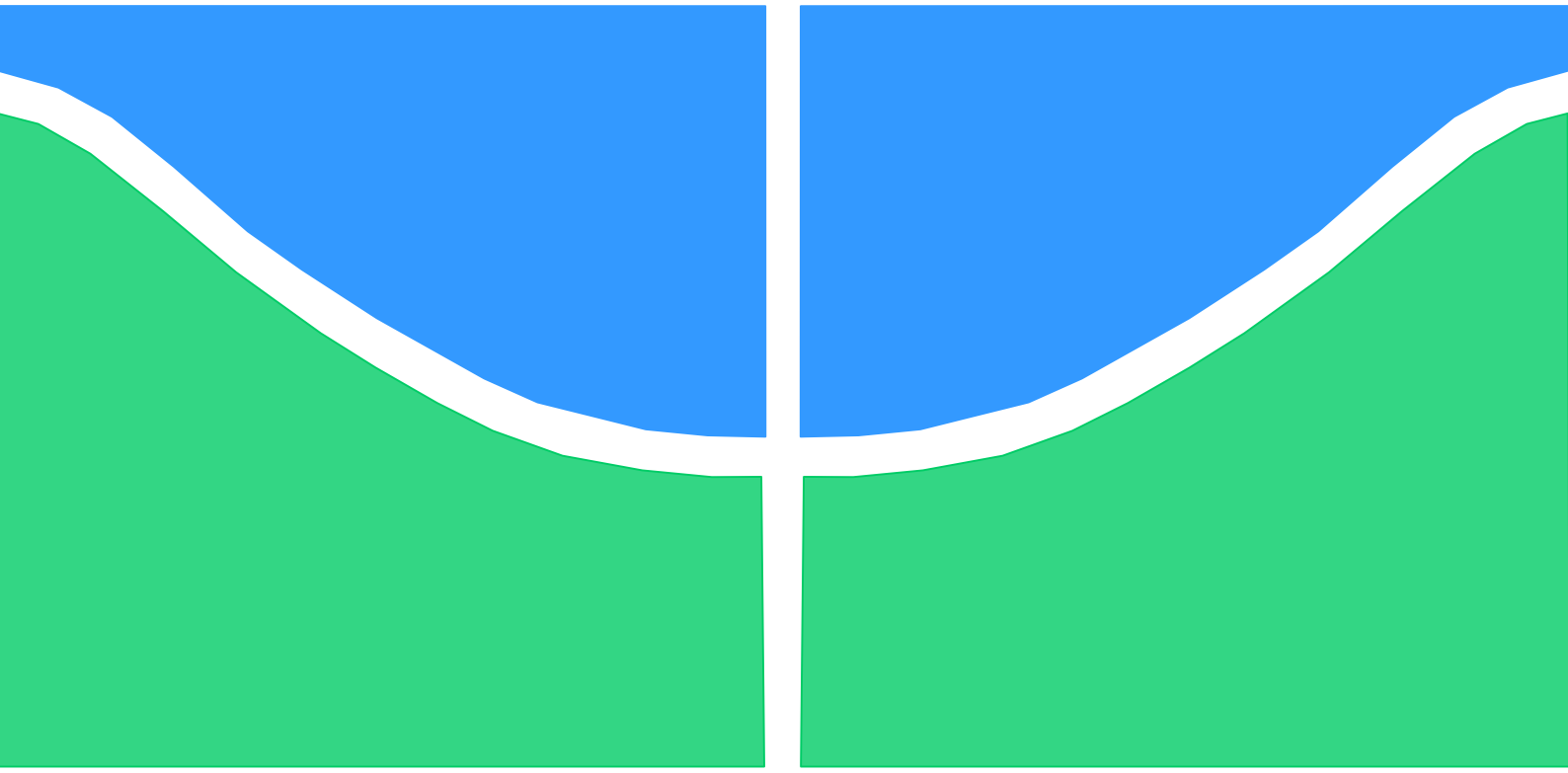

TRABALHO DE GRADUAÇÃO

Análise de Segurança em Redes sem Fio e Proposta de Solução para o Laboratório da Engenharia de Redes de Comunicação

Pedro Paulo Martins dos Santos

Brasília, Dezembro de 2015

UNIVERSIDADE DE BRASILIA

FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

UNIVERSIDADE DE BRASILIA
Faculdade de Tecnologia
Departamento de Engenharia Elétrica

TRABALHO DE GRADUAÇÃO

Análise de segurança em Redes sem Fio e proposta de solução para o Laboratório da Engenharia de Redes de Comunicação

Pedro Paulo Martins dos Santos

Relatório submetido como requisito parcial para obtenção do grau de Engenheiro de Redes de Comunicação

Banca Examinadora

Prof. Georges Daniel Amvame Nze, Dr., ENE/UnB
Orientador

Prof. Ugo Dias, Dr., ENE/UnB
Examinador

Prof. Diego Martins de Oliveira, Especialista, IFB Taguatinga
Examinador

Dedico este trabalho a minha família, a minha namorada, aos amigos de curso e aos professores que fizeram parte da minha formação.

Agradecimentos

Para a realização deste trabalho, muitas pessoas tiveram sua parcela e sem elas eu não conseguiria tê-lo finalizado e, por isso, faço alguns agradecimentos.

Agradeço ao Professor Georges Daniel Amvame Nze, por ter sido um orientador que sempre esteve à disposição quando precisei de ajuda; por ter dado apoio, incentivo, sugestões e críticas valiosas. Seu interesse e dedicação em realizar este projeto foram importantes para a conclusão do projeto.

Agradeço aos meus pais por terem incentivado e dado o apoio necessário para mim desde o início de minha educação até o ensino superior. Graças à dedicação de ambos é que devo grande parte do que pude realizar até hoje.

A minha namorada, por ter sido uma grande incentivadora e companheira durante a realização deste projeto.

Aos meus amigos de curso, por terem sido grandes companheiros ao longo da graduação, dando a base necessária para que pudéssemos seguir em frente durante ela.

E também aos professores da Universidade de Brasília, que tem sua parcela importante em minha graduação, trazendo o conhecimento, as ideias e o ensino necessário para a formação no curso de Engenharia de Redes de Comunicação.

Finalmente agradeço a Deus, pois sem Ele nada teria sido possível.

Pedro Paulo

RESUMO

Este projeto foi feito com o propósito de servir como um guia para se ter uma configuração segura quando se trata de WLAN. Especialmente nesse caso, é proposta uma solução segura para o Laboratório da Engenharia de Redes de Comunicação. Ainda que se tenham hoje soluções suficientemente seguras, é comum encontrar nos mais diversos locais, seja empresarial ou domiciliar, configurações de rede que tenham brechas para sofrer ataques e comprometer a rede. Sendo assim, neste projeto tem-se a análise de soluções para redes sem fio e, também, a análise de ataques conhecidos que podem ser lançados contra as mesmas. Para a análise da segurança dos tipos de redes, é usada a distribuição Kali Linux, a mais usada atualmente para testes de intrusão e auditoria em segurança. Busca-se aqui mostrar o que pode ser atingido quando se tem uma quantidade mínima de recursos, mostrando os riscos que se corre ainda que não haja atacantes com equipamentos sofisticados. Como resultado final, é apresentada a solução que melhor se protege contra os tipos mais comuns de ataques.

ABSTRACT

This project was made with the purpose to be a guide about secure configuration in WLAN. Although nowadays there is enough secure solutions, it is common to find in several places, either be residential or enterprise, network configurations that have flaws exploitable, compromising the network. Therefore, in this project, the analysis of wireless network solutions is made, and the analysis of known attack that can be launched against it. For the analysis of wireless security types, it is used the Kali Linux distribution, the most used currently for intrusion tests and security auditing. It is sought here to show what can be achieved when there is a minimal amount of resources, highlighting existent risks even if there is not sophisticated devices to perform attacks. As a result, it is presented the solution that bests mitigate the most common types of attacks.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

Siglas

AES	*Advanced Encryption Standard*
AP	*Access Point*
ARP	*Address Resolution Protocol*
BSS	*Basic Service Set*
CCMP	*Counter Mode with Cipher Block Chaining Message Authentication Code Protocol*
CRC	*Cycle Redundancy Check*
CSMA/CA	*Carrier Sense Medium Access/ Collision Avoidance*
DHCP	*Dynamic Host Configuration Protocol*
DoS	*Denial of Service*
EAP	*Extensible Authentication Protocol*
GTK	*Group Transient Key*
IEEE	*Institute of Electrical and Electronics Engineers*
IP	*Internet Protocol*
IV	*Initialization Vector*
LDAP	*Lightweight Directory Access Protocol*
LEAP	*Lightweight EAP*
MAC	*Medium Access Control*
MD5	*Message Digest 5*
MITM	*Man-in-the-middle*
MS-CHAPv2	*Microsoft Challenge Handshake Authentication Protocol version 2*
NAT	*Network Address Translation*
OSA	*Open System Authentication*
PEAP	*Protected EAP*
PMK	*Pair Master Key*
PSK	*Pre-Shared Key*
PTK	*Pair Transient Key*
RC4	*Rivest Cipher 4*
RSN	*Robust Security Network*
RSNA	*Robust Security Network Association*
SKA	*Shared Key Authentication*

TKIP	*Temporal Key Integrity Protocol*
TLS	*Transport Layer Security*
TTLS	*Tunneled TLS*
VoIP	*Voice over IP*
WEP	*Wired Equivalent Privacy*
WLAN	*Wireless Local Area Network*
WPA	*Wi-Fi Protected Access*

1 INTRODUÇÃO

Este capítulo apresenta considerações gerais preliminares relacionadas ao contexto do trabalho, aos objetivos e, finalmente, aos assuntos abordados ao longo do mesmo.

1.1 MOTIVAÇÃO

Hoje em dia, as redes sem fio são usadas largamente no dia-a-dia das pessoas. Seja no trabalho, seja em casa ou seja em lojas de atendimento ou em aeroportos, há muitas pessoas que desejam estar conectadas à internet. Devido à natureza do meio de transmissão das redes sem fio (aéreo), temos nossas informações propagadas pelo ambiente sem que possamos impedir que alguém capte os dados que são transmitidos. Naturalmente, há uma grande preocupação quanto aos métodos de se prover integridade e privacidade no uso destas redes. Ainda, é necessário levar em conta a proteção do cliente e da rede no momento da autenticação, fazendo com que tanto usuário como autenticador saibam que estão lidando com alguém legitimo.

A preocupação com a proteção nas redes *wireless* se deu desde o início do padrão IEEE 802.11, referente às redes sem fio, com o uso do *WEP* (*Wired Equivalent Privacy*). Como esse padrão não conseguiu fornecer a segurança necessária, surgiram outros métodos de segurança, o *WPA* (*Wi-Fi Protected Access*) e *WPA2*, que buscaram suprir as necessidades deixadas inicialmente. Com a adoção dos novos padrões, houve inclusive a divisão entre os métodos dependendo do ambiente usado para comportar a rede: ambientes empresariais (*WPA/WPA2-Enterprise*) e domiciliares (*WPA/WPA2-Personal*).

Ainda que se tenha buscado trazer o melhor da segurança para as redes sem fio, sabe-se que existem riscos que podem comprometer a rede, e os riscos são vários: uso de métodos de segurança já conhecidos por serem inseguros, roteadores configurados com senhas padrão ou senhas de fácil descoberta, além da falta de cuidado de usuários que acabam por expor a sua própria rede ao propagar informações quando não deveriam.

Assim, espera-se que o trabalho seja um guia para estabelecer-se uma rede segura de acordo com as melhores ferramentas e práticas disponíveis atualmente, visto que devido à falta de informação hoje em dia, muitas redes deixam vulnerabilidades que poderiam ser sanadas facilmente.

1.2 OBJETIVOS

Neste trabalho há o estudo e análise dos métodos atuais de segurança bem como os ataques existentes contra as mesmas. Portanto, as melhores práticas ao se criar uma rede sem fio são relatadas, sendo uma referência para aqueles que desejam saber o melhor método de segurança ao ser usado na criação de sua rede.

1.3 TRABALHOS RELACIONADOS

Quando se trata de segurança em redes sem fio, diversos trabalhos têm sido feitos visando verificar as vulnerabilidades ou até mesmo evitar que sejam exploradas. Eles são levados em conta para a proposta final de uma solução *WLAN (Wireless Local Area Network)*.

Em [1], é tratada a segurança do *WPA2-Enterprise*, visto como o método de segurança mais seguro. Ainda que fizessem uso do *EAP-PEAP* (autenticação considerada segura), foram capazes de usar o ataque do falso AP e quebra a chave de alguns usuários sem que percebessem (vulnerabilidade do *MS-CHAPv2* explorada), mostrando que um determinado atacante pode comprometer redes seguras.

Os algoritmos de encriptação também mostram vulnerabilidade, em [2], diversos ataques foram testados contra o método, sendo capazes de realizar ataque de negação de serviço (*DoS*), ataque de injeção de pacotes que possibilitou fazer *portscan* nos usuários da rede e também um ataque que fizesse a decriptação de pacotes arbitrários. Até mesmo o algoritmo *CCMP*, considerado o padrão mais seguro para encriptação em redes sem fio, pode apresentar vulnerabilidades. [3] explora a possibilidade do ataque *TMTO* (*Time memory Trade-Off*), no qual há a tentativa de comprometer o algoritmo de cifra usado.

Em outros trabalhos, trata-se ainda do uso de mecanismos que possibilitem a melhoria de segurança da rede. Recursos como filtro por *MAC*, lista de controle de acesso (*ACL*) ou ainda a não divulgação do *SSID* podem servir para atenuar os riscos, mas não para evitar um atacante habilidoso [4]. Em [5], a tentativa é mais eficaz: um método é proposto para evitar que ataques de desautenticação aconteçam, um ataque bastante explorado contra os métodos de segurança (*WPA2 Personal* e *Enterprise*).

1.4 ESTRUTURA DO TEXTO

No capítulo 2 é mostrada a fundamentação teórica necessária ao entendimento das redes envolvidas na análise e dos ataques realizados. O capítulo 3 detalha a metodologia dos cenários usados para configuração assim como usados para realização dos ataques. Resultados e análises são mostrados no capítulo 4, encerrando com as conclusões no capítulo 5.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo está presente todo o conteúdo sobre os temas relevantes ao trabalho para entendimento das tecnologias envolvidas.

2.1 IEEE 802.11 - WEP (WIRED EQUIVALENT PRIVACY)

Para que se possa entender segurança em *Wireless LAN (WLAN),* é necessário compreender a evolução dos principais métodos de segurança, incluindo como serviços de privacidade e integridade são providos. A necessidade de privacidade é altamente desejada, já que o meio de transmissão inerente em *WLAN* é compartilhado. Assim, assumindo que dados privados serão transmitidos em uma rede na qual outros podem capturar dados, é de extrema importância a garantia de que estes quadros de dados são protegidos com fortes controles de segurança.

WEP foi criado a fim de atingir um nível de segurança similar a rede cabeada. Com o uso de autenticação simples e técnicas de encriptação, os responsáveis pelo padrão 802.11 acreditavam que o objetivo da privacidade seria atingido.

Antes de mais nada, vale a pena mencionar os dois tipos de autenticação definidos no padrão IEEE 802.11 (1999): sistema aberto (*open system - OSA*) e chave compartilhada (*shared key - SKA*). O primeiro basicamente usa um algoritmo de autenticação "nulo". Já o segundo requer o uso da privacidade *WEP* para o funcionamento. Antes que uma estação seja capaz de se associar a uma *WLAN*, deve completar com sucesso a autenticação, que nesse contexto se refere a *open system* e *shared key*.

As trocas de pacotes, antes de qualquer troca de dados significante, para associação e autenticação são ilustradas a seguir:

Figura 2-1 Autenticação *WEP* com chave compartilhada.[6]

A autenticação *open system* basicamente tem dois passos: a estação envia um quadro de requisição de autenticação e o *Access Point (AP)* responde com um quadro de resposta.

A autenticação que faz uso de chave compartilhada faz uso de alguns passos a mais. Antes de a estação estar completamente autenticada, deve provar ao *AP* que é um dispositivo autorizado a fazer a tentativa de associação ao mesmo. Os passos são os seguintes: primeiro, a estação envia um *ARF* (*Authentication Request Frame*) mostrando a intenção de se juntar a *WLAN*; segundo, o *AP* responde com um desafio (*challenge sequence*) que é gerado usando um gerador de números pseudoaleatórios *WEP* com o tamanho de 128 *bytes*; a estação encripta o desafio com a chave compartilhada e estática *WEP*; finalmente, o *AP* decripta o quadro e verifica a carga. Se as chaves usadas forem as mesmas, resultará no sucesso da autenticação.

Percebe-se que por enviar o texto puro e o texto encriptado no meio sem fio, a autenticação *SKA* expõe mais vulnerabilidades até mesmo que o *OSA*, mostrando que o uso de *WEP* seria mais seguro com *OSA* do que com *SKA*.

WEP provê mecanismos tanto para autenticação de clientes (*shared key*) como para a encriptação dos dados de quadros *wireless*.

No processo de encriptação do *WEP*, várias informações são usadas como entrada. Temos os textos puros (carga útil de dados) a serem transmitidos nos quadros, o vetor de inicialização (*IV*), a chave compartilhada (privada).

Primeiramente, para gerar o *keystream WEP*, usa-se uma combinação da chave *WEP*, que pode ser de 40 bits ou de 104 bits, com o *IV*, que tem o tamanho de 24 bits. Essa combinação é usada no algoritmo do *RC4*, gerando o *keystream* para a encriptação. Como o *IV* tem um número de combinações limitado, surgem espaços para ataques que exploram esse reuso da chave estática em combinação com *IVs* que podem ser reusados.

Com o *keystream WEP* gerado, podemos combiná-lo com a carga útil de dados, que vem acoplada com o *ICV* (*Integrity Check Value*), usado para identificar se a integridade da mensagem foi mantida. No caso, o algoritmo *CRC-32* é usado. Finalmente, a operação XOR é usada para combinar o *keystream* com a carga útil + *ICV*. O esquema de encriptação *WEP* é mostrado na Figura 2-2.

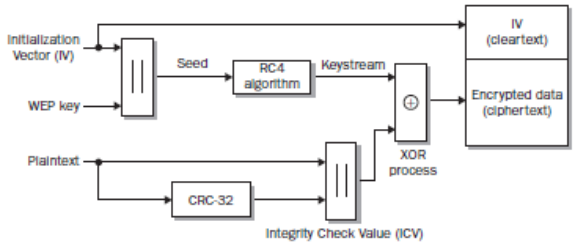

Figura 2-2 Fluxograma de encriptação *WEP*. [7]

WEP tem a opção de uso de chaves estáticas ou dinâmicas. Sabe-se que o uso de chaves estáticas carrega com si alguns perigos que expõem a segurança da rede. Usando-se sempre a mesma chave, o atacante que deseja sabe-la terá maior chance de obter sucesso. Como *IVs* podem ser reusados devido ao seu número limitado de combinações, podemos ter *keystreams* repetidas, deixando um tráfego que pode ser explorado por diversas ferramentas de ataque. Os tamanhos relativamente pequenos das chaves (40 ou 104 bits) deixa ainda uma faixa de combinações de chave de pequena extensão.O *ICV* usado (*CRC-32*) é relativamente fraco e a carga útil de dados poderia ser modificada sem que se percebesse.

Visando amenizar os diversos problemas do *WEP*, pode se escolher o uso de chaves dinâmicas *WEP*. Aqui, as chaves variam de acordo com o usuário e de acordo com a sessão. A ideia é que se mude regularmente a chave usada para uma mesma sessão de um usuário. Dessa forma, evita-se que um atacante consiga obter o número necessário de dados para uma mesma chave a fim de realizar a quebra da chave.

2.2 WI-FI PROTECTED ACCESS (WPA)

Após o lançamento do padrão IEEE 802.11, um novo grupo foi formado visando estabelecer um novo padrão que tratasse os problemas e fraquezas do *WEP*. Para tanto, seria estabelecido o padrão 802.11i. Por haver demora em estabelecimento de novos padrões, uma medida foi adotada provisoriamente e foi lançado o *WPA* (*Wi-Fi Protected Access*). O *WPA* chega com uma melhora no método de encriptação (*TKIP – Temporal Key Integrity Protocol*) além de prover a capacidade de dois métodos de autenticação: baseado em senha (*WPA-Personal*, para ambientes *SOHO – Small Office Home Office*) e o *802.1X/EAP* (*WPA-Enterprise*, para ambientes corporativos).

Vale lembrar que o *WPA* funcionou como um upgrade de *firmware* sendo executado no mesmo *hardware* usado pelo *WEP RC4*. Um número considerável de melhorias foi implementado no *WPA*. Para começar, a chave de 128 bits *WEP* usada no processo para a construção do *keystream* é gerada através da combinação de três elementos: o endereço *MAC* do transmissor, uma chave temporária e um contador sequencial *TKIP* (*TSC*). Isso garante que cada usuário, a cada quadro transmitido, estará usando um fluxo de chaves diferente. Também, por ser usado um contador sequencial por quadro, o receptor terá a chance de descartar quadros que tenham um número de sequência igual ou menor ao número do último quadro que tenha recebido corretamente. Esse fato também dá a oportunidade de se detectar ataques de repetição contra a *WLAN*. Outra importante melhoria é o uso de um *IV* do tamanho de 48 bits, contra os

24 bits usado no *WEP*. O número possível de *IVs* sobe de 17 milhões para 281 trilhões, tornando virtualmente inviável percorrer todos os valores de *IV* possíveis. Além disso, *TKIP* faz uso do *MIC* (*Message Integrity Code*) para checar a integridade das mensagens. Esse algoritmo é mais confiável que o anteriormente usado *CRC-32*, ajudando a detectar quadros adulterados.

O *WPA* foi projetado resultando em dois tipos de implementação voltados a atender tanto clientes de *SOHO* como clientes de largas companhias. Os dois tipos carregam as melhorias citadas anteriormente e se diferenciam pelos métodos de autenticação usados para permitir o acesso de usuários. Esses dois tipos se tratam do *WPA Personal* e do *WPA Enterprise*.

2.2.1 WPA-PERSONAL

O *WPA Personal* veio para suprir a necessidade daqueles que não contam com muita infraestrutura para fornecer serviços como *DHCP, DNS* e servidores de autenticação. Foi projetado para ser acessível para a comunidade como um todo, permitindo que os usuários deste tipo de *WPA* pudessem beneficiar das melhorias sem precisar investir em novos equipamentos requisitados para a sofisticada autenticação usada no *802.1X/EAP*.

Esse tipo de segurança *WPA* é chamado também de *WPA-PSK* (*Pre-shared key*) já que a senha é compartilhada por todos os usuários da rede. Foi projetado para atender usuários de *SOHO,* onde a rede é limitada a poucos *APs* e poucos usuários. *WPA-PSK* foi projetada, assim, para balancear a questão da usabilidade e da segurança. Ainda que remeta ao *WEP*, apresenta suas mudanças: a senha não é mais limitada a 13 caracteres alfanuméricos além de oferecer mais opções ao tipo de senha escolhida.

Um método de ataque contra o *WPA-PSK* seria a obtenção ou adivinhação da senha sendo usada, no qual haveria um falso cliente conectando a rede, obtendo acesso aos recursos da mesma. Com objetivo de atenuar tal possibilidade, algumas boas práticas são recomendadas. Primeiramente, não se deve usar senhas que sejam palavras de dicionários ou facilmente deduzidas. Existem ferramentas capazes de capturar quadros *WPA* de autenticação, possibilitando a execução do ataque de dicionário, na tentativa de roubar senhas fracas. Ainda, deve-se levar em consideração a mudança regular da senha usada na rede, além de manterem o registro de aparelhos da rede que tenham sido roubados ou perdidos, buscando não oferecer brechas para que outros consigam acessar a rede sem fio.

2.2.2 WPA-ENTERPRISE

Usuários de empresas tendem a terem mais precaução quanto à segurança de suas redes e, por isso, necessitam de controles mais robustos de segurança, principalmente no que diz

respeito ao método de autenticação. O padrão *WPA Enterprise* é capaz de atender a essa demanda na qual a rede é composta por inúmeros usuários, fazendo uso do *802.1X*/*EAP* para controle de acesso baseado em porta e autenticação extensível.

Ao contrário do *WPA-PSK*, que se baseia em uma mesma senha, no *WPA Enterprise* há a implementação de um cenário que pode fazer uso de diversos tipos de credenciais, tais como: *login* de usuário, senhas, *tokens* e certificados. Para suportar tal nível de autenticação, componentes adicionais precisam ser colocados na infraestrutura. Os três principais componentes são: o cliente (suplicante), o ponto de acesso (autenticador) e o servidor de autenticação (tipicamente, *RADIUS (Remote Authentication Dial-In User Service*). Abaixo há a ilustração do fluxo *802.1X*/*EAP*:

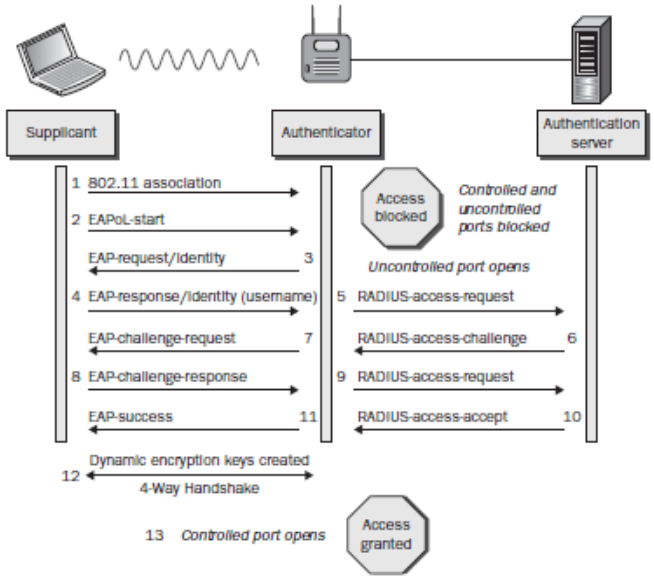

Figura 2-3 Autenticação *802.1X*/*EAP*. [7]

Suplicante: pode ser um laptop, um celular que tenha as credenciais apropriadas para autenticação, já que *WPA Enterprise* não se baseia em senha compartilhada.

Autenticador: é o *AP* mediador entre o servidor de autenticação e o suplicante. É responsável também por restringir o fluxo de tráfego via controle de acesso baseado em porta (802.1X) até que o cliente tenha sido autenticado.

Servidor de Autenticação: responsável por verificar as credenciais do suplicante. Tipicamente é um servidor *RADIUS*, ainda que possa ser de outro tipo com outros protocolos.

802.1X: este controle de acesso baseado em porta é que faz com que o suplicante consiga apenas ter contato com o servidor de autenticação visando sua autenticação. Até que o cliente tenha finalizado o processo, a única comunicação se dará por este tráfego.

EAP (Extensible Authentication Protocol): como o nome sugere, provê vários mecanismos para que o suplicante apresente as credenciais válidas ao servidor de autenticação.

Logo, *EAP* pode fazer uso de túneis *TLS (Transport Layer Security)*, certificados digitais, *tokens* ou simplesmente nome de usuário e senha. Isso permite que haja diversos modos de se utilizar o *EAP* de acordo com as necessidades da rede.

Dentre as principais comparações entre os dois métodos *WPA* (*Personal* e *Enterprise*), podemos notar as seguintes diferenças:

- *WPA Enterprise* permite o uso de credenciais únicas para cada usuário, uma configuração de usuário centralizada, autenticação mútua (cliente e servidor de autenticação), além de ser projetado para grandes empresas;
- *WPA Personal* tem uma maior facilidade e menor custo para a sua implementação, sendo que o uso do algoritmo *RC4*, do *TKIP*, do *MIC* e a proteção contra ataques de repetição estão presentes nos dois métodos.

2.3 802.11i (WPA2)

A emenda 802.11i foi ratificada e publicada como IEEE Std. 802.11i-2004, e padronizou forte encriptação e métodos melhores de autenticação. Essa emenda definiu uma rede de segurança robusta (*RSN*). O maior aspecto abordado por esse padrão, chamado também de *WPA2*, que o diferencia em comparação ao *WPA* é o seguinte:

- Reforço da privacidade dos dados: as necessidades com a confidencialidade foram supridas com a adoção do método de encriptação *Counter mode with Cipher Block Chaining Message Authentication Code Protocol (CCMP)*, que faz uso do algoritmo *Advanced Encryption Standard (AES)*. Como método opcional de encriptação, pode-se usar o *TKIP*, que usa o algoritmo de cifras *RC4* e é basicamente uma melhoria do *WEP*.

A seguir, na Tabela 2-1, mostra-se os padrões de segurança e suas certificações:

802.11 Standard	Wi-Fi Alliance Certification	Authentication Method	Encryption Method	Cipher	Key Generation
802.11 legacy		Open System or Shared Key	WEP	RC4	Static
	WPA-Personal	WPA Passphrase (also known as WPA PSK and WPA Pre-Shared Key)	TKIP	RC4	Dynamic
	WPA-Enterprise	802.1X/EAP	TKIP	RC4	Dynamic
802.11- 2007	WPA2-Personal	WPA2 Pass-phrase (also known as WPA2 PSK and WPA2 Pre-Shared Key)	CCMP (mandatory)	AES (manda-tory)	Dynamic
			TKIP (optional)	RC4 (optional)	
802.11-2007	WPA2-Enterprise	802.1X/EAP	CCMP (mandatory)	AES (manda-tory)	Dynamic
			TKIP (optional)	RC4 (optional)	

Tabela 2-1 Comparação entre padrões de segurança 802.11. [7]

O padrão IEEE 802.11-2007 definiu o que é conhecido como *RSN* e associação *RSN* (*RSNA*). Duas estações devem estabelecer um procedimento para autenticar e se associarem uma a outra assim como criar chaves dinâmicas de encriptação através de um processo conhecido como *4-Way Handshake*. Esta associação entre duas estações é conhecida como *RSNA*.

Assim, uma RSN é uma rede que permite a criação apenas de *RSNAs*. O padrão anteriormente citado permite também a criação de *Pre-RSNAs*, na qual medidas de segurança do antigo padrão 802.11 são usadas em um mesmo *BSS* (*Basic Service Set*). Essas redes mistas são chamadas de *Transition Security Network* (*TSN*), sendo capazes de adotar medidas de segurança definidas para as *RSNs* bem como de fazer uso padrões antigos como o *WEP*.

2.3.1 HIERARQUIA DE CHAVES

O padrão IEEE 802.11i *RSNA* tem dois tipos de hierarquia de chaves que são usadas para proteger ou tráfego *unicast* ou tráfego *multicast /broadcast*. Enquanto que para os quadros *unicast* temos a hierarquia de chave *pairwise*, para os *multicast/broadcast* temos a hierarqua de chave *group*. O objetivo que se busca com hierarquia de chaves é a busca pela constante mudança da chave usada. Fazendo uso de uma chave *master* para gerar chaves transientes, consegue-se evitar os ataques contra a criptografia usada [7].

Hierarquia de chave PAIRWISE

A chave que permanece no topo da hierarquia de chaves, tanto para o processo *802.1X*/*EAP* como para a senha *PSK* é a *Pairwise Master key (PMK)*. Essa chave é composta de 256 bits e é a base para que se gere a chave transientes (*PTK*). O tamanho da *PTK* irá depender do tipo de encriptação sendo usado (*CCMP* ou *TKIP*). É dessa chave que se geram as outras três:

- *KCK* (*Key confirmation key)* – são os primeiros 128 bits da *PTK* e é usada para fornecer serviços de integridade para os dois *handshakes*: *4-Way* e Group;
- *KEK* (*Key encryption key)* – tem também o tamanho de 128 bits e é usada para os serviços de confidencialidade durante os *handshakes*, encriptando o campo *Key Data*;
- *TK* (*Temporal key*) – tem vários papeis em vários mecanismos de segurança.

Hierarquia de chave GROUP

Esta hierarquia é composta de dois componentes apenas, a *Group Master Key (GMK)* e a *Group Temporal Key (GTK)*. Usadas apenas para proteger tráfego *multicast/broadcast*, essas

chaves são geradas pelo *AP* e distribuídas de forma segura. A *GTK* é disseminada usando o *handshake* apropriado, garantindo a confidencialidade dos tráfegos em questão.

Distribuição de Chaves

A distribuição de chaves é um processo sensível e deve ter um caminho seguro para que possa ser enviado para os participantes. Os processos de distribuição de múltiplas chaves estão definidos no padrão 802.11i e são categorizados em três áreas: o *4-Way handshake*, o *group key handshake* e o *STAKey handshake*. Para facilitar os três processos, quadros *EAPOL-Key* são usados para desempenhar serviços relacionados a chaves.

Cada um dos três processos serve para verificar que as estações participantes tem certa informação e/ou transferir chaves atualizadas entre os dispositivos. O *4-Way handshake* é usada para verificar a veracidade da *PMK* e para transferir seguramente a *GTK*. O *Group key handshake* atualiza o valor da *GTK*, enquanto o *STAKey Handshake* provê os meios de transmissão de informações *STAKey*.

Os quadros *EAPOL-Key* desempenham funções de verificação de chaves, atualização de chaves e distribuição das mesmas. Esses quadros têm diversos campos, dentre os quais pode-se citar: contador de repetição (número de sequência para detectar ataques de repetição contra os quadros), chave *RSC* (usada para identificar a chave *GTK* atual) e chave *MIC* (usada para checar a integridade dos quadros).

2.3.2 4-WAY HANDSHAKE

Parte essencial do *RSNA*, esse processo inclui a comunicação entre o suplicante (cliente) e o autenticador (*AP*) para que se desempenhe uma série de serviços de segurança para a rede sem fio. Este *handshake* é usado para verificar as *PMKs* atuais, distribuindo-as para os clientes. Também, pode ser usado para criar *PTKs*, comunicar uma *GTK* e verificar configurações de segurança.

Antes de que este processo ocorra, dois outros precisam ter acontecido:

- A autenticação e associação de baixo nível do padrão IEEE 802.11 (*open system*). É aqui que se trocam as informações *RSN EI*;
- Por segundo, a autenticação que é desempenhada pelo uso do processo *802.1X*/*EAP* ou do processo *PSK*. Estabelecida a autenticação, a construção da *PMK* é garantida.

O *4-way handshake* é literalmente composto de quatro mensagens entre o suplicante e o autenticador. O processo é ilustrado na Figura 2-4 e explicado em seguida:

Na primeira mensagem, o autenticador transmite uma mensagem contendo um *ANonce*, sem o *MIC* para o suplicante checar a integridade. É possível verificar o contador de repetição, validando a sequência do quadro. O suplicante cria o *SNonce* e usa a informação do mesmo para gerar a *PTK*.

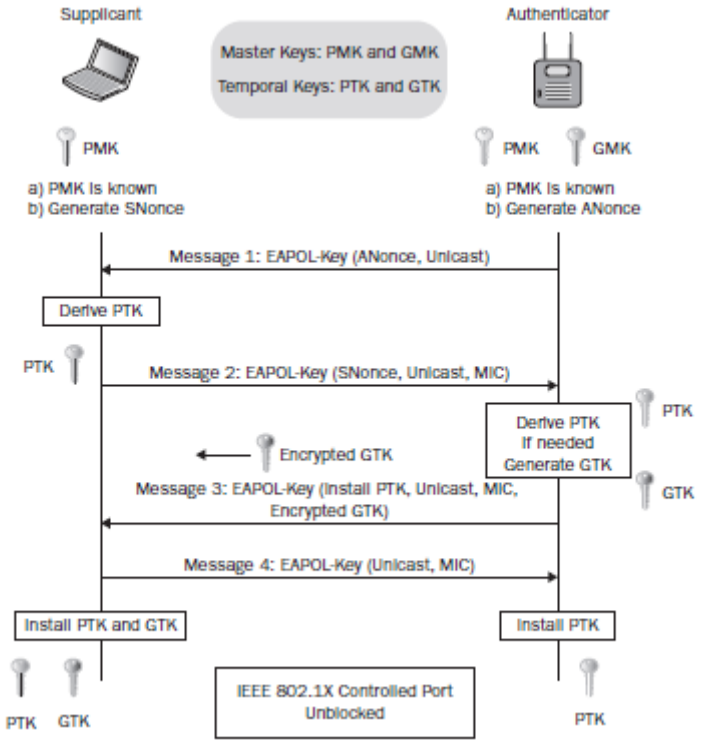

Figura 2-4 Autenticação *4-Way Handshake*.[7]

Em seguida, este *SNonce* é transmitido para o autenticador, para que ele também gere a *PTK*. Aqui, há a legitimação do suplicante, pois indica que o *AP* está se comunicando de forma correta com o cliente, além de haver a proteção contra o ataque do tipo *man-in-the-middle* já que a criação do *PTK* envolve o uso dos endereços *MAC*. Vale notar que a *PTK* não é enviada através da rede.

A terceira mensagem é transmitida para passar informações *RSN* e, se necessário, a *GTK* também será transmitida. Caso ela não tenha sido estabelecida, será criada. A mensagem prova que o suplicante está se comunicando com um autenticador que possui a mesma *PTK*.

Finalmente, a quarta mensagem tem o propósito de encerrar a comunicação, garantindo que ambos estão em posse da mesma *PTK*, realizando a instalação da chave para uso posterior durante o acesso à rede.

2.4 EXTENSIBLE AUTHENTICATION PROTOCOL (EAP)

Como já mencionado anteriormente, na estrutura do *802.1X* temos as validações das credenciais do suplicante por parte do servidor de autenticação. E os melhores métodos de autenticação *EAP* proveem meios de se ocorrer mútua autenticação. Ou seja, além de existir a autenticação do suplicante, podemos ter uma autenticação por parte do servidor também, garantindo uma maior segurança no processo para ambos.

Geralmente, os protocolos *EAP* usam certificados para autenticar os servidores de autenticação. Para tanto, deve-se instalar também esses certificados nos suplicantes ou até mesmo colocar a autoridade raiz responsável pelo certificado fornecido. O certificado servirá para dois principais propósitos:

- Validar o servidor de autenticação: o suplicante terá certeza de que está se comunicando com o servidor correto antes de começar a compartilhar dados sensíveis;
- Criar um túnel *TLS* encriptado: ao contrário de como é usado comumente, o túnel *TLS* se dará na camada 2 ao invés de ser usado na camada de transporte. Portanto, as credenciais apresentadas pelo suplicante poderão ser protegidas ao se encriptá-las antes de serem enviadas ao servidor de autenticação

Entre o autenticador e o servidor de autenticação, um segredo compartilhado é usado para o uso do protocolo *RADIUS*. Entre o suplicante e o autenticador, os quadros 802.11 usam a encapsulação *EAP Over LAN* (*EAPOL*) para se carregar os dados *EAP*. Entre o autenticador e o *AS (Authentication Server)* é usado o *RADIUS*, que encapsula os pacotes *EAPOL*. O segredo compartilhado serve para que cada um valide o outro.

