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

NÁSTROJE PRO PENETRAČNÍ TESTOVÁNÍ WI-FI A IPV4

TOOLS FOR WI-FI AND IPV4 PENETRATION TESTING

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE — Bc. David Jančík

AUTHOR

VEDOUCÍ PRÁCE — doc. Ing. Zdeněk Martinásek, Ph.D.

SUPERVISOR

BRNO 2024

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní program **Informační bezpečnost**

Ústav telekomunikací

Student: Bc. David Jančík ***ID:*** 223326

Ročník: 2 ***Akademický rok:*** 2023/24

NÁZEV TÉMATU:

Nástroje pro penetrační testování Wi-Fi a IPv4

POKYNY PRO VYPRACOVÁNÍ:

Hlavním cílem diplomové práce je návrh a implementace podpůrných nástrojů pro bezpečnostní penetrační testování bezdrátových sítí Wi-Fi a síťové infrastruktury IPv4. Výsledné nástroje budou naprogramovány v jazyce Python a výstup testování bude v JSON formátu. V teoretické části nastudujte současný stav problematiky (dostupné metodologie), navrhněte diagram obsahující postup penetračního testu pro výše uvedené oblasti. Na základě analýzy navrhněte a implementujte nástroje, které budou testovat specifické zranitelnosti Wi-Fi sítí a síťové infrastruktury IPv4. Funkčnost nástrojů ověřte ve virtualizovaném prostředí.

DOPORUČENÁ LITERATURA:

[1] KONDO, Tabu S.; MSELLE, Leonard J. Penetration testing with banner grabbers and packet sniffers. Journal of Emerging Trends in computing and information sciences, 2014, 5.4: 321-327.

[2] BRADBURY, Danny. Hacking wifi the easy way. Network Security, 2011, 2011.2: 9-12.

Termín zadání: 5.2.2024 ***Termín odevzdání:*** 21.5.2024

Vedoucí práce: doc. Ing. Zdeněk Martinásek, Ph.D.

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

ABSTRAKT

Diplomová práce se zabývá návrhem a implementací podpůrných nástrojů a metodologie pro bezpečnostní penetrační testování bezdrátových sítí Wi-Fi a síťové infrastruktury IPv4. Teoretická část se věnuje samotnému penetračnímu testování, přístupů, fázi a typům. Dále je popsán vývoj Wi-Fi sítí a jejich bezpečnostní protokoly. Jsou představeny různé penetrační nástroje pro Wi-Fi sítě a také typy útoků. V poslední teoretické části je popsán základní přehled o IPv4 a nástroje pro skenování IPv4. Nejprve se v praktické části navrhne vlastní metodologie pro Wi-Fi sítě a IPv4 a nástroje pro penetrační testování. Definuje se programovací jazyk Python a výstup jednotlivých nástrojů pro platformu Penterep. Proběhne rešerše nástrojů z teoretické části pro zvolení vhodných nástrojů pro nové podpůrné nástroje. Vlastní implementace penetračních nástrojů vychází z vytvořeného návrhu diagramu. Závěrem jsou shrnuty dosažené výsledky a návrhy na další rozšíření nástrojů pro Wi-fi a IPv4. Výsledkem této práce je implementace podpůrných nástrojů a návrh diagramu pro Wi-Fi sítě a IPv4.

KLÍČOVÁ SLOVA

penetrační testování, Wi-Fi, IPv4, metodologie, Penterep, nástroje, Python

ABSTRACT

The master thesis deals with the design and implementation of support tools and methodologies for security penetration testing of Wi-Fi networks and IPv4 network infrastructure. The theoretical part covers penetration testing itself, approaches, phases, and types. It also describes the development of Wi-Fi networks and their security protocols. Various penetration tools for Wi-Fi networks and types of attacks are introduced. In the last theoretical part, a basic overview of IPv4 and tools for IPv4 scanning is provided. Initially, in the practical part, a proprietary methodology for Wi-Fi networks and IPv4 and tools for penetration testing are proposed. The Python programming language is defined, along with the output of various tools for the Penterep platform. A review of tools from the theoretical part is conducted to select suitable tools for new support tools. The implementation of penetration tools is based on the design diagram created. The conclusion summarizes the results achieved and suggestions for further expansion of tools for Wi-Fi and IPv4. The result of this thesis is the implementation of support tools and the design diagram for Wi-Fi networks and IPv4.

KEYWORDS

Penetration Testing, Wi-Fi, IPv4, methodology, Penterep, tools, Python

Vysázeno pomocí balíčku `thesis` verze 4.07; `http://latex.feec.vutbr.cz`

Prohlášení autora o původnosti díla

Jméno a příjmení autora:	Bc. David Jančík
VUT ID autora:	223326
Typ práce:	Diplomová práce
Akademický rok:	2023/24
Téma závěrečné práce:	Nástroje pro penetrační testování Wi-Fi A IPv4

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné 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 závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné 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*

*Autor podepisuje pouze v tištěné verzi.

PODĚKOVÁNÍ

Chtěl bych vyjádřit upřímné poděkování svému vedoucímu práce, doc. Ing. Zdeňku Martináskovi Ph.D., za jeho cenné rady a podporu, stejně jako mé rodině za jejich neustálou motivaci a pochopení během mého studia. Dále děkuji kolegovi Mgr. Lukášovi Neudertovi za jeho odborné znalosti, které významně přispěly k mé diplomové práce. Vaše podpora a vedení byly nezbytné pro dosažení tohoto významného úspěchu.

Obsah

Seznam obrázků

Seznam tabulek

Seznam výpisů

Úvod

„Promiňte, dáte mi heslo od Wi-Fi?“ S touto obyčejnou otázkou se setkal alespoň každý jednou v dnešní době. Kamkoliv jdeme, můžeme využívat bezdrátovou síťovou technologii Wi-Fi (Wireless Fidelity) pro připojení k internetu. Může se jednat o veřejné sítě v knihovně, v městské hromadné dopravě, v kavárně či privátní sítě, například v práci nebo doma.

Wi-Fi usnadňuje uživatelům připojení k internetu z různých míst a na různé vzdálenosti. Obecné pravidlo pro domácí sítě je, že směrovače Wi-Fi pracující v pásmu 2,4 GHz mají dosah v interiéru až 45 metrů a 90 metrů venku. Díky této technologii směrovače nepotřebují síťové kabelové propojení. Novější směrovače 802.11n a 802.11ac, které pracují v pásmech 2,4 GHz i 5 GHz, dosahují větších vzdáleností [1]. Jednou z dalších výhod Wi-Fi je její jednoduché připojení k internetu, které nevyžaduje technické znalosti. Připojení k internetu prostřednictvím Wi-Fi je mnohem pohodlnější než přes fyzické porty ve směrovači, které jsou omezené počtem. Tímto způsobem se zlepšuje přístup pro zařízení IoT (Internet of Things). Nejnovější verzí je 802.11ax, známá také jako Wi-Fi 6E, která byla uvedena na trh v roce 2021. Tato technologie nabízí mimořádně vysokou rychlost, až 9,6 Gb/s a podporuje široké frekvenční pásma 2,4 GHz, 5 GHz a 6 GHz [2]. V neposlední řadě Wi-Fi snižuje náklady na mobilní data, protože uživatelé mohou využít Wi-Fi pro stahování nebo vysílání obsahu.

Přestože Wi-Fi sítě poskytují mnoho výhod, je kritické se zaměřit na zabezpečení a implementovat nezbytná opatření pro ochranu soukromí a dat. Taková nezbytná opatření můžou být náročnější z hlediska bezpečnosti, protože existují rizika, která nevyžadují fyzický přístup ke směrovači, včetně neautorizovaných sítí, útoky na hesla nebo útoky na šifrovací protokoly.

Když se osoba dostane do sítě, vznikají bezpečnostní rizika i v rámci IPv4 (Internet protocol version 4) protokolu, který umožňuje komunikaci mezi zařízeními v síti. IPv4 provoz může být snadno odposloucháván, což znamená, že útočníci mohou získat citlivé informace, jako jsou hesla a komunikační data. Bezpečnostní zranitelnosti v IPv4 sítích zahrnují také skenování portů, možnost útoků typu MITM (Man In The Middle) a další.

Cílem této práce bude navrhnout a implementovat podpůrné nástroje a metodologii pro penetrační testování Wi-Fi sítí a protokolu IPv4. V praktické části se práce zaměří na vytvoření těchto specifických nástrojů a diagramů, které slouží k provádění a vizualizaci procesu bezpečnostního testování.

První část práce se bude věnovat základním principům penetračního testování, přičemž bude kladen důraz na různé přístupy, typy a fáze, kterými může penetrační testování probíhat. Současně bude popsán samotný proces provádění těchto testů.

Druhá část práce se zaměří na bezdrátové Wi-Fi sítě, konkrétně na standard IEEE 802.11. Popíše se vývoj standardu 802.11 a jeho role v moderních bezdrátových komunikacích. Práce dále prozkoumá bezpečnostní protokoly, hrozby a techniky útoků, které mohou ohrozit bezdrátové sítě, a představí se dostupné nástroje pro realizaci těchto útoků.

Třetí část se bude věnovat protokolu IPv4, kde se popíšou jeho hlavní charakteristiky a nabídne se přehled nástrojů, které lze využít pro testování bezpečnosti IPv4 sítí.

Praktická část bude popsána od čtvrté části práce, kde se vyobrazí vlastní návrhy metodologie a nástrojů pro penetrační testování s důrazem na použití programovacího jazyka Python. Tato část bude zahrnovat příklad popisu výstupu nástrojů, specifický návrh pro Wi-Fi sítě včetně využití existujících nástrojů, fází penetračního testování a návrhu diagramu. Později se tato část zaměří na vlastní návrhy nástrojů pro testování sítí IPv4, opět s důrazem na využití existujících nástrojů a vytvoření návrhu diagramu.

V páté části budou popsány vlastní implementace penetračních nástrojů. Zahrnou se zde nástroje pro Wi-Fi sítě, jako jsou skenování přístupových bodů, zachycení handshaku, odhalení skrytých sítí, prolomení WPA (Wi-Fi Protected Access), útok opakováním, fragmentační a KoreK chopchop útok. Dále budou představeny nástroje pro IPv4 sítě, včetně skenování IP adres a portů a pro testování anonymního FTP (File Transfer Protocol) přístupu. Tato část poskytne podrobný přehled o vývoji a funkčnosti jednotlivých penetračních nástrojů.

V závěrečné části práce se shrnou dosažené výsledky, zhodnotí se efektivita jednotlivých nástrojů a navrhnou se další vylepšení. Budou představeny konkrétní návrhy na rozšíření funkcionality a optimalizaci stávajících nástrojů. Cílem této části bude poskytnout podklad pro možný budoucí vývoj této práce.

1 Penetrační testování

Penetrační testování, často nazývané etický hacking, představuje autorizovaný útok na systémy nebo sítě. Jedná se o simulovaný útok za účelem ověření bezpečnosti analyzovaného systému nebo prostředí. Penetrační testování se zaměřuje na vyhledávání a identifikaci zranitelností (Vulnerabilities) a jejich zneužití (Exploits), které existují v rámci IT (Information Technology) organizace infrastruktury a pomáhá potvrdit, zda jsou současná zavedená bezpečnostní opatření účinná [3].

Tyto testy se dělí na různé typy testování, například na síťové služby, webové aplikace, bezdrátové připojení, mobilní aplikace nebo sociální inženýrství [4]. Tato práce se bude primárně zaměřovat na standard IEEE (Institute of Electrical and Electronics Engineers) 802.11 a protokol IPv4.

Principem penetračního testování je identifikovat a zneužít bezpečnostní zranitelnosti, které by mohly být potencionálně zneužita neoprávněnými osobami. Během testování se používají různé automatizované nástroje, techniky a metody, které se využívají za účelem simulovat reálný útoky na systémy. Díky tomu se poskytují organizaci klíčové informace, které popisují, jakým stylem útoku mohou být jejich systémy zranitelné. Testy zahrnují různé simulace a odlišné typy útoků, které byly stanovené ve smlouvě, aby nedocházelo v případech kritických míst k ohrožení bezpečnosti organizace. Dalším úkolem je prozkoumat odolnost daného systému proti útokům jak z ověřených, tak neověřených uživatelů a to z hlediska neautorizovaného přístupu, běžných uživatelů či těch s administrátorskými právy. Rozhoduje se zde i role, zdali testování probíhá v rámci interní sítě, nebo pokrývá pouze komunikaci směřující z internetu. Jestliže se správně objasní rozsah, může dojít k testování prakticky k jakékoliv části systému. Během penetračního testování dochází k analýzám zranitelností, tj. analýza softwarových a hardwarových prvků v systému. To může zahrnovat, například hledání chyb v kódu aplikací, v operačním systému nebo dalších komponentech systému. Dále se tester snaží nalézt chyby v konfiguraci a řízení systému, nedostatečně zabezpečené nastavení nebo nedostatečné řízení přístupových práv [3] [4] [5].

Výstupem testování je následná zpráva, ve které se identifikují a zhodnocují zranitelnosti z hlediska rizik. To pomáhá organizaci určit jejich priority, které zranitelnosti jsou nejnaléhavější a vyžadují okamžitou pozornost a které naopak ne. Je zde také popsané, jaké bezpečnostní opatření by měla organizace zavést, aby nadále zabránila zneužití bezpečnostních rizik. Je možné zahrnout opravu zranitelností, úpravy konfigurace, aktualizace softwaru a firmwaru nebo jiné kroky k posílení bezpečnosti systému a ochraně před potenciálními útoky [3] [4] [6].

1.1 Metody penetračního testování

Penetrační testování lze dělit na 3 různé metody přístupu. První z nich je bílá skříňka (White-Box). V této skříňce se nacházejí veškeré informace o infrastruktuře a systémech. Další z nich je tzv. černá skříňka (Black-Box). V černé skřínce nejsou poskytnuty žádná data o infrastruktuře ani systémech. Posledním případem se stává šedá skříňka (Grey-Box), kde dochází ke kombinaci obou skříněk [3].

Bílá skříňka

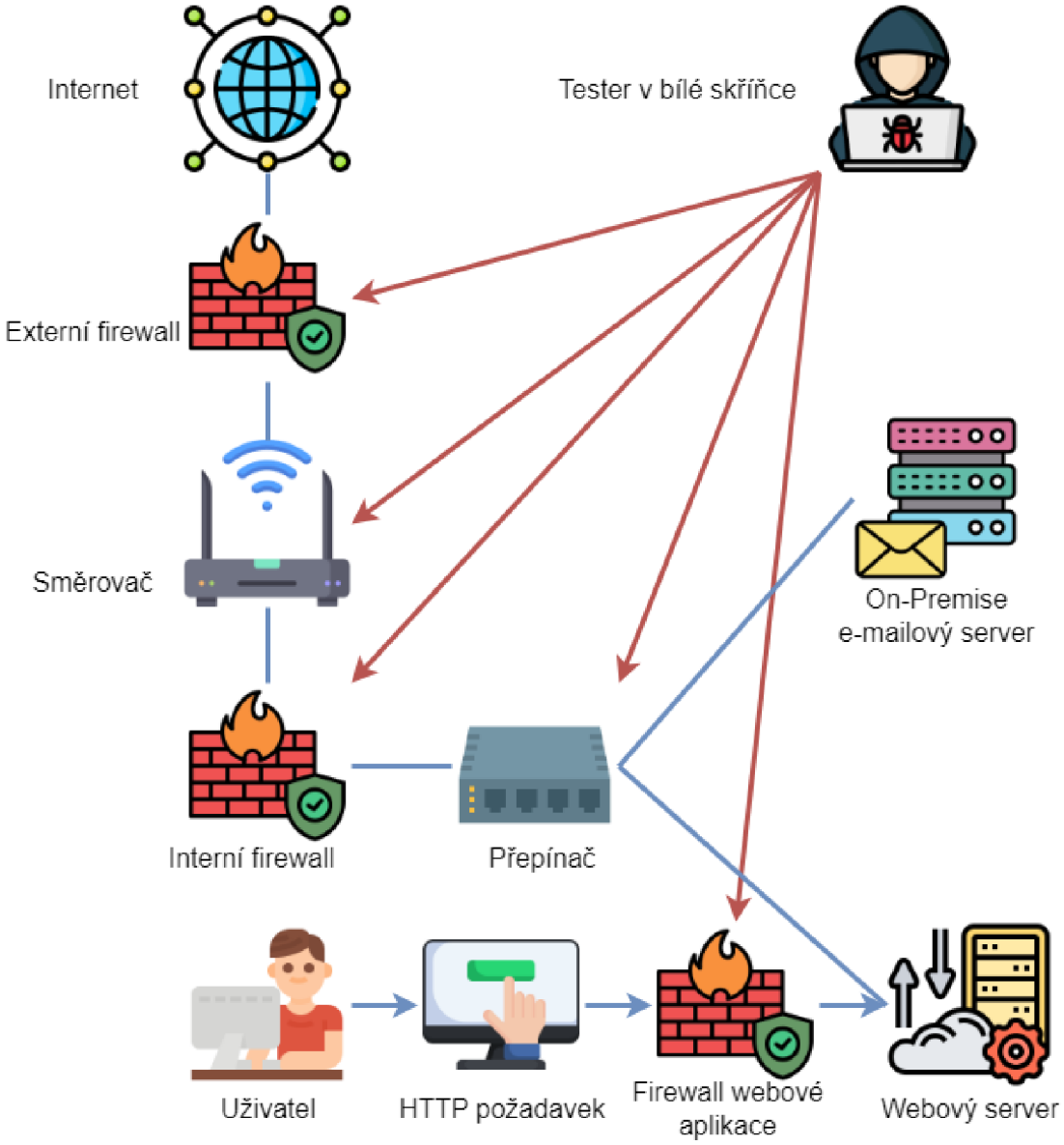
V případě bílé skříňky se jedná o testování zabezpečení, kdy je přístup k úplným informacím o systému, aplikaci nebo infrastruktuře která se testuje. Patří sem často informace o operačních systémech, schématu a rozvržení sítě, zdrojových kódů a případně i některá hesla a další technické aspekty softwaru. Často se provádí, když má organizace plný přístup k testovacím systémům (On-Premise), buďto ve vývojovému nebo testovacímu prostředí. Tato forma testování je velmi efektivní při identifikaci a odstraňování bezpečnostních rizik, protože se získají cenné informace o testovacím prostředí a může se provádět důkladná analýza tohoto prostředí. Toto schéma lze vidět na obrázku 1.1 [3] [4].

Výhodou této skříňky jsou detailní znalosti organizace, od které se odvíjejí testy, jelikož se detailně zná testovaný softwary, včetně kódu, infrastruktury a konfigurace. Dalším benefitem jsou lepší výsledky, které odhalí více zranitelností, které by jinak mohly zůstat nepovšimnuty. V neposlední řadě lze doporučit jak opravit zranitelnosti na úrovni kódu, což umožňuje rychlé nápravy [7].

Jednou z nevýhod je nerealistický pohled na testování, protože jsou zde neomezené přístupy a znalosti o systému, což nemusí vždycky odrážet reálné podmínky. Jiné nevýhody by mohli zahrnovat i v některých případech vyšší náklady, protože se musí investovat více času do analýzy a přípravy penetračního testování a mnohdy to může být i časově náročnější z důvodu podrobné analýze a testování na úrovni kódu [7].

Černá skříňka

Černá skříňka je metoda přístupu, kdy nejsou předány informace o interním fungování nebo infrastruktuře, která se testuje. Jsou nabídnuty pouze minimální anebo žádné informace o systému a jeho cílem je identifikovat a využít zranitelnosti systému [3]. Z obrázku 1.1 by bylo patrné, že by testující v černé skříňce neměl žádné informace o systémech a služeb, který běží v organizaci, tedy od testera by nevycházely žádný výstupy a vnitřní struktura by byla zamlžená.



Obr. 1.1: Schéma bílé skříňky.

Výhodou white-box testování je, že se testující chová a přemýšlí jako skutečný útočník, který nemá žádné předchozí znalosti o cílovém systému. Tímto způsobem se může zjistit, jakými prostředky může být systém napaden a jakým stylem je možné zlepšit jeho zabezpečení [4]. Další výhodou, zároveň i nevýhodou je, že zde není potřeba před začátkem testování chápat do detailu vnitřní síť organizace a nemusí se studovat znalost vnitřního kódu. Kvůli tomu se může stát, že nebude podchycena některá chyba na úrovni kódu anebo se nepokryje některé scénáře a chování aplikace či systému [7].

Šedá skříňka

Šedá skříňka spojuje zmíněné přístupy tím, že je poskytnutý částečný přístup k interním informacím a znalostem o systému, ale chybí například úplný přístup k infrastruktuře jako v případě bílé skříňky [3] [4]. Tyto částečné znalosti mohou zahrnovat omezený přístup ke zdrojovému kódu, některým konfiguračním informacím nebo dokumentaci systému. Tím se zvyšuje úroveň znalosti, které chybí v případě černé skříňky, ale stále se zachovává určitá autentičnost. Například na obrázku 1.1, by testující šedé skříňky mohl mít přístup k interní síti, věděl by, že tam běží webový server, ale nemusel by mít informace o lokálním e-mailovým serveru, tedy na obrázku by byl zamlžen.

Tímto způsobem může šedá skříňka vybalancovat pohled na bezpečnostní zranitelnosti aplikace nebo systému a umožňuje organizaci identifikovat a opravit potenciální rizika před tím než se stane cílem skutečného útoku.

1.2 Typy penetračního testování

Existují různé systémy, aplikace, služby nebo jiné zdroje, které jsou využívané v kybernetickém světě, na kterých závisí naše každodenní aktivita. Každý subjekt by měl mít zajištěnou ochranu jeho digitálních aktiv před potenciálními hrozbami a útoky. Jedná se hlavně o zdroje, které daná organizace využívá. Může zde spadat interní a externí penetrační testování, testování bezdrátové sítě, využití technik sociálního inženýrství, mj. testování API (Application Programming Interface), cloud computingu, mobilních zařízení a mnohem více [8]. Níže jsou stručně popsány některé základní typy penetračních testů.

Externí penetrační testování

Cílem externího penetračního testování je testovat organizaci z internetu. Nejprve se zaměřuje na dostupně veřejné informace a externě viditelná aktiva organizace. Snaží se pomocí zpravodajství z otevřených zdrojů, dále již OSINT (Open-Source Intelligence), získat co největší počet informací jako jsou IP (Internet Protocol) adresy, domény, subdomény, certifikáty, struktura e-mailových adres, o uživatelích, webové stránky, hesla a mnoho dalšího. Je zde snaha o identifikaci služeb o najití verzích služeb, známých zranitelnostech a jejich potenciálního zneužití [4] [8] [9] [10].

Interní penetrační testování

Interní penetrační testování probíhá již uvnitř sítě organizace. Jedná se o bezpečnostní proces, při kterém organizace prověřuje své vlastní informační systémy, sítě

a aplikace s cílem identifikovat zranitelnosti, které by mohly být zneužity útočníky zevnitř organizace. Toto testování simuluje útoky za bránou firewallu prováděním autorizovaným uživatelem se standardními přístupovými právy. Útoky simulované při interním penetračním testování odhalují zranitelnosti, které by mohli využít nejen zaměstnanci, ale i jiní uživatelé s přístupem do systému nebo útočníci, kteří si tento přístup zajistili dříve, například při externím penetračním testu [4] [8] [9].

Penetrační testování webových aplikací

Webové stránky jsou stále častějším zprostředkovatelem informací a komunikačním nástrojem, ale také jsou mnohdy zranitelné vůči útokům. Útočníkům mohou umožnit přístup k citlivým informacím, neoprávněný přístup k účtům nebo ke spuštění vzdáleného kódu, kdy může dojít až ke kompromitaci systému. Nejznámějšími útoky, jsou XSS (Cross-Site Scripting), který umožňuje vložit kód na dynamickou webovou stránku a SQL (Structured Query Language) injekce, která zahrnuje zavádění dotazů SQL do vstupních formulářů zranitelného serveru [11].

Sociální inženýrství

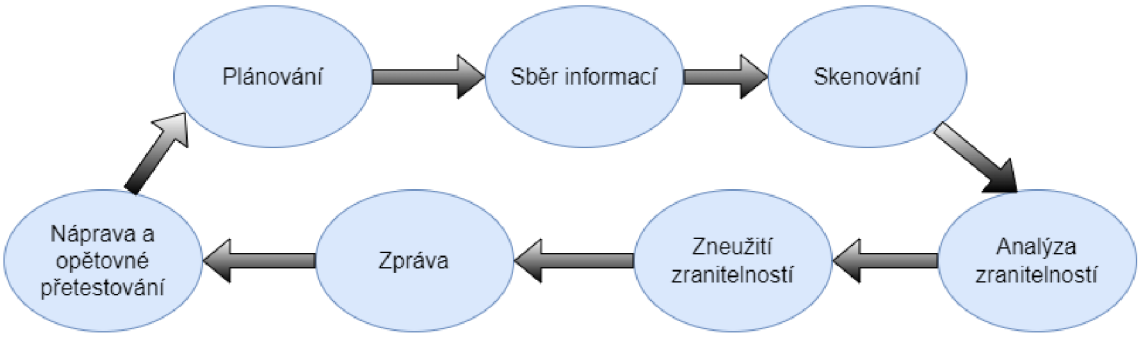
Sociální inženýrství je metoda, která popisuje, jak lze přesvědčit jednotlivce nebo firmy, aby vykonali akce, které jsou od nich vyžadované, např. odhalí důvěrné informace jako jsou hesla či interní data. Tento typ útoku využívá přirozenou lidskou tendenci důvěřovat a spoléhá se na sociální interakce lidí. Phishingové e-maily, kdy se osoba vydává za důvěryhodný zdroj, jsou jedny z nejznámějších metod, jak vylákat z obětí důvěrné informace. Kromě toho existují další metody, jako je tzv. vishing, podvodné telefonní hovory, nebo tzv. tailgating, neoprávněný vstup do budov či místností za někým, kdo má oprávnění, které útočníci používají k získání přístupu k citlivým údajům nebo fyzickým prostorům [12].

Penetrační testování bezdrátových sítí

Bezdrátové sítě používají rádiové vlny k přenosu dat, díky kterým čelí různým bezpečnostním hrozbám. Tato technologie může být zneužita k překonání firewallu, odposlechu citlivých informací, zachycování datových paketů nebo vysílání škodlivého obsahu. Cílem těchto testů je zjistit, jak dobře je síť schopna odolávat různým útokům a zneužití. Proces zahrnuje několik podstatných kroků, jako je identifikace sítě, sběr informací o dané síti, skenování pro identifikaci slabých míst, a vyhodnocení používaných šifrovacích technologií. Tyto kroky pomáhají zajistit, že citlivé informace přenášené bezdrátově jsou chráněny před neoprávněným přístupem a útoky [8] [13] [14].

1.3 Fáze penetračního testování

Penetrační testování se skládá z několika fází. Mezi ně patří plánování, sběr informací, skenování, analýza zranitelností, zneužití zranitelností a zpráva, popř. náprava a opětovné přetestování. Každá fáze má svůj význam a přispívá k celkovému pochopení bezpečnostních hrozeb v testovaném systému či sítí.



Obr. 1.2: Fáze penetračního testování.

Plánování

Počáteční fáze pro provádění penetračního testování, kde dvě strany, poskytovatel penetračního testování a cílová organizace, diskutují o právních důsledcích a pravidlech testování. Hlavním cílem této fáze je stanovení podstaty testu, tj. vyjasnění specifických požadavků organizace, včetně definování oblastí, které se nemají testovat. Také se zde vyjasňuje rozsah penetračních testů, což zajišťuje, že obě strany jsou obeznámeny v tom, co se bude testovat. Rozsah testů určuje, které aktiva jsou předmětem testování a jaké typy zranitelností budou zkoumány. Toto vymezení rozsahu má klíčový vliv na všechny následující fáze penetračního testování [15].

Sběr informací

V této fázi se shromažďují informace o cíli, jako jsou IP adresy, detaily domén, mailové servery a topologie sítě. Průzkum může být buď pasivní, využívající veřejně dostupné informace nebo aktivní zahrnující přímou interakci s cílovým systémem. Úkolem je shromáždit co nejvíce informací pro vytvoření účinné strategie útoku [15] [16].

Skenování

Pro třetí fázi se používají různé nástroje k prohledávání cílového systému za účelem identifikace otevřených portů, skenování zranitelností a analýzy síťového provozu.

Otevřené porty jsou potenciálně vstupní body pro útočníky. Skenování zranitelností zahrnuje testování aplikace nebo operačního systému na známé zranitelnosti, kde se používají automatizované nástroje nebo manuální testy pro testování aplikací nebo operačních systémů [15] [16].

Analýza zranitelností

Analýza zranitelností se zaměřuje na identifikaci potenciálních výskytů zranitelností v systémech. V průběhu analýzy zranitelností se využívají data získaná z průzkumu a skenování k určení možnosti zneužití identifikovaných zranitelností a určuje se jejich závažnost. Na základě rizika, které představují zranitelnosti, se upřednostní. Tento proces je podpořen globálně přijímanými metodami jako CVSS (Common Vulnerability Scoring System) a databáze jako NVD (National Vulnerability Database), které poskytují standardizované hodnocení závažností. Tyto fáze společně zajišťují, že každá zjištěná zranitelnost je řádně zanalyzována a adekvátně řešena [15] [16].

Zneužití zranitelností

Zneužití zranitelností umožňuje získat přístup k systému pomocí zranitelných míst odhalených v předchozích fázích. Jakmile se identifikuje vstupní bod, začne se zkoumat, jaké další zdroje lze přes něj získat. Během této fáze je nutné zachovat opatrnost, aby nedošlo k narušení nebo omezení obchodních činností. Následně se testují možné dopady tohoto vstupního bodu, jako je lokalizace citlivých dat, identifikace komunikačních kanálů a potenciální zneužití dalších systémů v síti [15] [16].

Zpráva

Do finální fáze penetračního testování patří zpracování zprávy, ve které jsou popsány nalezené zranitelnosti. Tato zpráva obsahuje podrobnosti o zranitelnostech, včetně jejich popisu, závažnosti, dopadu, důkazu konceptu a doporučení pro opravu zranitelnosti [15] [16].

Náprava a opětovné přetestování

Fáze nápravy a opětovného přetestování je realizována podle předchozího plánování a zájmu klienta. Závěrečné opětovné přetestování je nabízeno k potvrzení, že všechny zranitelnosti byly řádně opraveny a nezůstaly žádné nezajištěné bezpečnostní mezery [15].

1.4 Přístupy penetračního testování

Přístupy provádění penetračního testování zahrnuje kombinaci tří klíčových přístupů: manuálního, poloautomatizovaného a automatizovaného testování. Všechny tyto kombinace slouží pro identifikaci a vyhodnocení bezpečnostních rizik informačních systémů.

Manuální testování

Manuální testování aktivně hodnotí bezpečnost cílového systému. Díky tomu je umožněno identifikovat zranitelnosti, které mohou automatizované nástroje přehlédnout. Využívají se zde vlastní znalosti, zkušenosti a kreativity k objevení zranitelnosti a potenciálního způsobu útoku. Tato fáze často zahrnuje techniky, jako jsou analýzy kódu nebo logické chyby. Ať už se jedná o jakýkoliv manuální přístup k penetračnímu testování, zahrnuje to člověka nebo skupinu lidí, kteří musí pečlivě zkontrolovat dané systémy a najít v nich zranitelnosti [17].

Automatizované testování

Automatizované testování zahrnuje použití automatizovaných nástrojů a skriptů k provedení širokého otestování cílového systému. Tyto nástroje mohou hledat známé zranitelnosti, nesprávné konfigurace a další běžné bezpečnostní problémy. Příklady automatizovaných nástrojů pro penetrační testování zahrnují skenery zranitelností, skenery sítí nebo skenery webových aplikací. Automatizované testování je cenné pro rychlé identifikování jednoduše známých zneužitelných zranitelností a běžných bezpečnostních problémů [17].

Poloautomatizované testování

Poloautomatizované testování je kombinací dvou výše zmíněných procesů k otestování bezpečnosti informačních systémů. Zahrnuje použití automatizovaných nástrojů a skriptů k identifikaci známých zranitelností. Zároveň se provádí manuální testování pro odhalení potenciálních útoků a sofistikovanějších zranitelností jako jsou logické chyby.

2 Wi-Fi sítě

Wi-Fi, známá jako IEEE 802.11, což je sada standardů, která definuje komunikaci pro bezdrátové sítě a stanovuje architekturu a specifikace těchto sítí.

2.1 Vývoj Wi-Fi sítí

Od roku 1997 se vývoj standardů IEEE 802.11 neustále zdokonaloval. Pokrok se projevil ve zvýšené rychlosti přenosu dat, zabezpečení, šířce kanálu a byla zlepšena i spolehlivost. Jak byly původní standardy 802.11 rozšiřovány o další funkce, začaly být označovány svými změnami, např. 802.11a, 802.11b, 802.11g. Popis jednotlivých standardů lze nalézt v tabulce 2.1 a jsou stručně popsány níže.

Standard	Rok uvedení	Pásmo	Maximální přenosová rychlost	Šířka kanálu	Modulace
802.11a	1999	5 GHz	54 Mb/s	20 MHz	OFDM
802.11b	1999	2,4 GHz	11 Mb/s	22 MHz	DSSS, CCK
802.11g	2003	2,4 GHz	54 Mb/s	20 MHz	OFDM, DSSS, CCK
802.11n	2009	2,4/5 GHz	600 Mb/s	20/40 MHz	OFDM, MIMO
802.11ac	2013	5 GHz	6,8 Gb/s	20/40/80/160 MHz	OFDM, MU-MIMO
802.11ax	2019	2,4/5 GHz	9,6 Gb/s	20/40/80/160 MHz	OFDM, MU-MIMO
Wi-Fi 6E	2021	2,4/5/6 GHz	9,6 Gb/s	20/40/80/80+80/160 MHz	OFDM, MU-MIMO, MU-OFDMA
802.11be	2024	2,4/5/6 GHz	46 Gb/s	20/40/80/80+80/160/320 MHz	OFDM, MU-MIMO, MU-OFDMA, MLO

Tab. 2.1: Přehled IEEE 802.11 standardů.

Standard IEEE 802.11a byl uveden v roce 1999 a pracoval s maximální přenosovou rychlostí 54 Mb/s (megabit za sekundu) a šířkou kanálu 20 MHz (megahertz). V praxi dosahoval tento standard okolo 20 Mb/s. Jako modulační techniku používal OFDM (Orthogonal Frequency Division Multiplexing) místo DSSS (Direct Sequence Spread Spectrum). Byl vhodný pro použití ve stíněných rádiových prostředí, díky pásmu 5 GHz (gigahertz). Nakonec nebyl standard 802.11a široce přijaty kvůli vysokým nákladům, omezenému dosahu a nekompatibilitě s 802.11b [18] [19].

Standard IEEE 802.11b byl předveden na trhu podobně jako IEEE 802.11a. Maximální rychlost dat se přenášela 11 Mb/s, šířka kanálu byla 22 MHz a používala se metoda CSMA/CA (Carrier-sense multiple access with collision avoidance). Využívalo se nelicencované frekvenční pásmo ISM (Industrial, Scientific, and Medical), které pracovalo v 2,4 GHz. IEEE 802.11b se stal prvním standardem, který se začal masivně používat a tak došlo k popularizaci Wi-Fi technologie. Standard byl použit v konfiguraci bod-více bodů (Point-To-Multipoint), kdy přístupový bod komunikuje s mobilními klienty v dosahu přístupového bodu [18] [19].

Standard IEEE 802.11g používá stejnou technologii OFDM, která byla představena se standardem 802.11a. IEEE 802.11g kombinuje výhody standardu 802.11b

a 802.11a. Nabízí přenosovou rychlost až 54 Mb/s, šířka kanálu je 20 MHz a operuje na frekvenci 2,4 GHz. Tento standard je zpětně kompatibilní s 802.11b [18] [19].

Standard IEEE 802.11n nástupem nového standardu IEEE 802.11n se zvýšila rychlost přenosu dat. Zatímco standard 802.11g nabízel maximální teoretickou rychlost 54 Mb/s, 802.11n umožňuje rychlost až 600 Mb/s, což představuje navýšení výkonu oproti předešlým standardům. Další významnou změnou se stala implementace MIMO (Multiple Input, Multiple Output) technologie, která umožňuje využívat více antén na straně vysílače i přijímače [18] [19].

Standard IEEE 802.11ac byl zaveden v roce 2013 pro zvýšení rychlosti a šířky pásma. Dosahuje maximální přenosové rychlosti 6,8 GB/s, což je značné zvýšení oproti ostatním standardům, operuje v pásmu 5 GHz a šířka kanálu dosahuje až 160 MHz. Důležitým inovativním prvkem je také technologie MU-MIMO (Multi-user MIMO), která umožňuje směrování až osmi prostorových proudů k více klientům současně, což zlepšilo efektivitu sítě. Implementovalo se také tzv. beamforming, který umožňuje směrovat signál přesně k určitému zařízení, místo aby se signál šířil všemi směry. Nechybí zde ani zpětná kompatibilita [18] [19].

Standard IEEE 802.11ax, známý také jako WI-Fi 6, zlepšuje spolehlivost a výkon Wi-Fi sítě, zejména v prostředích s velkým počtem připojených klientů. Maximální přenosová rychlost nabývá 9,6 GB/s. Toho dosahuje pomocí modulace OFDMA a plánováním alokaci zdrojů. Tento standard také podporuje různé techniky pro efektivní využití spektra, včetně MU-MIMO pro obousměrné spojení a zlepšení spektrální účinnosti. Cílem je dosáhnout vyšší rychlosti, spolehlivosti a efektivity sítí ve stále rostoucím prostředí s mnoha zařízeními připojenými k Wi-Fi. Využívá pásma 2,4 GH a také 5 GHz.

Standard Wi-Fi 6E se zavedl v roce 2021 jako rozšíření standardu Wi-Fi 6 do nového 6 GHz pásma, což umožnilo využití dodatečného bloku spektra povolené FCC (Federal Communications Commission). Tento standard nabízí maximální přenosovou rychlost 9,6 Gb/s, šířku kanálu dosahující až 160 MHz a zlepšuje kapacitu sítě pro podporu více zařízení. Důležitou novinkou je nižší latence a podpora pro 21 kanálů, které snižují riziko interference v hustě obydlených oblastech. Wi-Fi 6E také využívá technologie MU-OFDMA (Multi-User Orthogonal Frequency-Division Multiple Access) pro lepší správu přenosů v sítích s vysokým počtem připojených zařízení [20] [21].

Standard 802.11be očekávaný v roce 2024, přinese ještě vyšší rychlosti dosahující až 46 Gb/s a bude operovat ve frekvenčních pásech 2,4, 5 a 6 GHz. Tento standard zavede EHT (Extremely High Throughput) a nabídne šířku kanálu až 320 MHz v 6 GHz pásmu, což je dvojnásobek maximální šířky kanálu 5 GHz pásma. Wi-Fi 7 také přinese novou technologii MLO (Multi-Link Operation), která umožňuje simultánní přenos a příjem dat přes různé frekvenční pásma a kanály, čímž vý-

razně zvyšuje propustnost a snižuje latenci. Díky těmto inovacím poskytne Wi-Fi 7 výrazně lepší uživatelskou zkušenost pro aplikace vyžadující vysokou šířku pásma a nízkou latenci, jako jsou virtuální a rozšířená realita, což představuje další krok v evoluci bezdrátových sítí [20] [21].

2.2 Bezpečnost Wi-Fi sítí

Bezpečnostní protokoly Wi-Fi sítě jsou technická opatření, která slouží k zajištění ochrany a soukromí dat přenášených přes bezdrátovou komunikaci. Tyto protokoly jsou navrženy tak, aby zabránily neoprávněnému přístupu a zneužití Wi-Fi sítě od neoprávněných uživatelů. Tyto protokoly zajišťují integritu a důvěrnost dat a celkovou dostupnost sítě.

	WEP	WPA	WPA2	WPA3
Rok uvedení	1997	2003	2004	2018
Metoda šifrování	RC4	TKIP založená na RC4 / AES	AES / TKIP	AES-GCMP
Velikost klíče relace	64 bitů/128 bitů	256 bitů	256 bitů	256 bitů
Typ šifry	Proudová	Proudová	Bloková	Bloková
Integrita dat	CRC-32	MIC	CBC-MAC	SHA-384
Managment klíčů	Není	4cestný handshake	4cestný handshake	SAE handshake
Autentizace	Sdílený klíč	PSK a 802.1x s EAP variantou	PSK a 802.1x s EAP variantou	SAE a 802.1x s EAP variantou

Tab. 2.2: Přehled bezpečnostních protokolů pro Wi-Fi sítě.

WEP (Wired Equivalent Privacy) se stal prvním šifrovacím protokolem používaným ve Wi-Fi sítích. Zavedl se v roce 1997 jako součást původního standardu IEEE 802.11 pro bezdrátové sítě. Metoda šifrování je založena na symetrickém šifrovacím algoritmu RC4 (Rivest Cipher 4). Algoritmus pracuje s tokenem, který se generuje pomocí klíče a IV (Initialization Vector). Jsou zde podporovány dvě různé délky klíčů relace – 64bitovou a 128bitovou velikostí. IV se skládá z 24 bitů, z čehož vyplývá, že klíč tvoří 40 bitů pro 64bitovou verzi a pro 128bitovou verzi se skládá ze 104 bitů, který se přenáší v každým rámci v čitelné podobě. Je zde také implementována slabá hashovací funkce pro integritu dat. Díky tomu mohou útočníci snadno manipulovat s datovými pakety. Tento bezpečnostní protokol neposkytuje silný mechanismus autentizace. Často se používá sdílené heslo, které je náchylné k útokům typu hrubou silou, což zapříčiňuje snadné prolomení hesla. WEP používá statický klíč, který se aplikuje pro připojení k bezdrátové síti s aktivovaným bezpečnostním režimem [22] [23].

WPA se zavedl v roce 2003 jako reakce na zranitelnosti ve WEP protokolu. Využívá TKIP (Temporal Key Integrity Protocol), založený na protokolu RC4, jako primární metodu šifrování, která má robustnější mechanismy zabezpečení oproti

WEP, příkladem je rotace klíčů. Pro metodu šifrování lze také využit AES (Advanced Encryption Standard). Nechybí zde ani podpora klíčů o délce 256 bitů. TKIP využívá nový klíč pro každý datový paket. Do WPA byla implementovaná pokročilejší integrita dat. Používají se klíčové protokoly pro generování a řízení klíčů. Jedním z hlavních prvků je použití PSK (Pre-Shared Key) nebo EAP (Extensible Authentication Protocol) pro autentizaci a výměnu klíčů mezi klienty a přístupovými body. Pro domácí sítě je možné využít jednoduché PSK, zatímco pro obchodní a firemní prostředí se nabízí možnost použití EAP. [22].

WPA2 byl realizován v roce 2004 jako následovník původního standardu WPA. Používá se symetrická bloková šifra AES (Advanced Encryption Standard) jako primární metoda šifrování, avšak stále lze nastavit šifrování pomocí metody TKIP. Nechybí zde podpora klíčů o délce až 256 bitů. Bezpečnostní protokol využívá šifrovací algoritmus AES-CCMP (Advanced Encryption Standard - Counter Mode with Cipher Block Chaining Message Authentication Code Protocol) pro šifrování datových paketů a zahrnuje taky integritu dat. WPA2 používá PSK nebo EAP pro autentizaci [22].

WPA3 byl představen v roce 2018 jako nejnovější bezpečnostní standard pro Wi-Fi sítě, zavádějící pokročilé šifrovací a autentizační technologie pro zvýšení ochrany dat. WPA3-Personal využívá metodu šifrování SAE (Simultaneous Authentication of Equals), která je základem pro bezpečnou výměnu klíčů mezi zařízeními, designovanou pro autentizační účely, a produkuje vysokoentropní PMK (Pairwise Master Key), sloužící jako vstup pro 4cestný handshake, který vytváří PTK (Pairwise Transient Key). Jedním z klíčových přínosů SAE je jeho odolnost proti offline slovníkovým útokům, což zajišťuje, že i v případě zachycení autentizačních dat nemohou být hesla efektivně prolomena. SAE je založeno na Dragonfly protokolu, což je symetrický klient-klient (Peer-To-Peer) protokol, umožňující bezpečnou výměnu symetrických klíčů z nízkoentropního sdíleného tajemství přes nezabezpečené veřejné kanály, založené na diskrétních logaritmech a eliptických křivkách nebo konečných polích (Finite Fields) kryptografie. WPA3-Enterprise, přestože nebyl zásadně změněn oproti WPA2 verzi, zahrnuje vylepšení a zvyšuje odolnost proti zneužití, nabízí volitelný 192bitový bezpečnostní režim používající 256bitový GCMP (Galois/Counter Mode Protocol) pro ověřené šifrování. MFP (Management Frame Protection) je technologie používaná v protokolu WPA3, která hlavně slouží k ochraně před útoky typu deautentizace. Tyto útoky umožňují v předešlých bezpečnostních protokolech vynutit odpojení uživatelů od přístupových bodů. Do WPA3 se implementoval i přechodový režim mezi WPA3 a WPA2, který umožňuje současnou podporu obou bezpečnostních specifikací s použitím stejného hesla, což usnadňuje zařízením, která nepodporují WPA3, připojení k síti s modernějšími zařízeními, která WPA3 podporují [24] [25] [26].

2.3 Nástroje pro penetrační testování Wi-Fi sítí

Existuje mnoho nástrojů, které mají za úkol testovat Wi-Fi sítě z hlediska penetračního testování. Tento proces zahrnuje použití speciálních nástrojů a technik pro identifikaci a využití zranitelností v bezdrátových sítích. Efektivními nástroji jsou např. sada nástrojů Aircrack-ng, Wifite2, Airgeddon a Fern Wifi Cracker, které umožňují provádět různé útoky, od dešifrování WEP/WPA/WPA2 hesel po analýzu síťového provozu.

2.3.1 Sada Aircrack-ng

Aircrack-ng představuje kompletní sadu nástrojů určených pro testování zabezpečení Wi-Fi sítí. Tato sada se soustředí na různé oblasti zabezpečení, včetně monitorování, které umožňuje zachytávání paketů, provádění útoků jako jsou deautentizační útoky, útoky přehráním, prolamování hesel bezpečnostních protokolů, promiskuitního režimu a mnohem více [27].

Jednou z hlavních funkcí nástroje Aircrack-ng je prolamování klíčů WEP a WPA. To je zásadní pro posouzení síly a zranitelnosti bezpečnostních protokolů sítě [28].

Airodump-ng a Airmon-ng

Airodump-ng je nástroj, který slouží k vypsání všech sítí v okolí a zobrazení užitečných informací o nich. Jedná se o tzv. paketový sniffer, takže je v podstatě určen pro zachytávání paketů a surových rámců (raw frames) Wi-Fi, když je nastavený bezdrátový síťový adaptér v promiskuitním režimu. Spouští se proti všem sítím okolo a shromažďuje užitečné informace, jako jsou BSSID (Basic Service Set Identifier), ESSID (Extended Service Set Identifier), název kanálů, typy šifrování. Lze spustit i proti konkrétnímu AP (Access Point) k zachycení paketů z určité Wi-Fi sítě a dokáže ukládat veškerá data do textových souborů jako jsou CSV (Comma-separated values), CAP (Capture), TXT (Text). Nástroj neopomíjí ani sílu signálu a počet přenášených paketů [29] [30] [31].

Aby bylo možné využívat nástroj Airodump-ng, musí být nejprve nastaven bezdrátový síťový adaptér na promiskuitní režim. Airodump-ng nastavuje automaticky rozhraní do promiskuitního režimu, jestliže není nastaven manuálně. Manuální nastavení promiskuitního režimu lze vidět ve výpisu 2.1.

Ve výpisu se také nachází příkaz `airmon-ng check kill`. Airmon-ng je nástroj ze sady Aircrack-ng, pro správu promiskuitního módu bezdrátových síťových rozhraní. Promiskuitní mód umožňuje síťovému rozhraní zachytávat všechny síťové pakety, které jsou v okolí. Následně je argument `check`, který zkontroluje zda běží na systému nějaké procesy, které by mohly narušit používání bezdrátových nástrojů

v sadě Aircrack-ng. Poslední argument `kill` pak ukončí procesy, které byly identifikovány jako potenciálně problematické. Typickým příkladem takového procesu je správa síťového připojení, který automaticky spravuje bezdrátová rozhraní [32].

Výpis 2.1: Sekvence příkazů pro promiskuitní režim a vypnutí rušivých procesů.

```
$ sudo ifconfig wlan0 down
$ sudo iwconfig wlan0 mode monitor
$ sudo airmon-ng check kill
$ sudo ifconfig wlan0 up
```

Aireplay-ng

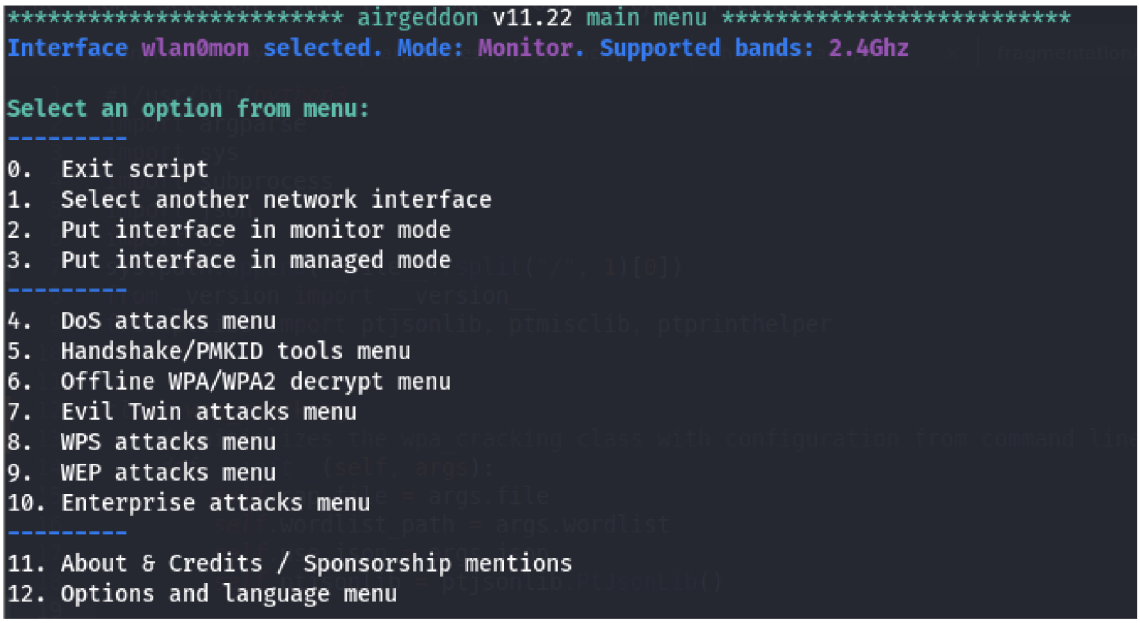
Aireplay-ng je nástroj používaný k vytvoření nelegitimního Wi-Fi provozu. Jeho primárním účelem je injektování (injection) rámců, což je důležité pro pozdější využití v Aircrack-ng k prolomení klíčů. Aireplay-ng umožňuje provádění několika útoků, jako je deautentizační útok, který pomáhá při zachytávání dat pro WPA handshake nebo útok falešné autentizace, při kterém jsou do přístupového bodu sítě injektovány pakety pro autentizaci za účelem vytvoření a zachycení nových IV. Mezi další typy útoků patří interaktivní opakování paketů, opakování ARP požadavků (ARP request replay), KoreK chopchop útok a útok pomocí fragmentování [33] [34].

2.3.2 Wifite2

Jedná se o nástroj pro auditování bezdrátových sítí, který je ovládaný v rozhraní příkazového řádku, prostřednictvím menu a navržen tak, aby zjednodušil a automatizoval proces testování zabezpečení bezdrátových přístupových bodů, který vychází ze sady nástrojů jako jsou Aircrack-ng, Reaver, Wash nebo Tshark. Byl navržen v jazyce Python s cílem zjednodušit proces skenování, auditování a detekce Wi-Fi klientů a zlepšit jeho přístupnost a efektivitu. Je zde implementovaný i proces pro lámání hesel. Nástroj je navržen tak, aby automaticky vybral nejlepší strategii útoku pro každou síť [35] [36] [37].

2.3.3 Airgeddon

Airgeddon je nástroj, který se ovládá pomocí rozhraní příkazového řádku, prostřednictvím menu, který slouží k auditu bezdrátových sítí. Je napsán v shellovém skriptu. Díky této volbě jazyka je flexibilní a relativně snadno pochopitelný, zejména pokud jde o způsob, jakým pracuje a vykonává své funkce. Pracuje jako wrapper (nadstavba) pro několik nástrojů třetích stran a integruje jejich funkce do jediného rozhraní. Obsahuje širokou škálu nástrojů pro skenování, útoky a zneužití bezdrátových sítí [38] [39]. Na obrázku 2.1 je zobrazeno úvodní menu s možnostmi výběru.



Obr. 2.1: Ukázka úvodního menu pro nástroj Airgeddon.

2.3.4 Fern Wifi Cracker

Fern Wifi Cracker využívá webové rozhraní navržené pro auditování Wi-Fi sítí. Tento nástroj, vytvořený pomocí programovacího jazyka Python a knihovny Python Qt GUI, provádí automatizovaný průzkum, identifikuje aktivní sítě v dosahu a používá nástroje jako je Aircrack-ng k proniknutí do sítě [40]. Obrázek 2.2 vyobrazuje ukázku úvodního menu pro tento nástroj.

2.3.5 Kismet

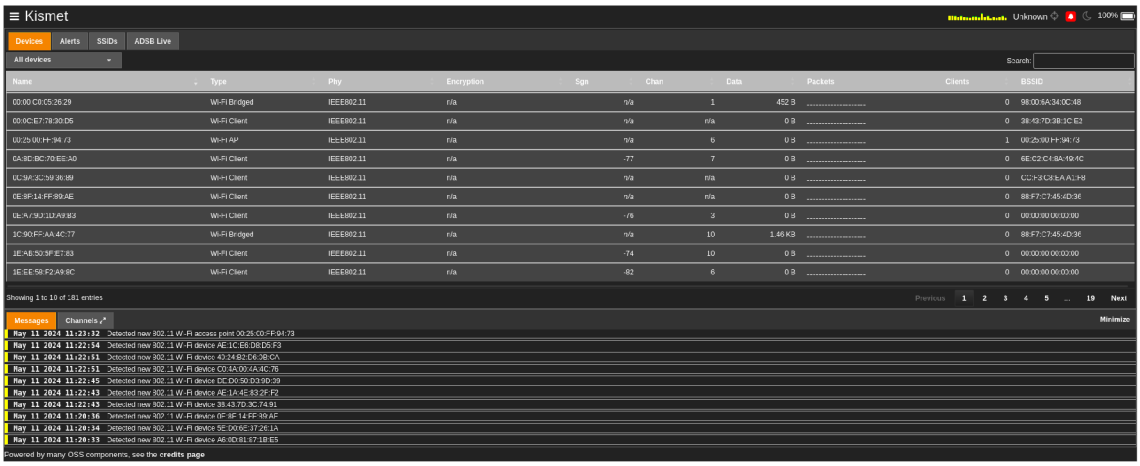
Kismet je primárně pasivní nástroj zaměřený na sběr a třídění bezdrátových dat. Tento nástroj funguje téměř zcela pasivně s několika výjimkami, jako je například režim skenování Bluetooth. Důležitým aspektem Kismetu je, že se obecně nejedná o nástroj pro útoky. Podporuje nejen Wi-Fi, ale i Bluetooth, Zigbee, RF (Radio Frequency) signály a další bezdrátové komunikace. Kismet je schopen pracovat s různými druhy hardwaru a ovladačů [41]. Silnou stránkou tohoto nástroje je schopnost identifikovat různé typy zařízení v síti. Umí detekovat typ zařízení, zda se jedná o Wi-Fi klienta, Wi-Fi AP, Wi-Fi v režimu Bridge a Wi-Fi zařízení. Na obrázku 2.3 lze vidět výstup z webového rozhraní Kismet.

Obr. 2.2: Ukázka úvodního menu pro nástroj Fern Wifi Cracker.

2.3.6 Wireshark

Slouží jako analyzátor síťových protokolů, který je široce používaný, jelikož zachytává a zobrazuje data paketů v reálném čase ze síťového rozhraní nebo souboru. Umožňuje uživateli zobrazit detaily síťového provozu pro řešení problémů, monitorování a analýzu výkonu sítě a bezpečnostních problémů. Má uživatelsky přívětivé grafické rozhraní a funguje ve více operačních systémech, včetně Linuxu, Windows, OS X a FreeBSD [42].

TShark je navržen speciálně pro uživatele, kteří dávají přednost práci v příkazovém řádku systému Linux. Jedná se v podstatě o terminálově orientovanou verzi programu Wireshark, která sdílí mnoho stejných funkcí, ale je optimalizována pro negrafické rozhraní [43].



Obr. 2.3: Ukázka webového rozhraní s odchytnutými daty pro nástroj Kismet.

2.3.7 Reaver a Wash

Reaver využívá zranitelností v implementaci WPS (Wi-Fi Protected Setup) k prolomení WPA/WPA2 zabezpečení Wi-Fi sítí. Nástroj Wash slouží k detekci bezdrátových sítí, které používají WPS, což je mechanismus usnadňující připojení nových zařízení k Wi-Fi síti. Wash skenuje bezdrátové sítě v dosahu a identifikuje ty, které mají aktivní WPS daemon, tedy proces na směrovači nebo přístupovém bodu, který spravuje WPS autentizaci [44] [45].

Wash také identifikuje obranné mechanismy směrovačů, jako jsou časová zpoždění mezi pokusy o PIN (Personal Identification Number) nebo dočasné deaktivace WPS po několika neúspěšných pokusech, což pomáhá pochopit, jak se směrovače brání proti takovým útokům. Pokud během útoku WPS daemon na směrovači přestane odpovídat, může to signalizovat přítomnost ochranných opatření [44] [45].

Nejprve se použije nástroj Wash k identifikaci směrovačů v dosahu, které pravděpodobně používají WPS daemon. Následně se využije nástroj Reaver k testování, zda je možné prolomit PIN zařízení metodou hrubé síly. Reaver potřebuje MAC (Media Access Control) adresu a kanál, na kterém zařízení komunikuje, a během procesu zobrazuje počet pokusů o uhodnutí PINu za sekundu a detekuje případná opatření, která zpomalují testování. Pokud WPS daemon na zařízení selže nebo přestane akceptovat PINy od Reavera, zobrazí se chybová hláška [44] [45].

2.3.8 Hcxdumptool, Hcxpcapngtool a Hashcat

Nástroj Hcxdumptool je speciálně navržen pro zachytávání paketů z Wi-Fi sítí. Tento nástroj se zaměřuje na protokoly WPA a WPA2 pro získávání dat, jako je handshake, která se používají pro útoky na základě hesel [46].

Primárním účelem Hcxpcapngtool, nyní často označovaný jako Hcxtools, je konverze a analýza zachycených paketů. Tento nástroj slouží k zpracování dat zachycených pomocí Hcxdumptool nebo podobných nástrojů, převádí je do formátů kompatibilních s dalšími nástroji pro testování zabezpečení a extrahuje klíčové informace potřebné pro útoky na hesla [47].

Hashcat slouží k urychlení procesu lámání hesel. Podporuje množství algoritmů šifrování a je výjimečně efektivní díky možnosti využití výpočetní síly GPU (Graphics Processing Unit). Hashcat umožňuje provádět různé typy útoků na hesla, včetně útoků hrubou silou a slovníkových útoků [48] [49].

2.3.9 Krackattacks-scripts

Krackattacks-scripts testují zabezpečení Wi-Fi klientů tím, že ověřují, jak dokážou klienti reagovat na různé útoky souvisejícími s instalací klíčů. Testy zahrnují ověřování, zda klient přijímá opakovaně odeslané vysílací rámce, správnou instalaci skupinových klíčů v procesu handshaku a zda nedochází k neoprávněné reinstalaci klíčů během 4cestného handshaku. Speciálnější testy se zaměřují na chování klientů při manipulaci s padělanými zprávami [50] [51].

2.3.10 Sada nástrojů pro WPA3

Dragonslayer, Dragondrain, Dragontime a Dragonforce jsou součástí pokročilé sady nástrojů určených pro bezpečnostní analýzu a testování protokolů WPA3. Každý z těchto nástrojů má specifickou úlohu ve výzkumu a testování zranitelností a společně poskytují komplexní přístup k identifikaci a řešení potenciálních zranitelností.

Dragonslayer provádí útoky na neplatné křivky proti klientům a serveru EAP-pwd. Tyto útoky obcházejí ověřování, tudíž útočníkovi stačí mít pouze platné uživatelské jméno [25] [52].

Dragondrain tento nástroj lze použít k testování, zda a do jaké míry je přístupový bod zranitelný vůči útokům typu DoS (Denial of Service) na SAE handshake protokolu WPA3 [25] [52].

Dragontime jedná se o experimentální nástroj pro provádění časových útoků proti SAE handshake, pokud jsou podporovány skupiny MODP 22, 23 nebo 24. Většina implementací WPA3 ve výchozím nastavení tyto skupiny nepovoluje [25] [52].

Dragonforce jedná se o experimentální nástroj, který využívá informace získané z útoků útočníka založených na načasováním nebo keše [25] [52].

2.4 Typy útoků na Wi-Fi sítě

S rozvojem Wi-Fi sítí začaly vznikat různé bezpečnostní hrozby, které potřebují pozornost v oblasti bezdrátových sítí. Tyto hrozby mohou nejen ohrozit integritu a soukromí dat, která se přenášejí přes Wi-Fi sítě, ale mohou například narušit dostupnost samotné sítě, což může nést vážné důsledky pro komunikaci a služby závislé na bezdrátovém připojení.

2.4.1 Prolomovací útoky na bezpečnostní protokoly

Prolomovací (Cracking) útoky na Wi-Fi sítě jsou techniky zaměřené na prolomení zabezpečení sítě, jejichž cílem je získat neoprávněný přístup. Tyto útoky jsou závislé na bezpečnostním protokolu, která daná síť využívá. Obecně se jedná o získání hesla. Při úspěšném prolomení hesla se obejdou bezpečnostní mechanismy a kompromituje se integrita síťové komunikace. V závislosti na zabezpečení implementované v bezpečnostních protokolech, jako jsou WEP, WPA, WPA2 nebo WPA3, se využívá různých metod a nástrojů pro průnik do sítě, zahrnuje to například zachytávání handshaku, na které se používají slovníkové útoky nebo útoky hrubou silou [53] [54].

Útok FMS (Fluhrer, Mantin, and Shamir) se stal prvním známým útokem na WEP. Tento útok se zakládá na způsobu, jakým WEP generuje svazky klíčů a také implementaci slabého IV, díky kterému se umožní shromáždit dostatečný počet paketů zašifrovaných těmito klíči, zanalyzovat je a získat klíč zpět. Počet IV, které je třeba shromáždit k dokončení útoku FMS, je přibližně 250 000 pro 40bitové klíče a 1 500 000 pro 104bitové klíče [53].

Progresivnějším útokem na WEP se stal útok PTW (Pyshkin, Tews, and Weinmann). Vychází z útoku FMS, ale nespoléhá na slabé IV jako útok FMS, díky čemuž se stal rychlým a účinným procesem útoku. Útok je schopen obnovit 104bitový klíč WEP s pravděpodobností úspěchu 50 % při použití méně než 40 000 rámců a s pravděpodobností 95 % při použití 85 000 rámců. Tento typ útoků vyžaduje shromáždění poměrně velkého množství rámců, což se provádí pasivně pomocí paketového snifferu, který monitoruje bezdrátový provoz na stejném kanále jako cílový přístupový bod a zachytává rámce. Problémem je, že za běžných podmínek se stráví poměrně dlouhou dobu pasivním sběrem všech paketů potřebných pro útoky, zejména u útoku FMS [53].

Slabé přihlašovací údaje mohou být příčinou proniknutí do sítě. Předsdílený klíč ve standardu WPA a WPA2 je náchylný k útokům pomocí slovníkových metod nebo metod hrubé síly [54].

V roce 2006 byla vyvinuta technika WPS (Wi-Fi Protected Setup) pro zjednodušení procesu připojování zařízení k bezdrátové síti. Tato technika umožňuje

automatickou konfiguraci zařízení buď stisknutím tlačítka, nebo zadáním krátkého PIN kódu. Primární metodou je útok hrubou silou na WPS PIN, což je většinou osmiciferný kód sloužící k autorizaci nových zařízení pro připojení k síti bez potřeby znalosti složitého hesla. Díky tomu, že poslední číslice PINu je kontrolní součet, efektivně se snižuje počet nutných pokusů k odhalení správného PINu. Dalším možným útokem je Pixie Dust Attack, který cílí na chyby v generování náhodných čísel při nastavení WPS, umožňující rychlé dešifrování WPS PINu a získání hesla k síti. V neposlední řadě je zde technika Null PIN Attack, využívající specifické chyby v implementaci WPS, kde úspěšný nastavení může být proveden bez poslání skutečného PINu [55].

WPA3 zahrnuje některé bezpečnostní funkce, které ztěžují útoky na hesla. Jeden z příkladu je SAE (Simultaneous Authentication of Equals), který slouží pro ochranu proti offline útokům hrubou silou [56].

2.4.2 Útok přehráním

Útokem přehráním (replay attack) dochází, když se odposlouchává zabezpečená síťová komunikace, ta se zachytí a poté se následně zpozdí nebo znovu odešle, aby byl uživatel naveden k tomu, co chce tester provézt [58].

Útok opakováním ARP požadavků

Útok je založen na opětovném vstřikování rámců (frame injection) do sítě, aby se v reakci na ně generoval provoz a rychleji se shromáždili potřebné IV. Typem rámce, který je pro tento účel nejvýhodnější, je požadavek ARP (Address Resolution Protocol), jelikož AP jej vysílá, pokaždé s novým IV [53].

Nejprve je zapotřebí zmonitorovat jaké bezdrátové sítě se nacházejí v dostupné vzdálenosti. To umožňuje nástroj Airodump-ng. První příkaz skenuje a analyzuje všechny sítě, které jsou dostupné v blízkosti. Jakmile se potvrdí, že v dané oblasti je dostupná testovaná Wi-Fi síť, v tomto případě `Diplomova Prace2023`, která používá bezpečnostní protokol WEP, tak se příkaz musí přerušit pomocí klávesové zkratky Ctrl + C.

Následně se přejde na specifickou Wi-Fi síť. Tato síť se identifikuje prostřednictvím její MAC adresy a přiřazeného komunikačního kanálu. MAC adresa konkrétní Wi-Fi sítě je určena pomocí parametru `--bssid`, zatímco používaný komunikační kanál je specifikován parametrem `--channel`. Pro účel prolomení hesla je také nutné uložit soubor použitím přepínače `--write`. V tomto kroku je možné vidět všechna zařízení, která jsou k tomuto přístupovému bodu připojena.

Pokud je Wi-Fi síť často používána a uživatelé se běžně připojují a odpojují, není nutné použít třetí příkaz. Falešná autentizace, prováděná pomocí přepínače

`--fakeauth` v nástroji Aireplay-ng, efektivně vytváří asociaci s cílovým přístupovým bodem. Je nezbytné zde definovat i přepínač `-h` určující MAC adresu zařízení, z kterého probíhá penetrační testování.

Čtvrtý příkaz začne generovat ARP pakety, které jsou odesílány na přístupový bod. Zatímco tyto pakety jsou odesílány, může být spuštěn poslední příkaz v dalším terminálu s nástrojem Aircrack-ng, který se snaží prolomit heslo ze souboru s příponou .cap. Celou sekvenci příkazů lze vidět ve výpisu 2.2. Je nezbytné a velmi důležité, aby příkaz s konkrétní Wi-Fi sítí a Aircrack-ng běželi synchronně v odlišných terminálech.

Výpis 2.2: Sekvence příkazů pro útok opakováním ARP požadavků.

```
$ sudo airodump-ng wlan0
$ sudo airodump-ng --bssid 1C:74:0D:D1:1D:9B --channel 1 --write
   arpRequestReplayAttack wlan0
$ sudo aireplay-ng --fakeauth 0 -b 1C:74:0D:D1:1D:9B -h 84:16:F9:19
   :BF:8A wlan0
$ sudo aireplay-ng --arpreplay -b 1C:74:0D:D1:1D:9B -h 84:16:F9:19:
   BF:8A wlan0
$ sudo aircrack-ng arpRequestReplayAttack.cap wlan0
```

2.4.3 Útok KoreK chopchop

KoreK chopchop útok, který dostal jméno podle svého tvůrce, se využívá v oblasti Wi-Fi sítích pro dešifrování datových paketů, jež jsou šifrovány pomocí protokolu WEP, a to i bez potřeby znát příslušný klíč. Ačkoliv neposkytuje možnost odhalit samotný klíč WEP, umožňuje odhalit nezašifrovaný obsah paketu. Útok probíhá pomocí zachycené šifrované zprávy z radiového frekvenčního proudu adresovanou cílovému přístupovému bodu. Útočník následně odhadne každý bajt datového zatížení (payload) paketu a správnost odhadu si ověří odesláním paketu a sledováním odpovědi od AP. Tento iterační proces umožňuje útočníkovi určit každý bajt datového zatížení [59] [60].

Rámec WEP se skládá z různých polí, jako jsou záhlaví, data a ICV (Integrity Check Value). Algoritmus ICV je implementací CRC32 (Cyclic Redundancy Check 32), který se vypočítává postupně pro každý bajt dat v rámci. Rámec využívá logickou operaci XOR (Exclusive Disjunction) s klíčovým proudem RC4. Při útoku chopchop se manipuluje s těmito prvky, aby se odvodila data. Útok spočívá v úpravě struktury rámce WEP a následném odhadnutí konkrétní hodnoty metodou pokus-omyl. Tato hodnota je rozhodující pro odvození jednoho bajtu dat a odpovídajícího bajtu proudu klíče. Opakováním tohoto postupu se odvodí celý proud klíčů [60].

První sekvence tří příkazů jsou stejná jak u útoku opakováním ARP požadavků, jelikož se opět tester snaží získat dostupné bezdrátové sítě. Čtvrtý příkaz aktivuje

chopchop útok pomocí přepínače `--chopchop`, který se snaží uhádnout obsah jednotlivých bajtů šifrovaného paketu. Pokud je paket úspěšně dešifrován, použije se nástroj Packetforge-ng k vytvoření ARP paketu. K tomu slouží přepínač `--arp`. Přepínač `-a` specifikuje BSSID cílového přístupového bodu a využívá soubor s příponou .xor, který byl vytvořen v předchozím kroku procesu. Přepínače `-k` a `-l` definují IP adresy odesílatele a cíle.

Předposlední krok zahrnuje použití přepínače `--interactive`, který spouští interaktivní režim odesílání paketů v nástroji Aireplay-ng. Tento režim umožňuje odesílat specifické pakety do sítě a přizpůsobovat injektování podle aktuální situace v síti. Přepínač `-r` specifikuje soubor s falešným paketem, který má být odeslán. Nakonec se opět pokusí Aircrack-ng v novém terminálu prolomit heslo podle zachyceného provozu v síti. Detailněji lze vidět sekvenci příkazů ve výpisu 2.3.

Výpis 2.3: Sekvence příkazů pro útok KoreK chopchop.

```
$ sudo airodump-ng wlan0
$ sudo airodump-ng --bssid 1C:74:0D:D1:1D:9B --channel 1 --write
   chopchopattack wlan0
$ sudo aireplay-ng --fakeauth 0 -b 1C:74:0D:D1:1D:9B -h 84:16:F9:19
   :BF:8A wlan0
$ sudo aireplay-ng --chopchop -b 1C:74:0D:D1:1D:9B -h 84:16:F9:19:
   BF:8A wlan0
$ sudo packetforge-ng --arp -a 1C:74:0D:D1:1D:9B -h 84:16:F9:19:BF:
   8A wlan0 -k 255.255.255.255 -l 255.255.255.255 -y chopchop.xor -
   w chop-forged-packet
$ sudo aireplay-ng --interactive -r chop-forged-packet wlan0
$ sudo aircrack-ng chopchopattack.cap wlan0
```

2.4.4 Fragmentační útok

Cílem fragmentačního útoku je získat dostatečné množství dat generovaných pomocí PRGA (Pseudo-Random Generation Algorithm), typicky okolo 1500 bajtů, která umožňují provádět injekční útoky. Útok začíná zachycením malého množství šifrovaných dat z paketů odeslaných přístupovým bodem. Tyto data obsahují části klíčů PRGA. Následně útočník vytváří a odesílá na AP specifické pakety, typicky ARP nebo LLC (řízení logického spoje), se známým nešifrovaným obsahem. Jestliže AP tyto pakety přijme a retransmituje je zpět do sítě, umožní to k získání dalších části PRGA klíčů z odpovídajících šifrovaných paketů. Tento proces se opakuje, dokud nejsou získány dostatečné informace pro rekonstrukci klíčů PRGA, což umožní dešifrovat a manipulovat s provozem v síti [61].

Fragmentační útok má obdobný postup jako útok chopchop. Příkazy se liší pouze ve čtvrtém a pátém příkazu. Cílem `--fragment` přepínače je získat dostatek dat

pro vytvoření nových paketů s použitím stejného šifrovacího klíče. Úspěšné provedení útoku vede k získání proudu klíčů, který se následně využije v nástroji Packetforge-ng, k vygenerování paketů pro další injekční útoky. Sekvence příkazů je zobrazena ve výpisu 2.4.

Výpis 2.4: Sekvence příkazů pro fragmentační útok.

```
$ sudo airodump-ng wlan0
$ sudo airodump-ng --bssid 1C:74:0D:D1:1D:9B --channel 1 --write
   chopchopattack wlan0
$ sudo aireplay-ng --fakeauth 0 -b 1C:74:0D:D1:1D:9B -h 84:16:F9:19
   :BF:8A wlan0
$ sudo aireplay-ng --fragment -b 1C:74:0D:D1:1D:9B -h 84:16:F9:19:
   BF:8A wlan0
$ sudo packetforge-ng --arp -a 1C:74:0D:D1:1D:9B -h 84:16:F9:19:BF:
   8A wlan0 -k 255.255.255.255 -l 255.255.255.255 -y fragment.xor -
   w fragment-forged-packet
$ sudo aireplay-ng --interactive -r fragment-forged-packet wlan0
$ sudo aircrack-ng chopchopattack.cap wlan0
```

2.4.5 Deautentizační útok

Deautentizační útok funguje na bázi deautentizačních rámců. Tyto rámce jsou součástí Wi-Fi sítí, které se používají legitimně k bezpečnému odpojení zařízení. Tímto způsobem se dají také zneužít k narušení spojení mezi uživateli a přístupovými body Wi-Fi, díky kterému může dojít až k DoS (Denial of Service) útoku, tedy všechna připojená zařízení budou opakovaně odpojena od sítě, což způsobí přerušení služeb a výpadky připojení pro legitimní uživatele. Dalším možným způsobem využití deautentizačního útoku je odpojení legitimního uživatele z Wi-Fi sítě pro získání WPA handshaku a následného pokusu o prolomení hesla k získání připojení k síti [62].

Přepínač `--deauth` předá nástroji Aireplay-ng parametr s číslicí, který značí, kolik deautentizačních rámců se má poslat na AP. V tomto případě se posílají 2 deautentizační rámce, aby uživatelé nepocítili výpadek konektivity. V případě hodnoty 0 by došlo k DoS, jelikož by se posílaly deautentizační rámce, dokud by nedošlo k ukončení příkazu v terminálu. Přepínač `-cm` následovaný MAC adresou určuje klienta připojené k cílovému přístupovému bodu, který má být odpojen. V tomto příkladu `84:16:F9:19:BF:8A` je MAC adresa cílového klienta.

Výpis 2.5: Příkaz pro deautentizační útok.

```
$ sudo aireplay-ng --deauth 2 -a 1C:74:0D:D1:1D:9B -c 84:16:F9:19:
   BF:8A wlan0
```

2.4.6 Wi-Fi spoofing

Wi-Fi spoofing spočívá ve vytvoření falešného síťového připojení pomocí zlého dvojčete (Evil Twin) nebo vytvoření nežádoucího přístupového bodu (Rogue Access Point), jež vypadá jako legitimní. Cílem spoofingu je ukrást citlivé informace, jako jsou přihlašovací údaje či finanční data od uživatelů [63] [64] [65].

Záměrem je vytvořit falešné přístupové body na místech s vysokou návštěvností, například kavárny, letiště či knihovny. Následně AP může vyzvat uživatele k vytvoření účtu pro přístup k určitým službám, včetně e-mailu a hesla. V praxi používá mnoho lidi stejné přihlašovací údaje na více platformách, díky kterému útočník získá přístup do jiných již zabezpečených platforem. Jakmile se uživatel připojí a vytvoří účet, může Wi-Fi spoofing dále eskalovat ke škodlivému kódu (Malware) nebo k přesměrování provozu. V neposlední řadě zde může dojít k útoku MITM, tedy útočník odposlouchává komunikaci mezi účastníky tak, že se stane aktivním prostředníkem [64] [65].

Jedním ze způsobů jak vytvořit falešné síťové připojení je využití zlého dvojčete. Hlavním cílem útočníka při útoku zlého dvojčete, je vytvořit falešný přístupový bod, který napodobuje legitimní přístupový bod se stejným ESSID. Jakmile se uživatel k tomuto přístupovému bodu připojí, tak všechna data, která v sítí uživatel sdílí, procházejí přes server ovládaný útočníkem. Útočník může vytvořit zlé dvojče pomocí chytrého telefonu nebo jiného zařízení podporující internet. Útoky tohoto typu jsou častější ve veřejných Wi-Fi sítích, které nejsou zabezpečené. Odhalit sítě zlých dvojčat může být mnohdy obtížné, protože útočník využívá sociální inženýrství k vytvoření falešných portálů, které kopírují obecné přihlašovací stránky používané mnoha veřejnými přístupovými body Wi-Fi [66] [67].

Dalším způsobem útoku je nežádoucí přístupový bod. Nežádoucí přístupový bod je jakýkoli bezdrátový přístupový bod v síti, který byl nainstalován na kabelovou infrastrukturu sítě bez souhlasu správce nebo vlastníka sítě [68].

2.4.7 Útoky znovuzavedením klíče

KRACK (Key Reinstallation Attack) se primárně týká zranitelnosti zařízeních připojených k Wi-Fi ve spojení s WPA2 protokolem. Útok využívá zranitelnosti v procesu 4cestného handshaku, který slouží pro ověření, že klient i přístupový bod mají správné přihlašovací údaje a zároveň zajišťuje vytvoření nového šifrovacího klíče pro šifrování veškeré následné komunikace. Při útoku KRACK se manipuluje a opakuje handshake zprávy, což klienta vede k nechtěné reinstalaci již používaného šifrovacího klíče. To může vést k opakovanému použití číselné hodnoty (Nonce) a narušení důvěrnosti dat [69] [70].

2.4.8 Útok Hole 196

Útok Hole 196 odhaluje zranitelnost ve standardech IEEE 802.11-2007, zejména v implementaci GTK (Group Temporal Key), který je určen k ochraně multicastového a broadcastového provozu, ale neposkytuje podporu pro párové klíče. Tato nedostatečnost umožňuje důvěryhodnému uživateli využít GTK pro spoofing multicastových zpráv, což vede k bezdrátovému ARP spoofingu. Tyto útoky umožňují zachytávat a manipulovat provoz tak, že se útočník umístí do pozice MITM prostřednictvím falšování ARP požadavků, aby se jevil jako síťová brána. Tato technika, zahrnuje oklamání stanic, aby odesílaly svůj šifrovaný provoz útočníkovi, který jej pak může dešifrovat, upravovat nebo zastavit [71].

2.4.9 Dragonblood

Dragonblood je označení pro sadu bezpečnostních útoků zaměřených na WPA3, nejnovější verzi bezpečnostního protokolu WPA, která měla zvýšit bezpečnost bezdrátových sítí. Tyto útoky se specificky soustředí na kompromitaci protokolu Dragonfly.

Útok postranním kanálem proti Dragonfly

Útok postranním kanálem je způsob, jakým lze získat citlivé informace z implementace systému, aniž by se tyto informace získávaly přímo z jeho výstupu. Tyto útoky využívají různé informace získané z fyzické implementace systému v tomto případě založené na času nebo právě podle aktivit v keši, aby odhalily citlivá data, která by jinak byla šifrována nebo na první pohled nezjistitelná [25] [52].

Metody Hash-To-Group a Hash-To-Curve umožňují bezpečné mapování hesel nebo jiných tajných dat na body v kryptografických skupinách, jako jsou eliptické křivky nebo multiplikativní skupiny. Tento proces je důležitý pro protokoly vyžadující silnou formu autentizace, jelikož převádí heslo na kryptografický klíč, přičemž je kritické minimalizovat riziko úniku informací o tomto hesle [25] [52].

Útoky založené na načasování (Timing Attacks) jsou umožněné díky proměnlivé době běhu algoritmů Hash-To-Group a Hash-To-Curve, která může odhalit informace o vstupních datech. Pokud algoritmus vyžaduje různý počet iterací pro různé vstupy, může delší doba běhu signalizovat útočníkovi, že první pokusy byly neúspěšné, což může poskytnout cenné informace o hesle. Toto riziko nastane v situacích, kdy implementace těchto algoritmů nejsou optimalizovány pro konstantní čas, což znamená, že doba běhu může variabilně záviset na konkrétních vstupních datech [25] [52].

Tato zranitelnost vychází z implementaci Hash-To-Curve algoritmu, kde proměnlivý počet iterací může odhalit informace, které mohou být využity v offline útoku

hrubou silou na heslo. Například, pokud algoritmus FreeRADIUS potřebuje více než 11 iterací, jeden z každých 2048 handshaku selže, což odhaluje, že heslový prvek nebyl nalezen v prvních 10 iteracích, a tím se poskytují cenné informace [25] [52].

Útoky založené na keš, známé také jako útoky postranním kanálem založené na keši, jsou specifické útoky využívající způsob, jakým procesory a operační systémy spravují keš paměť, k odhalení informací o prováděných operacích [25] [52].

Hlavním cílem útoků založené na keši je sledovat, jak dlouho trvá procesoru přístup k určitým paměťovým lokacím. Specificky se tyto útoky zaměřují na zjištění, zda byl test kvadratického zbytku (QR Test) v první iteraci Hash-To-Curve algoritmu úspěšný. Výsledek tohoto testu může útočníkovi prozradit informace o použitém hesle, což následně umožňuje provést offline útok na toto heslo. Útoky fungují na principu monitorování přístupových vzorů ke keši paměti. Moderní procesory uchovávají informace, ke kterým bylo nedávno přihlíženo, ve své keši, aby se urychlil proces opětovného získání těchto dat v budoucnu. Tato funkce může být využita k detekci, zda došlo k interakci s určitou částí paměti, na základě doby, která je potřebná k opětovnému načtení informací z této oblasti. Pokud je načtení rychlé, naznačuje to, že data byla uložena v keši díky nedávnému otevření. Pokud je proces pomalý, informace pravděpodobně nebyly v keši uloženy, což znamená, že k nim došlo bez nedávné aktivity [25] [52].

Degradační útok ve WPA3

Degradační útok umožňuje vynutit WPA3 k použití WPA2, což následně usnadňuje provedení slovníkových útoků. Tento typ útoku se týká primárně v režimech přechodu, kde jsou podporovány jak WPA3, tak WPA2, protože útočník může přesvědčit zařízení, že WPA3 není dostupné a donutit je tak k použití WPA2. Tím se otevírá možnost pro slovníkové útoky, které jsou proti WPA2 efektivnější díky nižší úrovni bezpečnosti [25] [52].

Jedním ze scénářů je vytvoření falešného přístupového bodu, který simuluje legitimní síť. Díky tomu útočník donutí zařízení k připojení k tomuto AP, který podporuje pouze WPA2, zachytí se handshake a následně se pokusí o jeho prolomení k odhalení hesla. [25] [52].

Z hlediska technické realizace degradačním útokům v kontextu Dragonfly handshake, který je používán ve WPA3 a EAP-pwd, lze vynutit použití slabší bezpečnostní skupiny během vyjednávání. Tím se snižuje celková bezpečnost komunikace. Při útoku na režim přechodu mezi WPA3 a WPA2 se můžou modifikovat signály tak, aby se zařízení přesvědčily, že pouze WPA2 je dostupné, což umožňuje zachytit dostatečné množství dat pro provedení slovníkového útoku [25] [52].

Ačkoliv WPA3 poskytuje lepší ochranu než WPA2, včetně opatření proti útokům

hrubou silou a vylepšené šifrování, existence režimu přechodu a možnost degradačního útoků otevírá cestu k potenciálnímu zneužití [25] [52].

2.4.10 Odepření služby ve WPA3

Útoky DoS (Denial of Service) proti WPA3 využívají specifické zranitelnosti, které se týkají vysoké výpočetní složitosti, resp. útokům postranním kanálem. Tyto obranné mechanismy, ačkoli jsou nezbytné pro zabezpečení, mají nežádoucí vedlejší efekt v podobě vysokého výpočetního zatížení, což může být zneužito pro realizaci DoS útoků [25] [52].

Mechanismus, který umožňuje DoS útoky, je tzv. anti-clogging mechanismus implementovaný v SAE. Tento mechanismus byl navržen k prevenci DoS útoků, které zaplavují oběť velkým množstvím commit rámců s padělanými MAC adresami. Na rozdíl od IP adres je padělání MAC adres triviální a v jakémkoli vysílacím prostředí, jako je Wi-Fi, se může snadno zachytit a znovu vyslat tajné cookies, který AP poslal klientovi a který klient musí odeslat zpět předtím, než AP zpracuje commit rámec od klienta. Proces autentizace vyžaduje významný výpočetní výkon jak ze strany klienta, tak ze strany přístupového bodu. Pokud útočník vygeneruje velké množství autentizačních požadavků, může to zapříčinit, že AP bude věnovat většinu svých výpočetních zdrojů na zpracování těchto požadavků, čímž se stává nedostupným pro legitimní uživatele. Tato vysoká výpočetní složitost se stává zranitelným bodem, který může být zneužit pro DoS útoky [25] [52].

V praxi se může iniciovat DoS útok na AP posíláním velkého množství autentizačních požadavků, které buď nemají být dokončeny nebo jsou specificky navrženy tak, aby byly výpočetně náročné. Tím se zatíží výpočetní zdroje AP, a sníží se jeho schopnost obsluhovat ostatní, legitimní uživatele. V důsledku toho může docházet k výpadkům služeb nebo výraznému snížení výkonu sítě [25] [52].

3 Internetový protokol verze 4

IPv4 je čtvrtá verze internetového protokolu. Jde o jeden z hlavních protokolů používaných v síti pro směrování internetového provozu, a to i přesto, že byl v posledních letech částečně nahrazen novější verzí IPv6 (Internet Protocol Version 6), kvůli předpokládanému vyčerpání adresního prostoru IPv4.

3.1 Základní přehled IPv4

Tato část se věnuje základnímu přehledu o IPv4, zahrnující adresní prostor, překlad síťových adres, beztřídní směrování, podsítě, TCP a UDP a v závěru síťové protokoly a služby.

3.1.1 Adresní prostor

Adresní prostor IPv4 je soubor unikátních adres používaných pro identifikaci zařízení připojených k síti. Každá adresa IPv4 se skládá z 32bitové čísla, což teoreticky umožňuje existenci přibližně 4,3 miliardy unikátních adres. V praxi je dostupných adres méně kvůli rezervaci pro speciální účely a efektivitě přidělování adres. Adresy jsou obvykle zapisovány jako čtyři desítková čísla oddělená tečkami. Každé z těchto čísel může nabývat hodnot od 0 do 255 [72].

Adresní prostor pro IPv4 se dělí do několika tříd adres, od A do E, na základě prvních několika bitů adresy. Toto rozdělení určuje, jak velká část adresy reprezentuje síť a jak velká část hostitele v rámci této sítě:

- Třída A: První bit je nastaven na 0, což umožňuje 128 možných sítí (1. bit je 0) a přibližně 16 milionů hostů na síť.
- Třída B: První dva bity jsou nastaveny na 10, což umožňuje 16 384 sítí (2 bity pro síť) s 65 534 hosty na síť.
- Třída C: První tři bity jsou nastaveny na 110, což umožňuje 2 097 152 sítí s 254 hosty na síť.
- Třída D: První čtyři bity jsou nastaveny na 1110, je vyhrazena pro multicastové adresování.
- Třída E: Je vyhrazena pro budoucí použití, experimenty a výzkum a není určena pro veřejné sítě [72].

3.1.2 Překlad síťových adres

NAT (Network Address Translation) pro IPv4 je technika, která umožňuje změnu IP adresy během jejího přenosu přes směrovač. Tím se zlepšuje bezpečnost a snižuje potřeba veřejných IP adres pro organizaci. NAT funguje tak, že mezi vnitřní

a vnější síť se zařadí brána s několika externě platnými IP adresami. Vnitřní zařízení s neveřejnými IP adresami tak mohou komunikovat s externím světem, aniž by byla jejich skutečná adresa odhalena. Tento proces zahrnuje překlad odchozího i příchozího provozu, čímž se zvyšuje bezpečnost sítě [73].

Existují různé typy NAT:

- Statický NAT: Přiřazuje jednu veřejnou IP adresu k jedné vnitřní.
- Dynamický NAT: Využívá fond veřejných IP adres pro více vnitřních zařízení.
- NAT s přetížením (PAT): Umožňuje mnoha vnitřním zařízením sdílet jednu veřejnou IP adresu pomocí různých portů [73].

3.1.3 Beztřídní směrování

CIDR (Classless Inter-Domain Routing) je metoda přidělování IP adres, která zefektivňuje distribuci adres tím, že nahrazuje systém založený na třídách A, B a C. Cílem CIDR bylo zpomalit nárůst směrovacích tabulek a snížit vyčerpávání adres IPv4. Tato metoda je založena na VLSM (Variable-Length Subnet Mask), umožňující efektivně dělit adresní prostor na podsítě různých velikostí. CIDR adresy obsahují předponu, která je binární reprezentací síťové adresy a přípony, který udává počet bitů v adrese. CIDR bloky sdílejí stejnou předponu a umožňují tvorbu větších či menších skupin adres. CIDR je nyní standardem pro směrování na internetu, podporovaným protokoly BGP (Border Gateway Protocol) a OSPF (Open Shortest Path First), a používá se k efektivnímu rozdělování adresního prostoru a optimalizaci síťového provozu [74].

3.1.4 Podsítě

Podsítě jsou segmenty větších sítí, které zlepšují správu a efektivitu sítě tím, že rozdělují síťový provoz do menších, lépe spravovatelných částí. Tento proces je základem pro minimalizaci zbytečného provozu a zvýšení rychlosti sítě. Každá IP adresa se skládá z prefixu sítě a ID hostitele, přičemž podsítě využívají podsíťovou masku pro identifikaci a oddělení různých segmentů sítě. Hlavní využití podsítí zahrnuje efektivnější rozdělování IP adres, snižování síťového provozu mezi specifickými skupinami zařízení a zlepšení bezpečnosti sítě [75].

3.1.5 TCP a UDP

TCP (Transmission Control Protocol) je protokol, který slouží pro výměnu dat mezi aplikacemi na internetu, zajišťuje spolehlivé a uspořádané doručení dat. TCP a IP tvoří základ internetové komunikace. TCP využívá spojovanou komunikaci, rozdělující data na pakety, spravující jejich přenos a zajišťuje bezchybné doručení pomocí

potvrzení příjmu a retransmise ztracených paketů. Na rozdíl od UDP, který je rychlejší ale méně spolehlivý, TCP se používá tam, kde je důležitá integrita dat, jako je prohlížení webových stránek, psaní e-mailů a přenos souborů [76] [77].

UDP (User Datagram Protocol) je protokol pro nízkolatenční a odolné spojení mezi aplikacemi na internetu, optimalizovaný pro situace, kde je důležitější rychlost než absolutní spolehlivost doručení. Na rozdíl od TCP, UDP nepotřebuje potvrzení příjmu dat, což umožňuje rychlejší přenos, ideální pro VoIP (Voice over Internet Protocol), DNS (Domain Name System) vyhledávání a vysílání médií. UDP umožňuje komunikaci mezi procesy bez zajištění doručení, pořadí paketů nebo opravy chyb, což je využíváno v aplikacích tolerantních ke ztrátě dat, jako jsou online hry, video konference a živé vysílání. Své místo má i v aplikacích vyžadujících rychlý přenos velkých objemů dat, kde aplikace sama zajišťuje integritu dat [76] [77].

3.1.6 Síťové protokoly a služby

Síťové protokoly jsou soubory pravidel, které určují, jak formátovat, odesílat a přijímat data, aby různá síťová zařízení mohla komunikovat navzdory rozdílům ve své infrastruktuře, designu nebo standardech. Protokoly jsou nezbytné pro fungování internetu a dalších sítí, protože bez nich by zařízení nevěděla, jak spolu vzájemně komunikovat. Síťové protokoly se uplatňují na různých úrovních modelu OSI (Open Systems Interconnection), který zahrnuje sedm vrstev od fyzické po aplikační. Každá vrstva modelu OSI má specifické úkoly, od přenosu dat po správu aplikací [78].

Důležité síťové služby a protokoly zahrnují TCP/IP pro internetovou konektivitu, HTTP (Hypertext Transfer Protocol) pro přenos hypertextu, SMTP (Simple Mail Transfer Protocol) pro odesílání e-mailů, FTP pro přenos souborů a HTTPS (Hypertext Transfer Protocol Secure) pro zabezpečenou komunikaci. Tyto protokoly umožňují spolehlivou a bezpečnou výměnu informací mezi zařízeními na síti. Navíc, protokoly jako SNMP (Simple Network Management Protocol) a ICMP (Internet Control Message Protocol) pomáhají ve správě a monitorování sítí, zatímco bezpečnostní protokoly jako SSL (Secure Sockets Layer) a TLS (Transport Layer Security) chrání data přenášená přes síť [78].

3.2 Nástroje pro skenování IPv4

Mezi široce používané nástroje pro skenování IPv4 patří Nmap, Masscan a Naabu. Každý z těchto nástrojů má své specifické vlastnosti a použití. Skenování IP adres jsou zásadní pro identifikaci aktivních zařízení na síti, rozpoznání otevřených portů a služeb běžících na těchto portech.

3.2.1 Nmap

Nmap (Network Mapper) je bezplatný nástroj pro objevování sítí a auditu bezpečnosti. Hodí se pro úkoly jako inventarizace sítě, správa aktualizací služeb a monitoring dostupnosti služeb. Nmap efektivně zjišťuje dostupné hostitele, jejich služby, operační systémy a typy firewallů. Optimalizuje skenování jak rozsáhlých sítí, tak jednotlivých hostitelů. Podporuje všechny hlavní operační systémy a poskytuje jak tradiční verzi pro příkazovou řádku, tak grafické rozhraní Zenmap [79].

3.2.2 Masscan

Masscan slouží ke skenování portů na úrovni internetu, který dokáže proskenovat celý internet za méně než 5 minut pomocí vysílání 10 milionů paketů za sekundu z jednoho stroje. Jeho použití a výstupy jsou podobné Nmapu. Masscan je optimalizován pro rychlé skenování velkého počtu strojů. Interně využívá asynchronní přenos. Nabízí flexibilitu v nastavení libovolných rozsahů portů a adres [80].

3.2.3 Naabu

Nástroj Naabu, který je vytvořen v programovacím jazyku Go, je navržen pro skenování a identifikaci otevřených portů na jednom nebo více hostitelích. Klíčovou funkcí je použití tzv. SYN/CONNECT probe, který zajišťuje rychlost a efektivitu při skenování. Nástroj je optimalizován pro jednoduché použití a minimální zatížení systémových zdrojů. Mezi hlavní výhody Naabu patří automatická manipulace s duplicitními hostiteli napříč různými subdoménami, což eliminuje potřebu ručního filtrování. Integrace s Nmap pro objevování služeb rozšiřuje jeho funkčnost, umožňuje detailnější analýzu skenovaných portů a identifikaci běžících služeb [81].

4 Vlastní návrh metodologie a nástrojů pro penetrační testování

Tato kapitola popisuje výběr nastávajících nástrojů, zmíněných v předchozích kapitolách 2.3 a 3.2, se využijí pro vlastní implementaci podpůrných nástrojů pro bezpečnostní penetrační testování bezdrátových sítí Wi-Fi a síťové infrastruktury IPv4. Definuje se vlastní návrh diagramu pro Wi-Fi sítě a IPv4 inspirovaný fázemi pro IPv4 1.3 a vytvoření jednotlivých fázích pro Wi-Fi sítě. Dále se vysvětlí, proč byl zvolen Python jako programovací jazyk a představí se struktura výstupu jednotlivých nástrojů, jelikož podpůrné nástroje budou integrovány do platformy Penterep. Samotná integrace nástrojů přímo do platformy Penterep není cílem této diplomové práce a proto implementace zůstane mimo rozsah.

4.1 Programovací jazyk Python

Python nabízí rozsáhlou knihovnu modulů a balíčků, které usnadňují tvorbu skriptů pro různé účely. Hlavní knihovnou pro podpůrné nástroje je `ptlibs` od platformy Penterep, která umožňuje formátovat JSON (JavaScript Object Notation) do vhodné podoby a dále jí zpracovat v AS (Autonomous System). K dalším důležitým knihovnám patří `scapy`, která umožňuje interaktivní manipulaci s pakety, `argparse` pro parsování argumentů příkazového řádku a `subprocess` pro spouštění systémových příkazů přímo z Pythonu.

Další výhody programovacího jazyka jsou práce s různými formáty dat, síťovou komunikaci, webové stránky a mnoho dalšího. Jedná se také o křížovou platformu, díky tomu lze psát skripty na jednom operačním systému a spouštět je na jiném, pokud je nainstalován Python interpretr. Python umožňuje snadno integrovat jiné jazyky, což umožňuje využít specializované knihovny nebo nástroje napsané v jiných jazycích. V neposlední řadě byl Python vytvořen kvůli čitelnosti a jednoduchosti.

Z těchto důvodů byl Python zvolen, v této práci, jako ideální programovací jazyk pro vytvoření vlastních nástrojů, které spojují efektivitu, přenositelnost a jednoduchost psaní kódu, což usnadňuje tvorbu a údržbu skriptů pro širokou škálu účelů.

4.2 Výstup jednotlivých nástrojů

Nástroje nabídnou dva možné formáty výstupu: standardní a JSON, který je definovaný pomocí přepínače `-j`. Zatímco standardní formát slouží pro běžné použití,

aby byl jednoduchý a přehledný v konzole, výstup ve formátu JSON poskytne komplexnější detaily, které bude využívat platforma Penterep. Klasický výstup lze vidět ve výpisu 4.1.

Výpis 4.1: Příklad klasického výstupu pro skenování dostupných Wi-Fi sití.

```
[*] AP Info:
[i] name: name
[i] netType: wifi
[i] bssid: bssid
[i] wifiChannel: channel
[i] wifiEncryption:encryption
[i] wifiCipher: cipher
[i] wifiAuth: auth
[i] wifiEssid: essid
```

V JSON formátu údaj `status` signalizuje AS, že program úspěšně dokončil svůj běh a je připraven k dalšímu zpracování. Tato hodnota se nastaví po ukončení nástroje na `finished`. Objekt `results` pak obsahuje další objekt `nodes`, které jsou zastoupeny v poli objektů. Tyto objekty poskytují podrobné informace o jednotlivých uzlech. Každý uzel se bude identifikovat pro Wi-Fi sítě jako `type net` a mít unikátní klíč `key`, který se generuje náhodně. Tyto uzly nebudou na nadřazených prvcích. Uzly budou mít, pro případ skenování dostupných Wi-Fi sití, definované `properties` specifické pro Wi-Fi sítě, tj. názvu `essid`, typu sítě `wifi`, identifikátoru sítě `bssid`, kanálu `channel`, šifrování `encryption`, šifrovacího algoritmu `cipher`, metody autentizace `auth` a názvu sítě `essid`. Navíc uzly budou ztotožňovat zranitelnosti, pokud budou nalezeny, označené kódem `PTV-XXX-XXXX`, což upozorňuje na možná bezpečnostní rizika spojená s tímto uzlem. Tyto informace jsou klíčové pro správu a bezpečnost sítě, protože umožňují detailní přehled o konfiguraci a potenciálních slabých místech. Příklad výstupu pro JSON formát lze vidět ve výpisu 4.2

4.3 Vlastní návrh pro Wi-Fi sítě

Vlastní návrh pro Wi-Fi sítě bude navazovat na rešerši týkající se Wi-Fi sítí k vytvoření vlastního diagramu, fází a podpůrných nástrojů.

4.3.1 Využití existujících nástrojů pro vlastní podpůrné nástroje

V tabulce 4.1 je uveden stručný přehled nástrojů a funkcionalit, které byly testovány a ověřeny. Červená barva s křížkem značí nedostatky jednotlivých nástrojů naopak zelená barva s fajfkou představuje výhody, kterými nástroj disponuje.

Výpis 4.2: Příklad JSON formátu pro skenování dostupných Wi-Fi sití.

```
{
    "satid": "",
    "guid": "",
    "status": "finished",
    "message": "",
    "results": {
        "nodes": [
            {
                "type": "net",
                "key": "randomly-generated-string",
                "parent": null,
                "parentType": null,
                "properties": {
                    "name": "bssid",
                    "netType": "wifi",
                    "bssid": "bssid",
                    "wifiChannel": "channel",
                    "wifiEncryption": "encryption",
                    "wifiCipher": "cipher",
                    "wifiAuth": "auth",
                    "wifiEssid": "essid"
                },
                "vulnerabilities": []
            }
        ],
        "properties": {},
        "vulnerabilities": []
    }
}
```

Z tabulky vyplývá, že žádný z těchto nástrojů neumožňuje automatickému ukončení rušivých procesů a formátování v požadovaném formátu. Některé nástroje jsou obtížně srozumitelné nebo jim chybí uživatelská přívětivost. To může mít za následky jako, složitost používaných nástrojů nebo nemusí být otestovaná určitá zranitelnost, která by měla být otestována. Avšak každý z těchto nástrojů plní hlavní cíl a budou sloužit jako wrapper pro vlastní nástroje.

Pro vlastní návrh nástroje se bude využívat sada nástrojů Aircrack-ng. Poskytuje širokou škálu funkcí pro monitorování, testování a lámání hesel. Díky tomu je umožněno provézt komplexní testy penetračního testování Wi-Fi sítí. Podporuje různé typy šifrování, včetně WEP, WPA a WPA2, které jsou nezbytné pro testování bezdrátových sítí. Sada Aircrack-ng poskytuje vysokou úroveň přizpůsobení a skriptování, což je užitečné pro vytváření vlastních nástrojů a automatizaci procesů. Tato sada nástrojů je také základem pro nástroje jako je Wifite2 a Airgeddon, které au-

Funkcionality	Sada nástrojů Aircrack-ng	Wifite2	Airgeddon
Promiskuitní režim	✓	✓	✓
Automatické ukončení rušivých procesů	×	×	×
Zachytávání paketů	✓	✓	✓
Generování síťového provozu	✓	✓	✓
Lámání hesel	✓	✓	✓
Lámání hesla pomocí GPU	×	✓	✓
Jednoduchost	×	✓	×
Automatizace	×	✓	✓
Uživatelsky přívětivé	×	✓	×
Přenositelnost	✓	✓	✓
Integrace nástrojů	×	✓	✓
Optimalizované ukládání dat na disk	×	×	×
Formátování	×	×	×

Tab. 4.1: Přehled dosavadních nástrojů pro Wi-Fi sítě.

tomatizují procesy penetračního testování Wi-Fi sítí.

Další nástroje, které se využijí, budou Reaver a Wash, díky jejich skvělé integrace s WPS. Wash také poskytuje cenné informace ohledně obranných mechanismech směrovačů. Reaver dokáže jednoduše prolomit WPS, pomocí různých druhů útoků.

Pro rychlejší lámání hesel se implementuje nástroj Hashcat a Hcxpcapngtool. Hcxpcapngtool bude sloužit ke konverzi formátu pro Hashcat, který dokáže rychleji a efektivněji lámat hesla.

Wireshark, jakožto analyzátor síťových protokolů, bude sloužit pro odposlech přenášených dat přes Wi-Fi sítě.

4.3.2 Fáze penetračního testování

Vlastní fáze penetračního testování by zahrnovaly sedm jednotlivých fází, které se postupně zaměřují na bezpečnost Wi-Fi sítí. Každá fáze má za cíl identifikovat a vyhodnotit potenciální bezpečnostní rizika a v závěru navrhnout efektivní opatření pro jejich řešení. Jednalo by se o:

- plánování
- průzkum dostupných Wi-Fi sítí
- identifikaci sítě
- průzkum dostupných zranitelností
- zneužití Wi-Fi sítí
- vytvoření zprávy
- nápravu a opětovné přetestování

z čehož vyplývá, že některé fáze by byly obdobné, jak byly popsány v kapitole 1.3.

Plánování

Plánování by se řídilo podle popisu v kapitole 1.3. Obsahem by byla diskuse o rozsahu testování a právních aspektech s cílem jasně definovat, které oblasti a aktiva budou testovány.

Průzkum dostupných Wi-Fi sítí

Tato fáze by se zaměřila na sběr informací dostupných Wi-Fi sítí, které jsou dostupné v blízkosti testované organizace. To zahrnuje identifikaci všech Wi-Fi sítí, které jsou vlastněné nebo používané organizací, stejně jako další sítě, ke kterým by se mohli připojovat její zaměstnanci.

Identifikace sítě

Zde by se zahrnula práce se seznamem konkrétních Wi-Fi sítí, aby bylo možné identifikovat a začít produkovat konkrétní data o každé z nich. Pro každou výše označenou síť by se začaly vytvářet individuální profily. Sítě, které by byly vybrány jako relevantní, by se zpracovala a zanalyzovala se specifickými informacemi, jako jsou bezpečnostní nastavení, síla signálu, typ šifrování a další technické parametry. Tato data umožní lépe pochopit možné zranitelnosti a připravit se na další fáze testování.

Průzkum dostupných zranitelností

Po identifikaci konkrétních Wi-Fi sítí by se provedla detailní analýza, zaměřená na odhalení zranitelností, které by mohly být potenciálně zneužity. Proces zahrnuje zachytávání datových paketů a nástroje pro lámání hesel. Cílem by bylo identifikovat zranitelná místa, jako jsou nedostatečně silná hesla nebo zastaralé šifrovací protokoly, které mohou představovat riziko pro organizaci.

Zneužití Wi-Fi sítí

Fáze zneužití Wi-Fi sítí by obsahovala realizaci útoků na dříve identifikované zranitelnosti s cílem získat kontrolu nad síťovými prvky organizace. Tento krok by byl vstupním bodem pro průnik do systému. Útoky mohou zahrnovat zneužití zranitelností na softwarové chyby, použití prolomeného hesla a další techniky, které demonstrují možné bezpečnostní hrozby. Tato fáze by ukázala, jak se reálně dá zasáhnout do provozu, odposlouchávat komunikaci nebo získat přístup k citlivým systémovým zdrojům.

Podání zprávy o výsledcích zneužití

Podání zprávy o výsledcích zneužití by probíhala podobně jako je popsáno v kapitole 1.3. V této fázi se zanalyzují všechna data získaná během předchozích kroků testování. Informace by byly kategorizovány na základě jejich závažnosti. Výsledná zpráva, která obsahuje detailní popis každé identifikované zranitelnosti a doporučení pro její řešení, se následně předá klientovi.

Náprava a opětovné přetestování

Fáze nápravy a opětovného přetestování by probíhala podle zájmu klienta pro potvrzení, že všechny zranitelnosti byly řádně opraveny.

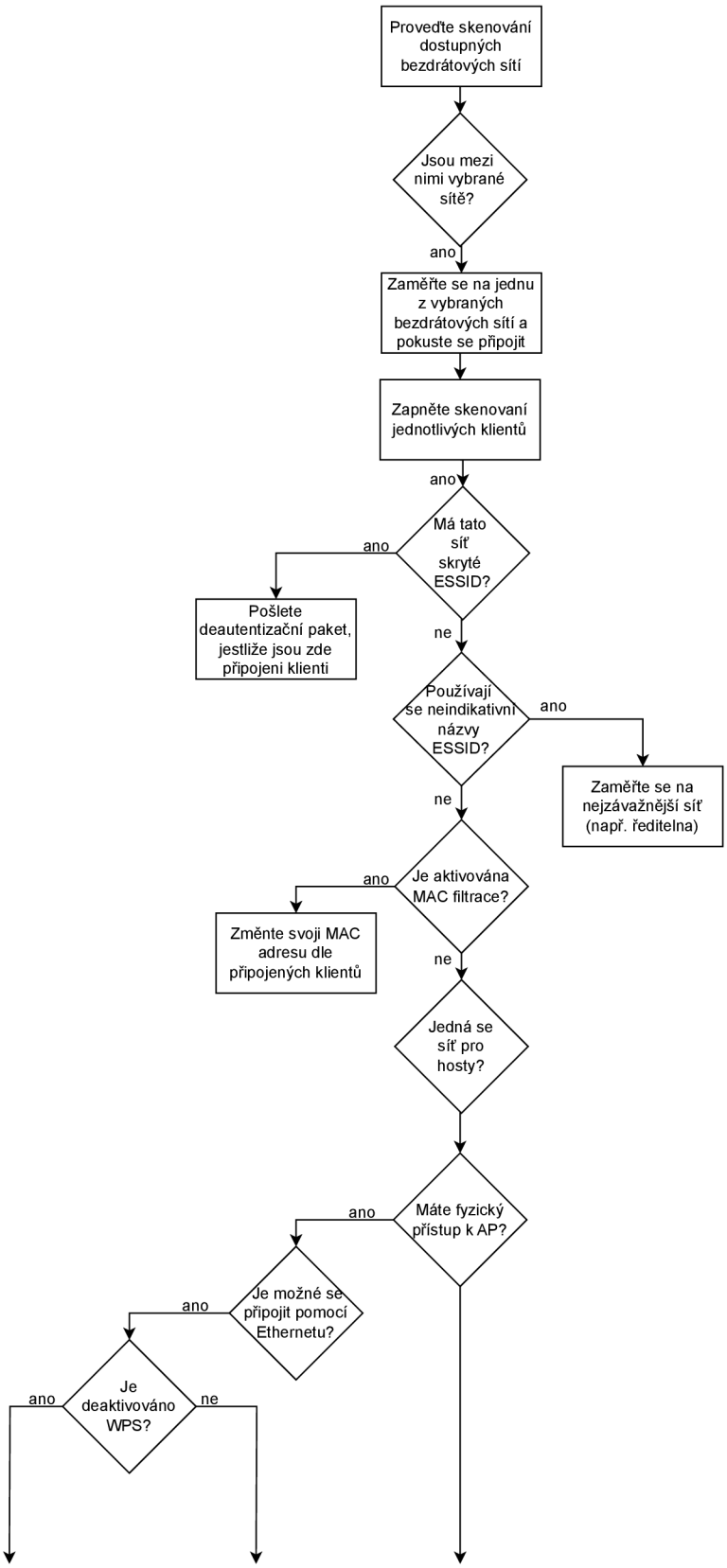
4.3.3 Návrh diagramu

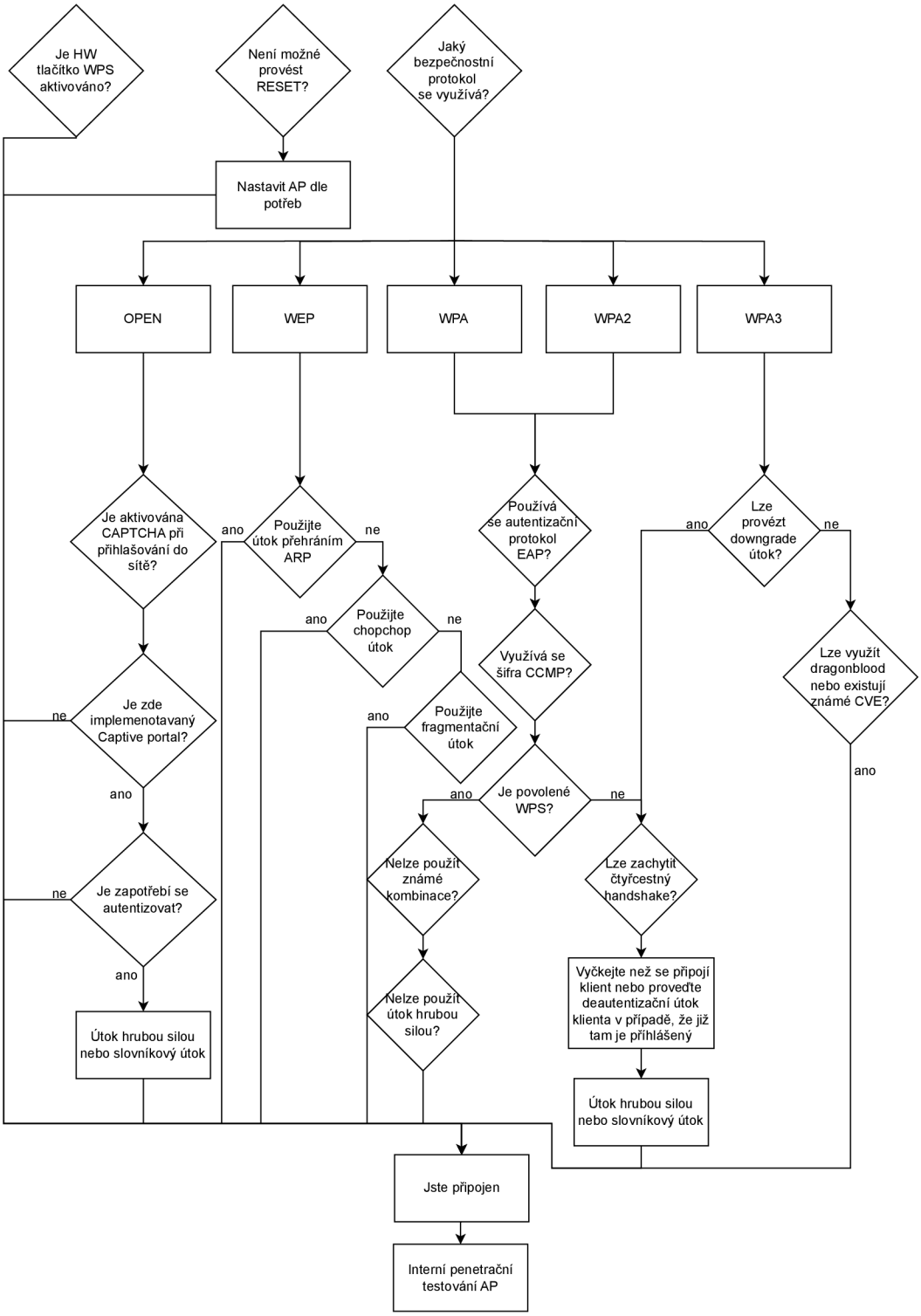
Cílem vlastního návrhu diagramu je navrhnout metodologii, která bude moci být postupně rozšířena a bude sloužit jako průvodce pro penetrační testery při testování Wi-Fi sítí. Návrh diagramu pro Wi-Fi sítě lze vidět na obrázku 4.1.

Diagram začíná fází průzkumu dostupných Wi-Fi sítí, který zahrnuje skenování prostředí za účelem identifikace všech dostupných bezdrátových sítí. Tento krok je zásadní, neboť poskytuje základní přehled o tom, jaké sítě jsou v dosahu a které z nich budou dále zkoumány. Při skenování jsou získávány základní informace o každé síti, například jaký typ zabezpečení používají, jaké mají bezpečnostní protokoly, na jakých kanálech fungují, jaká je jejich BSSID a ESSID. Po identifikaci dostupných sítí a základních informací je zaměřena pozornost na ty sítě, které jsou v dohodnutém rozsahu z fáze plánování.

Následně bude probíhat fáze s identifikací sítí, kde je prováděno podrobnější zkoumání klientů připojených k této síti. V tomto kontextu je podstatné získat základní informace o klientech, jako jsou například MAC adresy klientů, síla signálu, jejich připojení k přístupovým bodům a čas, kdy se klient připojil.

Další fáze se zaměřuje na zkoumání zranitelností různých aspektů Wi-Fi sítě. Zjišťuje se, zda některé sítě využívají skryté ESSID, což je technika, při které síť nevysílá své jméno s cílem, aby nebyla snadno rozpoznatelná. Při detekci sítě se skrytým ESSID může být nezbytné vyslat deautentizační rámec s účelem odpojit již připojené klienty, což povede k zobrazení ESSID.


Proveďte skenování dostupných bezdrátových sítí
Jsou mezi nimi vybrané sítě?
ano
Zaměřte se na jednu z vybraných bezdrátových sítí a pokuste se připojit
Zapněte skenovaní jednotlivých klientů
ano
Má tato síť skryté ESSID?
ano
Pošlete deautentizační paket, jestliže jsou zde připojeni klienti
ne
Používají se neindikativní názvy ESSID?
ano
Zaměřte se na nejzávažnější síť (např. ředitelna)
ne
Je aktivována MAC filtrace?
ano
Změnte svoji MAC adresu dle připojených klientů
ne
Jedná se síť pro hosty?
Máte fyzický přístup k AP?
ano
Je možné se připojit pomocí Ethernetu?
ano
Je deaktivováno WPS?
ano
ne



Obr. 4.1: Vlastní návrh diagramu penetračního testování pro Wi-fi sítě.

Pokud síť nevyužívá skryté ESSID nebo se již zjistil název sítě, zaměřuje se pozornost na sítě s indikativním názvem. Přednostně jsou zkoumány sítě, jejichž názvy nesou významný charakter, například sítě patřící vysokým představitelům nebo důležitým organizačním jednotkám.

V průběhu testování se kontroluje, zda síť využívá MAC filtraci, což je způsob, jak omezit přístup na základě fyzických adres zařízení. V případě, že je zjištěno využívání MAC filtrace, je možné toto bezpečnostní opatření obejít změnou vlastní MAC adresy na adresu, které již byl přístup povolen.

Dále je zkoumáno, zda existuje samostatná síť pro hosty a zda existuje fyzický přístup k jakémukoliv přístupovému bodu. To může umožnit připojení přes síťový kabel nebo využití hardwarového tlačítka WPS, pokud je aktivováno, díky čemuž by mohl být získán přístupu k síti bez potřeby znalosti hesla. V neposlední řade by mohlo dojít k resetování AP, což by vedlo k jeho přenastavení podle aktuálních potřeb.

Pozornost se následně zaměřuje na charakteristiky bezpečnostních protokolů implementovaných v sítích jako jsou WEP, WPA, WPA2, WPA3 a na situace, kdy síť heslo nevyžaduje, což je označováno jako OPEN a znamená to, že síť je otevřená, tudíž nevyužívá žádný bezpečnostní protokol. Každý z těchto protokolů vyžaduje specifický přístup k překonání bezpečnostních opatření.

V situaci, že Wi-Fi síť je nastavena jako OPEN, je důležité zjistit, zda je implementována CAPTCHA (Completely Automated Public Turing Test To Tell Computers And Humans Apart) a zda je součástí procesu i tzv. Captive portal. CAPTCHA se používá k ověření, že interakce probíhá s člověkem a ne s automatizovaným systémem. To zajišťuje, že přístup k internetu přes danou síť je chráněn před automatizovaným zneužitím. Dále je důležité zjistit, zda je součástí procesu Captive portal, který může sloužit pro autentizaci uživatelů nebo pouze vyžadovat souhlas s obchodními podmínkami.

V případě sítí zabezpečených protokolem WEP je zkoumána efektivita útoků, například útok přehráním ARP, KoreK chopchop útok anebo útok fragmentováním. Při těchto metodách je zabezpečení WEP prolomeno a umožněn přístup k síti za pomocí získaného klíče. Tyto techniky společně představují efektivní prostředky pro přístup do sítě WEP, což zahrnuje fázi zneužití bezdrátových sítí.

Pro sítě zabezpečené pomocí WPA a WPA2 se ověřuje, zda je pro autentizaci využíván EAP nebo PSK. Následně se posuzuje, jestli pro šifrování slouží TKIP nebo CCMP. Poté je ověřována přítomnost funkce WPS. Pokud je aktivní, jsou testovány známé kombinace. V situaci, kdy tyto pokusy neuspějí, přichází na řadu útok hrubou silou. Pokud je funkce WPS neaktivní je nutné odchytit 4cestný handshake, což lze dosáhnout čekáním na připojení nebo odpojení klienta prostřednictvím deautentizačního útoku. S odchycením handshaku začíná proces lámání hesla.

U zabezpečené sítě využívající WPA3 se zkoumá možnost provádění degradačního útoku, díky kterému by byly umožněny snadnější útoky hrubou silou nebo slovníkové útoky, tedy čekalo by se obdobně na handshake jak u WPA nebo WPA2. Kromě toho se zanalyzují další možné zranitelnosti známé jako Dragonblood, jež by mohla kompromitovat bezpečnostní prvky WPA3, stejně jako přítomnost jakýchkoli známých CVE (Common Vulnerabilities and Exposures), které by mohly otevřít cestu k útokům.

Návrh diagramu končí v okamžiku, kdy se podaří prolomit heslo nebo dostat přístup k síti prostřednictvím fyzického přístupu k zařízení. V tomto bodě dochází k připojení do interní sítě, což otevírá cestu k další fázi a tj. internímu penetračnímu testování AP. Tato etapa je zaměřena na konfiguraci AP, bezpečnosti interní sítě, identifikaci potenciálních zranitelných míst a ověření odolnosti proti různým typům útoků.

4.4 Vlastní návrh nástrojů pro IPv4

Vlastní návrh pro IPv4 bude navazovat obdobně jak v případě Wi-Fi sítích na rešerši týkající se IPv4 k vytvoření vlastního diagramu a podpůrných nástrojů.

4.4.1 Využití existujících nástrojů pro vlastní podpůrné nástroje

V tabulce 4.2 lze vidět stručný přehled nástrojů a funkcionalit, které byly testovány a ověřeny. Pravidla jsou stejná jak v případě tabulky 4.1, tj. červená barva s křížkem značí nedostatky naopak zelená barva s fajfkou představuje výhody.

Z tabulky vyplývá, že byly porovnány vlastnosti tří různých nástrojů: Nmap, Masscan, Naabu. Všechny tři nástroje podporují IPv4 i IPv6. Masscan vyniká v rychlosti skenování, zatímco Nmap a Naabu jsou v tomto ohledu méně efektivní. Pouze Nmap podporuje skriptování, díky čemu je možné detekovat verze služeb a operačního systému nebo získat tzv. banner. Všechny tři nástroje jsou použitelné přes CLI (Command Line Interface). Co se týče výstupu v požadovaném formátu, žádný z nástrojů v tabulce nevyhovuje očekáváním. Nmap i Naabu podporují překlad DNS, zatímco Masscan tuto funkci nemá.

Pro skenování, identifikaci síťových služeb, operačních systémů a jejich verzí se využije kombinace nástrojů Masscan a Nmap. Nástroje představují silnou strategii pro objevování a auditování síťových zdrojů. Masscan umožňuje skenovat celé rozsahy během několika minut, což je zvláště užitečné pro rychlé identifikování aktivních portů v rozsáhlých sítích. Jeho schopnost vysílat až 10 milionů paketů za sekundu z jednoho stroje ho činí ideálním pro první fázi bezpečnostního skenování, kdy je cílem rychle získat přehled o dostupných službách a otevřených portech.

Funkcionality	Nmap	Masscan	Naabu
Podpora adresního prostoru IPv4	✓	✓	✓
Podpora adresního prostoru IPv6	✓	✓	✓
Rychlost skenování dostupných hostů a portů	×	✓	×
Podpora skriptování	✓	×	×
Použitelnost v příkazové řádce	✓	✓	✓
Formátování	×	×	×
Překlad DNS	✓	×	✓

Tab. 4.2: Přehled dosavadních nástrojů pro skenování IPv4.

Na druhé straně Nmap nabízí podrobnější analýzu a pokročilé skenovací možnosti, včetně detekce operačních systémů, služeb a jejich verzí. Díky NSE (Nmap Scripting Engine), který umožňuje rozsáhlé možnosti skriptování, je Nmap vhodný pro cílené skenování jednotlivých hostitelů nebo služeb.

Použití Masscanu k rychlému zjištění, které systémy mají otevřené FTP porty a následné využití Nmapu pro hlubší analýzu těchto systémů k zjištění, zda podporují anonymní přístup, je příkladem efektivního využití obou nástrojů.

4.4.2 Návrh diagramu

Cílem vlastního návrhu diagramu je navrhnout metodologii, která bude moci být postupně rozšířena a bude sloužit jako průvodce pro penetrační testery při testování IPv4. Vlastní diagram pro IPv4 vychází z fází, které byly popsaný v kapitole 1.3, a je vyobrazený na obrázku 4.2.

Návrh vývojového diagramu pro IPv4 začíná skenováním dostupných hostů v síti, kde prvním krokem je identifikace aktivních hostů. Pro zjištění, zda jsou hosté aktivní, lze využít ICMP pakety. Nejprve se pošle zpráva s požadavkem, tzv. echo request, který umožňuje zjistit, zda je host živý. Pokud je host živý odpoví zprávou, tzv. echo reply. ICMP pakety jsou efektivní jak pro interní, tak pro externí sítě. V lokálních sítích lze použít ARP dotazy k zjištění aktivních IP adres, což je vhodné zejména v prostředí, kde je blokováno směrování ICMP. Efektivnější metodou je skenování otevřených portů, která začíná skenováním dvaceti nejběžnějších portů, pokračuje rozšířením na 100, 1000 portů a zakončí se to úplným skenováním všech portů. Tato technika sdělí všechny aktivní porty, které indikují běžící služby. Pokud skenování portů vrátí pozitivní výsledek, potvrzuje to, že host je živý a může být zařazen do další fáze penetračního testování. V případě negativního výsledku skenování hosta končí, což signalizuje, že daný host buď není přítomen, nebo má

všechny porty uzavřené. Pro skenování dostupných hostů lze využít nástroje jako jsou Nmap a Masscan.

V případě, že se obě strany domluvily v průběhu plánování, o zřízení uživatelského účtu pro penetrační testování, začnou se také kontrolovat vztahy v prostředí tzv. Active Directory. Active Directory slouží jako služba od Microsoftu pro správu sítí, která umožňuje správu uživatelských účtů a zdrojů, autentizaci a autorizaci v rámci Windows domén. Nejprve započne detekce doménových řadičů, která je klíčovým krokem v procesu zajištění síťové bezpečnosti, neboť tyto servery hrají zásadní roli ve správě uživatelských účtů a politik bezpečnosti uvnitř Windows domén. Tento proces skenování sítě je zaměřen na identifikaci serverů, které fungují jako centralizované autority pro autentizaci a autorizaci. Jelikož doménové řadiče ovládají přístupy uživatelů a nastavení bezpečnostních direktiv celé sítě, stávají se prioritními cíli penetračního testování. Nástroje jako Nslookup nebo Nmcli, při správném nastavení, umožňují efektivně lokalizovat doménové řadiče v síti.

Následně je možné použít nástroj BloodHoundAD a technika zvaná Kerberoasting, kde je možný využít nástroj Kerberoast. BloodHoundAD je analytický nástroj, který umožňuje vizualizaci vztahů a oprávnění v rámci Active Directory, čímž identifikuje potenciální slabá místa. Kerberoasting využívá slabost v implementaci služby Kerberos, což je standardní autentizační protokol v síťovém prostředí. Kerberoasting umožňuje útočníkům získat zašifrovaná hesla uživatelských účtů v síti, která poté mohou být dešifrována a zneužita k dalším útokům na síť. Pokud se získají přístupy do nových strojů je zapotřebí otestovat AD CS (Active Directory Certificate Services). K tomuto testování je určen nástroj Certipy. Tento nástroj může odhalit slabá místa v nastavení AD CS, a to bez ohledu na to, zda je zkoumaný účet administrátorský, běžný, nebo zda byl získán NTLM (NT LAN Manager) hash. Díky Certipy lze efektivně prověřit, jak je AD CS nakonfigurován a identifikovat potenciální rizika.

Po detekci živých hostů a získaného uživatelského účtu, jestliže byla již v průběhu plánování přiřazena, se provádí několik skenovacích a monitorovacích akcí paralelně, přičemž je klíčové, aby tyto operace nezatížily síť natolik, že by došlo k jejímu pádu. Proto je důležité přizpůsobit intenzitu skenování kapacitě a možnostem dané sítě. Tyto akce jsou, tzv. DNS Zone Transfer, skenování dostupných verzí a operačních systémů, skenování IPv6, skenování zranitelností a odposlechy komunikace na síti pomocí tzv. Listener.

Pokud se úspěšně provede DNS Zone Transfer, což je proces, při kterém se získává kopie databáze DNS serveru, může to pomoci identifikovat nově přítomné hosty v síti. Po získání těchto informací a pokud jsou mezi nimi objeveni noví hosté, se znovu spustí skenování otevřených portů na těchto hostech. Pro identifikaci DNS záznamů se může využít nástroj Nslookup nebo Dig, které slouží k získávání infor-

mací o DNS záznamech z konkrétního serveru.

Systémy mohou být konfigurovány tak, aby používaly výhradně IPv6, což znamená, že nemusí mít aktivní IPv4 adresy. V důsledku toho by tyto systémy nebyly detekovány během skenování, které jsou zaměřené pouze na IPv4. Nástroj Ptnetinspector od platformy Penterep je ideální pro adresování IPv6, neboť dokáže provádět skenování různými způsoby. Pasivně shromažďuje data bez zasahování do sítě, aktivně zkoumá síť pro získání detailních informací a agresivně, což zahrnuje pokročilé techniky pro analýzu síťových prvků. V případě, že budou opět objeveni hosti, proces se opakuje pro objevení otevřených portů.

Skenování dostupných verzí služeb a operačních systémů určuje specifické verze, které běží na síťových zařízeních. