PEDRO DINIZ DE MAGALHÃES

INTEGRAÇÃO DE FERRAMENTAS DE ANÁLISE E AUDITORIA EM REDES *WI-FI* COM A FINALIDADE DE QUEBRA DE CHAVES WEP/WPA

LAVRAS - MG

2011

PEDRO DINIZ DE MAGALHÃES

INTEGRAÇÃO DE FERRAMENTAS DE ANÁLISE E AUDITORIA EM REDES *WI-FI* COM A FINALIDADE DE QUEBRA DE CHAVES WEP/WPA

Monografia de Graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

Orientador

Prof. Luiz Henrique Andrade Correia

LAVRAS - MG

2011

PEDRO DINIZ DE MAGALHÃES

INTEGRAÇÃO DE FERRAMENTAS DE ANÁLISE E AUDITORIA EM REDES *WI-FI* COM A FINALIDADE DE QUEBRA DE CHAVES WEP/WPA

Monografia de Graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

Aprovada em *7 de junho de 2011*

Prof. Dr. Joaquim Quinteiro Uchôa

Profa. Ma. Juliana Galvani Greghi

Prof. Luiz Henrique Andrade Correia

Orientador

LAVRAS - MG

2011

Dedico aos meus pais.

AGRADECIMENTOS

Agradeço em primeiro lugar àqueles sem os quais eu não estaria presente. Aos meus pais, Crispim e Márcia, agradeço pelos anos de paciência e compreensão (e não foram poucos) e mais que isso, pelos anos de incentivo e pela confiança em mim depositados.

Ao meu orientador Professor Luiz Henrique por acreditar no potencial deste trabalho, sempre disponibilizando seu tempo para reuniões e revisões e simples conversas sobre redes.

À minha irmã, Luiza, agradeço pela existência, pelo companheirismo.

Aos meus amigos Daniel (Nanuque), Willian (Urso) e Mateus (Piloto) que foram os mais presentes durante os meus últimos semestres na Universidade e a todos aqueles que conheci e de alguma forma contribuíram para este trabalho, direta ou indiretamente, agradeço.

Agradeço à cafeína (sob a forma de Coca-Cola ou café) que me impediu de dormir quando eu mais queria e menos podia.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS E ACRÔNIMOS

ACK	-	*Acknowledgement*
AP	-	*Access Point*
APT	-	*Advanced Packaging Tool*
CCMP	-	*Counter Mode with Cipher Block Chaining Message Authentication Code Protocol*
CRC	-	*Cyclic Redundancy Check*
CSMA/CA	-	*Carrier Sense Multiple Acces with Colision Avoidance*
CTS	-	*Clear-To-Send*
DIFS	-	*Distributed Inter Frame Spacing*
DNS	-	*Domain Name System*
DoS	-	*Denial of Service*
EAP	-	*Extensible Authentication Protocol*
GUI	-	*Graphical User Interface*
ICV	-	*Integrity Check Vector*
IDE	-	*Integrated Development Environment*
IP	-	*Internet Protocol*
ISO	-	*International Standarization Organization*
IV	-	*Initialization Vector*
LLC	-	*Link Layer Control*
MAC	-	*Medium Access Control*
MIC	-	*Message Integrity Code*
MSK	-	*Master Session Key*
NAV	-	*Network Allocation Vector*
PMK	-	*Primary Master Key*
PSK	-	*Pre Shared Key*
PTK	-	*Pairwise Transient Key*
RTS	-	*Request-To-Send*
SIFS	-	*Short Interframe Spacing*
TKIP	-	*Temporal Key Integrity Protocol*
Wi-Fi	-	*Wireless Fidelity*
WEP	-	*Wireless Equivalent Privacy*
WPA	-	*Wi-Fi Protected Access*
WPA2	-	*Wi-Fi Protected Access Version 2*
WYSIWYG	-	*What You See Is What You Get*

SUMÁRIO

RESUMO

Este trabalho apresenta um estudo sobre a segurança nas redes sem fio 802.11, comumente chamadas de redes *Wi-Fi* (*Wi-Fi Alliance*). São abordados os protocolos de segurança WEP, WPA e WPA2 e ataques a redes sem fio. São também levantados os principais ataques específicos aos protocolos de segurança. Este trabalho tem como objetivo implementar uma ferramenta que integre ferramentas específicas para quebra de chaves WEP e WPA em modo texto, a uma interface gráfica desenvolvida na linguagem C++ com o auxílio do *framework* multiplataforma Qt. Esta ferramenta gráfica auxilia na auditoria e segurança das redes *Wi-Fi*.

Palavras-chave: WEP; WPA; *Wi-Fi*; Segurança; Aircrack; Quebra de Chaves.

ABSTRACT

This work shows firtly a study about security protocols in 802.11 wireless networks, also called Wi-Fi networks due to their relation to the certificate program provided by Wi-Fi Alliance. The main themes are the security protocols WEP, WPA, WPA2, and wireless networks attacks. Secondly, studies about specific attacks to the security protocols were made. The objective of this work is to implement a tool composed by a specific toolset that can brake WEP and WPA keys. This toolset is essencially text-mode, and a graphic user interface was developed using C++ and the Qt framework

Keywords: WEP; WPA; Wi-Fi; Security; Aircrack; Cracking.

1 INTRODUÇÃO

1.1 Contextualização

As redes sem fio estão se tornando cada vez mais populares em meios residenciais e corporativos devido à mobilidade e flexibilidade proporcionadas aos usuários. Tal popularidade implica em um aumento exacerbado do uso de dispositivos que acessam a rede, muitas vezes por meios não seguros (conexões não criptografadas, por exemplo) avivando a curiosidade de usuários avançados que tem em seu conhecimento uma arma para tomar o controle das redes, criando um ambiente de insegurança.

No Brasil, a popularização dos *notebooks*, *netbooks*, *smartphones* e *tablets* alavancou o uso das redes sem fio, fruto também das inúmeras promoções de vendas casadas de computadores e *access points*. Isso tem feito com que uma grande quantidade de usuários com pouco (ou nenhum) conhecimento em configuração de dispositivos *wireless* tenha utilizado métodos de autenticação e encriptação ultrapassados, como o WEP. Métodos WPA e WPA2, porém com chaves fracas, ainda não são barreiras às invasões e decriptações. Outros usuários, como acontece em muitos casos, utilizam a rede totalmente aberta facilitando ainda mais a sua invasão.

Em 2008 o site *Secure List* mantido pela empresa de *software* de segurança *Kaspersky*[1] publicou o resultado de um *wardriving*[2] feito na cidade de São Paulo. Segundo Dmitry Bestuzhev, o autor da publicação, os dados[3] não chegam a ser

[1] www.kaspersky.com

[2] Ato de sair pesquisando as características das redes sem fio onde elas se encontram disponíveis.

[3] http://www.securelist.com/en/analysis/204791997/Wardriving_in_Sao_Paulo_Brazil

alarmantes, mas há muito que melhorar. Das redes encontradas, 24% não usam métodos de encriptação e 50% utilizam o WEP. Ou seja, há um total de 74% de redes sem fio facilmente exploráveis na cidade.

1.2 Objetivos

Este trabalho tem como objetivo principal a implementação de uma ferramenta gráfica, para ambiente Linux, que auxilie na auditoria de redes sem fio quanto à força das senhas utilizadas pelos usuários e também os protocolos disponibilizados nos dispositivos. A ferramenta proposta será baseada nos aplicativos da família *Aircrack-NG*, amplamente utilizadas na invasão de redes *Wi-Fi*.

Os objetivos específicos são a análise superficial do funcionamento dos protocolos de segurança e das ferramentas de auditoria e ataques à redes sem fio bem como o levantamento dos requisitos de segurança em redes *Wi-Fi* para a compreensão de como se dá a quebra de chaves WEP e WPA/WPA2.

1.3 Estrutura do Trabalho

O presente trabalho está definido de forma a proporcionar uma sequência que auxilie no entendimento e desenvolvimento.

O Capítulo 1 apresenta o contexto de redes sem fio e a falta de segurança da mesma. Os Capítulos 2, 3 e 4 apresentam o referencial teórico da monografia, abordando, respectivamente, conceitos básicos de redes sem fio, segurança em redes sem fio e usabilidade, necessários ao desenvolvimento do mesmo. A metodologia do trabalho é mostrada no capítulo 5.

No capítulo 6 são apresentados os resultados juntamente com algumas considerações e o capítulo 7 finaliza o trabalho com as conclusões.

2 CONCEITOS EM REDES *WIRELESS*

2.1 Redes *Wireless*

Redes *wireless* ou redes sem fio são redes de computadores e/ou outros dispositivos que comunicam-se sem a necessidade de cabos, utilizando o ar como meio de transmissão de dados.

As soluções empregadas em redes sem fio fornecem conexões em uma área de cobertura limitada. Em geral essa área de cobertura varia de 10 a 100 metros de uma estação até um ponto de acesso. Essas soluções sem fio permitem transferência de dados por duas vias (ida e volta) tipicamente utilizadas em ambientes corporativos e domésticos (BARNES *et al.*, 2002). As redes sem fio também são comumente conhecidas por redes *Wi-Fi* devido ao programa de padronização de interoperabilidade de dispositivos proposto pelo órgão *Wi-Fi Alliance*.

O *Wi-Fi Alliance* é atualmente a maior associação de fornecedores de dispositivos necessários ao funcionamento de uma rede sem fio. Graças ao seu programa de certificados, dispositivos *wireless* de diferentes marcas e que seguem as especificações do padrão IEEE 802.11 funcionam de forma compatível, sem perda de desempenho ou de funcionalidades.

As redes *Wi-Fi* possuem algumas vantagens sobre as redes cabeadas, como mobilidade, facilidade e velocidade de implantação, flexibilidade e custo, todas decorrentes da ausência da necessidade de conexão de cabos. A implantação de uma rede *Wi-Fi* é facilitada, necessitando apenas do ponto de acesso e a distribuição das chaves de autenticação. Quanto à redução de custos, um bom exemplo seria fazer a conexão entre dois prédios: o gasto com cabos de um prédio a

outro seria mais alto do que os gastos com a criação uma conexão 802.11 (GAST, 2005).

Correia (2007) enumera desvantagens do uso de redes *Wi-Fi* : menor largura de banda, maior taxa de erros e topologia dinâmica. A baixa largura de banda já está sendo sanada com o desenvolvimento de padrões com velocidade mais alta como o IEEE 802.11n, que atinge a taxa nominal de 600 Mbps. A taxa de erros nas redes *Wi-Fi* são por vezes milhares ou até milhões de vezes maiores do que nas redes *ethernet* cabeadas, decorrentes das iinterferências das várias frequências presentes no dia-a-dia. O grande problema com a topologia dinâmica das redes é a necessidade de algoritmos de roteamento mais complexos, o que demanda dispositivos com maior capacidade de processamento.

Mesmo com suas desvantagens, as redes sem fio são alvo de pesquisas e desenvolvimento na indústria e no mercado. A escolha de qual padrão de rede sem fio utilizar depende da sua aplicação e de seu alcance. Redes de curta distância (até 10 metros) podem utilizar a tecnologia *bluetooth* (IEEE SOCIETY COMPUTER, 2005). As redes de dados para celulares, atualmente usam o 3G (CHEN; ZHANG, 2004). E para redes domésticas e corporativas utilizam-se as redes *Wi-Fi* que seguem o padrão IEEE 802.11 (IEEE SOCIETY COMPUTER, 2007).

2.2 O padrão IEEE 802.11

O 802.11 é um padrão para redes sem fio locais que define os modelos de implementação para as camadas física e de enlace do modelo de referência OSI/ISO (DAY; ZIMMERMANN, 1983)(GAST, 2005).

Na camada física são definidas as características físicas do sinal como frequência, taxa de transferência, e de transmissão e recepção dos dispositivos. A Tabela 2 abaixo mostra essas características físicas dos padrões encontrados atualmente no mercado.

Padrão do IEEE	Taxa	Frequência
802.11	1 e 2 Mbps	2,4 Ghz
802.11a	54 Mbps	5,8 Ghz
802.11b	11 Mbps	2,4 Ghz
802.11g	54 Mbps	2,4 Ghz
802.11n	600 Mbps	2,4 Ghz

Tabela 2: Características dos padrões 802.11.

A camada de enlace, assim como nas redes *Ethernet*, subdivide-se em duas outras camadas, LLC (*Logical Link Control* - Controle de Enlace Lógico) responsável por tratar os dados provenientes da camada física e enviar para as camadas superiores, e camada MAC (*Medim Access Control* - Controle de Acesso ao Meio) que trata os métodos de acesso ao meio.

O método de acesso ao meio empregado nas redes sem fio 802.11 é o CSMA/CA (*Carrier Sense Multiple Acces with Colision Avoidance* - Acesso Múltiplo por Detecção de Portadora com Prevenção de Colisão) que faz uma reserva do meio utilizando um diálogo em quatro passos distintos: RTS - CTS - DATA - ACK.

A Figura 1 ilustra o funcionamento do CSMA/CA. O transmissor escuta o meio e caso ele não esteja livre, a estação **A** transmissora espera um tempo aleatório denominado período de *backoff*. Cada vez que o canal fica livre de transmissões, inicia-se uma contagem regressiva a partir do *backoff*, caso contrário, a

contagem fica parada. Se o canal estiver ocioso ou a contagem regressiva terminar no momento em que o canal estiver sem transmissões, a estação **A** transmissora espera um tempo chamado de DIFS (*Distributed Inter Frame Spacing*) e envia um quadro de controle RTS (*Request to Send*) requisitando permissão para transmitir os dados à estação **B** receptora.

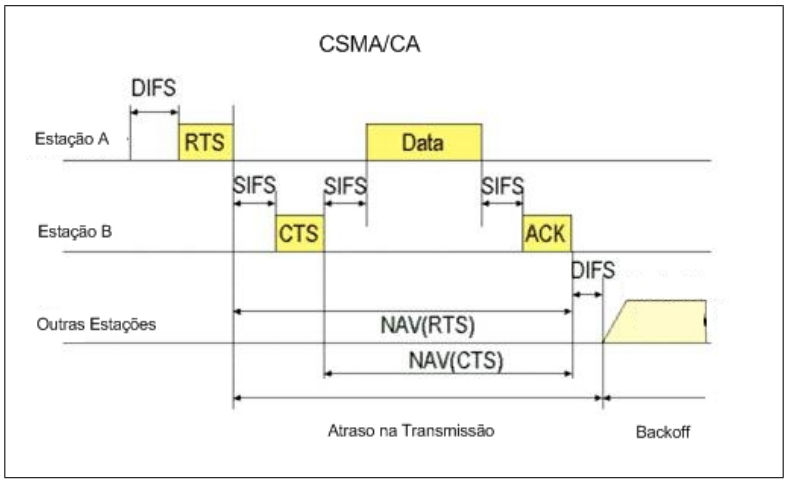

Figura 1: O método CSMA/CA (CORREIA, 2007).

O quadro RTS enviado por **A** possui informações como o tempo necessário à transmissão do quadro de dados e a recepção do quadro de controle ACK (*Acknowledgement*). As estações ao alcance de **A** recebem o RTS (enviado em *broadcast*) e ajustam o tempo de bloqueio de seus rádios, denominado NAV (*Network Allocation Vector*).

A estação *B* então envia um quadro CTS (*Clear to Send*) avisando que os dados podem ser enviados. As estações próximas a **B** também recebem o CTS que

contém as informações para que estas ajustem o período do NAV e bloqueiam seus rádios, evitando a transmissão durante esse tempo.

Para o cálculo do NAV das estações próximas a **A** são contabilizados o tempo estimado para a transmissão do quadro de dados e o recebimento da confirmação de recebimento, além do tempo que **A** e **B** esperam para enviar cada quadro do diálogo RTS - CTS - ACK - DATA, denominado SIFS (*Short Inter Frame Spacing*). Nas estações próximas a **B**, são considerados os tempos de SIFS, DATA e ACK a partir do recebimento do CTS.

Após o recebimento do CTS a estação **A** mais um período SIFS e então envia seu quadro de dados (DATA). Após o recebimento do quado de dados, **A** envia uma confirmação (ACK - *Acknowledgment*) que indica que a transmissão ocorreu corretamente.

Ao término desse processo as estações esperam um período DIFS e voltam a disputar o meio de transmissão.

O padrão 802.11 também fala sobre as topologias de uma rede *Wi-Fi*, na Seção 2.2.1 a seguir.

2.2.1 Topologia das redes 802.11

A topologia de uma rede 802.11 é composta pelos seguintes elementos, como mostrado por Kurose e Ross (2006):

- **Estação-base** - responsável pela criação do Conjunto de Serviços Básicos (BSS - *Basic Service Set*).
- **Estação sem fio** - estação que executa tarefas e se conecta a dispositivos sem fio. Um laptop é um exemplo de estação sem fio.

- **Enlace sem fio** - uma estação sem fio conecta-se a uma estação base por meio de um enlace sem fio.

A partir desses elementos, podemos definir duas formas de topologias para as redes sem fio. Quando as estações sem fio estão conectadas a uma estação-base, diz-se que estão operando em modo infra-estruturado, como na Figura 2, uma vez que os serviços de rede (atribuição de endereços, roteamento, por exemplo) estão sendo providos pela rede com a qual elas estão conectadas através da estação-base.

Neste caso, toda a comunicação entre as estações é intermediada pelo ponto de acesso raiz.

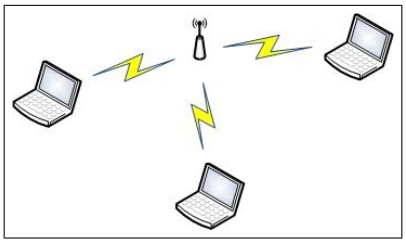

Figura 2: Rede sem fio em modo infra-estruturado.

Outra forma de comunicação é quando as estações estão conectando-se umas às outras, formando várias células de comunicação sem fio. Esta topologia é conhecida como *Ad-Hoc*. Nesse caso serviços como roteamento, atribuição de endereços, DNS entre outros, são providos pelas próprias estações.

Existe, ainda, uma topologia sendo padronizada pelo grupo IEEE 802.11s que forma uma rede híbrida, conhecida como redes em malha ou *Mesh*. Neste caso, os APs específicos para redes *Mesh* se comunicam uns com os outros formando um *backbone wireless*. As estações, que podem ser qualquer dispositivo que acesse

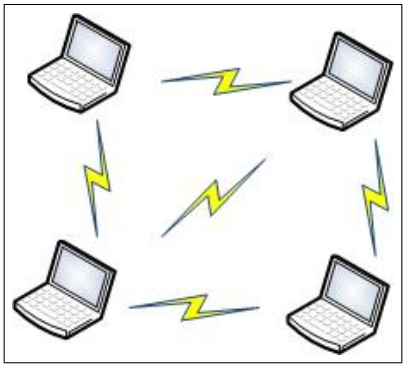

Figura 3: Rede sem fio em modo *Ad-Hoc*.

a rede *Mesh* (desde um notebook até mesmo um outro AP funcionando em modo infra-estruturado em conjunto com outras estações) acessam a internet por meio da rede (REDWAN; KIM, 2008). A Figura 4 mostra um exemplo.

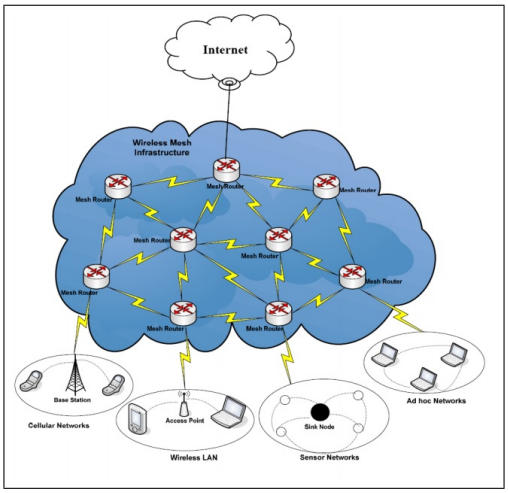

Figura 4: Rede *Mesh* (REDWAN; KIM, 2008).

Definidas as topologias de uma rede *Wi-Fi*, é necessário definir um padrão para os quadros trocados entre os elementos da rede. O padrão 802.11 possui um quadro definido de forma diferenciada descrito na Seção 2.2.2.

2.2.2 O formato do quadro MAC do padrão 802.11

A apresentação dos campos do protocolo se faz necessária para a compreensão das futuras aplicações, como a segurança em redes *Wi-Fi* descrita no próximo capítulo.

O formato dos quadros MAC do padrão 802.11 apresentam algumas diferenças em relação aos quadros de uma rede cabeada. Uma delas é a presença de 4 campos de endereço os quais muitas vezes não são todos usados. O formato do quadro 802.11 genérico é mostrado na Figura 5, bem como a explicação sobre seus campos. Os quadros são lidos da esquerda para a direita.

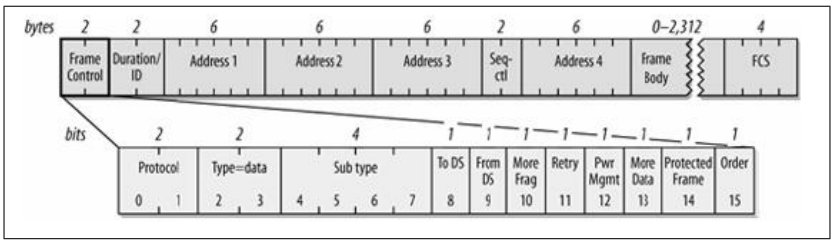

Figura 5: Quadro MAC 802.11 (GAST, 2005).

Campo *frame control* ou campo de controle de quadro: os quadros 802.11 iniciam-se com um campo de dois **bytes** denominado de controle de quadro. Os campos são descritos a seguir:

Protocol Version - este sub-campo de 2 bits de comprimento denomina qual a versão do protocolo 802.11 está sendo utilizada. Como atualmente apenas uma foi desenvolvida, esta recebe o valor 0.

Type* e *Subtype - esses campos em conjunto identificam o tipo de quadro que está sendo usado, como por exemplo um quadro CTS ou RTS. O campo *Type* possui 2 bits e identifica os tipos gerenciamento, controle e quadros de dados. O campo *Subtype* tem 4 bits e uma tabela completa sobre as combinações para o campo *Subtype* pode ser encontrada no manual do IEEE 802.11 (IEEE SOCIETY COMPUTER, 2007).

FromDS* e *ToDS - segundo Kurose e Ross (2006) esses valores podem ser interpretados da seguinte forma: *FromDS* mensagem oriunda do AP e *ToDS* que está sendo enviada para o AP. Quando os dois sub-campos estão com o valor 0, é considerado um sistema em modo *Ad-Hoc*. É utilizado um *bit* para cada sub-campo .

More Fragments bit - este *bit* indica se o quadro de dados teve que ser fragmentado para ser transmitido, em caso positivo este campo recebe o valor 1.

Retry bit - o valor 1 neste sub-campo indica que o pacote foi retransmitido e ajuda a eliminar duplicatas na estação receptora.

Power Management bit - indica se a estação transmissora entrará em modo de economia de energia ao terminar a transferência.

More Data bit - os *Access Points* utilizam este campo para indicar a uma estação receptora que existe pelo menos mais um pacote endereçado a ela. Isso

evita que a estação entre em modo de economia de energia antes do fim da transmissão.

Protected Frame bit - também conhecido como WEP *bit*. Indica se o dado sofreu encriptação antes de ser enviado.

Order bit - indica se os pacotes devem ser entregues em uma ordem ou não.

O campo *Duration/ID* vem logo em seguida do controle de quadro e utiliza 2 *bytes*. É utilizado para indicar o período de ocupação do meio de transmissão pelos quadros RTS/CTS. É também utilizado para o cálculo do ajuste do NAV.

Os campos de endereço logo após o campo *Duration/ID* são caracterizados da seguinte forma:

- *Address 1* - endereço MAC do receptor.
- *Address 2* - endereço MAC do transmissor.
- *Address 3* - depende dos bits *FromDS* e *ToDS*. Se *FromDS*=1, então é o endereço da estação original que transmitiu a mensagem. Se *ToDS*=1, é o endereço final da mensagem.

O campo *Sequence Control* possui 16 bits para controlar a sequência dos quadros pertencentes à mesma mensagem, e também faz o reconhecimento de mensagens duplicadas.

O campo de dados (*Frame Body*) possui até 2.312 *bytes* para dados e o campo de CRC faz o *checksum* de 32 bits do quadro.

3 SEGURANÇA EM REDES *WI-FI*

O fato de o meio de transmissão das redes sem fio ser aberto tem feito com que a segurança tenha sido um dos grandes focos de atenção de seus desenvolvedores. As redes sem fio não possuem limites físicos bem definidos, o que permite que qualquer pessoa que esteja dentro do alcance de uma certa rede possa capturar pacotes ou tentar alguma forma de invasão. Para tentar contornar essa situação, os dispositivos compatíveis com o padrão IEEE 802.11 e, consequentemente, com o padrão *Wi-Fi* possuem mecanismos que tentam assegurar a confidencialidade e integridade dos dados através de sistemas de autenticação.

3.1 Algumas Medidas de Segurança para Redes sem Fio

Wang e Zhang (2006) mostra algumas medidas de segurança que podem ser tomadas em redes sem fio. Em suma, a idéia é evitar o acesso de usuários não autorizados à rede.

Os filtros de endereço MAC tornam as redes restritas, por exemplo, apenas os endereços MAC registrados no *Access Point* (AP) podem ter acesso à rede, associando-se ao AP. Além disso, um sistema de autenticação e criptografia também se faz necessário seja ele WEP, WPA, WPA2 ou um servidor *Radius*, ou um misto dessas tecnologias (WANG; ZHANG, 2006).

Quanto ao problema da falta de limites bem definidos para redes sem fio, se faz necessário o controle da potência do rádio, evitando que o alcance da rede ultrapasse os limites desejados.

Loo (2008) enumera também as responsabilidades dos fabricantes de dispositivos *wireless*:

- os fabricantes de roteadores deveriam tornar as configurações de segurança um padrão, sendo possível desativá-las, mas o usuário seria avisado dos riscos.

- pequenos programas residentes que verificassem o número de usuários em um determinado tempo deveriam ser incorporados aos roteadores. Arquivos de *logs* podem ser úteis na verificação de intrusões.

- sempre que um método *hacker* fosse descoberto, atualizações de *software* e/ou *firmware* (se existente) deveriam ser lançadas assim que possível.

Da mesma forma que os fabricantes têm suas obrigações, os usuários também tem uma parcela de responsabilidades quanto a segurança das redes. Para Loo (2008), os usuários devem ser vigilantes quanto as suas informações e também estudar as formas de proteção para uma utilização mais segura da rede.

3.2 Protocolos de Segurança em Redes *Wi-Fi*

Conforme LASHKARI *et al.* (2009), a tecnologia de redes sem fio tem ganhado muita popularidade nos últimos anos. Porém, o uso de padrões de segurança rígidos não têm crescido na mesma velocidade. Existem atualmente três grandes padrões de segurança para redes 802.11: WEP (*Wired Equivalent Privacy*), WPA (*Wi-Fi protected Access*) e WPA2 (*Wi-Fi Protected Access version 2*) que serão descritos a seguir.

3.2.1 WEP - *Wired Equivalent Privacy*

O WEP (*Wired Equivalent Privacy*) foi concebido para prover a confidencialidade dos dados tal como ocorre em uma LAN (*Local Area Network* - Rede Local) cabeada. Isso é feito utilizando-se o algoritmo de encriptação RC4, como apresentado por Li *et al.* (2009). Foi desenvolvido em 1997 pela grupo de desenvolvedores do padrão 802.11b e foi o primeiro protocolo de encriptação a ser utilizado em redes sem fio, incorporando encriptação e chave compartilhada.

O WEP parte do princípio que a chave tem que ser conhecida por ambos os lados da conexão para que o sistema que criptografa/decriptografa os pacotes funcione. Essa chave conhecida por ambos os lados tem seu tamanho real reduzido devido à forma como o WEP usa o algoritmo RC4. A Figura 6 ilustra o funcionamento de um sistema WEP.

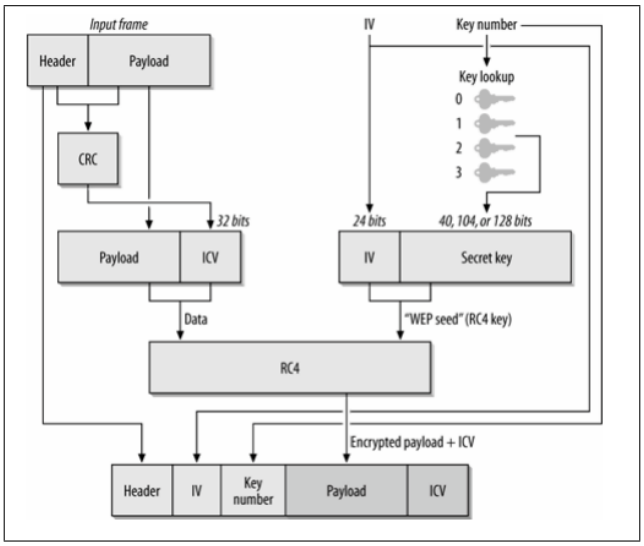

Figura 6: Operações criptograficas do protocolo WEP (GAST, 2005).

Primeiramente, uma chave WEP de 64 bits é constituída de 40 bits para os dados que são criptografados e 24 bits para o vetor de inicialização (IV - *Initialization Vector*), que é usado como uma chave de criptografia/decriptografia. A chave compartilhada e o IV são utilizados como semente em um gerador pseudo-randômico de números. O corpo e o cabeçalho da mensagem passam por um algoritmo de CRC (Cyclic Redundancy Check – verificação de redundância cíclica). O resultado do CRC é adicionado em um Vetor de Verificação de Integridade (ICV). Dessa forma o ICV e o os dados são concatenados. Os dados concatenados com o ICV e a chave gerada pelo IV e a chave compartilhada passam pelo algoritmo RC4. Ao fim são concatenados o resultado do RC4 e o ICV ao pacote.

O WEP apresenta dois modos de autenticação: *Open Sistem Authentication* e *Shared Key Authentication*. No modo *Open System Authentication*, Figura 7, primeiramente a estação envia uma requisição de autenticação ao *Access Point*. Este por sua vez responde simplesmente autenticando a estação, sem que haja a troca de informações ou uma negociação. Então a estação é associada ao AP. Para utilizar a rede, é necessário o conhecimento da chave WEP para a encriptação/decriptação dos pacotes.

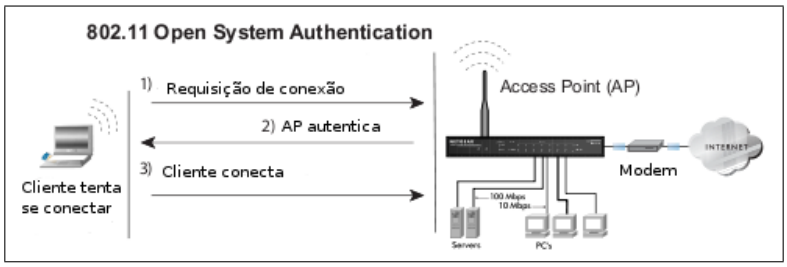

Figura 7: *Open System Authentication* (NETGEAR INC., 2005).

Quando os dispositivos estão utilizando o modo *Shared Key Authentication*, o processo decorre como ilustrado na Figura 8, conforme (NETGEAR INC., 2005):

1. A estação envia uma requisição de autenticação.
2. O *Access Point* responde enviando o que é chamado de *challenge text*, uma mensagem não criptografada.
3. A estação usa a chave WEP pré-configurada para criptografar a mensagem e enviá-la de volta ao *Access Point*.
4. O AP decripta então a mensagem enviada pela estação com a sua chave WEP pré-configurada. Se a mensagem decriptada for a mesma enviada no primeiro passo, a chave do AP é a mesma da estação e esta é autenticada.
5. A estação então se associa ao AP e pode usar a rede.

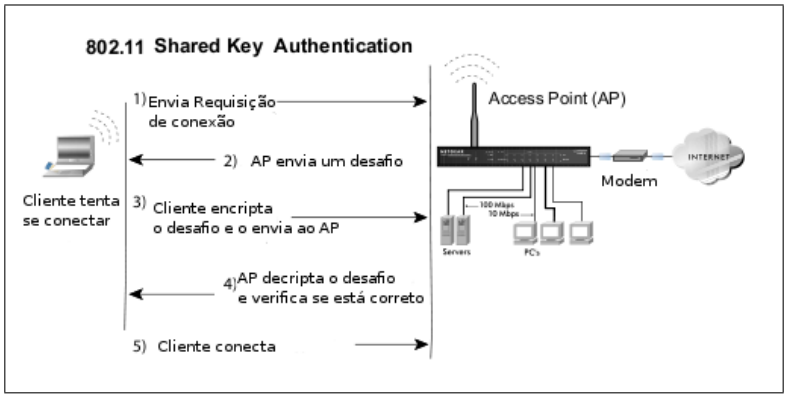

Figura 8: *Shared Key Authentication* (NETGEAR INC., 2005).

Uma grande falha do WEP se dá justamente pelo fato de o vetor de inicialização ser muito curto e passado de forma transparente, ou seja, ele não sofre

nenhuma encriptação. Uma senha WEP pode ser quebrada com a captura a partir de 5000 pacotes, uma vez que os vetores de inicialização são considerados curtos e se repetem com o tempo. Como o IV é um campo não criptografado, ele pode ser facilmente lido e utilizado em ataques de força bruta para quebra de senhas.

Outro problema do WEP é o uso do algoritmo CRC para garantir a integridade dos dados. O CRC é considerado uma boa forma de verificação de integridade, porém é falho para a criptografia por seu comportamento linear. Utilizando-se um tipo de indução, e coletando dados o suficiente, é possível gerar um dicionário de chaves e consequentemente descobrí-la (BROWN, 2003).

Descobertas as falhas do protocolo WEP, o *Wi-Fi Alliance* desenvolveu o WPA, descrito a seguir.

3.2.2 WPA - *Wi-Fi Protected Access*

O WPA foi criado pela *Wi-Fi Alliance* para suprir as falhas do WEP até que o grupo de estudos do IEEE 802.11 desenvolvesse um protocolo mais seguro (LI; GARUBA, 2008). O WPA foi desenvolvido com base nos *drafts*[4] do padrão 802.11i, que trata de formas mais seguras de autenticação do que as até então proporcionadas pelo padrão WEP de privacidade.

O protocolo WPA, assim como o WEP, utiliza o algoritmo de encriptação RC4 em seu processo de cifragem. O processo utilizado é diferente do empregado no WEP, permitindo uma encriptação mais complexa utilizando o protocolo TKIP (*Temporal Key Integrity Protocol*) e o algoritmo *Michael* ou MIC (*Message*

[4]Do inglês *draft.* Significa rascunho, trabalho não terminado

Integrity Code), descrito em (ALI; OWENS, 2010), que substitui o CRC para a verificação de integridade.

O protocolo TKIP implementa uma função que combina a chave inicial com o vetor de inicialização. O resultado dessa combinação passa pela criptografia RC4 juntamente com uma duplicata do vetor de inicialização. O resultado dessa passagem sofre uma operação de XOR[5] com os dados a serem criptografados, gerando por fim a mensagem a ser enviada (LASHKARI *et al.*, 2009), como mostrado na Figura 9.

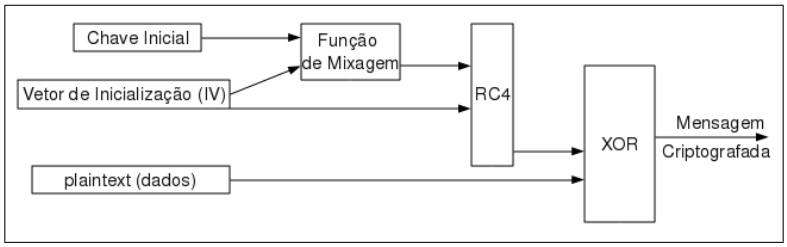

Figura 9: Encriptação com WPA-TKIP (LASHKARI *et al.*, 2009).

A Figura 10 ilustra a composição de um quadro 802.11 ao fim da execução do WPA. O campo de dados de um quadro MAC é composto pelo IV, um *byte* (na Figura detotado por Outro) para evitar IVs fracos, o IV estendido, os dados criptografados pelo algoritmo RC4, os MIC (*Message Integrity Code*) e o ICV (*Integrity Check Vector*). Na Figura 10, o endereço de destino, denotado por DA, e o endereço de origem, chamado de SA, são utilizados para compor o campo MIC juntamente com os dados e a chave de integridade do protocolo TKIP após passarem pelo algoritmo *Michael*.

[5]Operação lógica *Exclusive Or* (Ou Exclisivo). O retorno da operação somente é o valor verdadeiro se as entradas forem diferentes

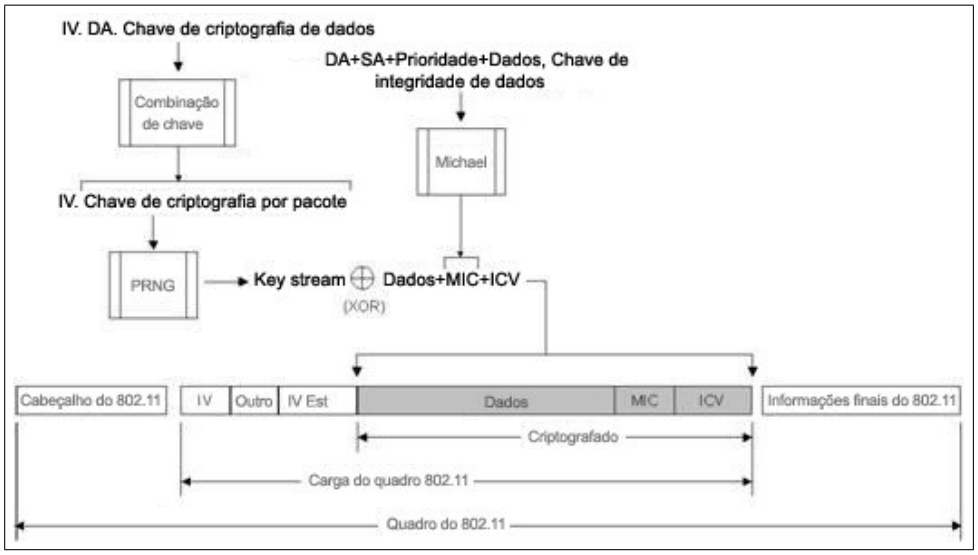

Figura 10: Composição de um quadro ao passar pela encriptação WPA (MICROSOFT, 2004).

Quanto à autenticação, o protocolo WPA implementa dois modos: PSK, ou *Pre-Shared Key* ou ainda WPA-Pessoal e WPA-Corporativo ou *WPA-Enterprise*. No modo WPA-PSK, toda a autenticação ocorre no *Access Point*. No modo WPA-Corporativo, o *Access Point* não autentica o usuário na rede. A autenticação é executada por um servidor e o AP fica responsável apenas por criar um canal seguro, utilizando o método de autenticação 802.1X aliado a um tipo de EAP (*Extensible Authentication Protocol*, para a troca de informações estação-servidor. O processo decorre seguindo a numeração da Figura 11, como descrito em (NETGEAR INC., 2005).

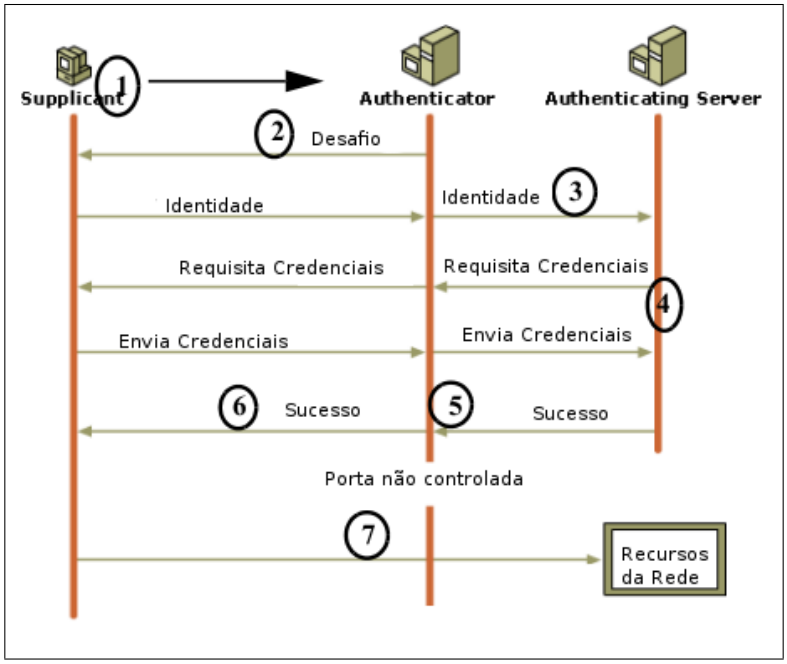

Figura 11: Processo de autenticação *WPA-Enterprise* (NETGEAR INC., 2005).

1. Uma estação descobre a rede através dos *beacons*[6] enviados pelo *Access Point* e solicita autenticação.

2. O AP responde solicitando uma identificação da estação.

3. A identificação é repassada ao servidor de autenticação. Nesse momento foi criada uma conexão segura (EAP) entre o servidor e a estação. Só trafegam pacotes EAP pela rede.

4. O servidor de autenticação verifica a identidade da estação.

[6]Beacon: quadro de gerenciamento contendo informações sobre a rede tais como SSID, tipo de encriptação e autenticação, entre outras.

5. Após a verificação o servidor envia uma mensagem de “sucesso”.

6. A mensagem de “sucesso” é repassada pelo AP à estação.

7. Se a estação foi aceita, é liberado o acesso aos serviços da rede.

Após a autenticação, inicia-se o processo de derivação da *Primary Master Key* (PMK) em que as chaves criptográficas serão estabelecidas em um processo conhecido como *4-way handshake* (Figura 12). Caso a autenticação tenha ocorrido no modo PSK, a PMK é a própria chave compartilhada. Se a autenticação foi pelo modo corporativo, então a PMK é a MSK. A PMK será utilizada apenas para a derivação das chaves temporárias (PTK - *Pairwise Transient Key*) utilizadas no protocolo TKIP. Ao final do *4-way-handshake*, é garantido que ambas as partes da comunicação possuem a mesma PTK e podem iniciar a troca de dados (VILELA; RIBEIRO, 2008).

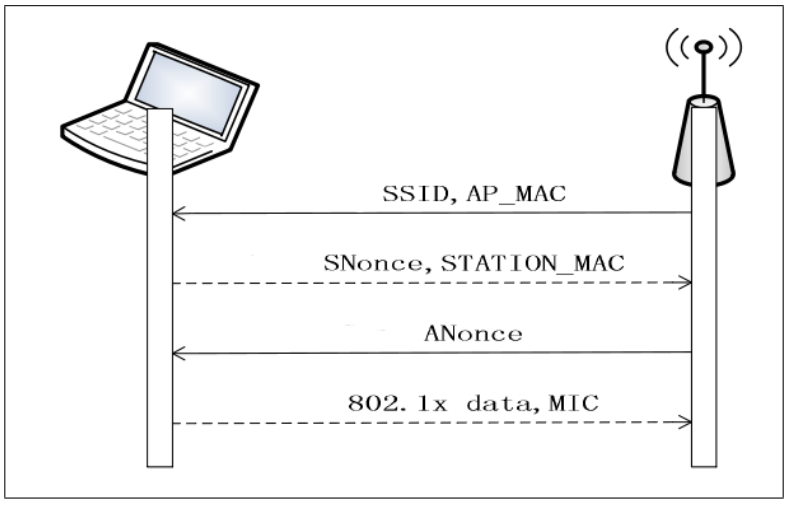

Figura 12: Processo *4-way handshake* (WANG *et al.*, 2010).

Em novembro de 2003, Robert Moskowitz lançou um artigo no qual explica a descoberta de uma falha no WPA em modo PSK que permite um ataque *offline* de dicionário lançado sobre os pacotes *4-way handshake* de uma rede sem fio. Conhecendo os dados obtidos no *4-way handshake* (Figura 12) e sabendo também quais as derivações utilizadas para se chegar ao MIC, pode-se então utilizar um dicionário composto por senhas em potencial para computar um MIC a partir da senha do arquivo e dos dados do *4-way handshake*. Se o MIC encontrado na computação for igual ao MIC passado no *4-way handshake*, então a senha foi quebrada e é justamente a senha utilizada na para computar o MIC (WANG *et al.*, 2010).

3.2.3 WPA2/802.11i

WPA2 é uma das certificações disponibilizadas pela *Wi-Fi Alliance* que certifica que o dispositivo sem fio é compatível com o padrão IEEE 802.11i. A meta da WPA2 é oferecer suporte aos recursos de segurança obrigatórios adicionais do IEEE 802.11i ainda não inclusos nos produtos com suporte a WPA (MICROSOFT, 2006).

O padrão 802.11i foi homologado em junho de 2004 e diz respeito a mecanismos de autenticação e privacidade. Para este padrão, foi implementado o protocolo de rede RSN (*Robust Security Network*), que permite meios de comunicação seguros (RUFINO, 2007).

A principal diferença entre os protocolos WPA e WPA2 está no processo criptográfico empregado em cada um. O WPA, como explicado na Seção 3.2.2 utiliza o método TKIP em conjunto com o algoritmo RC4. Em contra partida,

o WPA2 utiliza o AES (*Advanced Encription Standard*) na forma do protocolo CCMP (*Counter Mode with Cipher Block Chaining Message Authentication Code Protocol*) (MATHEWS; HUNT, 2007).

O protocolo CCMP funciona em dois modos: CBC-MAC e CTR. No modo CBC-MAC é calculado o MIC a partir do cabeçalho, do tamanho do cabeçalho e do campo *payload* do quadro. No modo CTR (*Counter*) ou *Counter Mode* o método criptográfico derivado do AES é utilizado para criptografar a mensagem (*payload* do quadro 802.11) e o MIC (CAM-WINGET *et al.*, 2002). A Figura 13 mostra onde cada fração do CCMP age.

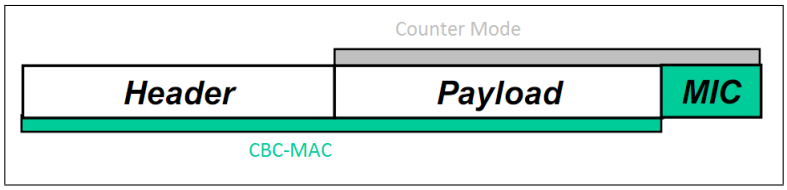

Figura 13: Etapas do protocolo CCMP (CAM-WINGET *et al.*, 2002).

Quanto à autenticação, a principal diferença entre o WPA e WPA2 é o fato de no WPA2 a estação ser autenticada em todos os APs do mesmo ESS, o que facilita a mudança de células (*roaming*).

Como o WPA e WPA2 compartilham da mesma forma de autenticação e derivação de chaves durante o *4-way handshake* propostos pelo 802.11i, o WPA em modo PSK também fica suceptível à ataques *offline* de dicionário sobre o *4-way handshake*.

3.3 Ataques comuns a redes sem fio

Existem diversos ataques inerentes a redes sem fio descritos na literatura. Os mais comumente executados devido às várias ferramentas de ataques são: *Sniffing*, *Spoofing*, *Hijacking* e DoS.

Originalmente, a escuta de rede (*Sniffing*) foi concebida como uma ferramenta de análise de tráfego. Atualmente, é uma das ferramentas de ataque mais eficazes seja para mapear a rede, seja para obter informações sem criptografia ou informações criptografadas, como parte de um ataque maior (BARNES *et al.*, 2002).

As ferramentas de escuta foram desenvolvidas inicialmente em uma época em que as tecnologias de rede cabeadas disponíveis, como *hubs* e repetidores, não direcionavam os pacotes diretamente ao seu destinatário. Os dados eram enviados em *broadcast*. Tendo isso em mente, o *Sniffing* se tornou uma ferramenta importante também para as redes sem fio, uma vez que os dados também não são enviados para uma determinada estação e sim para o meio de transmissão, o ar, não sendo nem mesmo necessário estar associado à rede para interceptar os dados.

Spoofing é um tipo de ataque que consiste em enganar o equipamento de rede, fazendo com que ele "pense" que a estação está conectada. Para isso existem algumas técnicas simples como a modificação de endereços MAC e/ou IP (BARNES *et al.*, 2002).

O principal motivo deste tipo de ataque é invadir redes fechadas. Uma rede pode ser fechada por filtro de MAC, filtro de IP e autenticação. No caso de uma rede fechada apenas por MAC e/ou IP, pode-se fazer uma análise das estações conectadas e então simplesmente trocar o endereço MAC e/ou IP da estação invasora.

Ataques de *Hijacking*, ou roubo de sessão são executados falsificando a tabela ARP do equipamento wireless (um AP por exemplo). O atacante então pode se associar a um IP válido na rede. A partir daí todos os dados direcionados para o AP passam pelo atacante primeiro. Esse ataque também é conhecido como o ataque do homem do meio, ou interceptador (ULBRICH; VALLE, 2009).

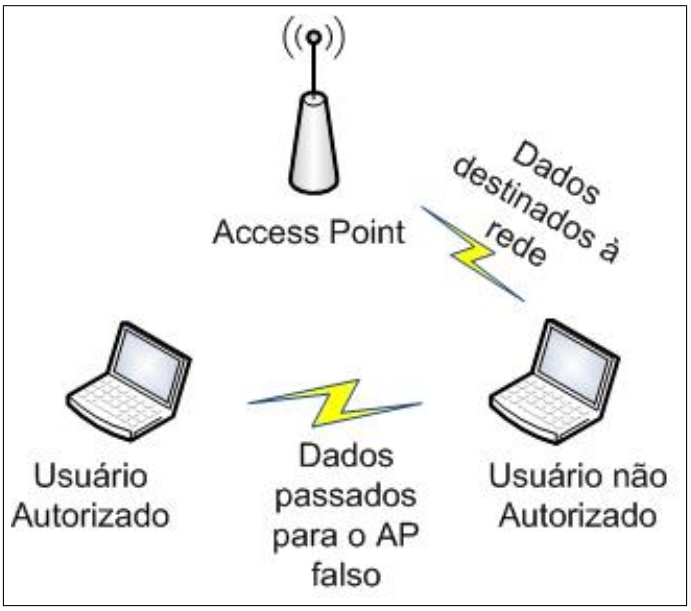

Figura 14: Ataque *Man-In-The-Middle*.

Os ataques de negação de serviço (DoS - *Denial of Service*) são ataques que dificultam ou até mesmo impedem o uso da rede. Podemos citar alguns ataques deste tipo:

- **interferência no sinal de rádio** - este ataque é realizado simplesmente apontando um sinal de rádio no dispositivo a ser atacado. Usa-se um sinal no

mesmo canal e com potência mais alta. É comumente chamado de *jamming* de sinal (CHINTA *et al.*, 2009).

- **flooding ou inundação de pacotes** - consiste em inundar a rede com pacotes de maneira a diminuir a transferência de dados (BELLAICHE; GREGOIRE, 2008).

- **ataques de desassociação** - é executado enviando pacotes de pedido de desassociação ao AP.

A maioria dos ataques a redes sem fio são executados com ferramentas específicas, não sendo necessário modificar pacotes manualmente. Algumas das ferramentas fazem todo o ataque, deixando que o usuário apenas usufrua dos resultados. Na próxima seção serão abordadas algumas das ferramentas necessárias à realização dos ataques.

3.4 Ferramentas para Ataques a Redes sem Fio

As ferramentas de auditoria e testes de segurança em redes sem fio são também amplamente utilizadas para realizar ataques. Também são apresentadas algumas ferramentas com a função específica de ataque.

Uma das ferramentas mais famosas e utilizadas no mercado, o *Wireshark* (Figura 15) é um analisador de rede com suporte aos mais variados protocolos, sendo também utilizado como *sniffer*. É um programa portável a sistemas operacionais como *Windows*, *Linux*, BSDs, *Solaris* e MAC OS (WIRESHARK FOUNDATION, 2011).

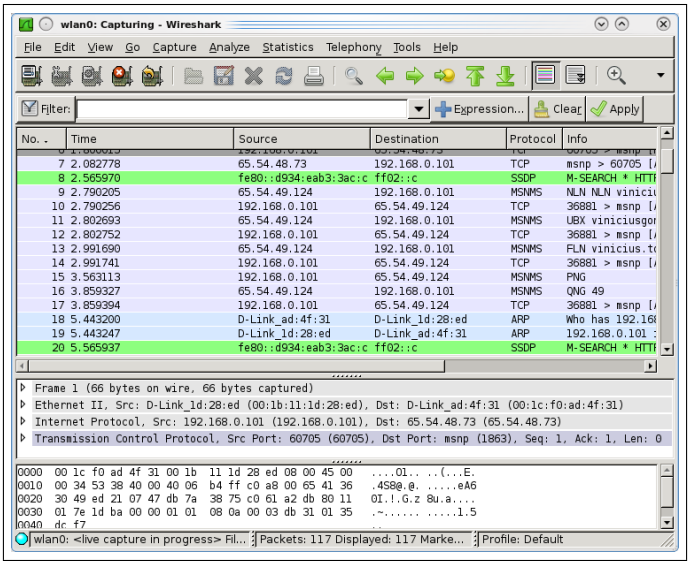

Figura 15: Wireshark em modo de captura de pacotes.

O *tcpdump* (THE TCPDUMP TEAM, 2011) é um analisador de pacotes tão poderoso quanto o *Wireshark* porém sem interface gráfica, talvez por isso tão pouco intuitivo. Também possui a funcionalidade de captura de pacotes, filtros de captura e leitura de arquivos pré-capturados.

O *macchanger* (ORTEGA, 2007) é apenas um *frontend* para a ferramenta *ifconfig* do Linux, sua única função é modificar o endereço MAC dos dispositivos sem fio. Não é exatamente uma ferramenta de segurança, mas sua função pode ser descrita como uma ferramenta de *spoofing* para o endereço MAC, necessário para cobrir rastros em ataques. Ele pode também modificar o MAC de acordo com o fabricante desejado.

A ferramenta Weplab, mostrada na Figura 16, realiza ataques utilizando um dicionário de senhas ou ataques estatísticos para a quebra de senhas. Para tal é

necessário um arquivo de captura com pacotes de dados suficientes. Uma interface gráfica encontra-se disponível no site do projeto (MARTÍN, 2005).

Figura 16: Ajuda do Weplab.

A família de aplicativos *Aircrack-NG* (AIRCRACK-NG, 2010) possui ferramentas que executam uma simples escuta na rede até ataques de injeção de pacotes com o intuito de gerar tráfego ou um ataque de DoS. A utilização dessas ferramentas em conjunto, e se corretamente utilizadas, pode levar à quebrar uma chave WEP ou WPA dependendo das condições de tráfego. São ferramentas multiplataforma executadas em apenas modo-texto. Em ambiente MAC OS esse conjunto de ferramentas pode ser substituído pelo KisMAC (KISMAC TEAM, 2011), um programa *open source* assim como o *Aircrack-NG*, porém disponível apenas para MAC OS.

Existe, ainda, uma ferramenta de total gerenciamento de redes sem fio com funções de detecção de redes, captura de pacotes e detecção de intrusos. Essa ferramenta, chamada *Kismet* (Figura 17) (KERSHAW, 2011) possui uma interface pseudo-gráfica baseada na biblioteca ncurses. É uma ferramenta quase tão completa quanto o *Aircrack-NG* (não possui função de quebra de chaves) com a vantagem de uma interface que dispensa memorização de comandos.

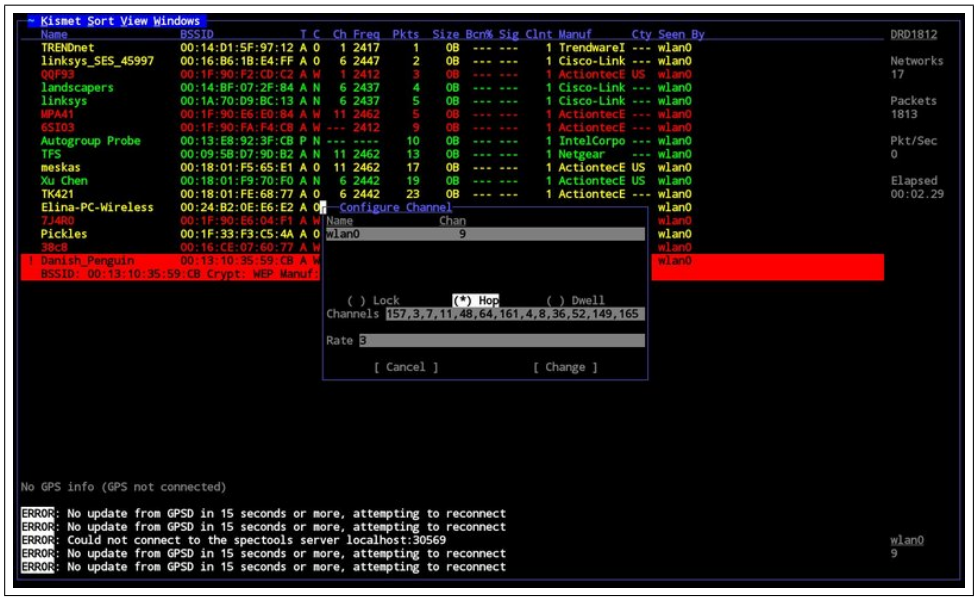

Figura 17: Interface do Kismet.

As ferramentas mostradas nesta seção não possuem interfaces gráficas ou então possuem uma interface em modo texto também comumente chamada pseudo-gráficas. É conhecido que, quando bem projetadas, interfaces gráficas geralmente são mais práticas do que interfaces em modo texto. O próximo capítulo trata de interfaces gráficas juntamente com alguns conceitos de usabilidade.

4 USABILIDADE EM INTERFACES GRÁFICAS

Em computação, uma interface gráfica para o usuário (GUI do *inglês Graphical User Interface*) é um tipo de interface que permite interagir com o computador por meio de imagens e dispositivos apontadores (um *mouse*, por exemplo) e também de forma muitas vezes limitada pelo teclado (THE LINUX INFORMATION PROJECT, 2004). Interfaces gráficas estão presentes em qualquer tipo de dispositivos que precisam de interações com usuários, como é o caso de computadores e dispositivos móveis de comunicação (*smartphones*, por exemplo). Em suma, uma GUI pode ser entendida como aquilo que o usuário vê na tela de um dispositivo e consegue interagir com o sistema através da interface gráfica, de forma mais simples do que a visualização de um terminal em modo texto.

As interfaces gráficas modernas permitem a manipulação direta, aumentando dramaticamente a produtividade dos usuários. Isso ocorre se a interface não se interpor entre o usuário e a tarefa a ser executada como uma barreira (GOEDICKE; SUCROW, 1996). Para tal, alguns conceitos de usabilidade devem ser aplicados ao desenvolvimento.

A usabilidade foi definida pela *International Standarization Organization* (ISO) como uma medida pela qual o *software* pode ser utilizado por usuários específicos para atingir objetivos específicos com eficácia, eficiência e satisfação dentro de um contexto específico (ISO 9241-11) de uso como mostrado na Figura 18 (RAMLI; JAAFAR, 2008).

De um ponto de vista mais técnico, Hix e Hartson (1993) citam que a usabilidade é a combinação de algumas características orientadas ao usuário. São elas:

1. Facilidade de aprendizado.

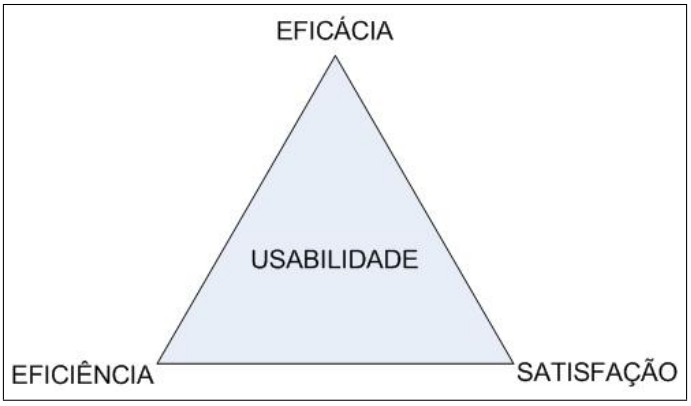

Figura 18: Definição de usabilidade (RAMLI; JAAFAR, 2008).

2. Alta velocidade do usuário ao realizar uma tarefa.

3. Baixa taxa de erros do usuário.

4. Satisfação subjetiva do usuário.

5. Memorização das funcionalidades.

Hix e Hartson (1993) propõem uma redefinição da formalização proposta pela ISO dizendo que a usabilidade é relacionada à eficácia e eficiência do usuário e também à reação do mesmo frente a interface.

5 METODOLOGIA

5.1 Tipo de Pesquisa

O presente trabalho pode ser classificado quanto a sua natureza como uma pesquisa aplicada, pois o objetivo principal se concentra na geração de conhecimento para a solução de um problema prático.

Gil (1999) diz que as pesquisas podem ser classificadas quanto aos seus objetivos como exploratória, que visa formular novas teorias sobre o problema ou descritiva, quando visa descrever as características de determinando fenômeno.

Pode-se dizer que este documento tem um caráter descritivo quanto aos procedimentos, pois visa analisar e descrever os acontecimentos registrados a partir da ferramenta implementada.

5.2 Procedimentos Metodológicos

5.2.1 Etapas da Pesquisa

Este trabalho consiste em uma pesquisa aplicada, desenvolvida em quatro etapas: levantamento bibliográfico, implementação, testes e coleta de dados e documentação

A primeira etapa consistiu de um levantamento bibliográfico destinado ao aprendizado de conceitos ainda não fixados e estabelecimento de um referencial teórico sobre redes sem fio 802.11. A pesquisa bibliográfica foi realizada com base em fontes primárias e secundárias a partir do mês de março do ano de 2010.

Após realizada a pesquisa bibliográfica, foi implementada uma ferramenta de ataque a redes sem fio com base nas ferramentas da família de aplicativos *Aircrack-*

NG. A implementação desta ferramenta foi feita utilizando-se a linguagem de programação C++, com o auxílio do *framework* Qt para a produção de interfaces gráficas.

Ao finalizar a implementação da ferramenta, foi iniciada a etapa de testes comparativos com as atuais ferramentas de ataque que se destinam à quebra de chaves WEP/WPA, no caso as ferramentas do *Aircrack-NG*.

A justificativas da escolha da linguagem e do *framework* a serem utilizados são apresentados a seguir.

5.2.2 C++/QT vs. Java/Swing

Após o levantamento bibliográfico e documental, foi implementada uma interface gráfica integradora para os aplicativos de auditoria em redes sem fio. Essa interface foi desenvolvida na linguagem C++ utilizando o *framework* Qt (NOKIA CORPORATION, 2008).

O Qt é um *framework* multiplataforma destinado ao desenvolvimento de aplicações, para *desktop* ou sistemas embarcados, com interface gráfica em C++. Por ter sido desenvolvido em C++ não é necessário o uso de uma máquina virtual, como a usada pela linguagem de programação Java.

Em (DALHEIMER, 2006) são mostradas algumas das vantagens de se trabalhar com o *framework* Qt para C++ ao invés de utilizar o *framework* Java/Swing:

- menor quantidade de linhas de código para a resolução de um mesmo problema.
- menor consumo de memória devido a não utilização de uma máquina virtual.

- a aparência das interfaces gráficas criadas com o Qt apresentam um *design* mais agradável do que as criadas com Java/Swing.

Apesar das facilidades inerentes à linguagem Java, as pesquisas relacionadas em (DALHEIMER, 2006) mostram que a produtividade dos programadores está ligada à maturidade e a preferência dos mesmos.

Há ainda dois fatores agravantes ao uso de Java para o nosso propósito: o projeto será desenvolvido para uma distribuição GNU/Linux que não possui a máquina virtual Java (JRE) instalada por padrão e as ferramentas que deseja-se integrar (descritas a seguir), são escritas em C/C++ o que torna o uso da linguagem desejável, porém não obrigatório.

5.2.3 A Família de Ferramentas *Aircrack-NG*

As ferramentas presentes no *software Aircrack-NG* (AIRCRACK-NG, 2010) destinam-se a ataques à redes sem fio 802.11 e aos protocolos de segurança WEP, WPA e WPA2. São ferramentas escritas na linguagem C/C++ e estão disponíveis para os sistemas *Windows* (com algumas restrições) e Linux. As ferramentas relevantes para este trabalho são: *Aircrack-NG*, *Airodump-NG*, *Aireplay-NG* e *Airmon-NG*.

O *Aircrack-NG* (Figura 19) é uma ferramenta que consegue quebrar chaves WEP e WPA uma vez que os pacotes necessários são capturados. É a única ferramenta que funciona sem nenhuma restrição no sistema *Windows*, pois não utiliza os *drivers* dos adaptadores de rede sem fio.

A ferramenta *Airodump-NG* é a ferramenta de captura de pacotes. Para a quebra de chaves WPA, é necessário apenas um pacote específico (a negociação

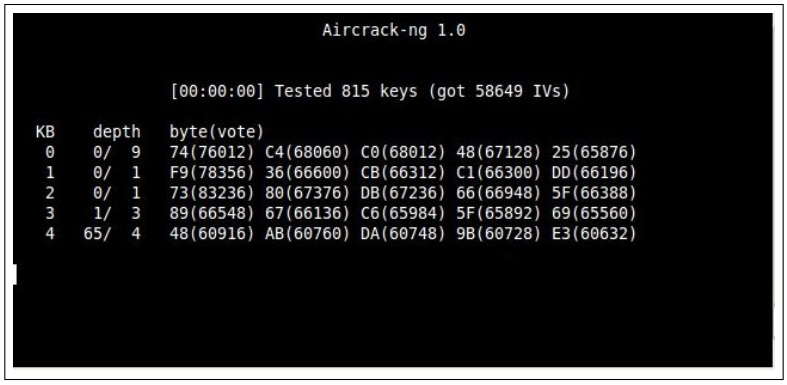

Figura 19: *Aircrack-NG* - Quebra de chaves WEP.

de autenticação) e para WEP quanto mais pacotes melhor. Para a captura dos pacotes necessários algumas vezes é necessário injetar pacotes na rede. Para tal, usa-se em conjunto com o *Airodump-NG* a ferramenta *Aireplay-NG*.

O *Aireplay-NG* é uma ferramenta que implementa a injeção de diversos tipos de pacotes, como de desautenticação, autenticação falsa e ainda injeta pacotes definidos pelo usuário. Essa ferramenta cria as condições de tráfego necessárias à quebra das chaves WEP e WPA.

O *Airmon-NG* simplesmente coloca a interface *wireless* em modo monitor, uma condição necessária para que todos os canais de uma rede sem fio sejam capturados. É justamente essa ferramenta que apresenta problemas com a plataforma *Windows*. São necessários *drivers* especiais para as interfaces, que muitas vezes não são aceitos pelo sistema.

Excetuando-se o *Aircrack-NG*, todas as ferramentas aqui utilizadas necessitam que a interface de rede esteja em modo monitor, o que acaba tornando seu uso por vezes inviável ao se utilizar o sistema operacional *Windows*.

A partir do conhecimento obtido por meio das ferramentas, pode-se então executar a quebra seguindo os passos mostrados na Figura 20. O usuário deverá habilitar o modo monitor em sua interface de rede sem fio. Feito isso, procura-se a rede alvo e inicia-se a captura de pacotes. Caso seja necessário, um ataque de injeção de pacotes deve ser lançado. Por fim, então é feita a quebra sobre os pacotes capturados.

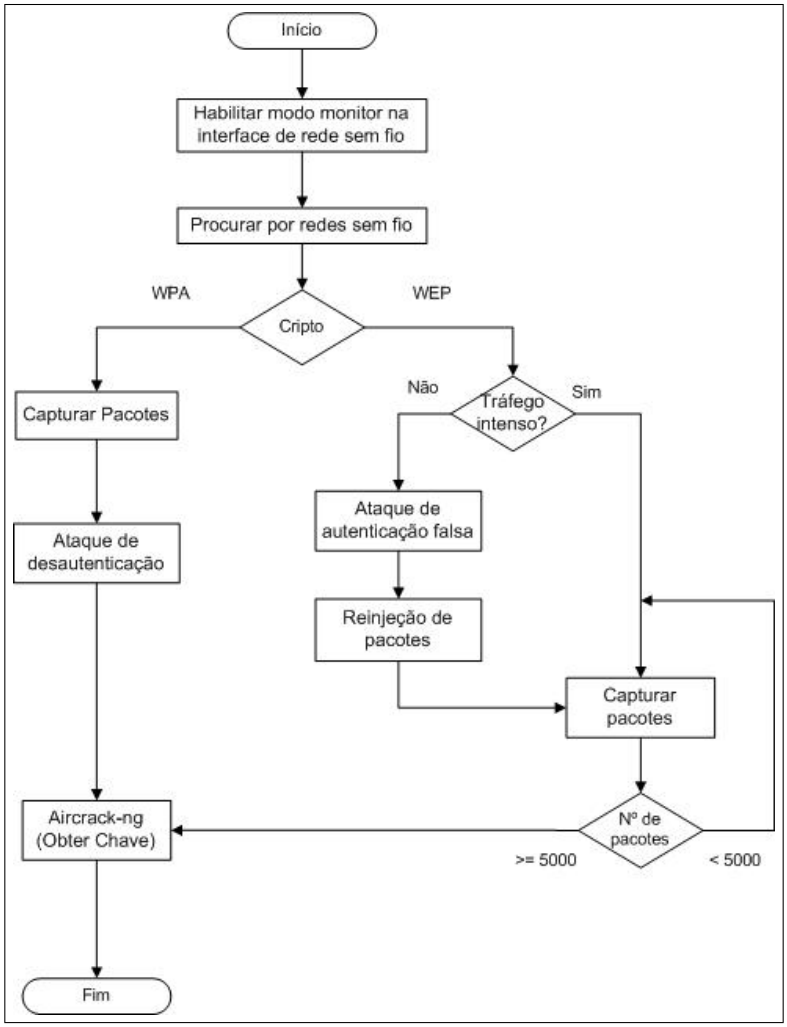

Figura 20: Diagrama de blocos representando uma quebra de chave WEP ou WPA.

5.2.4 Ambiente

A implementação foi executada utilizando o sistema operacional GNU/Linux Ubuntu[7] versão 10.04.1. A versão do *kernel* Linux utilizado foi o 2.6.32.24, o mesmo que acompanha a distribuição, sem modificações. O principal motivo da não atualização do kernel para o desenvolvimento do trabalho foi o suporte deficiente aos *drivers* utilizados pelo *Aircrack-NG* nas versões atuais desta distribuição Linux.

Devido à ausência de *drivers* que permitem habilitar o modo monitor da interface *Wi-Fi* em sistemas *Microsoft Windows*, optou-se por não implementar o programa para tal sistema, pois o mesmo não seria funcional, embora a interface gráfica possa ser recompilada para sistemas não-Linux.

Com o intuito de agilizar o desenvolvimento, optou-se por utilizar uma IDE que permitisse "desenhar" a interface gráfica sem a necessidade de escrever o código completo para a mesma. A IDE (*Integrated Development Environment* –Ambiente de Desenvolvimento Integrado) escolhida foi o Qt *Creator*[8] que permite a programação na linguagem C++ utilizando os recursos do Qt. Foi necessário também a instalação de um compilador da linguagem C++, foi escolhido o g++ por ser nativo de sistemas Linux, e também o suíte *Aircrack-NG* que funciona como *back-end* para a aplicação desenvolvida. O Ubuntu disponibiliza o gerenciador de pacotes APT (*Advanced Packaging Tool* –Ferramenta de Empacotamento Avançada), que tornou possível a instalação das últimas versões dos *sofwares* supracitados, disponíveis nos repositórios da distribuição.

[7] Disponível em www.ubuntu.com

[8] Mantida pela *Nokia* e disponível em http://qt.nokia.com

As versões dos softwares instalados através do comando `apt-get` foram as seguintes:

- Qt 4.6.2.
- *Aircrack-NG* 1.0.
- *Qt Creator* 1.3.1.
- g++ 4.4.3.

Foi utilizado ainda para testes um adaptador USB para redes *Wi-Fi* sem marca com as seguintes especificações:

- Potência: 500mW.
- Rádio IEEE 802.11 b/g.
- Modos de Operação: *Managed*, *Ad-Hoc*, *Monitor*.
- Chipset Realtek RTL8187B.

5.2.5 Implementação da Interface Gráfica

A implementação da interface gráfica foi facilitada pelo uso do *Qt Creator*, pois este usa o sistema WYSIWYG (*What You See Is What You Get*) comum em outras IDEs. Este sistema permite que o programador "desenhe" a interface sem a necessidade de programar (escrever) as linhas de código correspondentes às declarações de cada elemento de uma interface.

Ao criar um novo projeto no *Qt Creator*, o programador escolhe uma classe base para sua aplicação GUI (*Graphical User Interface*). Esta classe base implementa o elemento base da interface, funcionando como um contêiner que irá acomodar os outros itens gráficos, chamados de *Widgets*.

A interface gráfica desenvolvida consiste de um *QDialog* como classe base e para "acomodar" as várias etapas de um processo de quebra de senha, utiliza-se um conjunto de abas implementadas pela classe QTabWidget. Em cada uma das abas são criados botões que implementam as rotinas para cada fase da quebra de uma senha, muitas vezes executando comandos do linux ou mesmo o *Aircrack-NG*, ou seja, comandos externos à aplicação. Os botões são implementados pela classe *QPushButton* do Qt.

Foram utilizados, também do Qt, os outros itens que recebem e fornecem dados às ações executadas pelos botões. São eles o *QLineEdit*, *QListWidget*, *QComboBox* e *QLabel*.

Em alguns *Widgets* a propriedade *toolTip* foi alterada para que, durante a passagem do ponteiro do mouse, fosse exibido um pequeno texto explicativo sobre utilidade dos mesmos, como mostrado na Figura 21.

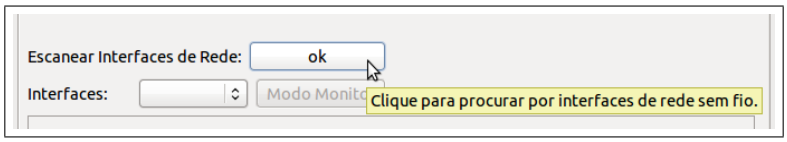

Figura 21: Propriedade toolTip: textos explicativos

5.2.6 Implementação do Sistema

As rotinas implementadas durante o desenvolvimento deste trabalho baseiam-se na execução de comandos (comandos do próprio sistema ou do suíte *Aircrack-NG*) externos à aplicação desenvolvida. Tais comandos são executados fazendo uso da classe *QProcess* do Qt, que implementa métodos que permitem a execução de comandos, aplicações ou *scripts* externos à aplicação em desenvolvimento.

A Figura 22 mostra um trecho de código que utiliza o QProcess para executar uma aplicação externa ao programa do trecho de código em questão. Após instanciar o objeto, é chamado o método *QProcess::start(const QString & program, OpenMode mode = ReadWrite)*, que espera que o programador passe como parâmetro uma string com o caminho do programa a ser executado. Como a aplicação já se encontra no PATH do sistema, não é necessário explicitar o caminho completo da mesma. O método *start* então cria um novo processo que será responsável por executar a aplicação "gedit".

```
1 #include <QProcess>
2
3 QProcess *proc = new QProcess();
4 proc->start("gedit");
5
```

Figura 22: Utilização do QProcess.

Alguns comandos necessitam de um terminal para serem executados, como é o caso dos comandos `ifconfig` e `iwconfig` por exemplo. Para executá-los, é necessário então que seja chamado um terminal para que este terminal faça a chamada ao comando. Para tal, foi escolhido um terminal que suporte a chamada de comandos via parâmetros, o XTERM.

O XTERM permite a chamada de qualquer outra aplicação quando utilizado com a opção "-e" da seguinte forma:

```
$ xterm -e <aplicação>
```

Para que o *QProcess* agora possa executar um comando que necessite do uso do XTERM, utilizamos a função sobrecarregada *QProcess::start(const QString & program, const QStringList & arguments, OpenMode mode = ReadWrite)*, da forma como mostrado na figura

```
1 #include <QProcess>
2
3 QProcess *pIfconfig = new QProcess;
4 QStringList arguments;
5 char command[256];
6
7 sprintf(command,"ifconfig -a > %s","/tmp/arqIfconfig.dat");
8 arguments << "-e" << command;
9
10 pIfconfig->start("xterm",arguments);
11 pIfconfig->waitForFinished(60000);
12 delete pIfconfig;
13
```

Figura 23: Utilização do QProcess com Xterm.

Nota-se agora que o método *start* pede além do caminho do comando, uma lista de parâmetros.

Descrito como foi utilizado o *QProcess* durante a implementação , o programa consiste basicamente em executar os programas do suíte *Aircrack-NG* e tomar os dados de sua saída, desviando-os da saída padrão (terminal) para um arquivo texto, como na Figura 23, para então serem processados.

Após o processamento dos arquivos de texto, os dados obtidos são exibidos para o usuário em suas posições correspondentes e são utilizados nas próximas tarefas.

A Seção 5.3, a seguir, apresenta a metodologia de testes de usabilidade utilizada para a avaliação da interface desenvolvida.

5.3 Metodologia Para a Avaliação da Interface

Conforme Rocha e Baranauskas (2003), as avaliações são necessárias para o conhecimento do que querem os usuários e quais os problemas enfrentados por eles quando da utilização de um *software*. Quanto mais informações sobre os usuários os desenvolvedores obtiverem, melhor será o *design* da interface.

Para a avaliação da interface proposta no presente trabalho, foi utilizada a avaliação heurística apresentada por Nielsen e Molich (1990).

Na avaliação heurística, o avaliador olha e interage com todas as telas da interface e, a partir dessa utilização, tenta formar uma opinião sobre o que é bom ou ruim no que foi avaliado a partir das dez heurísticas segundo Nielsen (1993).

Os avaliadores escolhidos foram cinco alunos do curso de Ciência da Computação da Universidade Federal de Lavras (UFLA). Todos os avaliadores haviam cursado as disciplinas de Interfaces Homem-Máquina e Redes de Computadores e, portanto, tinham os conhecimentos mínimos para realizar a avaliação da interface.

Os testes decorreram da seguinte forma:

- A interface é apresentada aos avaliadores com uma explicação de seu funcionamento.

- É apresentada aos usuários uma ficha de avaliação (anexo). Junto à ficha, foi anexada uma página com as dez heurísticas segundo Nielsen (1993) traduzidas em (ROCHA; BARANAUSKAS, 2003).

- Os avaliadores realizam a avaliação assinalando quais heurísticas estão sendo afetadas, descrevendo em que parte da interface ocorre o problema e com qual grau de severidade isto ocorre. Se possível, apresentam também uma solução.

As dez heurísticas entregues aos avaliadores são as seguintes:

1. **Visibilidade do status do sistema:** O sistema deve sempre manter os usuários informados sobre o que está acontecendo através de *feedback* apropriado, em um tempo razoável.

2. **Compatibilidade entre sistema e mundo real:** O sistema deve utilizar a linguagem do usuário, com palavras, frases e conceitos familiares para ele, ao invés de termos específicos de sistemas. Seguir convenções do mundo real, fazendo com que a informação apareça em uma ordem lógica e natural.

3. **Controle e liberdade para o usuário:** Estão relacionados à situação em que os usuários freqüentemente escolhem as funções do sistema por engano e então necessitam de "uma saída de emergência" claramente definida para sair do estado não desejado sem ter que percorrer um longo diálogo, ou seja, é necessário suporte a *undo* e *redo*.

4. **Consistência e padrões:** Referem-se ao fato de que os usuários não deveriam ter acesso a diferentes situações, palavras ou ações representando a mesma coisa. A interface deve ter convenções não-ambíguas.

5. **Prevenção de erros:** Os erros são as principais fontes de frustração, ineficiência e ineficácia durante a utilização do sistema.

6. **Reconhecimento em lugar de lembrança:** Tornar objetos, ações, opções visíveis e coerentes. O usuário não deve ter que lembrar informações de uma parte do diálogo para outra. Instruções para o uso do sistema devem estar visíveis ou facilmente acessíveis.

7. **Flexibilidade e eficiência de uso:** A ineficiência nas tarefas pode reduzir a eficácia do usuário e causar-lhes frustração. O sistema deve ser adequado tanto para usuários inexperientes quanto para usuários experientes.

8. **Projeto minimalista e estético:** Os diálogos não devem conter informações irrelevantes ou raramente necessárias. Cada unidade extra de informação em um diálogo compete com unidades relevantes e diminui sua visibilidade relativa.

9. **Auxiliar os usuários a reconhecer, diagnosticar e recuperar erros:** Mensagens de erro devem ser expressas em linguagem natural (sem códigos), indicando precisamente o erro e sugerindo uma solução.

10. **Ajuda e documentação:** Mesmo que seja melhor que o sistema possa ser usado sem documentação, pode ser necessário fornecer ajuda e documentação. Tais informações devem ser fáceis de encontrar, ser centradas na tarefa do usuário, listar passos concretos a serem seguidos e não ser muito grandes. A ajuda deve estar facilmente acessível e *on-line*.

Os avaliadores opinam sobre o grau de severidade do problema encontrado, que está diretamente relacionado com qual severidade a heurística foi afetada. As severidades variam de 1 a 5 da seguinte forma:

1. Não concordo que seja um problema de usabilidade.
2. Problemas relacionados ao projeto estético.
3. Problemas de pouca importância, que não atrapalham a execução das tarefas.
4. Problemas graves, que atrapalham a execução da tarefa de maneira sensível.
5. Problemas gravíssimos, que atrapalham severamente ou até impedem a tarefa de ser executada.

Um exemplo de problema com grau de severidade **1** para um software de quebra de senha pode ser justamente a não quebra da senha, quando da utilização de um dicionário pobre. O problema não se encontra na interface, mas no dicionário usado. Pode-se usar o exemplo da senha não ser quebrada também como um problema com grau de severidade **5**, porém agora o usuário foi impedido em algum momento de realizar alguma tarefa que levasse à quebra.

A Figura 24 apresenta o modelo de ficha de avaliação usada pelos avaliadores. Eles descreveram o problema encontrado, quais heurísticas este problema afetou, qual a severidade do problema e, por fim, apresentam uma recomendação para solucionar o problema. São preenchidas quantas fichas os avaliadores julgarem necessário.

Problema			
Heurística(s) afetada			
Local		Severidade	
Recomendação			

Figura 24: Modelo de ficha de avaliação.

Ao fim de todas as avaliações os dados foram coletados para a interpretação e os resultados são conferidos na Seção 6.3.

6 RESULTADOS E DISCUSSÕES

6.1 Considerações Sobre a Interface Gráfica Desenvolvida

A aplicação desenvolvida com interface gráfica foi produzida com o intuito de obter chaves WEP e WPA de redes *Wi-Fi*. Decidiu-se por nomeá-la YAWKC (*Yet Another Wi-Fi Key Cracker*) em função dos programas já existentes com mesma funcionalidade, como *KisMAC* (KISMAC TEAM, 2011), e o próprio *Aircrack-NG* (AIRCRACK-NG, 2010).

A aplicação YAWKC exime o usuário de qualquer execução em linha de comando, proporcionando interações em sua grande maioria via dispositivo apontador (*mouse*). A interface do YAWKC proporciona algumas vantagens sobre o uso do *Aircrack-NG* em modo texto. Ao utilizar a interface gráfica, o usuário fica desobrigado da memorização de todos os comandos bem como suas opções. A quantidade de erros do usuário diminui, uma vez que os erros causados por digitação não existe.

Algumas interfaces gráficas para o *Aircrack-ng* ainda necessitam que o usuário digite as informações das redes encontradas, ou então de seus adaptadores de redes sem fio. Ao utilizar o YAWKC o usuário se depara com caixas de texto contendo listas de opções, que são escolhidas com apenas um clique do mouse. Em apenas um momento da utilização é necessário o uso do teclado (mas também é possível fazê-lo utilizando o mouse).

Da mesma forma que existem vantagens no uso de interfaces gráficas, as desvantagens também se fazem presentes principalmente para usuários avançados. O *Aircrack-ng* (em modo texto) possui uma ampla gama de opções e de filtros para

captura de pacotes e ataques mais complexos que podem ser utilizadas de forma a conseguir a quebra das chaves em situações mais adversas.

Existem formas de automatizar alguns comandos também em linha de comando, através de *scripts*, como é o caso do *Airoscript*[9](Figura 25), que também não obrigam o usuário a memorizar os comandos e suas opções. Ainda assim a interface gráfica é mais agradável ao usuário iniciante do que uma interface em modo texto, como é o caso do *Airoscript*.

Figura 25: Interface em modo texto *Airoscript*.

Por fim, ao mesmo tempo que uma interface gráfica facilita muito a execução de algumas tarefas, ela também tira do usuário a responsabilidade de aprender os fundamentos da execução destas tarefas.

6.2 Apresentando o YAWKC e comparando com o *Aircrack-NG*

O software produzido como resultado deste trabalho utiliza-se da execução dos programas do suíte Aircrack-NG através de uma interface gráfica. Assim como ao utilizar o *Aircrack-ng*, o usuário deve executar o YAWKC como superusuário.

[9]Disponível em http://code.google.com/p/airoscript/

Ao executar o programa, o usuário se depara com uma tela de apresentação (Figura 26), contendo um texto explicativo, bem como um mini-tutorial. O usuário poderá ler o texto e então seguir para a próxima aba com título “Procurar Dispositivos”, em que será definida a interface a ser usada e quais suas configurações. O processo todo segue como mostra o fluxograma da Figura 20 da Seção 5.2.3.

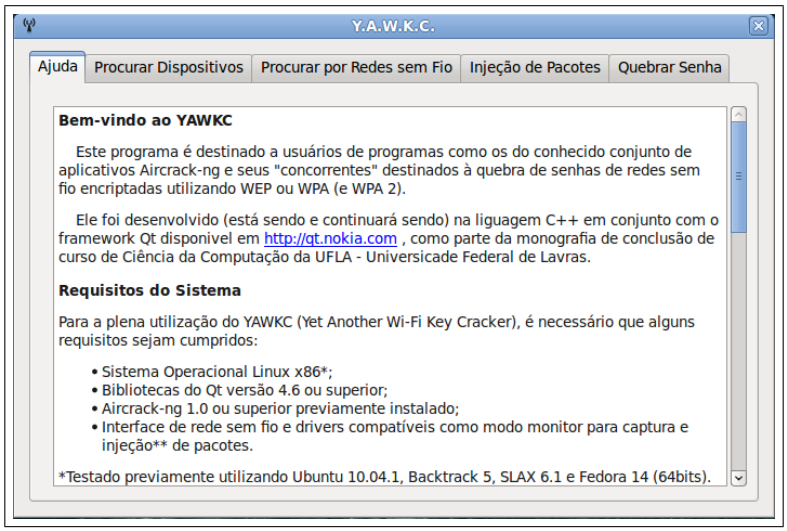

Figura 26: Tela de Ajuda.

Ao clicar na aba “Procurar Dispositivos” (Figura 27) o usuário deve clicar no botão que irá executar a busca por dispositivos de rede sem fio. Não é necessário conhecimento prévio das interfaces presentes. Ao clicar, é apresentada uma lista com as interfaces das quais o usuário deverá escolher uma e ativar o modo monitor, clicando no botão “Modo Monitor”. Ao fim da operação o usuário deve selecionar a interface em modo monitor e seguir para o próximo passo. Caso o usuário não

escolha uma interface em modo monitor, as abas subsequentes ficam bloqueadas e, portanto, sua utilização fica impossibilitada.

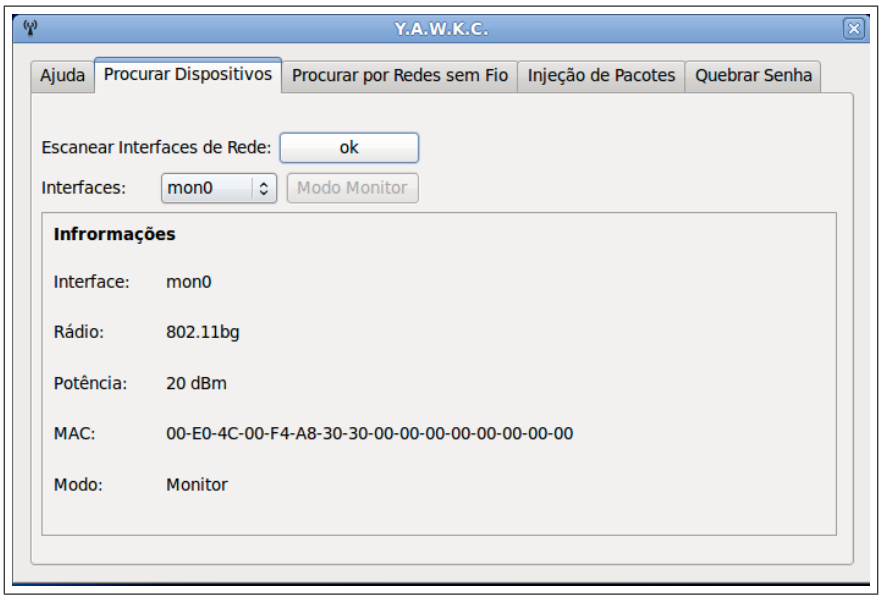

Figura 27: Procura por dispositivos - YAWKC.

O mesmo processo executado no programa *Airmon-NG* exige que o usuário conheça previamente suas interfaces de rede sem fio através dos comandos `ifconfig` e `iwconfig`. Supondo que a interface escolhida pelo usuário para a execução de toda a tarefa seja a interface `mon0`, executa-se o comando `airmon-ng start wlan0` como superusuário. A resposta do comando é mostrada na Figura 28.

Com o modo monitor ativado é possível então procurar por redes sem fio e iniciar a captura dos pacotes. No YAWKC a procura da rede alvo é feita clicando no botão “Procurar Redes” que executa uma busca por quinze segundos e então

```
Interface       Chipset         Driver

wlan0           RTL8187         rtl8187 - [phy0]
                                (monitor mode enabled on mon0)
```

Figura 28: Resposta ao comando `airmon-ng start wlan0`.

retorna para o usuário uma lista com todas as redes encontradas. O botão destinado a busca de redes então é desativado. Seleciona-se a rede desejada e então clica-se no botão "Capturar Pacotes"(Figura 29).

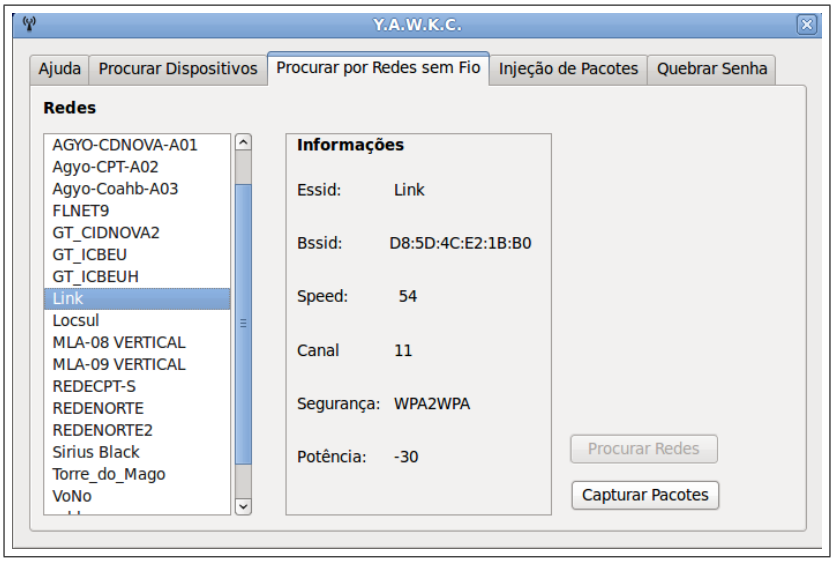

Figura 29: Procura por redes sem fio - YAWKC.

A procura de redes pode ser executada por linha de comando tambémj por um superusuário. Supondo que a interface em modo monitor do usuário em questão é a `mon0`, usa-se o comando `iwlist mon0 scan` ou pelo comando `airodump-ng mon0` (Figura 30).

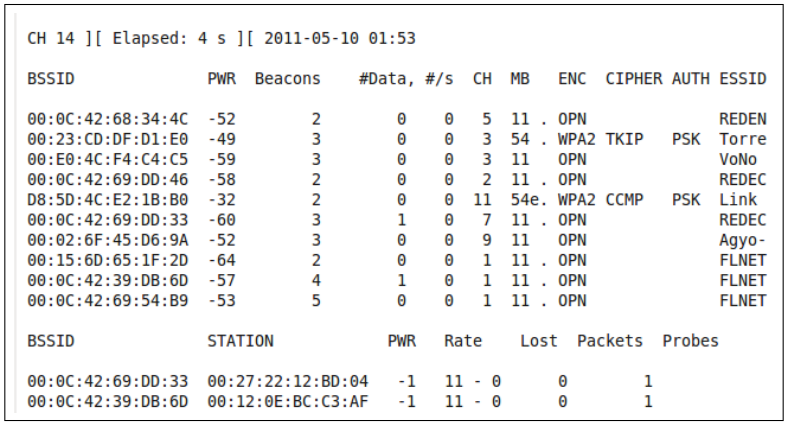

Figura 30: Procura por redes sem fio em modo texto.

Já a captura se torna um pouco mais complicada uma vez que o usuário deve conhecer os parâmetros da rede que deseja capturar os pacotes. Executa-se o comando `airodump-ng -c 11 -bssid D8:5D:4C:E2:1B:B0 -w /tmp/dump -output-format pcap mon0` para obter o mesmo resultado do clique no botão do YAWKC , no qual a opção `-c` pede que seja passado o canal da rede como parâmetro, `-bssid` o endereço MAC do *access point*, `-w` o local onde será salvo o arquivo contendo as capturas e `-output-format` o formato do arquivo. Por último é passada a interface em modo monitor. A resposta a esse comando pode ser conferida na Figura 31.

O próximo passo consiste em atacar a rede injetando pacotes. O programa *Aireplay-NG* possui nove tipos de ataques de injeção de pacotes, dos quais foram escolhidos os três mais simples e mais usados para serem inseridos no YAWKC. Estes ataques são suficientes para criar as condições necessárias a obtenção de uma chave WEP ou WPA em uma rede com usuários conectados. No YAWKC,

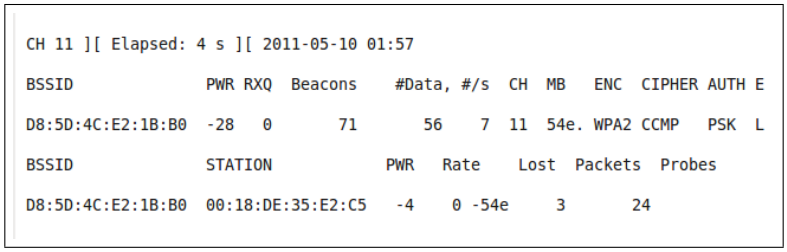

Figura 31: Captura de pacotes em modo texto.

o usuário deve escolher uma estação conectada à rede e preencher campo “MAC de um Cliente”, escolher um tipo de ataque e clicar no botão “Injetar Pacotes”, conforme ilustrado na Figura 32.

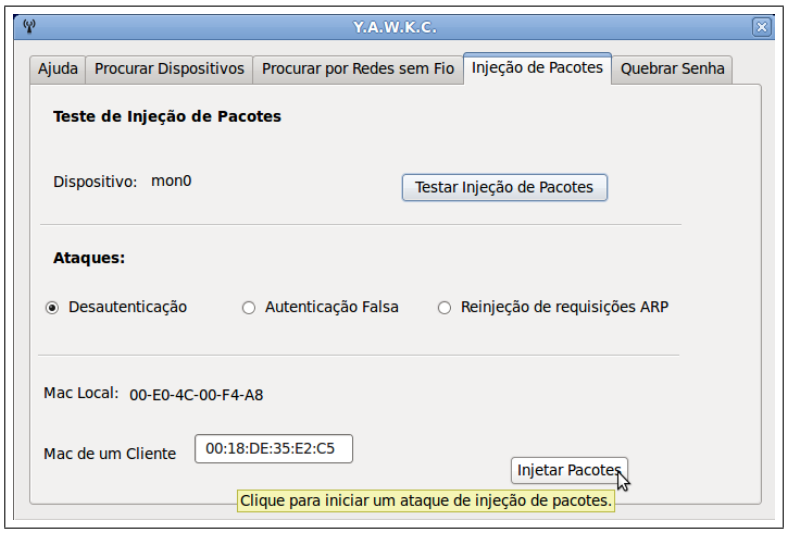

Figura 32: Tela de Injeção de Pacotes - YAWKC.

O usuário também tem a opção de esperar o acúmulo de 5000 pacotes para quebrar uma chave WEP ou então capturar um *4-way-hadshake* no caso

do WPA. O comando em modo texto equivalente ao botão do YAWKC que injeta pacotes de desautenticação de estações é `aireplay-ng -deauth 15 -a D8:5D:4C:E2:1B:B0 -c 00:18:DE:35:E2:C5 mon0`, no qual:

- `D8:5D:4C:E2:1B:B0` é o endereço MAC do *access point*.
- `00:18:DE:35:E2:C5` é o endereço MAC de uma estação a ser desautenticada.
- `mon0` é a interface *Wi-Fi* em modo monitor.

A resposta a esse comando, também presente no YAWKC é mostrada na Figura 33.

```
01:59:26  Waiting for beacon frame (BSSID: D8:5D:4C:E2:1B:B0) on channel 11
01:59:27  Sending 64 directed DeAuth. STMAC: [00:18:DE:35:E2:C5] [121|124 ACKs]
01:59:28  Sending 64 directed DeAuth. STMAC: [00:18:DE:35:E2:C5] [126|118 ACKs]
01:59:28  Sending 64 directed DeAuth. STMAC: [00:18:DE:35:E2:C5] [127|117 ACKs]
01:59:29  Sending 64 directed DeAuth. STMAC: [00:18:DE:35:E2:C5] [124|114 ACKs]
01:59:30  Sending 64 directed DeAuth. STMAC: [00:18:DE:35:E2:C5] [124|119 ACKs]
01:59:30  Sending 64 directed DeAuth. STMAC: [00:18:DE:35:E2:C5] [130|118 ACKs]
01:59:31  Sending 64 directed DeAuth. STMAC: [00:18:DE:35:E2:C5] [126|120 ACKs]
01:59:32  Sending 64 directed DeAuth. STMAC: [00:18:DE:35:E2:C5] [121|118 ACKs]
01:59:33  Sending 64 directed DeAuth. STMAC: [00:18:DE:35:E2:C5] [126|125 ACKs]
01:59:33  Sending 64 directed DeAuth. STMAC: [00:18:DE:35:E2:C5] [128|119 ACKs]
01:59:34  Sending 64 directed DeAuth. STMAC: [00:18:DE:35:E2:C5] [124|121 ACKs]
01:59:35  Sending 64 directed DeAuth. STMAC: [00:18:DE:35:E2:C5] [10|107 ACKs]
01:59:35  Sending 64 directed DeAuth. STMAC: [00:18:DE:35:E2:C5] [ 0|119 ACKs]
01:59:36  Sending 64 directed DeAuth. STMAC: [00:18:DE:35:E2:C5] [ 6|115 ACKs]
01:59:37  Sending 64 directed DeAuth. STMAC: [00:18:DE:35:E2:C5] [ 0|119 ACKs]
```

Figura 33: Resposta do ataque de desautenticação - YAWKC e *Aircrack-NG*.

Após capturar os pacotes necessários, é hora de obter a chave. O usuário seleciona o tipo de criptografia (no YAWKC) e clica no botão “Quebrar Senha”. Caso a criptografia seja WPA, há o diferencial que o usuário deve escolher um arquivo de dicionário[10]. A Figura 34 mostra a última tela do programa.

[10] Arquivo contendo senhas em potencial.

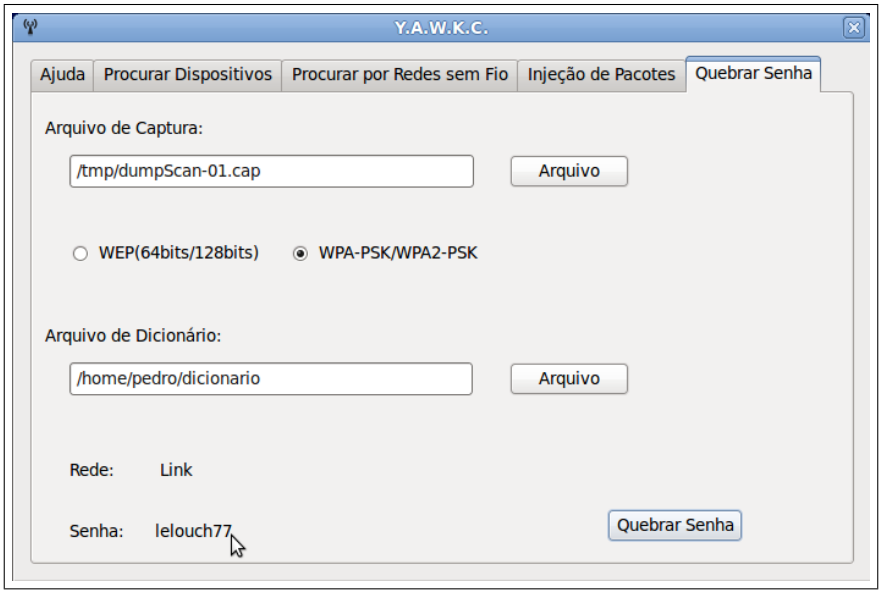

Figura 34: Tela de Quebra de Senhas - YAWKC.

Novamente é possível executar um comando em modo texto para executar a obtenção da chave. Para chaves WEP usa-se `aircrack-ng -a 1 <arquivo de captura>` no qual a opção `-a` indica o tipo de criptografia (1 - WEP, 2 - WPA) e passa-se ao fim do comando o nome do arquivo de captura. Para chaves WPA ou WPA2 o comando é `aircrack-ng -a 2 -w <dicionário> <arquivo de captura>` passando como parâmetros do comando o tipo de criptografia, o arquivo de dicionário e o nome do arquivo de captura. A Figura 35 mostra o *Aircrack-NG* ao quebrar uma senha WPA2.

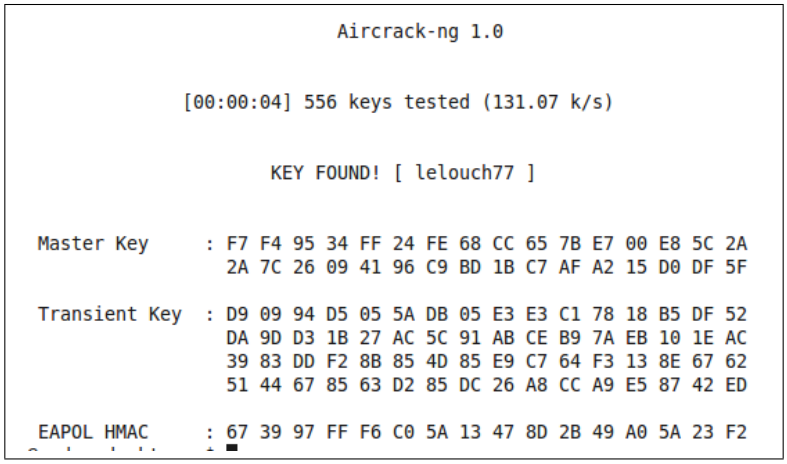

Figura 35: Quebra de chave WPA do *Aircrack-NG*.

6.3 Avaliação da Interface

Após efetuada a avaliação heurística apresentada na Seção 5.3 os dados foram tabelados e, a partir da tabela obtida, foi feito um gráfico para melhor apresentação dos dados.

Os critérios avaliados foram a frequência com que as heurística eram afetadas e a severidade das heurísticas assinaladas pelos avaliadores. Para análise dos problemas de usabilidade da interface considerou-se a heurística afetada com maior severidade e a heurística afetada o maior número de vezes segundo os avaliadores. A Figura 36 apresenta o gráfico com os dados obtidos pela avaliação.

Conforme o gráfico, pode-se inferir que as heurística com maior severidade (severidade 5) foram as de número 3, 7 e 9 que são respectivamente **Controle e liberdade para o usuário**, **Flexibilidade e eficiência de uso** e **Auxiliar os usuários a reconhecer, diagnosticar e recuperar erros**(ver Seção 5.3). Estas

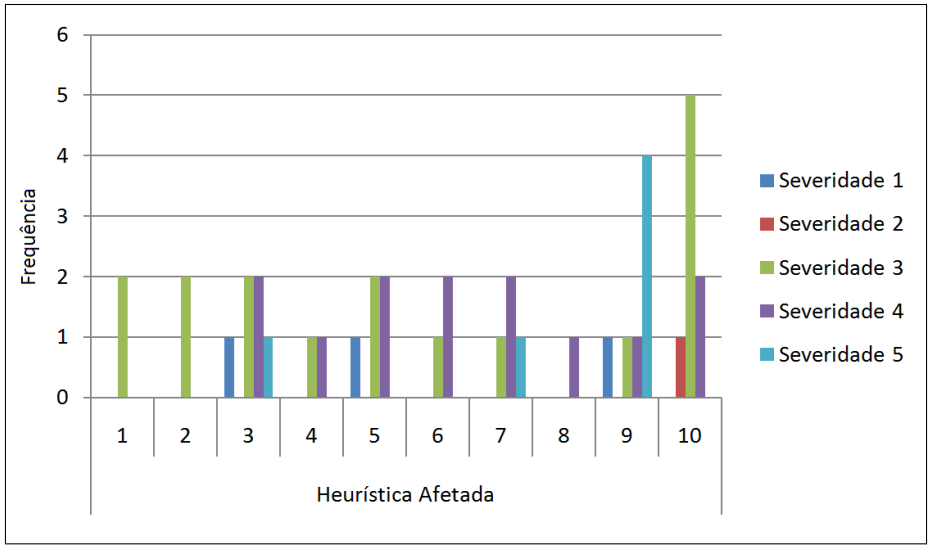

Figura 36: Gráfico: Frequência das heurísticas afetadas e severidade das mesmas segundo os avaliadores.

três heurísticas foram afetadas com serveridade 5 na tela "Procurar por Redes sem Fio" do YAWKC (Figura 37). O problema desta tela ocorre quando o usuário seleciona uma rede sem fio e então inicia a captura de pacotes. O botão de procura é bloqueado, e o usuário fica impossibilitado de executar uma nova procura e escolher uma nova rede, ferindo as heurísticas de número 3 e 7. Além disso, o usuário não recebe qualquer mensagem de erro durante este ocorrido, ferindo a heurística 9.

Nota-se também que no gráfico da Figura 36 a heurística que mais assinalada durante as avaliações foi a de número 10, **Ajuda e documentação**. Os avaliadores queixaram-se da localização da tela de ajuda e também da forma como o texto foi escrito.

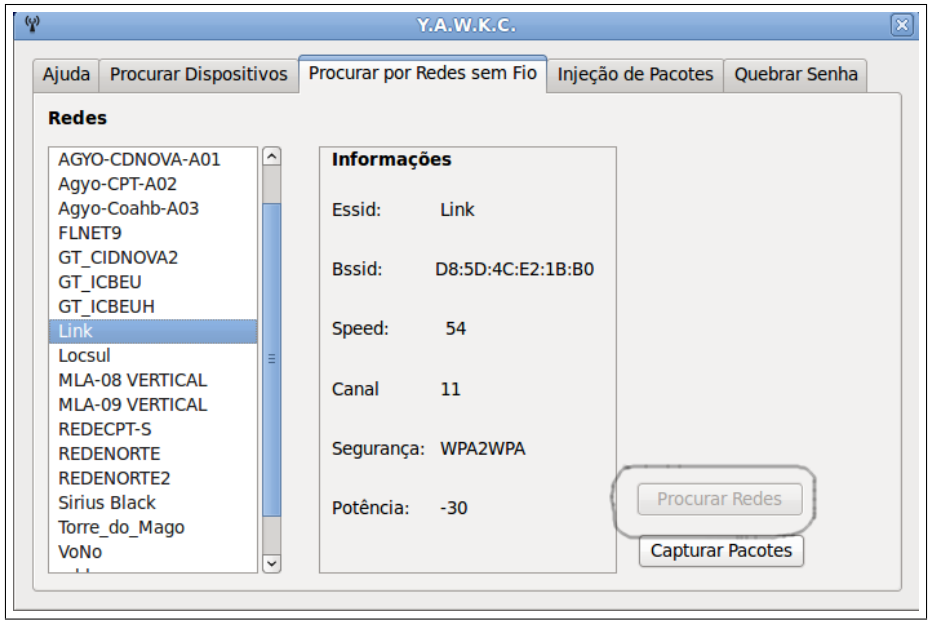

Figura 37: Problema de Usabilidade. Ponto sem volta.

6.4 Distribuição do YAWKC e Requisitos do Sistema

Para utilizar o YAWKC é necessário que alguns requisitos sejam cumpridos. O usuário deve utilizar uma distribuição GNU/Linux com o suíte *Aircrack-NG* devidamente instalado e com o *drivers* que suportam a utilização da interface de rede *Wi-Fi* em modo monitor e que também suporte injeção de pacotes (opcional).

É necessário também que o sistema possua as bibliotecas do Qt versão 4.6 ou superior instaladas e a interface de rede *Wi-Fi* suporte pelo menos o modo monitor.

Para a distribuição do YAWKC foi montada uma distribuição Linux usando o SLAX como base, adicionando os módulos referentes ao *Aircrack-ng* e ao Qt

4.7. O SLAX foi escolhido pelo fato de ser extremamente leve, pequena (250 MB) além de ser uma distribuição *live CD* e *live Pendrive*[11].

O capítulo a seguir apresenta as conclusões obtidas com o desenvolvimento deste trabalho, bem como propostas para trabalhos futuros ainda baseado na manutenção da interface gráfica para o *Aircrack-ng*.

[11] Uma distribuição Live CD aceita ser utilizada diretamente do CD sem a necessidade de ser instalada no *Hard Drive*. A Live Pendrive tem a mesma característica, sendo utilizada diretamente do pendrive.

7 CONCLUSÕES E TRABALHOS FUTUROS

Este capítulo apresenta ao leitor as conclusões alcançadas com o desenvolvimento do trabalho em relação à interface gráfica produzida. Na Seção 7.2 é apresentada uma proposta para trabalho futuro.

7.1 Conclusões

O objetivo do presente trabalho foi o estudo dos protocolos de segurança e a implementação de uma aplicação com interface gráfica com o intuito então de auditar chaves WEP e WPA de redes *Wi-Fi*.

A aplicação foi desenvolvida com sucesso com o uso da linguagem C++ juntamente com o *framework* Qt e o ambiente de desenvolvimento *Qt Creator*, que facilitaram a criação da interface e a execução dos comandos do suíte *Aircrack-NG*. A interface encapsula comandos por vezes extensos, que constantemente são digitados errados pelo usuário devido a dificuldade em memorizá-los. Mesmo executando a quebra de senhas quando corretamente ao ser utilizado no fluxo correto de uma quebra de chaves WEP/WPA (ver o fluxograma da Figura 20), o YAWKC apresentou alguns problemas de usabilidade quando efetuada uma avaliação heurística da interface proposta por (NIELSEN; MOLICH, 1990).

A aplicação é destinada a todo o público, porém é direcionada a usuários com conhecimentos básicos em redes sem fio. Por este motivo, não foram implementadas todas as opções do *Aircrack-NG*, bem como os ataques mais avançados.

Ao fim da implementação foi montada uma distribuição Linux para o uso do programa desenvolvido. A distribuição funciona sem a necessidade de instalação

no *hard-drive* do usuário, bastando executar o *boot* a partir de um *pendrive* ou de um CD.

Apesar de simples, e com alguns problemas de usabilidade, a interface atingiu o objetivo de encapsular completamente os comandos em modo texto, permitindo também a inserção de novas funcionalidades e novos ataques.

7.2 Uma Proposta para Trabalho Futuro

Como trabalho futuro, propõe-se inicialmente a correção dos problemas de usabilidade encontrados na interface. Após este passo importante para os usuários, pode-se fazer uma atualização da aplicação YAWKC com a implementação de todas as opções, e uso de todos os outros programas, da família *Aircrack-ng* tornando-a uma ferramenta mais genérica no uso de auditoria de redes *Wi-Fi*, e não somente destinada à quebra das senhas.

Pode-se ainda portar o programa para o uso em *smartphones*, desta vez usando a linguagem Java ou o *kit* de desenvolvimento da plataforma. Porém para isso é necessário que os *drivers* das interfaces de rede sem fio do dispositivo móvel em questão suportem o modo monitor e a injeção de pacotes, o que é necessário à captura de pacotes em qualquer canal e a criação de condições que acelerem o processo de captura, respectivamente.

Referências

AIRCRACK-NG. 2010. Acessado em: 4 mai. 2011. Disponível em: <http://www.aircrack-ng.org>.

ALI, K. M.; OWENS, T. J. Access mechanisms in Wi-Fi networks state of art, flaws and proposed solutions. In: *Telecommunications (ICT), 2010 IEEE 17th International Conference on*. [S.l.: s.n.], 2010. p. 280–287.

BARNES, C.; BAUTTS, T.; LLOYD, D.; OUELLET, E.; POSLUNS, J.; ZENDZIAN, D. M. *Hack Profing Your Wireless Network*. [S.l.]: Syngress Publishing, Inc., 2002.

BELLAICHE, M.; GREGOIRE, J. C. Measuring defense systems against flooding attacks. In: *Wireless Communications and Mobile Computing Conference, 2008. IWCMC '08. International*. [S.l.: s.n.], 2008. p. 600–605.

BROWN, B. 802.11: the security differences between b and i. *IEEE Potentials*, v. 22, n. 4, p. 23–27, 2003.

CAM-WINGET, N.; MOORE, T.; STANLEY, D.; WALKER, J. *802.11i Overview*. 2002. Acessado em: 4 mai. 2011. Disponível em: <http://csrc.nist-.gov/archive/wireless/S10_802.11i Overview-jw1.pdf>.

CHEN, J.-C.; ZHANG, T. *IP-Based Next-Generation Wireless Networks: IP-Based Next-Generation Wireless Networks*. [S.l.]: John Wiley & Sons, Inc, 2004.

CHINTA, R. T.; WONG, T. F.; SHEA, J. M. Energy-efficient jamming attack in ieee 802.11 mac. In: *Military Communications Conference, 2009. MILCOM 2009. IEEE*. [S.l.: s.n.], 2009. p. 1–7.

CORREIA, L. H. A. *Tópicos em Tecnologias de Comunicação sem fio*. [S.l.]: FAEPE, 2007.

DALHEIMER, M. K. *Qt vs. Java: A Comparison of Qt and Java for Large- Scale, Industrial-Strength GUI Development*. [S.l.], 2006.

DAY, J. D.; ZIMMERMANN, H. The OSI Reference Model. In: *PROCEEDINGS OF THE IEEE*. [S.l.: s.n.], 1983. p. 1334–1340.

GAST, M. *802.11 Wireless Networks The Definitive Guide*. 2nd. ed. [S.l.]: O'Reilly Media, Inc., 2005.

GIL, A. C. *Métodos e Técnicas de Pesquisa Social*. [S.l.]: Atlas, 1999.

GOEDICKE, M.; SUCROW, B. E. Towards a formal specification method for graphical user interfaces using modularized graph grammars. In: *Software Specification and Design, 1996., Proceedings of the 8th International Workshop on*. [S.l.: s.n.], 1996. p. 56–65.

HIX, D.; HARTSON, H. R. *Developing User Interfaces: Ensuring Usability Through Product & Process (Wiley Professional Computing)*. [S.l.]: Wiley, 1993. ISBN 0471578134.

IEEE SOCIETY COMPUTER. *IEEE Std. 802.15.1 IEEE Standard for Information technology — Specific requirements Part 15.1: Wireless medium*

access control (MAC) and physical layer (PHY) specifications for wireless personal area networks (WPANs). [S.l.], jun 2005.

IEEE SOCIETY COMPUTER. *IEEE Std. 802.11 IEEE Standard for Information technology — Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. [S.l.], jun 2007.

KERSHAW, M. *Kismet*. 2011. Acessado em: 4 mai. 2011. Disponível em: <http://www.kismetwireless.net/>.

KISMAC TEAM. *KisMAC*. 2011. Acessado em: 4 mai. 2011. Disponível em: <http://www.kismac-ng.org>.

KUROSE, J. F.; ROSS, K. W. *Redes de Computadores e a Internet: Uma abordagem top-down*. 3. ed. [S.l.]: Pearson Addison Wesley, 2006.

LASHKARI, A. H.; MANSOORI, M.; DANESH, A. S. Wired Equivalent Privacy (WEP) versus Wi-Fi Protected Access (WPA). *2009 International Conference on Signal Processing Systems*, n. 445 - 449, p. 445–449, may 2009.

LI, C.; WU, H.; CHEN, S.; LI, X.; GUO, D. Efficient implementation for MD5-RC4 encryption using GPU with CUDA. *3rd International Conference on Anti-counterfeiting, Security, and Identification in Communication*, p. 167–170, aug 2009.

LI, J.; GARUBA, M. Encryption as an Effective Tool in Reducing Wireless LAN Vulnerabilities. *Fifth International Conference on Information Technology: New Generations*, p. 557–562, 2008.

LOO, A. The Myths and Truths of Wireless Security. *Communications of the ACM*, v. 51, n. 2, p. 66–71, feb 2008.

MARTÍN, J. I. S. *WepLab, analyzing WEP encryption security on wireless networks*. 2005. Acessado em: 4 mai. 2011. Disponível em: <http://weplab-.sourceforge.net/>.

MATHEWS, M.; HUNT, R. Evolution of wireless lan security architecture to ieee 802.11i(wpa2). In: *Proceedings of Communication Systems and Networks ,AsiaCSN 2007*. [S.l.: s.n.], 2007.

MICROSOFT. *Integridade e criptografia de dados do WPA*. 2004. Acessado em: 4 mai. 2011. Disponível em: <http://technet.microsoft.com/pt-br/library-/bb878126.aspx>.

MICROSOFT. *A atualização WPA2 (Wi-Fi Protected Access 2)/WPS IE (Wireless Provisioning Services Information Element) para Windows XP com Service Pack 2 está disponível*. 2006. Acessado em: 3 set. 2011. Disponível em: <http://support.microsoft.com/kb/893357>.

NETGEAR INC. *Wireless Networking Basics*. [S.l.], 2005. Acessado em: 3 set. 2010. Disponível em: <http://docs.netgear.com/reference/sve/wireless/pdfs-/FullManual.pdf>.

NIELSEN, J. *Usability Engeneering*. [S.l.]: Morgan Kaufmann, 1993.

NIELSEN, J.; MOLICH, R. Heuristic evaluation of user interfaces. In: *Proc. ACM CHI'90 Conference*. [S.l.: s.n.], 1990.

NOKIA CORPORATION. *Qt*. 2008. Acessado em: 3 set. 2010. Disponível em: <http://qt.nokia.com>.

ORTEGA, A. L. *GNU Mac Changer*. 2007. Disponível em: <http://www.alobbs-.com/macchanger/>.

RAMLI, R.; JAAFAR, A. e-RUE : A cheap possible solution for usability evaluation. In: *Information Technology, 2008. ITSim 2008. International Symposium on*. [S.l.: s.n.], 2008. v. 3, p. 1–5.

REDWAN, H.; KIM, K.-H. Survey of Security Requirements, Attacks and Network Integration in Wireless Mesh Networks. In: *New Technologies, Mobility and Security, 2008. NTMS '08.* [S.l.: s.n.], 2008. p. 1–5.

ROCHA, H. V. da; BARANAUSKAS, M. C. C. *Design e Avaliação de Interfaces Humano-Computador*. [S.l.]: NIED, 2003.

RUFINO, N. M. de O. *Segurança em redes sem fio: aprenda a proteger suas informações em ambientes Wi-Fi e Bluetooth*. 2nd. ed. [S.l.]: Novatec, 2007.

THE LINUX INFORMATION PROJECT. *GUI Definition*. 2004. Acessado em: 4 mai. 2011. Disponível em: <http://www.linfo.org/gui.html>.

THE TCPDUMP TEAM. *Tcpdump*. 2011. Acessado em: 4 mai. 2011. Disponível em: <http://www.tcpdump.org/>.

ULBRICH, H. C.; VALLE, J. D. *Universidade Hacker*. [S.l.]: Digerati Books, 2009.

VILELA, R. R. da S.; RIBEIRO, D. da S. *SEGURANÇA EM REDES WIRELESS Estudo comparativo entre os protocolos WEP E WPA para implementação de segurança em Empresas e Residências*. 2008. Acessado em: 3 set. 2010. Disponível em: <http://www.sucesumt.org.br/mtdigital/anais/files-/RedesWirelessWEP.pdf>.

WANG, Q.; ZHANG, C. J. Analyse of the application schemes for the wireless network security. *2006 IET International Conference on Wireless, Mobile and Multimedia Networks*, p. 1–4, nov 2006.

WANG, Y.; JIN, Z.; ZHAO, X. Practical defense against wep and wpa-psk attack for wlan. In: *Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference on*. [S.l.: s.n.], 2010. p. 1 –4.

WIRESHARK FOUNDATION. *Wireshark*. 2011. Acessado em: 4 mai. 2011. Disponível em: <http://www.wireshark.org/>.