Abaixo há uma breve descrição para os principais tipos de *EAP* usados em *WLANs*.

EAP-MD5

Foi um dos primeiros tipos de *EAP* a surgir. Era usado em autenticação por porta antes de começar a ser usado em redes *WLAN*. Contudo, contém diversas vulnerabilidades: apenas o suplicante é validado, o servidor não. Como para a construção de chave dinâmica é necessária autenticação de ambas as partes, o método de encriptação é *WEP* estático ou até mesmo nenhum; o nome de usuário é sempre enviado em texto claro, sendo susceptível ao ataque de força bruta ou técnicas de engenharia social; a senha é ‘hasheada’ por meio da função *hash MD5*, que já é sabida como fraca.

EAP-LEAP

Este método apareceu com a novidade usar *WEP* dinâmico como método de encriptação. Possui um método de autenticação pseudo mútuo. Aqui, a função *hash* usada é o *MS-CHAPv2*. Além do desafio feito pelo servidor no momento de autenticação, o cliente faz um desafio similar ao servidor, buscando "autenticá-lo". Assim como o *EAP-MD5*, deixa o nome do usuário em texto claro e a função *hash MS-CHAPv2* é vulnerável também ao ataque de dicionário.

EAP-PEAP

No *PEAP*, um túnel de encriptação *TLS* é criado para encriptação das credenciais do usuário, não havendo a vulnerabilidade existente no *LEAP* e *MD5*. Para que isto seja possível, um certificado digital por parte do servidor deve ser apresentado ao cliente. Ou seja, o uso do certificado digital se torna a autenticação por parte do servidor e garante a proteção dos dados do cliente. Dentro do túnel *TLS*, outros métodos do *EAP* são usados para autenticar o cliente. Neste caso, pode ser usado tanto o tradicional nome de usuário e senha (*MS-CHAPv2*) como a apresentação de certificado por parte do cliente (*EAP-TLS*).

EAP-TTLS

O *EAP-TTLS* é bem similar ao *EAP-PEAP*, criando um túnel de encriptação *TLS* para autenticação do cliente. A diferença é que o *EAP-TTLS* suporta muito mais métodos de autenticação dentro do túnel *TLS*. No entanto, o mesmo não tem suporte nativo para sistemas *Windows*, o que faz com que sua penetração no mercado seja menor.

EAP-TLS

É considerado um dos mais seguros. Aqui, obrigatoriamente o cliente deve apresentar um certificado no momento da autenticação. Primeiramente, o servidor apresenta seu certificado, podendo o cliente recusá-lo. De modo similar, o servidor pode recusar o certificado apresentado pelo usuário. Nesse método, não é necessário o estabelecimento de um túnel *TLS* de encriptação. No entanto, por exigir um certificado para cada suplicante, o *EAP-TLS* se torna complexo de se estabelecer.

2.5 REMOTE AUTHENTICATION DIAL IN USER SERVICE (RADIUS)

RADIUS é um serviço distribuído que provê gerenciamento centralizado de segurança e de controle de acesso de usuário. Pode gerenciar e proteger uma rede *WLAN*. Usuários são autenticados pelo servidor *RADIUS*, que faz uso de um banco de dados com dados de perfis

como senhas, tipos de acesso, etc. O usuário tem acesso aos recursos da rede de acordo com as regras associadas a ele [8].

Baseado no modelo cliente-servidor, o *RADIUS* faz trocas de mensagens via *UDP*. O Servidor de Acesso à Rede (*NAS*) comporta-se como cliente *RADIUS*, passando as requisições para o servidor. Como exemplo de *NAS*, temos os pontos de acesso, roteadores e switches. O servidor *RADIUS* é capaz de desempenhar autenticação, autorização e contabilidade (AAA) após receber a requisição do cliente. A comunicação entre ambos é encriptada por meio de segredo compartilhado, já previamente configurado, não havendo necessidade de transmitir esta chave de um para o outro.

O *RADIUS* suporta vários métodos de autenticação e pode ser integrado com diversos tipos de banco de dados como *Structured Query Language (SQL)* ou *Lightweight Directory Access Protocol (LDAP)*. Neste caso, o servidor *RADIUS* irá checar a requisição de autenticação/autorização com as informações das bases de dados.

Este protocolo é extensível e é capaz de aumentar a lista de atributos usados sem afetar a implementação, o que se torna vantajoso para vendedores que desejam fazer seus próprios atributos. *RADIUS* tem suporte também para o *EAP*, havendo mais opções para protocolos de autenticação. Esta combinação (*RADIUS* + *EAP*) permite o vasto uso dos dois em *WLAN*. As mensagens *EAP* podem ser encapsuladas em pacotes *RADIUS*, não havendo necessidade de um *AP* ter suporte para o *EAP*. A Figura 2-5 mostra o processo.

Figura 2-5 Autenticação com servidor *RADIUS*.[6]

2.5.1 FREERADIUS

FreeRADIUS é o servidor de autenticação mais utilizado para sistemas Linux, por ser o responsável pela autenticação de pelo menos um terço dos utilizadores de internet do mundo,

inclusive pela comunidade acadêmica, incluindo o *eduroam*. No site do projeto "freeradius.org", há a descrição do mesmo, clamando ser uma implementação modular do protocolo *RADIUS*, de alto desempenho e rica em opções e funcionalidades. Ainda, é especializado para redes sem fio e controle remoto de usuário por meio de AAA [9]. Por ser *open source* e de ampla aceitação e uso, o *FreeRADIUS* foi escolhido como implementação *RADIUS* para este trabalho.

O *FreeRADIUS* utiliza o protocolo *RADIUS* (RFC 2865 e 2866). A comunicação entre cliente e servidor se faz pelo uso do protocolo *UDP*, sendo intermediada pelo *NAS*, no caso de uma *WLAN*. Durante a comunicação entre cliente e servidor *FreeRADIUS*, o cliente faz uma requisição ao servidor que por sua vez enviará uma mensagem de *Challenge* como resposta.

2.5.2 DALORADIUS

Para facilidade de uso, o *FreeRADIUS* pode ser configurado com o uso de scripts ou com o uso de outro projeto *open source*, o *daloRADIUS*. *DaloRADIUS* é uma aplicação de gerenciamento *RADIUS* via web voltada para a gestão de *hotspots* em WLANs. Dispõe de gerenciamento de usuários, relatórios em gráficos, contabilidade, mecanismos de cobrança e se integra com o GoogleMaps para geolocalização dos *APs* da rede.

O *daloRADIUS* é uma plataforma web escrita em PHP, HTML, CSS e *JavaScript* e utiliza a camada de abstração de banco de dados, o que na prática faria com que suportasse vários sistemas de banco de dados, mas na prática é mais voltado para banco de dados MySQL.

daloRADIUS é essencialmente uma plataforma web para gerenciar um servidor *RADIUS* de forma que teoricamente pode gerenciar qualquer servidor *RADIUS* mas especificamente ele gere *FreeRADIUS* e sua estrutura de banco de dados. Como uma aplicação web, *daloRADIUS* atua como um console de gerenciamento para controlar todos os aspectos de um servidor *RADIUS*, bem como proporcionar características comerciais e profissionais estendidos como contabilidade de informações, relatórios em gráficos, uma ferramenta de cobrança e integração interna para GoogleMaps, serviço para geo-localização de Servidores *NAS* e hotspots.

2.6 DD-WRT

Segundo [*dd-wrt.com*], DD-WRT é um desenvolvimento de um *firmware* não oficial sob os termos da GPL para vários roteadores wireless IEEE 802.11 a/b/g/h/n baseado nos chips de referência Broadcom ou Atheros.

Um roteador DD-WRT adiciona mais funções e capacidades para o roteador. No caso, o roteador deste trabalho é um *Linksys* WRT54G, o qual o DD-WRT provê suporte. As funções listadas a seguir são algumas das quais o *firmware* de fábrica do *Linksys* não oferece:

- *IPs* estáticos: o DD-WRT permite que *IPs* estáticos sejam configurados baseado no endereço *MAC* do dispositivo que conecta, sendo o *IP* fornecido pelo *DHCP*. A associação pode ser feito por meio do nome do hospedeiro;
- Repetidor/*Bridging*: havendo dois ou mais roteadores DD-WRT na rede, é possível conectá-los, sendo que o roteador primário irá rodar o servidor *DHCP* e o secundário será um repetidor. Isso permite que as pessoas conectadas à rede se movam sem desconectar-se;
- *SSH*/*Scripting*/*Cron Jobs*: permite o acesso via *SSH (Secure Shell)*, configurando sem o uso de interface gráfica (*GUI*); permite a implementação de scripts para realizar tarefas ou informar o estado de algum parâmetro; finalmente, permite que os scripts sejam rotineiramente executados (*Cron Jobs*).

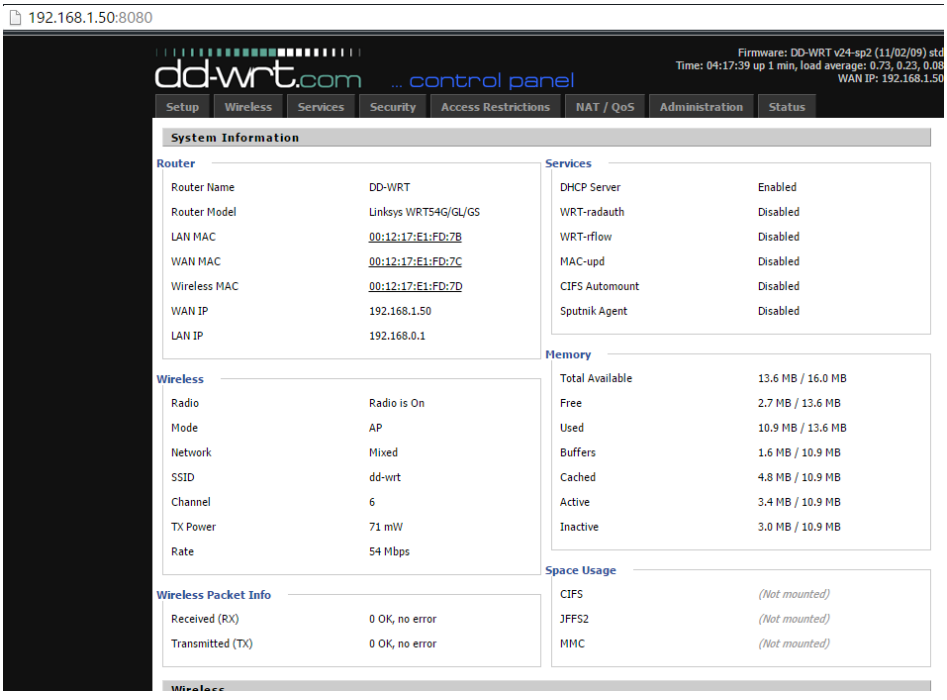

Figura 2-6 Interface gráfica do DD-WRT.

2.7 KALI LINUX

Segundo [*docs.kali*], Kali Linux é uma avançada distribuição especializada em Testes de Intrusão e Auditoria de Segurança. É uma distribuição resultante de uma reconstrução completa do BackTrack Linux, que adere totalmente aos padrões de desenvolvimento Debian. Alguma das ferramentas presentes no Kali são apresentadas abaixo:

- AIRODUMP-NG: é uma ferramenta utilizada para a captura de pacotes que trafegam na rede sem fio e é particularmente adequada para coletar quadros *IVs* (vetor de inicialização *WEP*), pacotes do *4-Way Handshake*, detectar redes que não divulgam o *SSID*, podendo usar os dados coletados para quebra de chave com o aircrack-ng;
- AIRCRACK-NG: é um programa de violação de chaves *WEP* e *WPA-PSK* 802.11 que pode recuperar chaves depois de capturar pacotes suficientes. A ferramenta é um conjunto de ferramentas de auditoria de redes sem fio, capaz analisar os pacotes capturados pelo Airodump-ng e, consequentemente, quebrar as chaves.
- AIREPLAY-NG: é usado para injetar pacotes na rede. Sua função principal é gerar tráfego para uso posterior no aircrack-ng para quebra de chaves *WEP* e *WPA-PSK*.

2.8 VULNERABILIDADES EXISTENTES

As redes sem fio do padrão IEEE 802.11 são intrinsicamente inseguras devido à natureza do meio usado para transmitir (aéreo), que é compartilhado e não tem especificamente uma fronteira (o sinal se propaga até onde for possível). Desta forma, o sinal é compartilhado não só com os usuários da mesma rede, mas com qualquer dispositivo que seja capaz de estar na mesma faixa de transmissão da rede, o que traz diversos riscos, aumentando assim os riscos de segurança e ataques contra *WLANs*.
Alguns desses ataques serão listados a seguir [7].

ROGUE ACCESS

Um dispositivo trapaceiro (podendo ser um *AP*) é qualquer dispositivo que tenha rádio *WLAN* e esteja conectado na infraestrutura cabeada sem gerência dos administradores da rede, ou seja, sem autorização. De acordo com [CWSP], muitas vezes são os próprios usuários da rede que acabam por instalar um AP não autorizado, ao invés de um hacker. O problema é que isso pode não estar devidamente seguro, abrindo uma brecha para que alguém possa ganhar acesso a infraestrutura da rede que uma companhia deseja proteger, tais como servidores *LDAP* (*Lightweight Directory Access Protocol*), AAA e banco de dados. Alguns destes usuários

podem nem saber também que estão comprometendo a rede, faltando um maior reforço da política de segurança vigente na empresa.

Também há o caso em que se faz uso da comunicação *ad hoc* (*IBSS – Independent Basic Service Set*), onde um cliente com acesso a rede cabeada pode estabelecer uma conexão sem fio com outro cliente, garantindo acesso ao mesmo para a rede cabeada. Similarmente, câmeras de segurança e impressoras sem fio podem ser usados como meio de acesso à rede.

- Roubo de dados
- Destruição de dados
- Perda de serviços
- Inserção de dados maliciosos
- Ataques terceirizados (usar a rede comprometida para atacar outra)

Espionagem

Comunicações em *WLAN* podem ser capturadas ao se estar na mesma frequência do canal da comunicação. Essas comunicações podem ser monitoradas de duas formas: espionagem casual ou espionagem maliciosa.

Casual:

Neste caso, um espião casual pode simplesmente usar um rádio 802.11 para capturar pacotes de gerência das redes sem fio. Assim, podem ter acesso a informações como: *SSID*, endereço *MAC*, taxas de dados suportadas, e outras capacidades do *BSS* (canal, encriptação, etc.). Ferramentas capazes de analisar as informações das redes sem fio ao redor estão disponíveis na distribuição Kali, como por exemplo, o Kismet e o airmon-ng (parte da família do aircrack-ng).

Maliciosa:

Aqui, ocorre a captura dos dados de qualquer comunicação na rede sem fio, o que é considerado normalmente como ilegal. Análise de protocolo e análise de pacotes geralmente são usadas para diagnosticar problemas de comunicação na rede, identificar padrões de tráfego e achar gargalos na rede. Assim, muitas ferramentas existem (*Wireshark*, por exemplo) para que um administrador de uma rede seja capaz de analisar e diagnosticar os problemas de sua rede. No entanto, um analisador de protocolos pode ser usado para fins maliciosos, como quebra de chave e decriptação de dados.

Ataques de autenticação

Como já mencionado anteriormente, em sistemas antigos de redes sem fio, existia a *Open System Authentication*, na qual todos os clientes são autenticados. Por conseguinte, surgiu

o *WEP* com a autenticação baseada em chave. Como mais recente tipo de autenticação usada pelo *WPA2*, temos a autenticação *PSK* (uso de chaves) e *802.1X*/*EAP*. O *EAP* tem vários tipos como já mencionado, os quais se diferenciam por usar diferentes credenciais tanto por parte do cliente como por parte do servidor de autenticação, o que torna alguns tipos de *EAP* mais forte que outros.

Tem-se o *LEAP*, o qual é susceptível ao ataque off-line de dicionário, devido ao hash da senha de resposta ser quebrável.

Fazendo parte do *WPA* e *WPA2-Personal*, o método de autenticação *PSK* é consideravelmente fraco, sendo susceptível ao ataque de dicionário de força bruta. Chaves compartilhadas podem ser facilmente obtidas também por meio de técnicas de engenharia social, manipulando pessoas que as possuem para que compartilhem informações da rede, no caso, a chave. Uma consequência da exposição da chave da rede, é que a chave de encriptação *TKIP*/*RC4* ou *CCMP*/*AES* dinamicamente gerada passa a ser quebrável, expondo os dados trafegados na rede. Isso será possível se o atacante capturar o processo de *4-Way Handshake*. Com a chave e o processo capturado em mãos, é possível realizar o processo de decriptação anteriormente mencionado.

O maior risco com ataques de autenticação é que ao serem comprometidas certas credenciais legítimas, toda a rede se torna vulnerável.

Ataques DoS (Denial of Service)

Um ataque de negação de serviço não visa roubo de dados ou informações. Especificamente, ele é capaz de desativar completamente a rede, se tornando uma séria preocupação. Impede que usuários sejam capazes de se conectar à rede, o que se torna crítico quando se tem serviços de extrema importância não acessíveis. Acesso à internet é bloqueado, uma ligação *VoIP (Voice over IP)* é terminada e o acesso a qualquer base de dados não é possível. E algo que agrava ainda mais esse tipo de ataque é que nada pode ser feito para evitar o seu início.

Layer 1 DoS attack

Esse é o tipo de ataque DoS mais simples de fazer, no qual é necessário que se tenha algum dispositivo que gere sinal na faixa de uso da rede sem fio. Esse sinal pode ser um que ocupe grande parte da banda ou que concentre especificamente em algum canal do espectro. Devido à natureza do tipo de acesso ao meio usado na rede sem fio (*CSMA*/*CA – Carrier Sense Multiple Access – Collision Avoidance*), caso haja uma contínua transmissão de sinal interferente nos canais, a comunicação não será possível enquanto houver a interferência.

3 METODOLOGIA

Este capítulo envolve a metodologia usada no trabalho para a configuração e simulação dos cenários dos ataques às redes bem como o modo como os ataques foram realizados.

3.1 DELIMITAÇÃO DO TEMA

Este trabalho tem como proposta a construção de um guia para entendimento da segurança em redes *WLAN*. Especificamente, aborda aspectos dos tipos de seguranças hoje disponíveis (*WEP*, *WPA* e *WPA2*) e as vulnerabilidades existentes. Assim, com o uso de ferramentas como DD-WRT, *FreeRADIUS*, *daloRADIUS* e *Kali Linux*, busca-se abordar todos os meios necessários para se ter uma rede consideravelmente segura.

O fluxograma da Figura 3-1 mostra as etapas seguidas para a consolidação do trabalho.

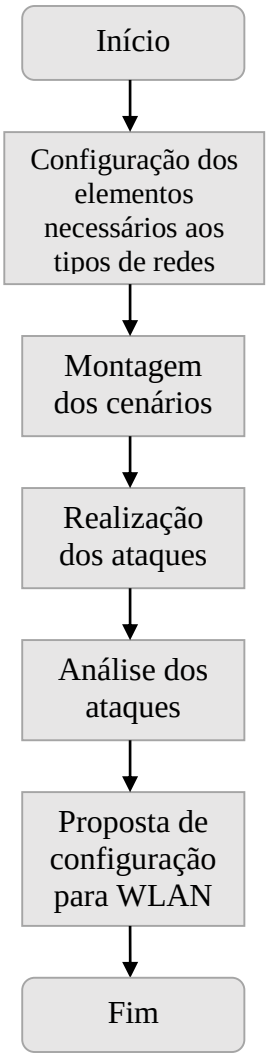

Figura 3-1 Fluxograma das etapas do trabalho.

Etapa 1 – Configuração dos elementos necessários aos tipos de redes: aquisição dos dispositivos (adaptador *wireless*, roteador *wireless*) e configuração dos mesmos bem como configuração das ferramentas usadas nos tipos de redes testadas (*FreeRADIUS*, DD-WRT, *Kali Linux*, *Oracle VM Virtual Box*).

Etapa 2 – Montagem dos cenários: dependendo do tipo de rede escolhida para teste (*WEP*, *WPA*, *WPA2*), os dispositivos e ferramentas necessárias são preparadas para os testes da segurança da rede escolhida.

Etapa 3 – Realização dos ataques: com o tipo de segurança escolhido, as vulnerabilidades já conhecidas são exploradas, visando constatar a falha de segurança de cada tipo de rede.

Etapa 4 – Análise dos ataques: verificação dos resultados obtidos e análise da eficácia dos ataques.

Etapa 5 – Proposta de configuração para *WLAN*: com os resultados dos ataques e análise em mãos, analisa-se uma proposta de rede segura para *WLANs*.

3.2 MATERIAL UTILIZADO

Para os testes deste trabalho, foi usado o roteador *Linksys* WRT54G v2.2 (Fig. 3-2), sendo que o *firmware* usado foi o DD-WRT *VINTAGE*: standard, capaz de fazer uso de todos os tipos de segurança (*WEP*, *WPA*, *WPA2-Personal/Enterprise*) bem como os modos de encriptação.

Figura 3-2 Roteador *Linksys* WRT54G v2.2.

Figura 3-3 Adaptador *wireless* TP-Link TL-WN722N.

O adaptador USB *Wireless* TP-LINK TL-WN722N (Fig. 3-3), com uma antena de 4dBi e compatível com os padrões 802.11n/b/g foi usado como adaptador wireless para captar os dados e realizar os ataques por meio da distribuição *Kali Linux*.

O *Kali Linux* foi utilizado como máquina virtual, sendo que o *software* era o *Oracle VM VirtualBox*. Portanto, sabe-se que por se tratar de uma máquina virtual, podem ter ocorrido limitações quanto ao desempenho nos ataques e capturas.

Ainda, para comportar o servidor *RADIUS* usado no modo de segurança *WPA/WPA2-Enterprise*, uma outra máquina virtual rodando a distribuição Linux foi usada. No caso, o servidor *RADIUS* adotado foi o *FreeRadius*, que é um *software* de código livre, em conjunto com o *MySql*, responsável pelo armazenamento dos dados. Ainda, para o gerenciamento do servidor *RADIUS*, fez-se uso também do *daloRADIUS*, capaz de prover uma interface gráfica e também o gerenciamento dos mais diversos recursos (contabilidade, autenticação, geolocalização (Fig. 3-5), autorização, etc.).

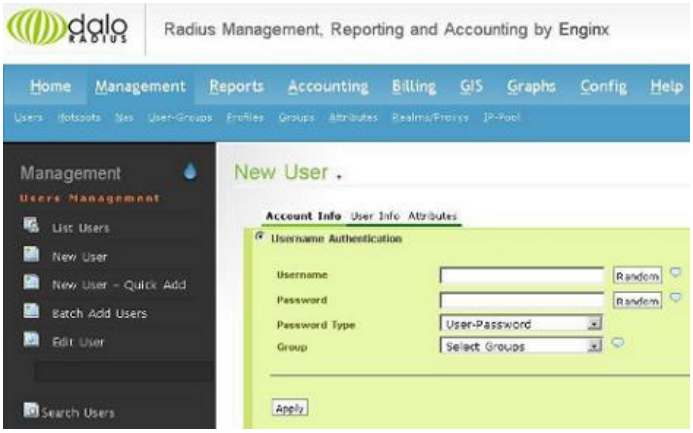

Figura 3-4 Interface gráfica do *daloRADIUS* – Adição de usuários. [15]

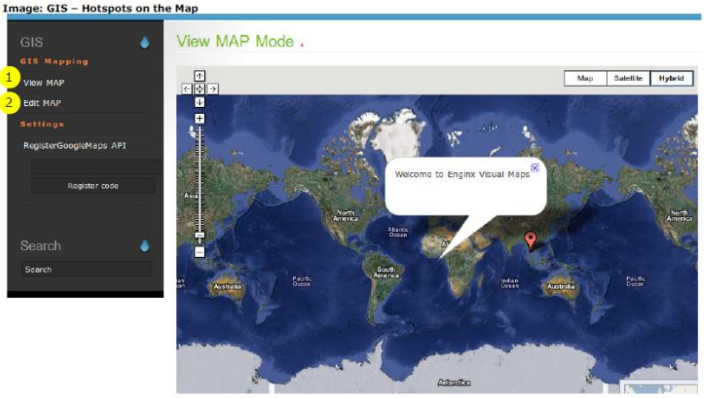

Figura 3-5 Monitoramento da localização dos *APs* com GoogleMaps.[15]

3.3 CONFIGURAÇÃO DA REDE

Fazendo uso de um modem para ter de fato uma conexão à internet, a conexão entre o modem e o roteador DD-WRT foi feita da seguinte forma (Fig. 3-6):

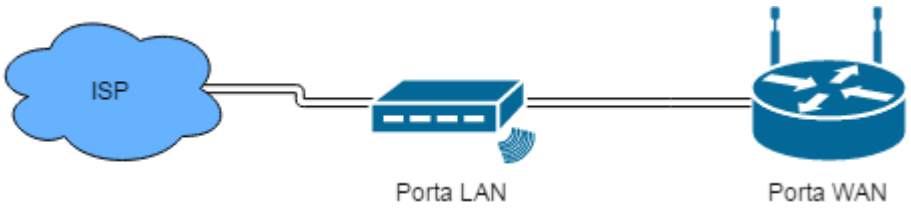

Figura 3-6 Topologia - roteador Linksys conectado ao modem.

Assim, a rede do *AP* usado no trabalho pertencia a uma faixa de endereços *IP* diferente da do modem, ou seja, o cabo foi conectado em uma porta *LAN* do modem e a outra ponta foi conectada na porta *WAN* do roteador, com o propósito de ser possível distinguir entre clientes de uma rede e outra. A Figura 3-7 mostra as configurações de *IP* do DD-WRT.

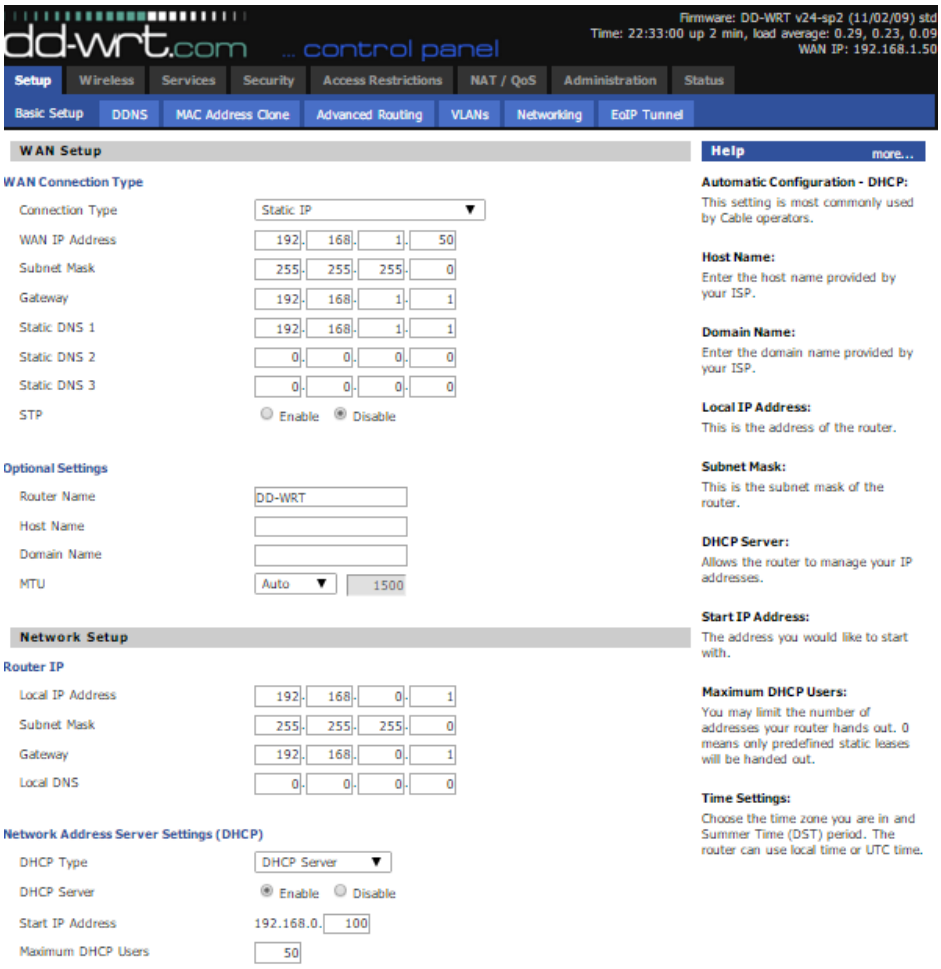

Figura 3-7 Interface gráfica do DD-WRT com configurações de *IP* e *DHCP*.

3.3.1 CENÁRIO 1: WEP

Neste caso, foram necessários o uso do roteador *Linksys*, um modem que provesse acesso à internet e um cliente no cenário em que há equipamentos conectados à rede. No caso de o cenário ser um no qual não há clientes, apenas o atacante figura no cenário junto com o roteador. O cenário é ilustrado na Figura 3-8.

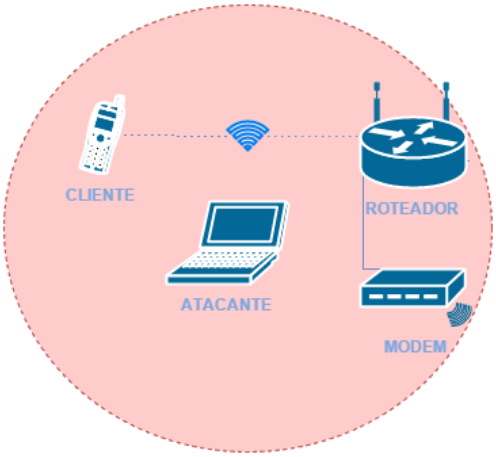

Figura 3-8 Cenário para ataques contra WEP e *WPA/WPA2-Personal*.

Para que o *WEP* esteja como método de segurança, a modificação é feita via interface web do DD-WRT como mostra a figura a 3-9.

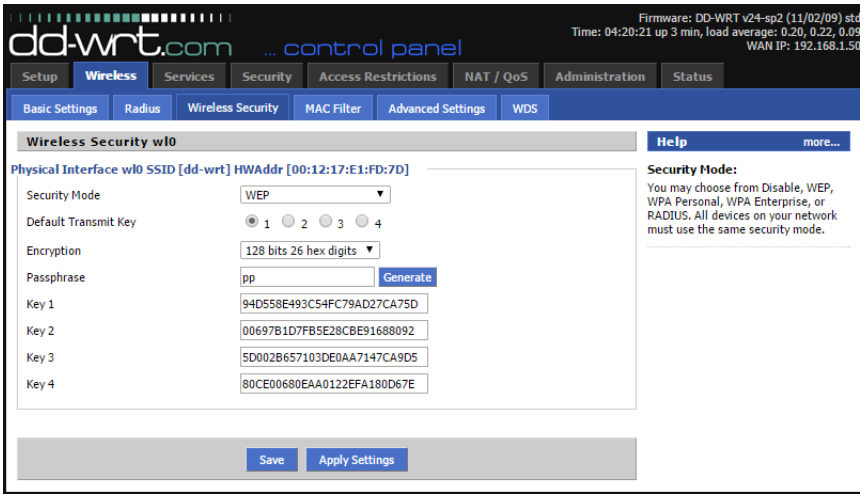

Figura 3-9 Configuração no DD-WRT para o cenário *WEP*.

3.3.2 CENÁRIO 2: WPA/WPA2-PERSONAL

Nessa configuração, a mudança física na topologia não ocorre. A única mudança necessária no caso é a configuração de segurança no roteador. Lembrando que pode ser escolhido como algoritmo de encriptação tanto o *TKIP* (*WPA*) como o *CCMP* (*WPA2*). O método de encriptação não fará a diferença para os ataques realizados neste trabalho. Porém, como já foi mostrado em (referência de ataque *TKIP* e *CCMP*), há vulnerabilidades que podem ser exploradas.

3.3.3 CENÁRIO 3: WPA/WPA2-ENTERPRISE

Este cenário, por sua vez, tem o adicional de ter o servidor *RADIUS* ajudando na autenticação no processo da autenticação *802.1X/EAP*. Atacante e servidor foram hospedados em máquinas virtuais separadas em um mesmo laptop. No entanto, por usarem adaptadores de rede diferentes, não houve interferência no processo entre os mesmos. O cenário é mostrado na Figura 3-10 e a configuração do DD-WRT na Figura 3-11.

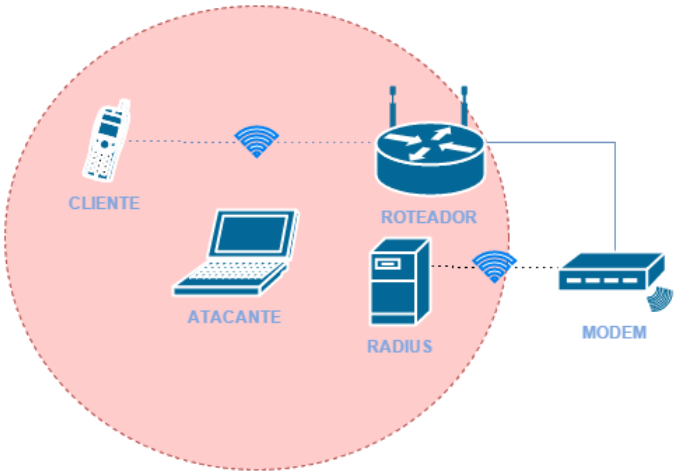

Figura 3-10 Cenário para ataques contra *WPA/WPA2-Enterprise.*

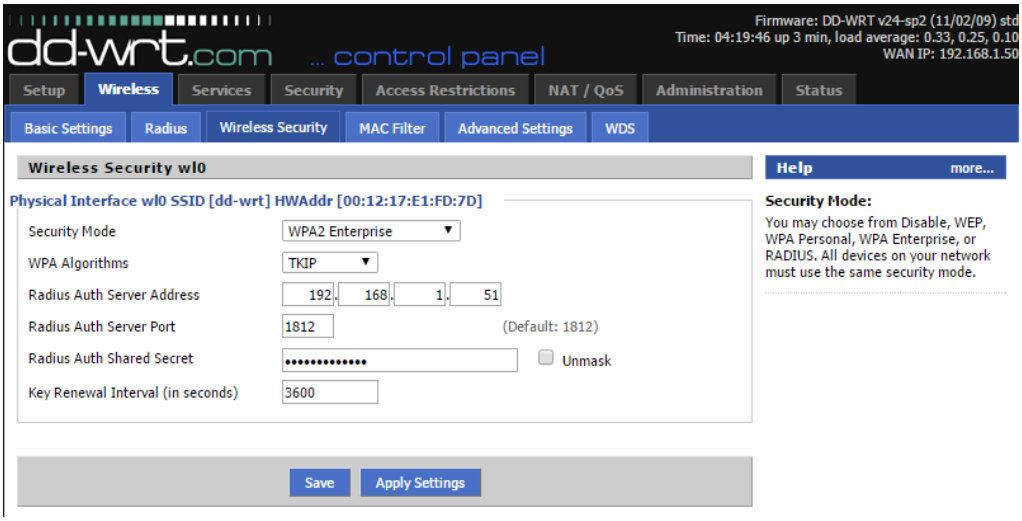

Figura 3-11 Configuração no DD-WRT para *WPA/WPA2-Enterprise*.

3.4 ATAQUES

Cenário WEP

Para os ataques realizados contra o tipo de segurança *WEP*, foram conduzidos os ataques com os seguintes cenários:

- Chave *WEP* de 64 bits com cliente conectado à rede;
- Chave *WEP* de 64 bits sem cliente conectado à rede;

- Chave *WEP* de 128 bits com cliente conectado à rede;
- Chave *WEP* de 128 bits sem cliente conectado à rede.

No caso de haver cliente(s) conectado(s) à rede, o ataque é realizado da seguinte forma [7]:

Primeiramente, passa-se a monitorar a rede alvo do ataque, executando o seguinte comando:

```
airodump-ng -c 1 -b 00:12:17:e1:fd:7d -h 14:5a:05:6a:ef:5e --write wep wlan1mon
```

Como o número de dados trafegados pode ser baixo e como é necessário coletar uma alta quantidade de pacotes para ter uma quantidade considerável de vetores de inicialização (*IV*) a fim de que se descubra a chave, utiliza-se do recurso de injeção de pacotes *ARP request*:

```
aireplay-ng -3 --b 00:12:17:e1:fd:7d -h 14:5a:05:6a:ef:5e wlan1mon
```

Para que essa injeção de pacotes aconteça, é necessário que se tenha algum cliente conectado à rede. Assim, ao se captar um pacote *ARP* deste cliente, o processo tem início, utilizando-se o endereço *MAC* do cliente conectado à rede. Desta forma, a coleta de pacotes para que se tenha uma longa base de *IVs* é acelerada. O teste das chaves é realizado de acordo com a coleta de pacotes, sendo que os processos de coleta, injeção de pacotes e teste de chaves acontecem de forma simultânea.

```
aircrack-ng -b 00:12:17:e1:fd:7d wep wlan1mon
```

Para o cenário onde não existem clientes conectados à rede, o processo é um pouco diferente. De começo, simula-se uma falsa autenticação no *Access Point*, autenticação é essa que faz uso apenas do endereço *MAC* da interface de rede que está em modo promíscuo:

```
aireplay-ng --fakeauth 0 -o 1 -e dd-wrt -a 00:12:17:E1:FD:7D -h F8:D1:11:B3:D8:8E wlan1mon
```

Assim, trata-se de uma falsa autenticação *Open System*. Como não se tem posse da chave de encriptação, utilizam-se duas técnicas (*Fragmentation* e *ChopChop*) para a recuperação do *keystream* dos pacotes encriptados pela rede *WEP*:

```
aireplay-ng --fragment -b 00:12:17:E1:FD:7D -h F8:D1:11:B3:D8:8E wlan1mon
#(fragmentation)
```

```
aireplay-ng --chopchop -b 00:12:17:E1:FD:7D -h F8:D1:11:B3:D8:8E wlan1mon
#(chopchop)
```

Essas técnicas são possíveis graças ao fato do *AP* enviar pacotes de divulgação da rede ainda que não existam clientes conectados a ele. Em posse do *keystream* é possível realizar a injeção de pacotes *ARP request*, forjando também o suposto *IP* que o "falso" cliente conectado obteve:

```
packetforge-ng --arp -a 00:12:17:E1:FD:7D -h F8:D1:11:B3:D8:8E -k
192.168.1.100 -l 192.168.1.1 -y fragment-0325-172339.xor -w arp-request
aireplay-ng --interactive -r arp-request wlan1mon
```

A partir desse passo, o processo acontece de forma semelhante ao primeiro cenário, onde a coleta de dados, a injeção de pacotes *ARP* e o teste de chaves acontecerão simultaneamente. O fluxograma é ilustrado na Figura 3-12.

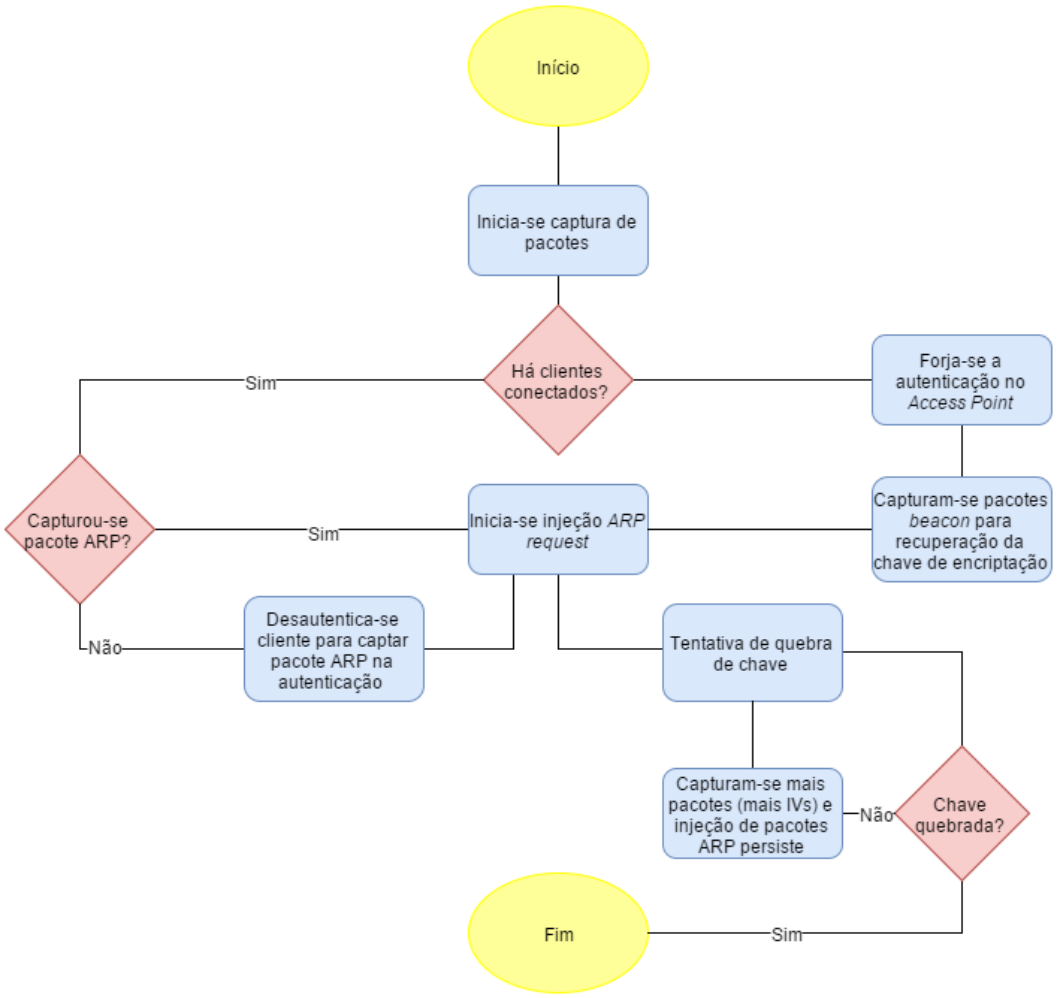

Figura 3-12 Fluxograma de ataque contra WEP.

Cenário WPA/WPA2-Personal

Para os ataques contra a rede *WPA*/*WPA2-Personal*, os cenários para os ataques foram os seguintes:

- Segurança *WPA* com criptografia *TKIP*;
- Segurança *WPA* com criptografia *AES*;
- Segurança *WPA2* com criptografia *TKIP*;
- Segurança *WPA2* com criptografia *AES*.

Em todos os cenários de ataque, os procedimentos realizados foram os mesmos. Diferentemente do tipo de segurança *WEP*, no *WPA/WPA2-Personal*, não existem muitas brechas para serem explorados a fim de se obter a chave da rede. Portanto, a eficiência do ataque dependerá principalmente de dois fatores: da senha configurada para a rede e da base de dados do atacante de possíveis senhas. Desta forma, constata-se que a melhor forma de se defender de ataques em uma rede *WPA/WPA2-Personal* é configurando senhas de difícil combinação e que não tenha caracteres conhecidos por serem constantemente utilizadas em senhas, como por exemplo "123456", "senha" e "admin". Levando em conta essas sugestões na hora de se configurar a rede, dificilmente haverá espaços até mesmo para atacantes determinados.

Quanto ao processo do ataque, ele é simples e acontece como explicado em seguida. Após iniciar-se o monitoramento da rede atacada, é necessário que se tenha ao menos um cliente conectado ao *AP*, pois para que o ataque aconteça, o processo do *4-Way Handshake* tem que ser capturado. Esta parte é essencial pois a tentativa de quebra de chave só é possível com a posse de alguns parâmetros utilizados no estabelecimento da chave (*PTK*), que são eles: *A-Nonce, S-Nonce* e endereços *MAC*. Como o processo de autenticação pode demorar a acontecer, podemos forçar uma desautenticação de alguém que esteja conectado, utilizando o seguinte comando:

```
aireplay-ng --deauth 1 -c 98:52:B1:3B:32:58 -a 08:7A:4C:83:0C:E0 mon0
```

Onde o campo –c simboliza o *MAC* do cliente atacado, o –a simboliza o *MAC* do *AP* , –deauth corresponde ao ataque realizado e mon0 simboliza a interface usada para o ataque.

Após a execução deste comando, o cliente será desconectado da rede e provavelmente tornará a conectar, realizando o processo de *Handshake*. Como a captura já terá sido iniciada, o processo será gravado e pronto para ser usado no ataque. O passo final é fazer a tentativa por meio de um ataque de força bruta, onde se usa um arquivo que contenha inúmeros possíveis senhas que possam estar sendo usadas pelas redes:

```
aircrack-ng -w wparockyou.txt wpa_crack-01.cap
```

Sendo que wparockyou.txt trata-se do arquivo com possíveis senhas e o wpa-crack-01.cap é o arquivo com os pacotes capturados. Dentre esses pacotes, há os do *handshake*.

No caso, quanto melhor se escolher a base de dados usada, maior a chance de se ter sucesso no ataque. Vale citar também aqui a importância de se terem usuários cientes dos perigos que correm ao exporem de forma indireta ou direta a senha de acesso à rede. Deve-se tomar cuidado quanto à essas questões, que fazem parte da área de engenharia social.

O fluxograma do ataque é ilustrado na Figura 3-13.

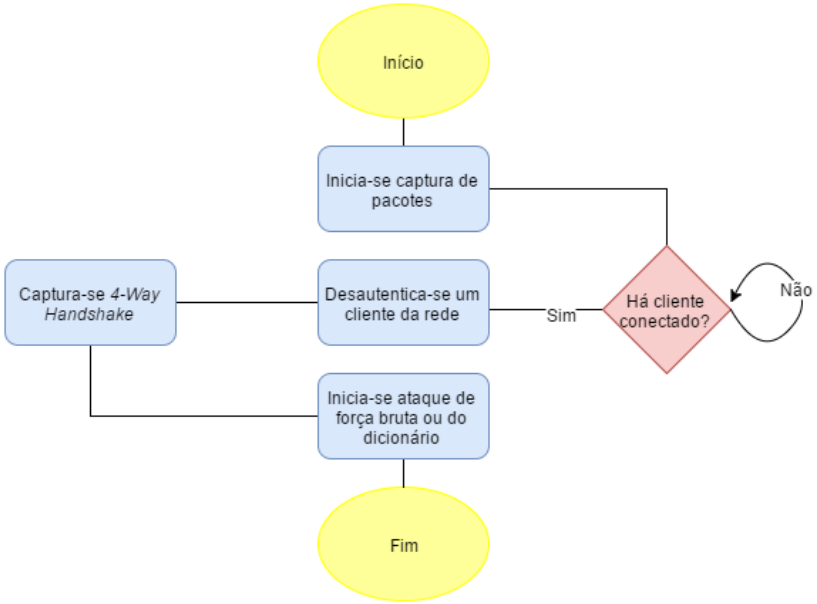

Figura 3-13 Fluxograma de ataque contra *WPA/WPA2-Personal*.

Para os ataques contra este modo de autenticação, foi estabelecido o cenário onde o tipo de *EAP* usado foi o mais comumente empregado, o *PEAP* (*Protected EAP*). Com o objetivo de realizar um ataque contra o *EAP*, primeiramente captura-se os pacotes de *handshake* do *EAP*, fazendo uso do airodump-ng:

```
airodump-ng wlan1mon –channel 1 –ssid
```

É necessário que se saiba o tipo de *EAP* utilizado antes de tentar atacá-lo. Para tanto, inspeciona-se os pacotes capturados por meio do *Wireshark* após colocar a interface de rede sem fio em modo de monitoramento, como é mostrado no comando acima. Dentro dos campos correspondentes à autenticação *802.1X*, é possível ver os detalhes do *EAP* em questão. Para realizar isso, basta ir em *Statistics* → *Protocol hierarchy*. Selecionando *802.1X Authentication*, apenas os pacotes relacionados ao *EAP* aparecerão como mostram as figuras 3-14 e 3-16.

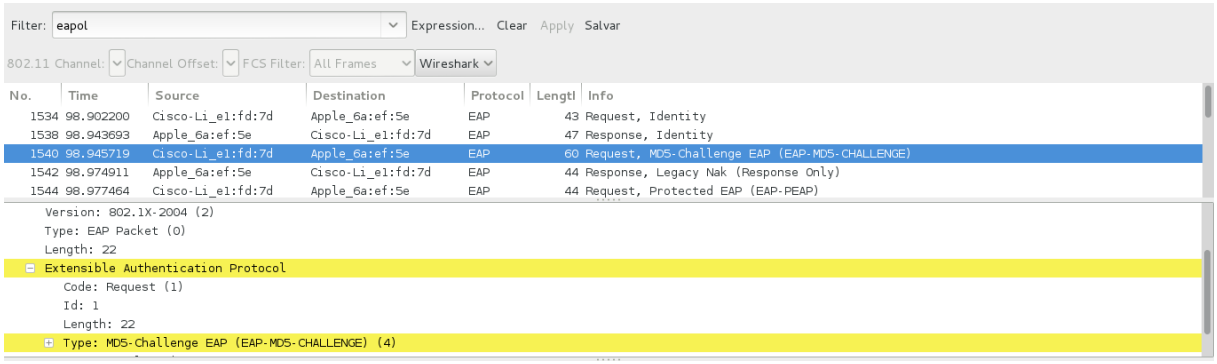

Figura 3-14 Identificação do tipo de *EAP* usado no cenário (*EAP-MD5*).

Como é possível perceber e como já foi falado, os dois tem vulnerabilidades semelhantes e, portanto, estão correndo o risco de sofrerem do mesmo ataque. Já que não há uso de um túnel de encriptação para passar os dados do protocolo de autenticação, é necessário apenas que haja a desautenticação de um cliente seguido de um ataque de força bruta. Aqui o ataque tem muita semelhança com o ataque contra o método *PSK* já mostrado. As informações são coletadas durante o processo de *4-Way Handshake* [10].

Para dar início ao processo de quebra de chave, inicia-se a desautenticação por meio do seguinte comando:

```
aireplay-ng --deauth 1 -a <MAC-AP> -c <MAC-CLIENT> <INTERFACE>
```

Em seguida, presta-se atenção na tela de captura (airodump-ng), esperando que apareça uma mensagem no canto superior direito da tela a mensagem de que foi capturado o processo de *handshake* do cliente que está sendo atacado.

Já descoberto o tipo de *EAP*, pode-se prosseguir para o ataque dependendo do tipo de *EAP* em questão. Caso se trate do *EAP-LEAP* ou *EAP-MD5*, usa-se as ferramentas *asleap* ou *eapmd5pass*, as quais são capazes de realizar um ataque de dicionário através dos seguintes comandos:

```
genkeys -r wordlist.txt -f wordlist.hash -n wordlist.idx
asleap -r eap_crack-01.cap -f wordlist.hash -n wordlist.idx
```

O primeiro comando é responsável por gerar um arquivo de *hash* das palavras contidas no dicionário indicado(wordlist.txt). O segundo é a tentativa de fato da senha usada pelo usuário que teve seu *handshake* capturado(eap_crack-01.cap).

Portanto, todo o processo de ataque para o *EAP-MD5* e o *EAP-LEAP* é semelhante ao *PSK* do *WPA Personal* e a ilustração dos passos é similar ao do fluxograma acima.

O *EAP-TTLS* e o *EAP-PEAP* podem se tornar vulneráveis caso os clientes da rede não autentiquem de fato o certificado dos servidores que são apresentados. No caso, o atacante pode configurar um falso *AP* e impersonificar o legitimo, podendo enganar o cliente e fazer com que o mesmo se conecte ao falso *AP* ao ser desautenticado pelo atacante. O segredo do ataque aqui é que o túnel *TLS* de encriptação é quebrado, permitindo o acesso às informações trocadas pelo *EAPOL* (*challenge* e *response*). O fluxograma do ataque é ilustrado na Figura 3-15.

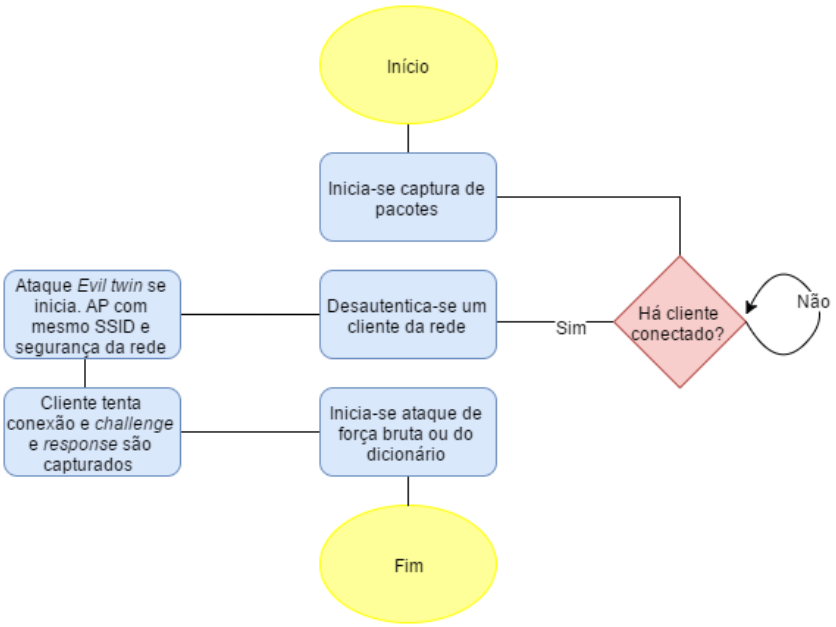

Figura 3-15 Fluxograma de ataque contra *WWPA/WPA2-Enterprise*.

Inicialmente, para constatar que se trata de um *EAP* com túnel de encriptação *TLS*, inspeciona-se os pacotes de autenticação da rede em questão. No caso, busca-se a informação do mesmo modo como no ataque anteriormente mencionado. A Figura 3-16 mostra que de fato se trata de um *EAP* com encriptação *TLS* (*EAP-PEAP*).

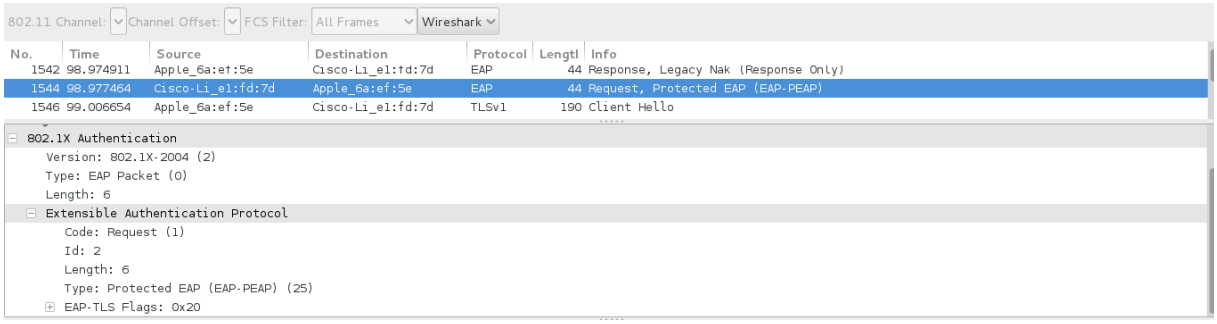

Figura 3-16 Identificação do EAP-PEAP via *Wireshark*.

Antes de tudo, é necessário que se tenha uma infraestrutura que fará parte do processo de convencer o cliente de que a rede falsa do atacante se trata de uma rede legítima. Para isto, deve ser criado tanto um falso servidor *RADIUS* como um falso ponto de acesso, sendo que o

sinal do falso *AP* deve ser mais forte do que o do legítimo, de forma que o cliente tente conectar-se ao falso.

Os *softwares* usados para o falso *AP* e *RADIUS* não vêm já instalado no Kali. São eles: o *freeradius-server-2.1.12*, o *freeradius-wpa (Wireless Pwnage Edition) patch* e o *hostapd v2.0*. Os passos de instalação e configuração estão presentes no Apêndice deste projeto (A.6).

Após a configuração dos *softwares*, inicia-se o servidor *RADIUS* com o seguinte comando: radiusd –X, onde o –X simboliza que o servidor está sendo executado no modo *debugging*, permitindo que seja visto as operações que o servidor realiza. A Figura 3-17 mostra o comportamento esperado ao ser executado o comando:

Figura 3-17 Servidor *RADIUS* executado em modo *debugging*.

A outra parte da infraestrutura é o falso *AP*, que pode ser baseado em um *hardware* ou simplesmente através do uso de um *software* (que é o caso deste projeto), usando o adaptador de rede sem fio como o rádio do *AP*. Para isto, o *software hostapd AP* é usado. O *software* é inciado com o comando hostapd ./hostapd.conf. A Figura 3-18 mostra a tela esperada:

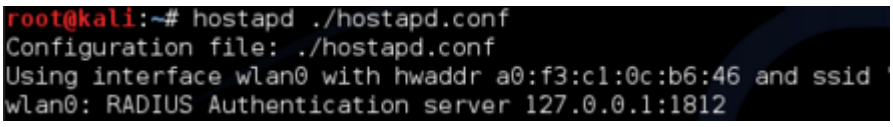

Figura 3-18 Execução do falso AP (*hostapd*).

Com os dois elementos da rede falsa funcionando, é possível que o ataque seja iniciado, de acordo com o fluxograma ilustrado acima.

Primeiramente, ocorre a desautenticação do cliente como já foi feito nos outros ataques mencionados. A diferença aqui é que não é desejado que o cliente torne a se conectar no *AP* em que estava conectado. O cliente é levado a pensar que está conectando-se na mesma rede de

antes. É neste processo de engano que o atacante consegue captar as credenciais que são encriptadas no túnel *TLS*.

Não é simples levar o cliente a tentar conectar-se na rede falsa estabelecida. O sinal precisa ser mais forte, o que pode ser complexo se o ambiente é de acesso restrito, não permitindo uma aproximação à vítima do ataque.

No caso de êxito do ataque, tanto a tela do servidor *RADIUS* como a tela do *hotsapd* mostrarão o tráfego entre suplicante e servidor. A parte que interessa estará presente no arquivo de log gerado pelo *FreeRADIUS* e pode ser acessado pelo seguinte comando:

```
tail –f /usr/local/var/log/radius/freeradius-server-wpe.log
```

Apesar da autenticação não ser realizada com sucesso, a informação desejada já terá sido captada (Fig. 3-19), ou seja, o desafio e resposta da função *MS-CHAPv2* estará à disposição para que o ataque de força bruta ou de dicionário seja executado.

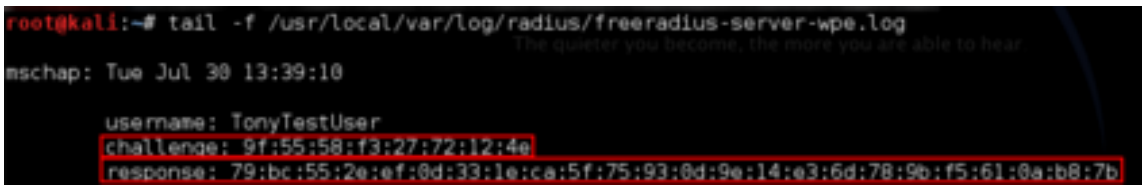

Figura 3-19 Captura do *challenge/response* do método *EAP-PEAP*.

O *asleap*, então, poderá ser usado para tentativa de quebra de chave.

4 RESULTADOS E ANÁLISE

Este capítulo apresenta os resultados obtidos com os ataques e análise dos mesmos e das soluções usadas para confecção do trabalho, visando estabelecer uma configuração ideal de rede sem fio.

4.1 RESULTADOS

Os resultados dos ataques *WEP* e *WPA/WPA2-Personal* estão listados nas Tabelas 4-1 e 4-2, seguidas de um gráfico de comparação.

	Segurança *WEP*			
	Chave 64 bits		Chave 128 bits	
Tempo de ataque	Com cliente	Sem cliente	Com cliente	Sem cliente
	3min	3min 15s	4min 56s	6min

Tabela 4-1 Tempo de ataque contra WEP para quebra de chave.

	Segurança *WPA/WPA2-Personal*			
	TKIP		*AES*	
Tempo de ataque	*WPA*	*WPA2*	*WPA*	*WPA2*
	1min 33s	1min 40s	1min 16s	1min 21s

Tabela 4-2 Tempo de ataque contra *WPA/WPA2-Personal* para quebra de chave.

Os dados mostrados são as médias de cinco ataques feitos para cada caso. O tempo começou a ser contado a partir do momento em que se iniciava a execução dos comandos necessários. Como cliente, foram usados um celular *iPhone 4* de 16GB com sistema operacional *iOS 6.2* e um notebook com *Windows 8* com processador Intel® Core™ i5-3210M CPU @ 2.50GHz e 6GB de memória *RAM*.

Quanto aos ataques *WEP*, como pode ser visto, o tempo de quebra de chave foi extremamente rápido, tanto com cliente ou sem clientes conectados. O tamanho de chave teve pequena influência no tempo de quebra, não sendo um obstáculo para a obtenção da chave. O

fato de tanto o ataque que havia cliente conectado como o ataque que não havia cliente conectado terem tido tempos próximos de quebra se deve à razão de que o ataque realizado só se diferencia na falsa autenticação realizada quando não se tem cliente, não sendo um processo demorado e de difícil realização.

Esta é mais uma prova de como é vulnerável uma rede sem fio configurada como segurança *WEP* nos dias de hoje. Mesmo com fatores que acabam por diminuir o desempenho nos ataques (uso de máquina virtual e uso de adaptador sem fio de baixo alcance), não existiram dificuldades para obtenção da chave de acesso.

Para o ataque contra a rede *WPA/WPA2-Personal*, partiu-se do pressuposto que a chave estava presente no dicionário utilizado para o ataque. O resultado, neste cenário, resulta em uma quebra de chave mais rápida até mesmo do que no caso da segurança *WEP* (Fig. 4-1). Isso mostra o quão é necessário adotar as boas práticas contra vazamento de informações que possam comprometer a rede sem fio e até mesmo a rede local (*LAN*):

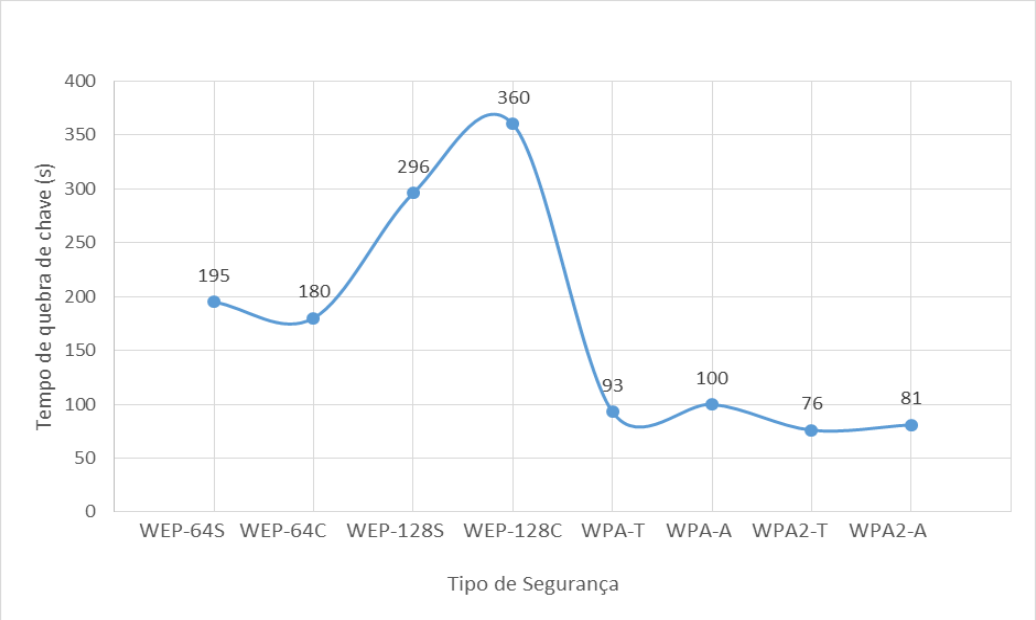

Figura 4-1 Tempo de quebra de chave de acordo com o método de segurança.

Portanto, adotando-se boas práticas de uso tais como chaves longas (possível de 8 a 64 caracteres) e o uso de caracteres alfanuméricos, periodicamente mudando e evitando informar para qualquer pessoa, contribuirá de forma essencial para a contínua segurança da rede. Prova da dificuldade imposta quando não se tem alguma dica da chave usada foi constatado por meio de um ataque de força bruta em uma chave onde já se sabiam 4 caracteres e o tamanho da chave (10 caracteres). O tempo levado para quebrar tal chave foi de quase 37 horas, levando em conta que apenas letras maiúsculas foram usadas. Garantindo que a senha varie com seus caracteres,

com combinações improváveis, certamente irá impedir um atacante com recursos limitados, similar aos recursos utilizados neste trabalho. Mesmo para um bom atacante, a chance é grande de ser impraticável a quebra de chave.

Para o caso do *WPA/WPA2-Enterprise*, todos os ataques no final se basearam em um ataque de força bruta com uso de dicionários e, sendo assim, todos dependem da senha usada para conseguir a quebra de chave. Como já foi falado sobre o ataque de força bruta, dado os recursos disponíveis, podem levar muito tempo para serem efetivos, isto dependerá da senha usada pelo usuário.

Nos ataques contra o *EAP-MD5* e o *EAP-LEAP*, os métodos foram iguais. O ataque de desautenticação foi utilizado visando levar o cliente a entender que o *AP* tenta desconectá-lo da rede, resultando no rompimento da conexão.

Assim como no caso do *WPA/WPA2-Personal*, não houve dificuldade em capturar o processo de *handshake*, já que estas soluções não apresentam defesas nativamente contra ataques do tipo *DoS*, o que pode prejudicar muito os serviços da rede, já que além de haver o risco de ter um intruso na rede, acessando todos os dados, ele poderá no mínimo causar uma interrupção do acesso aos serviços disponibilizados.

Para o restante dos tipos de *EAP* testados, *EAP-PEAP* e *EAP-TTLS*, a dificuldade se tornou maior, visto que o sucesso da obtenção do *challenge/response* depende muito da diferença da força dos sinais do *AP* legítimo e do *AP* falso. Isto porque mesmo após o cliente ser desconectado por meio de pacotes 802.11 de gerência forjados, o dispositivo ao tentar reconectar-se enviará pacotes *probe request* buscando a rede na qual estava conectada e fará sua escolha baseado na força e qualidade do sinal dos pacotes de *probe response* recebidos (dos *AP* falso e do *AP* legítimo) pelo cliente.

Os resultados foram positivos quanto a tentativa de conexão do cliente desconectado ao *AP* falso. Deixando o *AP* falso mais próximo do que o *AP* legítimo do cliente, o sinal do primeiro prevaleceu sobre o segundo, fazendo que fosse possível a captura do *challenge/response* no processo de autenticação.

A Tabela 4-3 faz uma comparação entre os cinco tipos de *EAP* mencionados neste projeto levando em conta quatro importantes aspectos: autenticação do servidor, autenticação do cliente, facilidade de implantação e segurança.

4.2 ANÁLISE DAS SOLUÇÕES

Quanto ao uso do *WEP* em redes sem fio, hoje em dia não é nada seguro utilizá-lo como solução. As vulnerabilidades são inúmeras: há o reuso de chaves, devido ao tamanho do vetor

de inicialização (*IV*) ser não muito grande (24 bits) e ser mandado em texto claro, prejudicando a integridade das mensagens; não há mudança constante das chaves, que deve ser configurada em cada estação toda vez que se muda, dificultando o processo; os algoritmos usados (*RC4*, *CRC-32*) são vulneráveis e não trazem proteção considerável; um atacante pode até mesmo descobrir o *keystream* usado para criptografar os pacotes sem precisar saber a senha.

Logo, não deve ser usada em situação nenhuma. No entanto, caso essa seja a única opção disponível, algumas melhorias podem ser usadas, ainda que frágeis também. Primeiramente, é possível configurar a rede para que não seja divulgado o *SSID* em pacotes *broadcast*. Isto garante que apenas clientes que conhecem a rede irão acessá-los. Porém, ao autenticar-se, o cliente manda o nome da rede em texto claro, sendo passível de ser capturado por alguém que esteja capturando os pacotes. Outra medida que pode ser adotada é a filtragem pelo endereço *MAC*. Tenta-se aqui garantir que apenas usuários com dispositivos autorizados irão acessar à rede. Contudo, é fácil de forjar o endereço *MAC* de qualquer dispositivo, e se algum atacante tem ciência do endereço *MAC* de um dispositivo ele será capaz de autenticar-se na rede sem problemas.

O *WPA* e o *WPA2-Personal* apresentaram resultado melhores em comparação ao *WEP*, porém sofrem da vulnerabilidade do ataque da força bruta e de negação de serviço (desautenticação). Apesar de a senha poder ser estabelecida de 8 a 64 caracteres, hoje em dia existem até serviços em nuvem capazes de realizar com um poder de processamento muito maior que de um simples notebook. Assim, nunca se deve economizar quando se trata do tamanho e complexidade da chave usada.

	EAP-MID5	EAP-LEAP	EAP-TLS	EAP-TTLS	PEAP
Server Authentication	None	Password hash	Certificate	Certificate	Certificate
Client Authentication	Password hash	Password hash	Certificate	MSCHAP(v2), EAP, CHAP	EAP
Ease of Deployment	Easy	Difficult	Difficult	Moderate	Moderate
Security	Insecure	Insecure	Secure	Secure	Secure

Tabela 4-3 Comparação entre os diferentes tipos de *EAP*.[14]

EAP-MD5 e *EAP-LEAP* confirmam por ser os mais inseguros já que enviam parte das credenciais do usuário em texto claro (nome do usuário) seguido do *challenge/response* em *hash*, sendo vulnerável ao ataque de força bruta. Ainda que o *EAP-LEAP* faça uma espécie de autenticação por parte do servidor, evitando a chance de um ataque *Man-in-the-middle*, no qual o atacante encaminha tráfego entre as duas partes, capturando tudo, ele faz uso do *MS-CHAPv2*,

que é o responsável por enviar as mensagens de credenciais sem encriptação. Além disso, é um protocolo proprietário (Cisco), dificultando sua implementação.

O *EAP-TTLS* e o *EAP-PEAP* são mais seguros do que os anteriormente citados, por agregarem o túnel *TLS* de encriptação, acabando com o problema do ataque de força bruta. No entanto, os dois são vulneráveis ao *MITM*, já que em certos casos o estabelecimento do túnel pode não acontecer, deixando brecha para que um atacante possa capturar a sessão. Além disso, com um uso de falso *AP* é possível que o atacante consiga obter as mesmas credenciais que o *EAP-LEAP* e o *EAP-MD5* não protegem, tornando a cair no problema do ataque de dicionário.

O *EAP-TLS* é o mais seguro de todos, por mudar o tipo de autenticação por parte do cliente: ao invés de fazer uso de nome de usuário e senha, faz uso de certificado digital, assim como o servidor o faz. Isso evita que o ataque de força bruta aconteça. Porém, isso o torna de difícil implementação também, sendo necessário um grande esforço para gerenciar os certificados de cada cliente, além de distribuí-los a todos [11].

4.3 PROPOSTA DE SOLUÇÃO

Levando em conta os resultados obtidos com os ataques e o conhecimento de outros ataques a que estão sujeitos os mais diversos métodos de segurança e autenticação em redes sem fio, a Figura 4-2 mostra a topologia considerada ideal para um ambiente minimente empresarial, como por exemplo o LabRedes, onde se tem um grande fluxo de pessoas (alunos, funcionários e professoes) e onde não se tem uma certa restrição de acesso. O ideal então é que se tenha uma solução na qual os usuários possuem suas credenciais, possibilitando maior restrição quanto ao uso dos recursos da rede e maior controle sobre as ações realizadas na rede por parte dos usuários (consumo de banda, quantidade de acessos, etc.), o que pode ser monitorado por meio do *daloRADIUS*. Buscou-se usar as soluções de *software* livre, que são de larga aceitação e não trazem custos para implantação.

O método de segurança é o *WPA2-Enterprise*. *WPA2* porque conta com o algoritmo de encriptação *CCMP*/*AES*, que é sabido por ser mais seguro que o *TKIP*/*RC4* (pertencente ao padrão *WPA*). Ressalta-se aqui que o uso do *Enterprise* deve ser empregado em qualquer ambiente que não seja domiciliar e de comércio (restaurantes, consultórios, etc.). Hoje em dia, até mesmo nas residências, o uso da rede sem fio é alto, devido ao poder de penetração desta tecnologia. Em ambientes de grande fluxo de pessoas, como universidades, empresas de grande porte, ou seja, locais que além de fornecerem acesso à internet, possuem uma infraestrutura onde informações sensíveis são armazenadas, tais como conta de usuários e banco de dados com informações sigilosas.

A topologia desta rede é ilustrada na Figura 4-2.

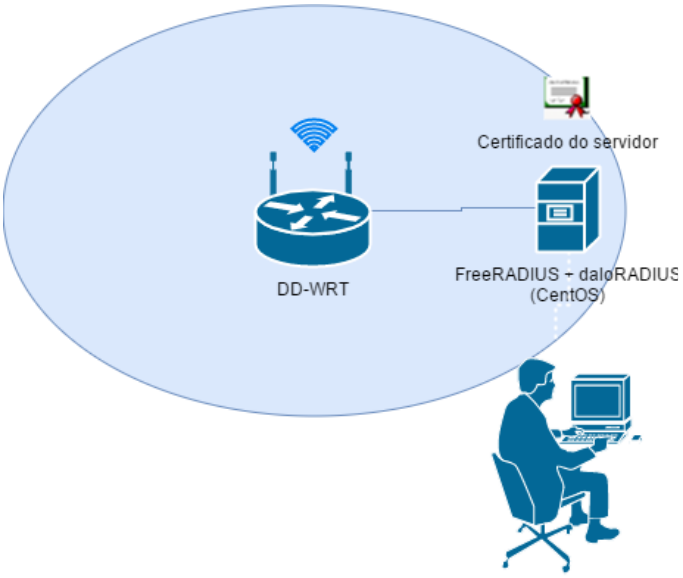

Figura 4-2 Topologia proposta para uso no Laboratório de Redes.

Para o roteador, foi escolhido o *firmware* DD-WRT, que é de *software* livre e amplamente utilizado, contando com as mais diversas características e com vasto suporte para várias marcas de roteador, como *Linksys, D-Link, TP-Link e Asus*. É capaz de fazer controles avançados de *QoS*, tem suporte para *VPN, WDS* wireless bridging, autenticação via *RADIUS*, implementação de portal para acesso à rede (*Chillispot*), suporte a *VLAN* e suporte ao *NAT*.

Como servidor *RADIUS*, foi escolhido o *FreeRADIUS*, que é responsável atualmente por cerca de um terço das autenticações na rede mundial. Tem suporte para os principais tipos de *EAP* (*EAP-TLS*, EAP-*TTLS* e *EAP-PEAP*), podendo ser integrado com o *daloRADIUS*, uma interface gráfica do servidor que possibilita o gerenciamento de usuários (contabilidade, autorização e autenticação via web), além de prover o controle da localidade dos *APs* (integração com GoogleMaps) e, também, inclui outras ferramentas como gerenciamento do banco de dados e interface de acesso do usuário, fazendo com que seja possível que o próprio cliente veja seu uso na rede e outras informações (faturamento, dados pessoais, etc.).

O método de autenticação desta topologia é o *802.1X/EAP* baseado em porta, no qual o *EAP* utilizado é o *EAP-PEAP*. A escolha foi feita levando-se em conta o custo de implementação e segurança provida. Este método faz uso do túnel de encriptação *TLS*, autenticação por parte do usuário via nome e senha, além da haver autenticação por parte do servidor *RADIUS* com o uso de certificado digital.

Ainda, vale lembrar o cuidado que se deve ter com a gerência do servidor *RADIUS* e do(s) roteador(es) usados dentro da rede. Visto que é necessário realizar configurações,

verificações do funcionamento de ambos os elementos da rede, é fundamental que se faça uso de meios seguros quando se realiza operações nos mesmos. Portanto, protocolos que cifram a troca de dados sensíveis dentro da rede, no caso dados de gerência e configuração, devem ser usados. Como principais exemplos confiáveis, temos o acesso remoto por *SSH* (*Secure Shell)*, acesso via web por *HTTPS* (*HyperText Transfer Protocol Secure*) e a gerência dos dispositivos de rede com o uso do *SNMPv3* (*Simple Network Management Protocol version 3*). Além disso, as senhas e permissões de acessos aos dispositivos devem ter a mesma cautela de quando se estabelece a senha da rede.

Finalmente, sempre a política de segurança deve ser conhecida por todos os que acessam à rede. Os métodos de segurança utilizados protegem de forma eficiente a rede, tendo vulnerabilidades que não são fáceis de serem exploradas, como a quebra de chave. Todavia, usuários que não tem conhecimento dos cuidados que devem tomar com as credenciais que usam, podem comprometer a segurança de toda a rede. Logo, o administrador da rede deve criar políticas de segurança que garantam o cuidado por parte dos usuários com suas contas e escolha de senhas. Técnicas de engenharia social que visam enganar clientes legítimos da rede estarão sendo evitadas assim.

CONCLUSÕES E TRABALHOS FUTUROS

As redes sem fio constituem hoje o principal portal de acesso à internet para a maioria das pessoas, mais especificamente as redes *WLAN*. Cada vez mais os usuários dependem de seu uso e cada vez mais se tem informações sensíveis na rede. É fundamental que as redes sem fio continuem a evoluir e entregar métodos de autenticação e de encriptação cada vez mais seguro, garantindo integridade, autenticidade e privacidade.

Os padrões de segurança atuais podem ser suficientemente seguros quando empregados de forma correta como um todo, ou seja, empregados corretamente na configuração, na acessibilidade e o estabelecimento de políticas de segurança que garantam a contribuição dos usuários para manutenção da rede, não a comprometendo.

Foi levando em conta esses fatores que se chegou a uma proposta de solução para o Laboratório da Engenharia de Redes de Comunicação, onde o acesso à *WLAN* acontece de forma livre na rede cabeada e na rede sem fio se dá por meio de uma senha somente. Considerando as informações e dados guardados na rede, deve-se ter uma maior precaução com a mesma. Esta solução engloba os aspectos necessários ao tipo de ambiente que é o laboratório: garante que cada um tenha suas credenciais de acesso à rede; provê uma plataforma de gerenciamento dos usuários e de contabilidade do uso da rede; traz um roteador capaz de exercer as mais avançadas funções, como *QoS*, *NAT*, *Access Control List* (*ACL*), portal para *login* e muitos outros; faz uso de um servidor *RADIUS* amplamente usado e com boa capacidade para expansão de processamento.

Porém, existem ataques capaz de poderem comprometer até mesmo uma rede considerada segura. Sendo assim, como trabalho futuro, pretende-se verificar os elementos de rede que podem ser agregados para garantir uma segurança maior, tais como: sistema detector de intruso sem fio, capaz de evitar ataques como desautenticação, negação de serviço, falso *AP*, dentre outros [5]; um *firewall*, que seja capaz de bloquear tráfegos maliciosos contra a rede. Também, trabalhos vêm sendo feito buscando encontrar padrões nos ataques de forma que se consiga entrar a origem do mesmo [12]. Finalmente, a infraestrutura empregada será aproveitada para implementar autenticação baseada em porta na rede cabeada.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] A. Cassola, W. Robertson, E. Kirda, G. Nubir. A Practical, Targeted, and Stealthy Attack Against WPA Enterprise Authentication. Northeastern University, College of Computer and Information Science.

[2]M. Vanhoef, F. Piessens. Practical Verfiicationi of WPA-TKIP Vulnerabilities.

[3] M. Junaid , Muid Mufti, M.Umar Ilyas. Vulnerabilities of IEEE 802.11i Wireless LAN CCMP Protocol. International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol:1, No:11, 2007

[4]G. M. Silva, J. N. Souza. Uma análise dos mecanismos de segurança de redes locais sem fio e uma proposta de melhoria. FC, UFU.

[5]T. D. Nguyen, D. H. M. Nguyen, B. N. Tran, H. Vu, N. Mittal. A lightweight solution for defending against deauthentication/disassociation attacks on 802.11 networks. Vietnam National University, University of Texas at Dallas.

[6]A. G. Linhares, P. A. da S. Gonçalves. Uma Análise dos Mecanismos de Segurança de Redes IEEE 802.11: WEP, WPA, WPA2 e IEEE 802.11w. UFPE, CIn.

[7]D. D. Coleman, D. A. Westcott, B. E. Harkins, S. M. Jackman. CWSP – Certified Wireless Security Professional. Sybex, 2013.

[8]D. Szilagyi, A. Sood, T. Singh. *RADIUS*: A Remote Authentication Dial-In User Service. *InSight*: RIVIER ACADEMIC JOURNAL, VOLUME 5, NUMBER 2, FALL 2009

[9]V. H. L. Antunes, Frontend 2.0 para Gestão de *RADIUS*. Faculdade de Eng. Da Univ. de Porto. Julho, 2009.

[10]W. L. Pritchett, D. D. Smet. Kali Linux Cookbook. Packt Publishing, 2013.

[11] B. Antoniewicz. 802.11 Attacks. Disponível em: http://www.mcafee.com/us/resources/white-papers/foundstone/wp-80211-attacks.pdf. Acesso em: 18/10/2015.

[12]R. Ragupathy, R. Sharma. Detecting Denial of Service Attacks by Analysing Network Traffic in Wireless Networks. International Journal of Grid Distributing Computing, Vol. 7, no. 3, 2014.

[13]D. van der Walt. FreeRADIUS Beginner's Guide. Packt Publishing, 2011.

[14]M. Alamanni. Kali Linux Wireless Penetration Testing Essentials. Packt Publishing, 2015.

[15]L. Tal. daloRADIUS User Guide Version 0.9-9. Enginx, May, 2011.

[16]A.S. dal Pont, Portal de Autenticação com Monitoramento na Infraestrutura do Instituto Federal Catarinense. IFC, 2013

APÊNDICE

Neste capítulo, são apresentadas as principais configurações usadas para realização do trabalho, a fim de que se apresente todos os passos necessários para repetir os cenários e redes configuradas.

A.1 CONFIGURAÇÃO FREERADIUS

Para que seja possível instalar sem nenhum problema, fará-se a autenticação no servidor como *root* para ter todos os privilégios de administrador executando o comando `#sudo su`, e preenchendo a senha de super usuário. Por padrão, as placas de redes estão configuradas por *DHCP*, no arquivo interfaces, localizado no diretório `/etc/network/`, que será editado para o modo manual e estático, como mostra na Figura A.1-1. Para concretizar a alteração destas configurações nas interfaces, executa-se o comando `#/etc/init.d/networking restart` para reiniciar os serviços da rede. Para confirmar as configurações que foram feitas, utiliza-se o comando `#ifconfig`.

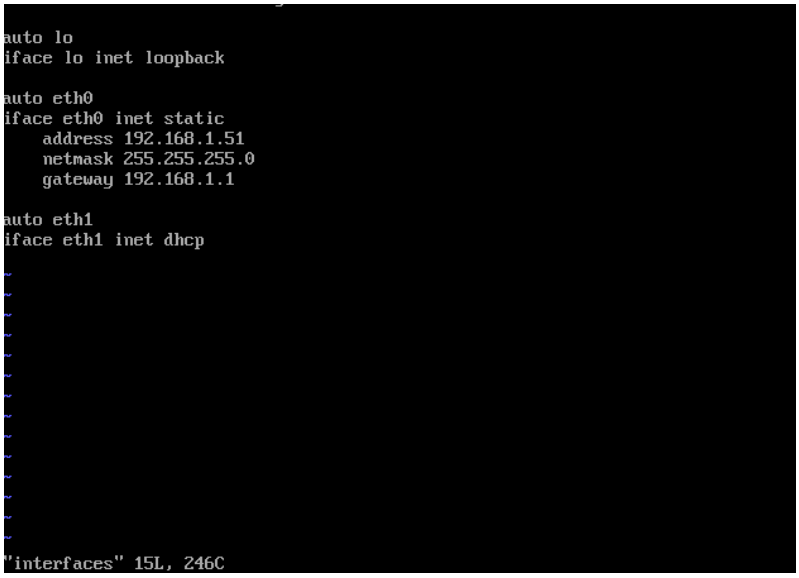

Figura A.1-1 Configurações de rede no servidor do *FreeRADIUS*.

Antes de instalar as ferramentas, é feita uma verificação e atualização do banco de dados do repositório do sistema operacional Ubuntu com o comando `#apt-get update && apt-get upgrade`, para deixar o repositório completo. Após o repositório atualizado, e antes da instalação das ferramentas, autenticação e monitoramento, instala-se algumas aplicações para o correto funcionamento dos mesmos. Através do comando `#aptitude install apache2 ssl-cert php5-gd php-db php-pear libapache2-mod-php5 mysql-server`

`mysql-client freeradius freeradius-mysql phpmyadmin freeradius-utils gcc` logado como *root*, o *Apache2*, php, *FreeRADIUS* e o *MySQL* serão instalados para autenticar, guardar no banco de dados e fazer o *hotspot*.

Após a instalação usa-se os seguintes comandos para poder ter acesso ao navegador via *HTTPS*: executam-se os comandos `#a2enmod ssl, #a2ensite default-ssl`. Em sequência, o comando `#service apache2 restart`, para reiniciar os serviços do Servidor Web, com as alterações sendo efetuadas.

Com o *FreeRADIUS* instalado, cria-se o banco de dados *MySQL* usado para guardar as autenticações dos usuários logados, dentre outras informações de autorização e contabilização. Para criar um banco de dados, entra-se como usuário *root* em seu terminal usando os seguintes comandos [13]:

```
#MySQL –u root –p
#Enter password
mysql> CREATE DATABASE radius → comando para criar o banco de dados;
mysql> quit
```

Os scripts SQL *schema.sql* e *nas.sql* devem ser adicionados com os comandos a seguir:

```
#mysql –u root –p radius < /etc/freeradius/sql/mysql/shema.sql;
#mysql –u root –p radius < /etc/freeradius/sql/mysql/nas.sql.
```

Para modificar e cadastrar no banco de dados usa-se alguns privilégios com os comandos:

```
#mysql –u root –p
Enter password
mysql> GRANT ALL PRIVILEGES ON radius.* TO 'radius'@'localhost'
IDENTIFIED BY 'mysqlsecret';
mysql> FLUSH PRIVILEGES → banco de dados com os devidos privilégios;
mysql> quit
```

O primeiro arquivo a ser editado é o `sql.conf` que fica no diretório `/etc/freeradius/` modificado para o conteúdo da Figura A.1-2.

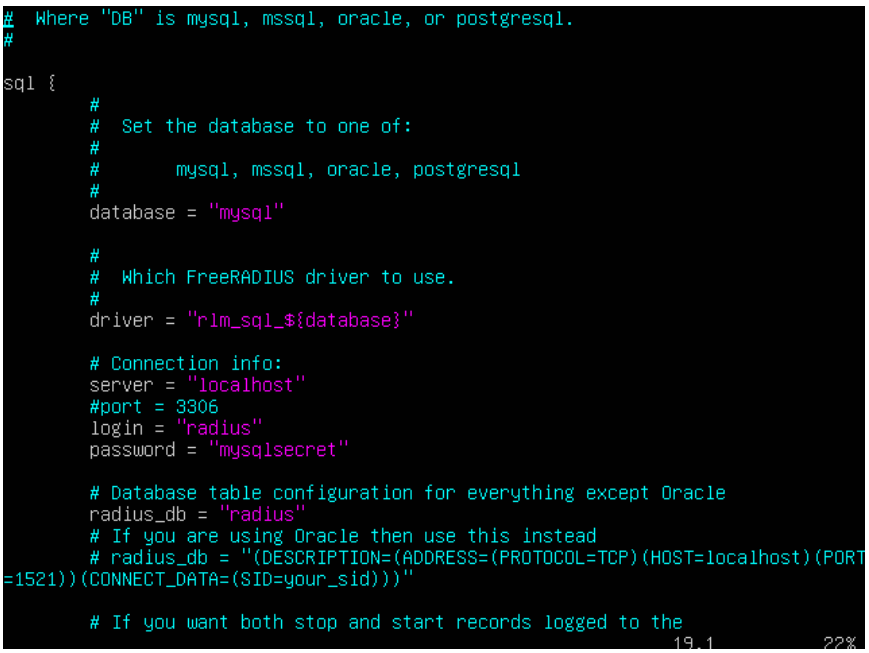

Figura A.1-2 Configuração de conexão do banco de dados com o servidor *RADIUS*.

O segundo arquivo a ser editado está em `/etc/freeradius/client.conf`, que altera-se para a palavra chave *mysqlsecret* (ou qualquer outro a escolha), como mostra a Figura A.1-3.

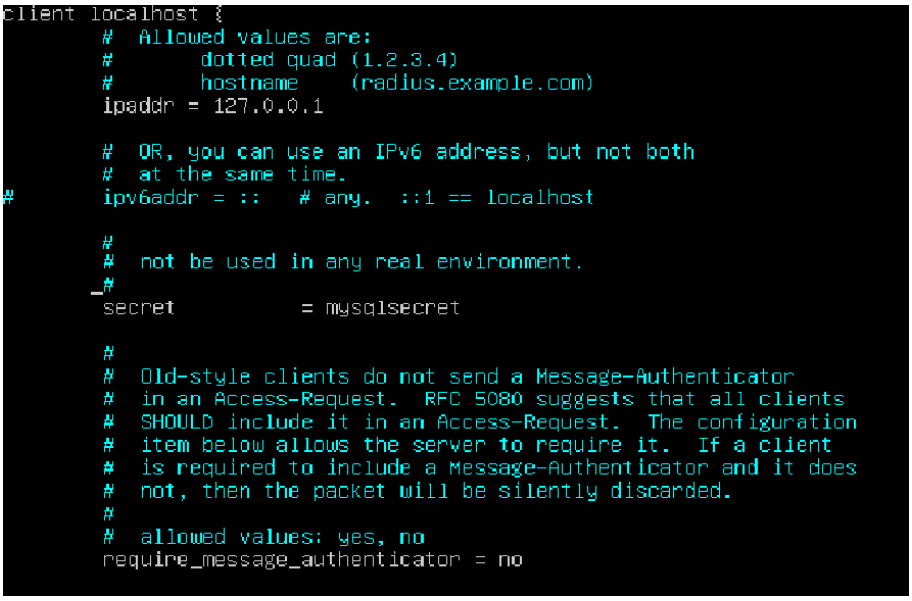

Figura A.1-3 Configuração do segredo compartilhado entre *AP* e *RADIUS*.

Todos os usuários do *FreeRADIUS* são cadastrados ou alterados no arquivo *users* no diretório `/etc/freeradius/` após o cadastramento reinicia-se o sistema operacional com o comando `#reboot`. Após o reinício do SO interrompe-se o serviço *FreeRADIUS* com o comando `#/etc/init.d/freeradius stop`, para a depuração e verificação de erros com o comando `#freeradius -XXX`. Se o resultado da depuração mostrar êxito, executa-se o comando pelo teclado `control+c` para sair.

Aqui nestas configurações adiciona-se os usuários no próprio arquivo do *FreeRADIUS*. Para segurança e armazenamento destes usuários, cadastra-se direto no banco de dados *MySQL* que foi criado nas etapas anteriores. Para fazer esta ligação teremos que editar novamente o arquivo `sql.conf` no diretório `/etc/freeradius/`, tirando o comentário da linha `readclients=yes`. Para finalizar a configuração da autorização do *FreeRADIUS* com o banco tem que se editar o arquivo `radius.conf` no diretório `/etc/freeradius/` tirando o comentário da linha `INCLUDE sql.conf`.

Nesta etapa faz-se a inserção de um usuário e senha no *MySQL* para fazer o teste na conexão. Após reinicializar o serviço do *FreeRADIUS*, usa-se o comando radtest com o nome e a senha cadastrada, o *IP* ou nome do servidor e a porta, e por último a palavra-chave do *FreeRADIUS* para comprovar o correto funcionamento da integração do mesmo e o banco MySQL como mostra a Figura A.1-4.

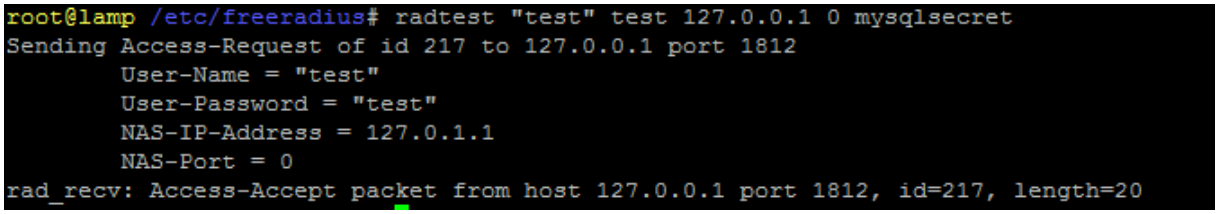

Figura A.1-4 Teste do servidor *RADIUS* com usuário de teste.

A.2 CONFIGURAÇÃO daloRADIUS

O *daloRADIUS* é uma ferramenta *Web* para gerencia o servidor *FreeRADIUS*. A ferramenta possui gerenciamento de cadastros de clientes, relatórios gráficos e faturamento. Para o funcionamento do *daloRADIUS* é necessário instalar um servidor *Web*, que já vem previamente instalado atualmente nas diversas distribuições Linux (CentOS, RedHat, etc.).

Os comandos abaixo mostram como foi instalado o *daloRADIUS* (*FreeRADIUS* já instalado). Inicialmente acessamos a pasta de temporários com o comando:

```
# cd /tmp
```

O seguinte comando, baixa o pacote do *daloRADIUS* dentro da pasta temporários e deve ser executado em apenas uma linha:

```
#wget
'http://downloads.sourceforge.net/project/daloradius/daloradius/dalo
radius-0.9-8/daloradius-0.9-8.tar.gz'
```

O comando abaixo foi usado para descompactar o arquivo:

```
# tar xvzf daloradius-0.9-8.tar.gz
```

O próximo comando move o arquivo para outro local a pasta do servidor *web*:

```
# mv /tmp/daloradius-0.9-8 /var/www/daloradius
```

Para mudar o proprietário e o grupo dos arquivos de modo que o servidor web possa acessá-los o comando usado foi o seguinte:

```
# chown -R www-data:www-data /var/www/daloradius
```

Assim, a interface gráfica do *daloRADIUS* poderá ser acessada digitando no browser o IP da máquina virtual que o hospeda.

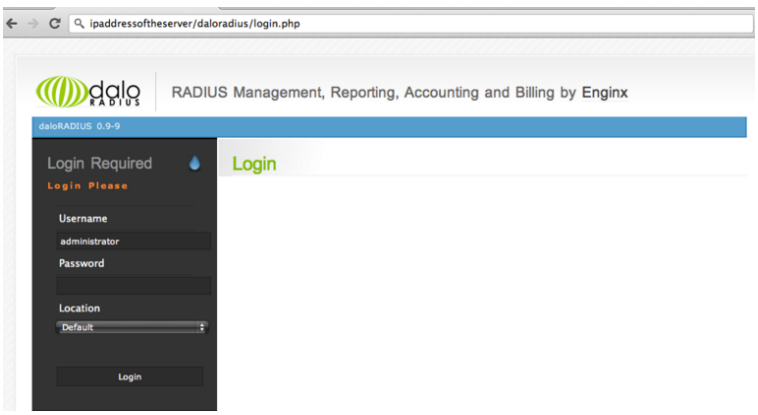

Figura A.2-1 Interface de autenticação do *daloRADIUS* para gerenciador da rede.

Uma opção alternativa existente é o uso da *appliance* fornecida pelo próprio criador do *daloRADIUS* (Lirian Tal). Ele disponibiliza um arquivo .ova onde tanto *daloRADIUS* como

FreeRADIUS estão já configurados e pronto para serem usados, havendo apenas necessidade de colocar a interface de rede no modo *Bridge* (vide A.4) para que a máquina virtual com o servidor *RADIUS* seja visível na rede.

Além da plataforma de gerenciamento do *daloRADIUS*, esta *appliance* vem com outras funcionalidades, como mostra a Figura A.2-2:

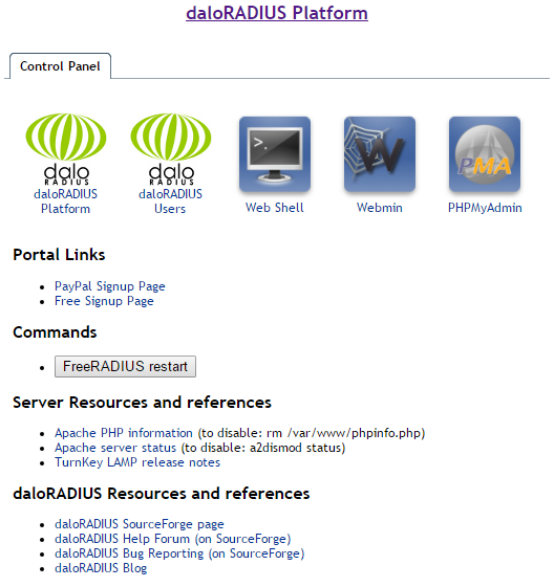

Figura A.2-2 Ferramentas disponíveis para gerência no daloRADIUS.

Há a plataforma para acesso dos usuários também, onde podem editar configurações de sua conta (dados pessoais, senha, etc.), além de poderem verificar outras funcionalidades que podem ser implementadas tais como acompanhamento do uso dos dados, o plano de dados e faturas.

Ainda, há a possibilidade de fazer uso do *Web Shell* para acessar mais facilmente a interface *shell* do sistema sem a necessidade de realizar uma conexão via *SSH*. *Webmin* e *PHPMyAdmin* servem para gerenciar componentes do servidor e banco de dados, respectivamente.

A.3 CONFIGURAÇÃO ADAPTADOR WI-FI

Neste trabalho, era necessário um adaptador sem fio exclusivo para o *Kali Linux*, de modo que ela funcionasse em modo promíscuo para que seja capaz de capturar e injetar pacotes nos momentos de ataque às redes sem fio. Como servidor *RADIUS* e *Kali Linux* estavam sendo usados no mesmo hospedeiro (diferentes máquinas virtuais com diferentes IPs), o adaptador TP-LINK WN722N foi usado para servir como interface de rede sem fio do Kali.

Para que seja possível utilizá-lo na máquina virtual, os seguintes passos devem ser seguidos:

Passo 1: após inserir o adaptador *wi-fi* no hospedeiro, vá nas configurações de sua máquina virtual por meio da interface do *VirtualBox*. Clique em *USB* e ao clicar no ícone de adição de novo dispositivo *USB*, irá aparecer o adaptador *wi-fi* inserido no hospedeiro.

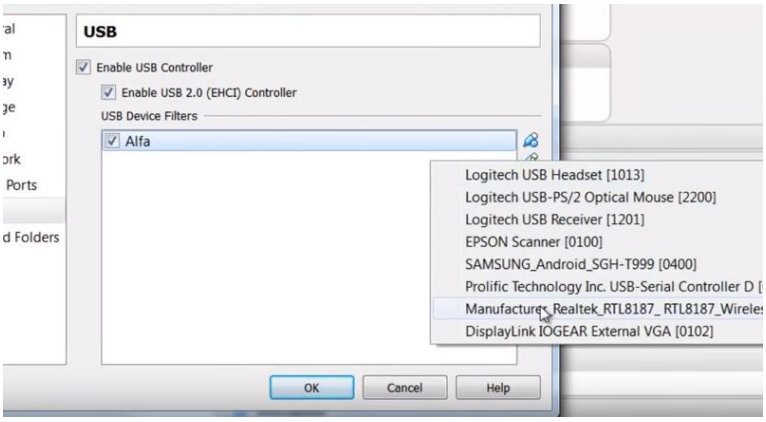

Figura A.3-1 Identificação do adaptador de rede sem fio.

Passo 2: Após clicar para adicionar o adaptador como dispositivo *USB*, clique sobre ele e aparecerão os dados sobre o mesmo. Dê especial atenção aos campos *Vendor ID* e *Product ID*, que serão usados no processo.

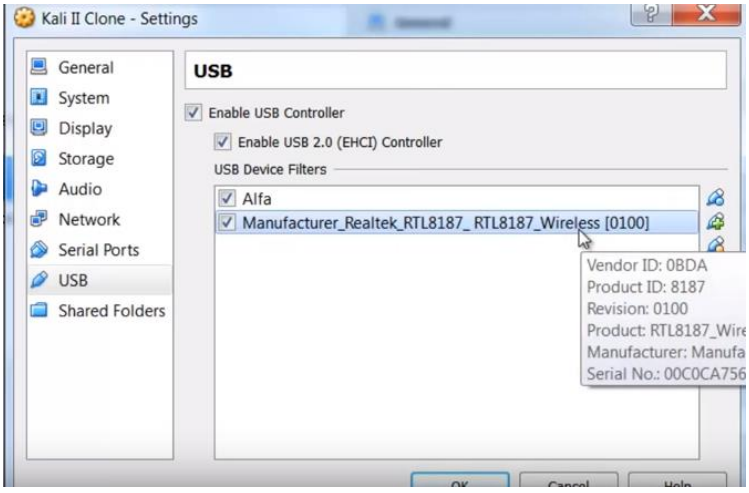

Figura A.3-2 Identificação dos campos *Vendor ID* e *Product ID*.

Passo 3: Remova o adaptador da lista e clique no ícone de adição manual de um dispositivo *USB*. Nomeie o mesmo como quiser e em seguida preencha os campos de *Vendor ID* e *Product ID* com os dados anotados no passo anterior.

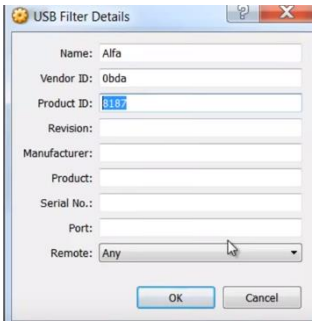

Figura A.3-3 Adição do adaptador de rede sem fio à máquina virtual do Kali.

Passo 4: Remova fisicamente o adaptador *wi-fi* do hospedeiro e inicie a máquina virtual do Kali. Feito isso, reinsira o adaptador no hospedeiro. Aparecerá um ícone no canto inferior direito da tela indicando que algum dispositivo foi conectado.

Figura A.3-4 Reconhecimento do adaptador feito pela máquina virtual.

Após digitar *ifconfig* no terminal do Kali, deverá aparecer a interface de rede sem fio do adaptador inserido, como mostra a imagem a seguir.

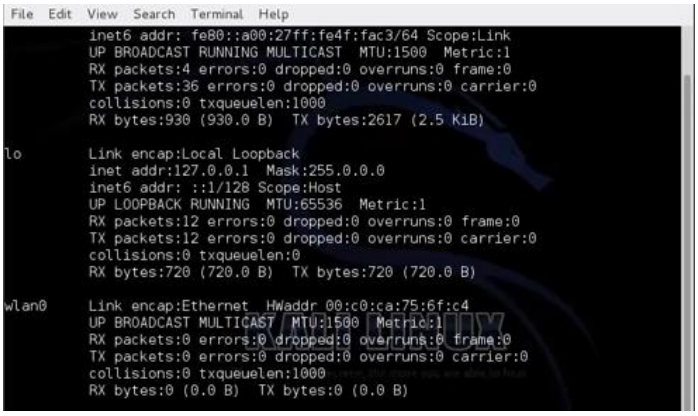

Figura A.3-5 Adaptador *wireless* conectado à máquina virtual (wlan0).

Passo 5: Para que o adaptador esteja em modo promíscuo, digite no terminal o seguinte comando: airmon-ng start wlan0. A interface de rede será renomeada para *mon0* ou *wlan0mon*, basta checar pelo comando do ifconfig. Desta forma, o adaptador estará pronto para ser usado pelas ferramentas do Kali de injeção e captura de pacotes.

A.4 CONFIGURAÇÃO ORACLE VM VIRTUALBOX

Para construção da topologia deste trabalho, alguns componentes foram necessários (roteador, adaptador *wi-fi*, servidor *RADIUS*, laptop para ataques) e, portanto, visando suprir a necessidade da aquisição de diversos componentes físicos, foi feito o uso de máquinas virtuais para o servidor *RADIUS* e para o uso da distribuição *Kali Linux*.

As configurações a fazer na ferramenta mais relevantes para este trabalho são as seguintes:

- Configuração do adaptador 1 no modo *Bridge*:
    - Neste caso, usa-se o adaptador da rede sem fio para realizar um *Bridge* e obter um *IP* diferente do hospedeiro para que se possa distinguir a máquina virtual do hospedeiro (que tem apenas o papel de prover a conexão da internet às maquinas virtuais, isto é, servidor *RADIUS*)

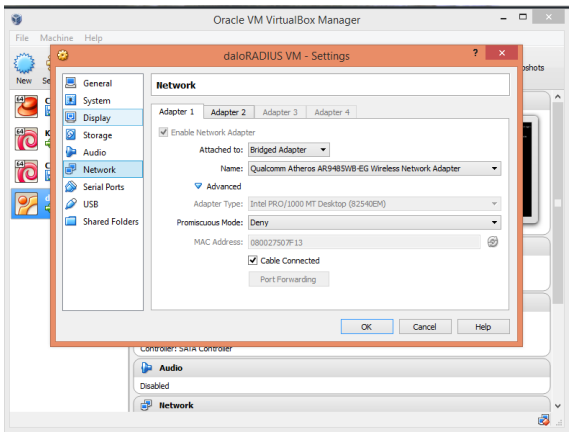

Figura A.4-1 Adaptador 1 da máquina virtual em modo *Bridge*.

- Configuração do adaptador 2 no modo *Virtualbox Host-Only Ethernet*:
    - Aqui, usa-se o adaptador *Host-only* para que se possa ter uma rede isolada das demais. Desta forma, é possível que se crie uma restrição de acesso às maquinas virtuais, sendo possível garantir acesso apenas a dispositivos da mesma rede. É vantajoso também por evitar a necessidade de uma interface de rede física no hospedeiro, conectando as máquinas virtuais e hospedeiro.

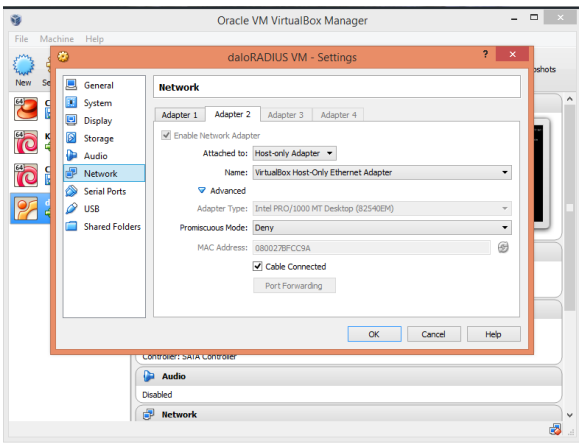

Figura A.4-2 Adaptador 2 da máquina virtual em modo *Host-Only Adapter*.

A.5 CONFIGURAÇÃO CERTIFICADOS DIGITAIS

Nos padrões *WPA* e *WPA2-Enterprise*, alguns métodos de autenticação *EAP* necessitam de um certificado digital apresentado pelo servidor de autenticação, para que o suplicante possa ter garantia de que está tentando se conectar ao servidor correto. Lembrando que os tipos de *EAP* que fazem uso de certificados digitais são os seguintes: *EAP-PEAP*, *EAP-TLS* e *EAP-TTLS*, sendo que o *EAP-TLS* é o único que trabalha com certificado digital no cliente também. É o de maior custo para implementação, porém traz uma garantia a mais de segurança, podendo o servidor verificar a legitimidade do cliente. Abaixo estão os passos necessários para a construção deste certificado.

Os arquivos de configuração de certificação do *FreeRADIUS* estão localizados na pasta /etc/raddb/certs/*.cnf. A maioria do conteúdo destes arquivos podem ser deixados como estão, já que eles configuram vários parâmetros do *OpenSSL*. Alguns campos (que serão indicados a seguir em negrito) devem ser editados para suprir suas necessidades.

```
ca.cnf

...
[ req ]
prompt = no
distinguished_name = certificate_authority
default_bits = 2048
input_password = any
output_password = any
x509_extensions = v3_ca

[certificate_authority]
countryName = BR
stateOrProvinceName = Radius
localityName = Somewhere
organizationName = Example Inc.
emailAddress = admin@admin.com
commonName = "Certificate Authority Sample"
...
```

```
server.cnf

...
[ req ] prompt = no
distinguished_name = server
default_bits = 2048
input_password = any
```

```
output_password = any

[server] countryName = BR
stateOrProvinceName = Radius
localityName = Somewhere
organizationName = Example Inc.
emailAddress = admin@admin.com
commonName = "Server Certificate Sample"
...
```

Será necessário editar o arquivo `client.cnf` apenas se o *EAP-TLS* for utilizado.

Após terem sido editados os arquivos `ca.cnf` e server.cnf, é necessário recriar os certificados do servidor e o do *CA* (*Certification Authority*). Neste processo, qualquer certificado anterior será deletado, portanto, salve quaisquer certificados previamente criados que deseje guardar:

```
# cd /etc/raddb/certs
# make
```

Caso a mensagem "make: Nothing to be done for 'all'" apareça, será necessário deletar alguns arquivos manualmente:

```
# rm –f  *csr *key
# make
```

Caso contrário, deverá aparecer a criação de chaves e certificados por parte do *OpenSSL* como mostrado abaixo:

```
openssl req -new -x509 -keyout ca.key -out ca.pem -config ./ca.cnf
Generating a 2048 bit RSA private key
.....................................................
etc.
```

Para criar certificados aos clientes, o processo é similar. É necessário editar o arquivo client.cnf, e modificar os campos pertinentes na seção [client] no fim do arquivo. Após salvar o arquivo, execute o seguinte comando:

```
# make client
```

Será criado um novo certificado para cliente em `client.pem`. Ele pode ser importado para o cliente e usado no *EAP-TLS*.

A fim de que se criem vários certificados para outros clientes, devem ser mudados os valores dos campos `emailAddress` e `commonName`. O *OpenSSL* cria certificados únicos para cada cliente, e irá reclamar caso tente criar diferentes certificados onde esses campos se repitam.

O passo final é importar os certificados *CA* para a máquina dos clientes e o jeito mais simples para fazê-lo é importando o arquivo `ca.der` da pasta `/etc/raddb/certs` para a área de trabalho *Windows* do cliente. Dando um clique duplo no arquivo deverá iniciar o processo de importar o certificado para o acervo de certificados do Windows.

A.6 CONFIGURAÇÃO FALSO AP E RADIUS

No ataque de força bruta a uma rede *WPA/WPA2-Enterprise*, pode ser o caso de se deparar com uma autenticação onde o *EAP-TTLS* ou *EAP-PEAP* esteja sendo usado, e as credenciais estarão sendo encriptadas em um túnel *TLS*, devido ao uso de certificado por parte do servidor. Para que algum atacante seja capaz de ter acesso aos dados de autenticação (desafio e resposta), o uso de um falso *AP* e um falso *RADIUS* é necessário e, para isso, os *softwares freeradius-wpe* e *hostapd* devem ser configurados. Os passos a seguir mostram como.

Primeiramente, o *freeradius-server-2.1.12* pode ser instalado copiando e executando os seguintes comando como root no terminal:

```
wget ftp://ftp.freeradius.org/pub/radius/old/freeradius-server-
2.1.12.tar.bz2
wget https://raw.github.com/brad-anton/freeradius-
wpe/master/freeradius-wpe.patch
tar -jxvf freeradius-server-2.1.12.tar.bz2
cd freeradius-server-2.1.12
patch -p1 < ../freeradius-wpe.patch
./configure
make
make install
ldconfig
```

Depois de executados os comandos, verifique se a instalação foi feita corretamente executando o comando `radiusd –v`. Isso deverá mostrar *FreeRADIUS-WPE* na saída deste comando

```
root@kali:~/freeradius-server-2.1.12# radiusd -v
radiusd: FreeRADIUS-WPE Version 2.1.12, for host i686-pc-linux-gnu,
built on Nov 22 2015 at 13:12:55
```

Em seguida, o *hostapd v2.0* deve ser instalado. Para tanto, os seguintes comandos devem ser executados como *root*:

```
wget http://hostap.epitest.fi/releases/hostapd-2.0.tar.gz
tar -zxvf hostapd-2.0.tar.gz
cd hostapd-2.0/
cd hostapd/
cp defconfig .config
apt-get install libnl-dev
apt-get update
apt-get install libssl-dev
make && make install
```

Para que as credenciais sejam capturadas com sucesso de um usuário *Windows*, uma mudança deve ser feita no arquivo `/usr/local/etc/raddb/modules/mschap`. A linha

`with_ntdomain_hack` não deve estar comentada e deve estar como sim. Desta forma, as informações de desafio/resposta podem ser capturadas com sucesso:

```
# Windows sends us a username in the form of
# DOMAIN\user, but sends the challenge response
# based on only the user portion.  This hack
# corrects for that incorrect behavior.
with_ntdomain_hack = yes
```

Deste passo em diante, o servidor *RADIUS* já pode ser executado. Execute-o com o comando `radiusd -X`.

Com o servidor *RADIUS* já configurado, é necessário que se tenha o *AP* divulgando uma falsa rede para o cliente que está sendo atacado. O *hostapd* será usado em conjunto com o adaptador de rede sem fio (funcionando como rádio) para a divulgação de tal rede.

Este *software* requer um arquivo de configuração e os dados a seguir podem ser inseridos neste arquivo para o uso do mesmo:

```
interface=wlan0
driver=nl80211
ssid=CorpNetwork
logger_stdout=-1
logger_stdout_level=0
dump_file=/tmp/hostapd.dump
ieee8021x=1
eapol_key_index_workaround=0
own_ip_addr=127.0.0.1
auth_server_addr=127.0.0.1
auth_server_port=1812
auth_server_shared_secret=testing123
wpa=2
wpa_key_mgmt=WPA-EAP
channel=1
wpa_pairwise=TKIP CCMP
```

Estes dados podem ser salvos em um arquivo chamado `hostapd.conf` (o nome fica a critério de quem o cria).

Finalmente, para inciar o *AP*, o seguinte comando é executado:

```
hostapd ./hostapd.conf
```

A tela do terminal deverá mostrar a seguinte saída, simbolizando que o *hostapd* está pronto para aceitar conexões dos clientes:

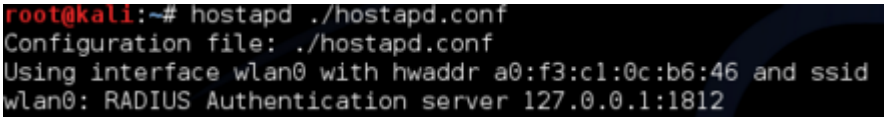

Figura A.6-1 Execução do *hostapd*.