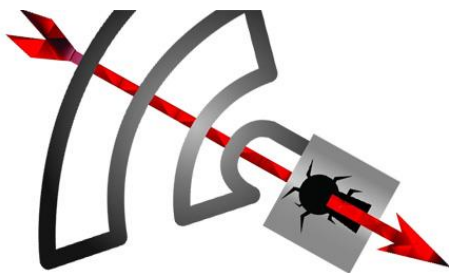


RELATÓRIO TRABALHO PRÁTICO #1 SEGURANÇA 2017/2018



WIRESHARK



- Nuno Fidalgo - 21140369
- Ricardo Calvão - 21210839
- Telmo Cristóvão - 21160536

Índice

Introdução.....	3
O que são redes sem fios	3
Porque é necessário encriptação nas redes sem fios	4
Encriptação WiFi	5
O que é a encriptação WPA2	5
Como funciona o WPA2	6
Vulnerabilidade WPA2	7
Principais sistemas afetados	9
Como podemos proteger-nos	10
Demonstração.....	10
Diagrama	10
Testes efetuados	11
Teste #1: Sem vulnerabilidades	11
Teste #2: Vulnerabilidade detetada na group key	12
Teste #3: Completamente vulnerável.....	14
Capturas Wireshark.....	15
Conclusão	16
Referências.....	16
Recursos	16

Introdução

O tema deste trabalho tem como objetivo falar sobre segurança de redes sem fios, com especial atenção à encriptação **WPA2** e a vulnerabilidade que foi descoberta no início de 2017 por Mathy Vanhoef.

Vamos abordar o que são redes sem fios e como é feita a proteção nestas redes, assim como a necessidade de haver mecanismos de encriptação como forma de proteção principal dos dados transmitidos.

Explicaremos como funciona a vulnerabilidade que foi descoberta, tal como as suas consequências, quais os sistemas mais afetados e como nos podemos proteger.

O nosso trabalho tem como principal fonte de informação disponível o *website* oficial <https://www.krackattacks.com> e foram usados os programas e scripts disponíveis no **github** <https://github.com/vanhoefm/krackattacks-scripts> utilizados para uma breve demonstração ao vivo que iremos executar no final da apresentação deste trabalho.

Para mais informações sobre o tema recomendamos que aceda diretamente ao trabalho de investigação disponível em <https://papers.mathyvanhoef.com/ccs2017.pdf>.

O que são redes sem fios

As redes sem fios são redes que interligam diversos computadores através de comunicação rádio, por isso se chama redes sem fios, porque ao contrário do que era normal até à data, apenas existiam comunicações através de ligações com fios, o denominado cabo de rede, ou **RJ-45**, temos ainda as ligações telefónicas, coaxiais e óticas.

As redes sem fios (wireless) têm dominado o mercado nos últimos anos devido à sua rápida evolução, quer em velocidade, facilidade de implementação, de uso quotidiano, baixa manutenção e a diminuição de cabos usados para ligar um computador, fator de grande reclamação por parte de utilizadores.

Este tipo de rede funciona de vários modos, entre estes temos o modo **ad-hoc**, **mesh** e ponto de acesso, como os mais comuns, sendo este último o mais utilizado.

Cada modo usa o protocolo **802.11** com variantes das normas **N/B/G/AC**, na qual cada norma tem como principais características a velocidade de comunicação, a distância máxima e frequência espectral usada.

Porque é necessário encriptação nas redes sem fios

Como falamos no tópico anterior as redes sem fios são uma forma de comunicação através de ondas de rádio por algoritmos de modelação de frequências.

Este tipo de comunicação funciona muito similar as ondas de rádio AM/FM para as quais só precisamos de saber qual a frequência para poder sintonizar e ouvir uma determinada estação rádio e/ou informação que pretendemos ouvir/aceder.

Nas comunicações de computadores não é bem este tipo de comportamento que pretendemos e apesar do comportamento acima descrito esteja correto, de forma bastante básica, se não existir algum género de proteção, qualquer computador pode sintonizar e ter acesso a todo os dados das transmissões vizinhas, para os quais pode até manipular e aceder a dados não autorizados.

Por esta razão é que temos que proteger os dados das comunicações das redes sem fios e para isso recorremos a encriptação, neste caso do meio físico que é a transmissão sem fios ou pela encriptação da modelação de frequências de sinal.

De momento existem dois tipos de encriptação de redes sem fios, a encriptação **WEP** (Wired Equivalent Privacy) e a **WPA** (Wi-Fi Protected Access) sendo que esta tem de momentos duas versões e o âmbito do nosso trabalho incide apenas na versão **WPA2** como iremos aprofundar no próximo capítulo.

Encriptação WiFi

Neste capítulo vamos apresentar o que é encriptação **WPA2** (Wi-Fi Protected Access 2), como funciona a vulnerabilidade descoberta e os principais sistemas afetados.

O que é a encriptação WPA2

O **WPA2**, criado em 2003, corresponde à versão final do protocolo **802.11i** que é um novo certificado para as redes sem fios, tornando-as mais confiáveis e seguras. Esta tecnologia foi baseada no modelo WPA ao qual ainda é utilizado hoje em dia e que mantém uma boa margem de segurança na sua utilização.

O **WPA2** é um protocolo que utiliza o **AES** (Advanced Encryption Standard), sistema de encriptação mais seguro e mais pesado do que o **WPA** original. O **WPA** utiliza o algoritmo **RC4**, modelo de encriptação utilizado no **WEP**, que troca com frequência a chave de encriptação para a segurança da comunicação utilizando o **TKIP**. Esta é a principal diferença entre o **WPA** anterior em relação ao **WPA2**.

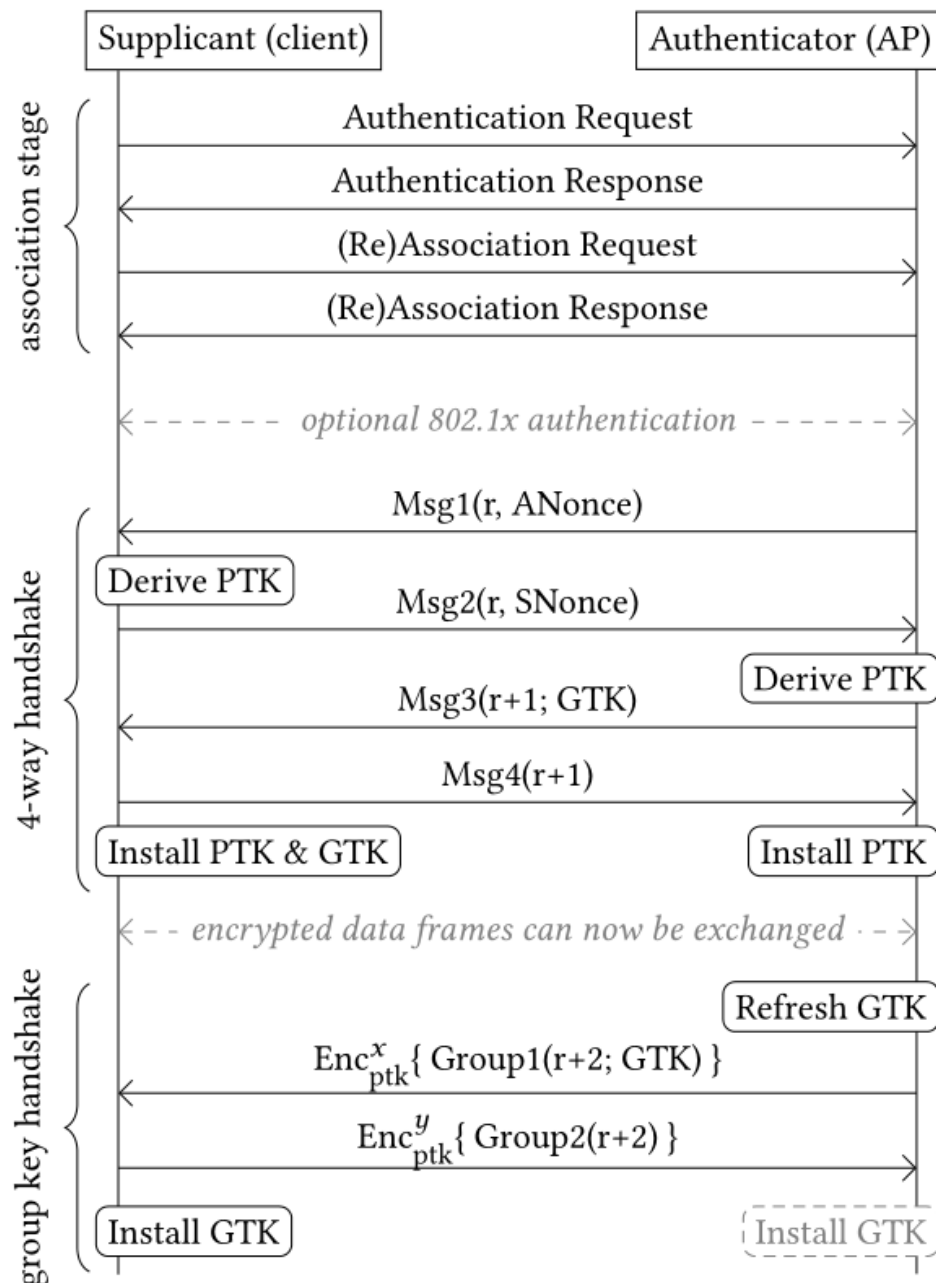
O **AES** é um sistema de encriptação bastante seguro, baseado na utilização das chaves de 128 a 256 bits que torna a tarefa de invasão dos sistemas, mesmo diante de uma falha no algoritmo, bem mais complicada.

Uma desvantagem do **WPA2** é que o **AES** exige mais processamento para ser executado, logo o seu uso é recomendado para quem necessita de um elevado padrão de segurança. De qualquer forma, as máquinas atuais são capazes de suportar este processamento extra sem grandes problemas.

O **WPA2** utiliza protocolos como o **RADIUS**, **802.1x**, **EAP**, **TKP**, **AES** e **RSN** (Robust Security Network) e oferece os modos de operação **Enterprise** (Infraestrutura) e **Personal** (Preshared Key) e possui características adicionais de segurança em relação ao **WPA** para qual o objetivo principal do **WPA2** é fornecer suporte aos produtos para que possam utilizar todos os recursos de tecnologia não presentes no sistema **WPA**. Ambos os sistemas fornecem autenticação e criptografia, disponibilizando a garantia de confidencialidade, autenticidade e integridade em redes sem fios.

Como funciona o WPA2

Esta imagem foi retirada do estudo feito e mostra como funciona o 4way-handshake que estabelece as comunicações iniciais, de forma a criar uma key única para dois dispositivos comunicarem.



Vulnerabilidade WPA2

A vulnerabilidade descoberta que apresentamos neste trabalho é resultante de uma má implementação no software **wpa_supplicant** em **Linux** que consequentemente afetou o **Android**, visto este sistema ser baseado em **Linux** e software open-source.

Esta vulnerabilidade explora uma funcionalidade do protocolo **WPA2** que tem a ver com o **4WayHandshake** feito inicialmente quando um cliente se liga a um Ponto de Acesso (AP, Access Point) e estabelece as chaves de encriptação próprias para cada cliente, de forma a poder comunicar de forma segura.

O protocolo tem como referência que no **4WayHandshake** inicial pode ocorrer falhas de comunicação e pede para repetir a terceira mensagem sucessivas vezes até conseguir estabelecer as chaves de encriptação para a comunicação com este cliente.

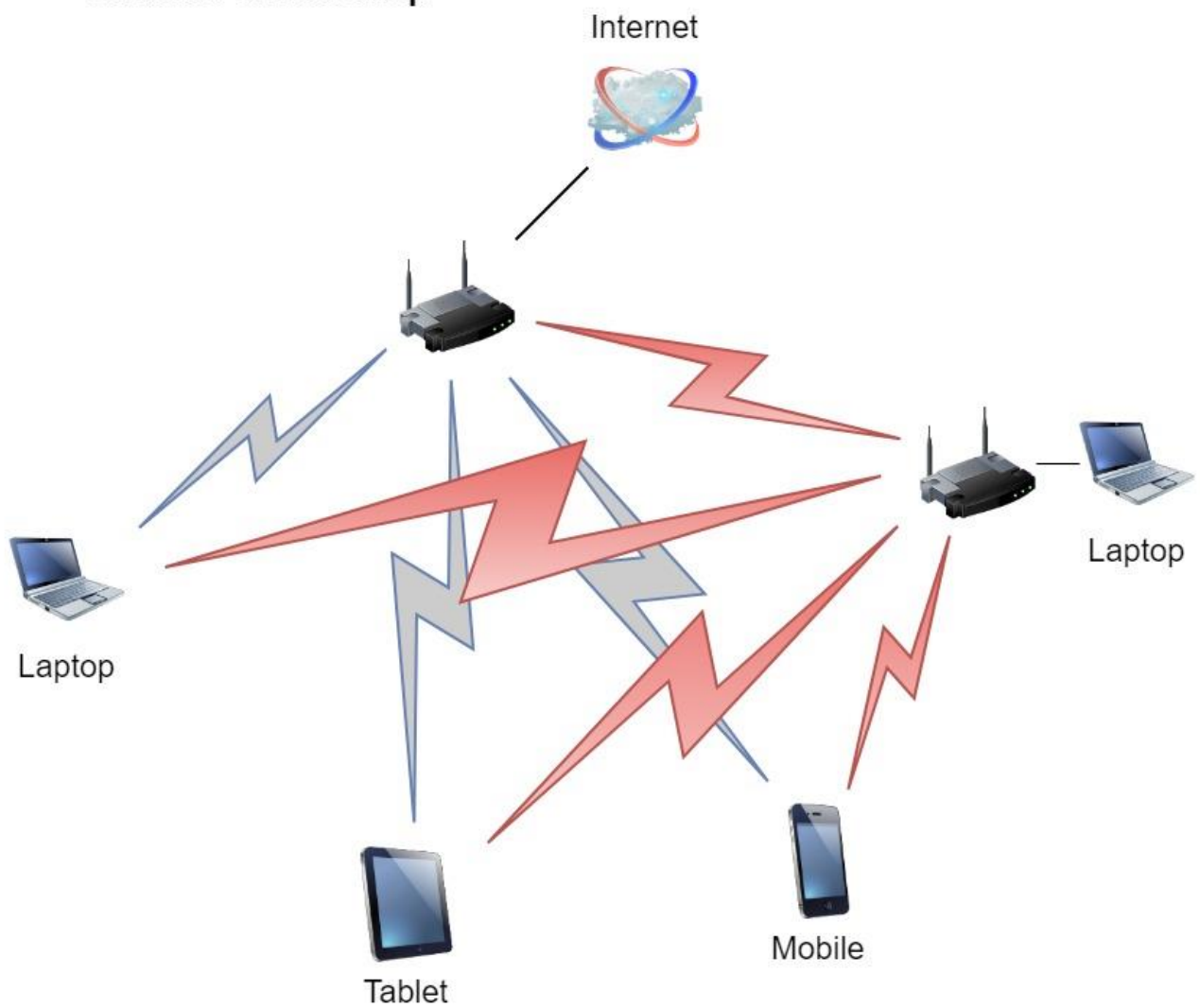
Através de uma falha na implementação do software **wpa_supplicant** versões anteriores à 2.6 é possível mudar a terceira mensagem colocando-a totalmente a zeros (0) ou uns (1), em binário, o que faz com que seja possível fazer uma reinstalação da chave de comunicação entre clientes e ponto de acesso, tendo desta forma acesso a dados protegidos e encriptados.

No *website* oficial tem uma boa explicação deste fenómeno, assim como um vídeo de demonstração, onde usam um computador com uma placa de rede sem fios extra de forma a clonar um **AP** (ponto de acesso), dando acesso à *Internet*, e onde usam um telemóvel/Tablet com a versão 6 do **Android** ao qual este se liga a este falso AP e com a reinstalação da chave de encriptação conseguem aceder a todo o tipo de tráfego entre o cliente e a *Internet*, até mesmo aos dados de login de um qualquer site que se aceda no *Internet*.

Uma breve nota na qual, é usado um outro software chamado **SSL_trip**, que tem como função desabilitar as conexões SSL para a *Internet*, visto ser um outro tipo ataque/vulnerabilidade à qual foge ao âmbito deste trabalho.

Esta demonstração tem como principal forma de ataque o chamado **Homem-no-meio (Man-in-the -Middle - MitM)**, como mostramos no diagrama em seguida.

Cenário: coffee shop



Principais sistemas afetados

Como falamos anteriormente esta falha/vulnerabilidade apareceu devido a um erro de implementação de um *software* em Linux, e como tal, muitos dos sistemas que usam esse software também ficaram vulneráveis, mas assim que esta questão foi tornada pública, rapidamente saíram as devidas correções.

O quadro em seguida mostra testes efetuados a vários sistemas que foram estudados:

Implementation	Re. Msg3	Pt. EAPOL	Quick Pt.	Quick Ct.	4-way	Group
OS X 10.9.5	✓	✗	✗	✓	✓	✓
macOS Sierra 10.12	✓	✗	✗	✓	✓	✓
iOS 10.3.1 ^c	✗	N/A	N/A	N/A	✗	✓
wpa_supplicant v2.3	✓	✓	✓	✓	✓	✓
wpa_supplicant v2.4-5	✓	✓	✓	✓ ^a	✓ ^a	✓
wpa_supplicant v2.6	✓	✓	✓	✓ ^b	✓ ^b	✓
Android 6.0.1	✓	✗	✓	✓ ^a	✓ ^a	✓
OpenBSD 6.1 (rum)	✓	✗	✗	✗	✗	✓
OpenBSD 6.1 (iwn)	✓	✗	✗	✓	✓	✓
Windows 7 ^c	✗	N/A	N/A	N/A	✗	✓
Windows 10 ^c	✗	N/A	N/A	N/A	✗	✓
MediaTek	✓	✓	✓	✓	✓	✓

^a Due to a bug, an all-zero TK will be installed, see Section 6.3.

^b Only the group key is reinstalled in the 4-way handshake.

^c Certain tests are irrelevant (not applicable) because the implementation does not accept retransmissions of message 3.

Como podemos proteger-nos

Como podemos verificar, bastantes sistemas estão vulneráveis a esta falha, mas temos que ter em atenção que devemos continuar a usar o WPA2, visto não haver nenhuma alternativa e irá haver um **WPA3** mas de momento a maioria dos equipamentos não tem um mecanismo melhor.

Certifiquem-se de que os vossos sistemas são atualizados com as devidas correções de software para este efeito e temos que ter atenção a que redes ligamos os nossos dispositivos, tendo em especial atenção a sítios com redes públicas partilhadas sem qualquer tipo de autenticação, sendo o cenário ligar a redes públicas de cafés e outros do género será onde corremos o maior risco.

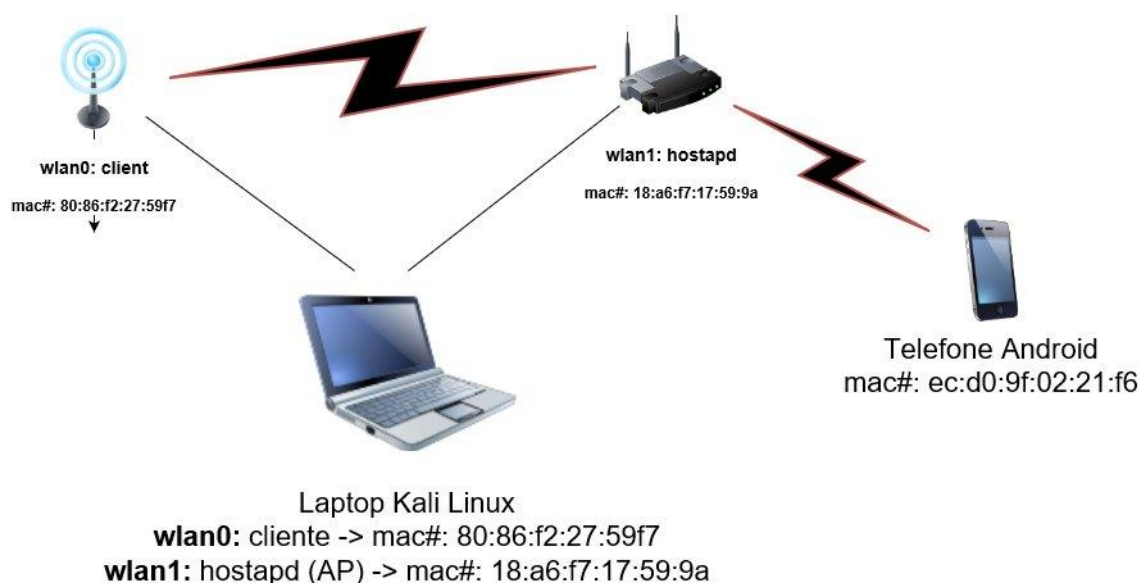
Demonstração

Para a demonstração vamos usar um portátil com **Kali Linux** instalado e uma placa de rede **TP-Link WN722N V1**, onde tivemos a sorte um dos membros do grupo ter esta placa e ter sido uma das razões para a escolha do tema, referimos que é a mesma placa que foi usada para criar os scripts que estão disponíveis no **github** de forma a efetuarem os testes e a demonstração do vídeo.

Diagrama

Demonstração:

Portátil Kali Linux como AP + cliente com segundo cliente telefone Android



Testes efetuados

Foram efetuados três testes com dispositivos diferentes como vamos mostrar em seguida, todos os screenshots e capturas apresentadas são relativas à interface que está a fazer de *Access Point* como *Man in the Middle*.

Teste #1: Sem vulnerabilidades

Telemóvel Xiaomi Redmi Note 4 com LineAge OS 14.1 (Android 7.1.2)

The screenshot shows the 'Phone status' application interface. The top status bar displays the time as 02:50 and battery level at 65%. The app title 'Phone status' is at the top. The interface is divided into two columns. The left column lists various status items, and the right column displays their corresponding values.

Item	Value
SIM 1 status Vodafone	Device model Redmi Note 4
SIM 2 status NOS	Android version 7.1.2
IMEI information	LineageOS version 14.1-20180411-NIGHTLY-mido
IP address fe80::eed0:9fff:fe02:21f6 192.168.6.174	LineageOS API level Guava (7)
Wi-Fi MAC address ec:d0:9f:02:21:f6	Android security patch level 5 March 2018
Bluetooth address Unavailable	Baseband version 953_GEN_PACK-1.112222.1.119607.1
Serial number 2be068a30104	Kernel version 3.18.31-perf-g3180dc0 jenkins@agrippa.acc.umu.se #1

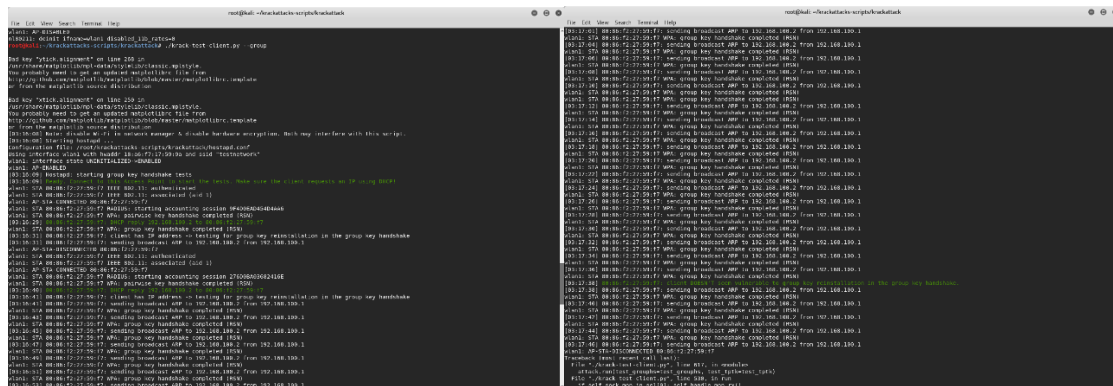
TESTE TPTK KEY:

Neste teste foi usado o script: `./krack-test-client.py -tptk`

[illegible]

Como vemos pelos screenshots este dispositivo não tem vulnerabilidade usado a *tptk key*.

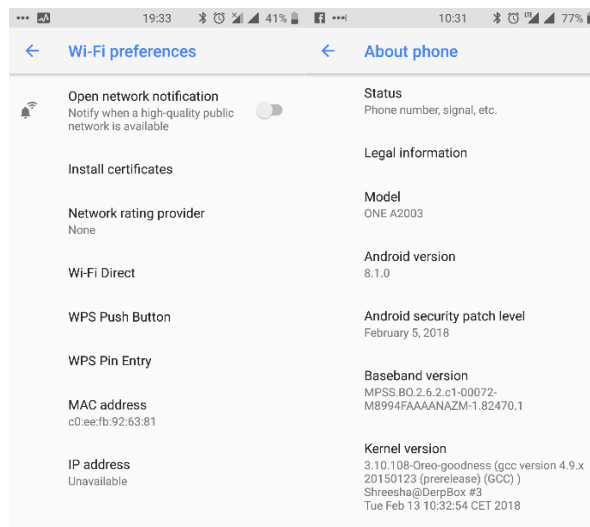
Neste teste foi usado o script: `./krack-test-client.py -group`



O mesmo resultado usado uma *group key*.

Teste #2: Vulnerabilidade detetada na group key

Telefóval One Plus 2 – Android 8.1.0



[illegible][illegible]

Teste #3: Completamente vulnerável

Telefóvel Oukitel K6000 Pro – MTK Android 7.0

TESTE TKTP KEY:

Neste teste foi usado o script: `./krack-test-client.py -tptk`

```

File Edit View Search Terminal Help
root@kali: ~ # ./krackattacks-scripts/krackattack.py
[49:48:11] Note: disable Wi-Fi in network manager & disable hardware encryption. Both may interfere with this script.
[49:48:12] Starting hostapd ...
Configuration file: /root/.krackattacks-scripts/krackattack/hostapd.conf
Using interface wlan0 with hwaddr 18:ae:47:17:39:9a and ssid "testnetwork"
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED

[49:48:13] wlan0: STA 00:00:00:00:00:00 IEEE 802.11: authenticated
wlan0: STA 00:27:15:c5:34:b8 IEEE 802.11: associated (aid 1)
[49:49:02] 00:27:15:c5:34:b8: Hostapd: Resetting TX IV of group key and sending Mgmt/4
wlan0: AP-STA-CONNECTED 00:27:15:c5:34:b8
[49:49:04] 00:27:15:c5:34:b8: RADIUS: starting accounting session 111763770150084
[49:49:04] 00:27:15:c5:34:b8: Hostapd: already installing pairwise key
[49:49:04] 00:27:15:c5:34:b8: Hostapd: Injecting Mgmt (with same ANonce) before Mgt/3 to test TPDK construction attack
[49:49:04] 00:27:15:c5:34:b8: Hostapd: Resetting TX IV of group key and sending Mgmt/4
[49:49:04] 00:27:15:c5:34:b8: WPA: received EAPOL-Key mgt/4 - invalid state (10) - dropped - MIC -1
[49:49:04] 00:27:15:c5:34:b8: usage of all zero key detected (WPA=seq=26). Client is vulnerable to (re)installation of an all-zero key in the pairwise
[49:49:04] 00:27:15:c5:34:b8: !!! Other tests are unreliable due to all-zero key usage, please fix this first !!!
[49:49:04] 00:27:15:c5:34:b8: Hostapd: Injecting Mgmt (with same ANonce) before Mgt/3 to test TPDK construction attack
[49:49:06] 00:27:15:c5:34:b8: Hostapd: Resetting TX IV of group key and sending Mgmt/4
[49:49:08] 00:27:15:c5:34:b8: Hostapd: Injecting Mgmt (with same ANonce) before Mgt/3 to test TPDK construction attack
[49:49:10] 00:27:15:c5:34:b8: Hostapd: Resetting TX IV of group key and sending Mgmt/4
[49:49:10] 00:27:15:c5:34:b8: Hostapd: Injecting Mgmt (with same ANonce) before Mgt/3 to test TPDK construction attack
[49:49:12] 00:27:15:c5:34:b8: Hostapd: Resetting TX IV of group key and sending Mgmt/4
[49:49:12] 00:27:15:c5:34:b8: Hostapd: Injecting Mgmt (with same ANonce) before Mgt/3 to test TPDK construction attack
[49:49:14] 00:27:15:c5:34:b8: Hostapd: Resetting TX IV of group key and sending Mgmt/4
[49:49:14] 00:27:15:c5:34:b8: Hostapd: Injecting Mgmt (with same ANonce) before Mgt/3 to test TPDK construction attack
[49:49:16] 00:27:15:c5:34:b8: Hostapd: Resetting TX IV of group key and sending Mgmt/4
[49:49:18] 00:27:15:c5:34:b8: Hostapd: Injecting Mgmt (with same ANonce) before Mgt/3 to test TPDK construction attack
[49:49:18] 00:27:15:c5:34:b8: Hostapd: Resetting TX IV of group key and sending Mgmt/4
[49:49:20] 00:27:15:c5:34:b8: Hostapd: Injecting Mgmt (with same ANonce) before Mgt/3 to test TPDK construction attack
[49:49:22] 00:27:15:c5:34:b8: Hostapd: Resetting TX IV of group key and sending Mgmt/4
[49:49:22] 00:27:15:c5:34:b8: Hostapd: Injecting Mgmt (with same ANonce) before Mgt/3 to test TPDK construction attack
[49:49:22] 00:27:15:c5:34:b8: Hostapd: Resetting TX IV of group key and sending Mgmt/4
Ctrl-Cback (most recent call):
wlan0: interface state ENABLED->DISABLED

```

Neste teste verificamos que este dispositivo tem a vulnerabilidade da *tptk key* e ainda que podemos fazer a reinstalação da chave com os valores todos a zero, sem esta a maior falha existente na implementação do WPA2.

TESTE GROUP KEY:

Neste teste foi usado o script: `./krack-test-client.py -group`

[illegible]

Este dispositivo também está vulnerável com a *group key*.

Capturas Wireshark

Temos os ficheiros com as capturas do wireshark que entregamos com o trabalho, mas não exploramos profundamente, devido a demonstração que fizemos mostra se o cliente esta ou não vulnerável, tal como provam os screenshots anteriores.

Conclusão

Neste trabalho limitamos-nos a comprovar o excelente trabalho desenvolvido por *Mathy Vanhoef*, onde demonstramos como podemos efetuar testes em vários dispositivos de forma a avaliarmos pontos de falha em redes wireless nas quais podemos a vir interagir ou pretendemos fazer auditorias futuras.

Temos de ter a noção que esta vulnerabilidade não é das mais críticas existentes atualmente e que devemos obrigatoriamente continuar a usar encriptação WPA2, no sentido em que temos de ter alguns cuidados extra na forma como usamos os nossos dispositivos e que devemos tentar ao máximo certificarmo-nos que as redes onde nos ligamos são redes confiáveis.

Nos casos em que os dispositivos são bastante vulneráveis, como no caso de dispositivos com chips MediaTek, então teremos de ser mais ainda mais cuidadosos porque este fabricante fez claramente uma má implementação do protocolo, tanto em software como hardware.

Outro fator a ter em conta é que esta vulnerabilidade não é uma falha no protocolo/standard do **WPA2**, mas sim maioritariamente uma falha na implementação em software/hardware por parte de alguns fabricantes.

Outra dificuldade encontrada a quem queira explorar mais este sobre este tópico, é na forma em como o ataque é efetuado, sendo um ataque do tipo Man in the Middle, o atacante tem de ter alguns recursos de equipamento e/ou estar perto da rede para implementar o ataque de forma transparente e sem ser detetado.

Para concluir, devemos então ser utilizadores mais consistentes e atentos no uso que fazemos diariamente e devemos continuar a usar a **WPA2**, de forma a termos comunicações seguras e fiáveis, e de que a maior parte dos fabricantes já tem esta ao corrente deste problema e já implementou as devidas correções e quanto mais tempo passar mais redes vão ser atualizadas no sentido de mitigar a falha encontrada.

Referências

- <https://www.krackattacks.com>
- <https://github.com/vanhoefm/krackattacks-scripts>
- <https://www.kali.org/news/kali-on-krack>
- <https://fossbytes.com/krack-attack-security-patch>
- <https://papers.mathyvanhoef.com/ccs2017.pdf>

Recursos

- Screenshots: Em anexo ao numa pasta com todos os screenshots recolhidos;
- Wireshark: Em anexo todos as capturas efetuadas;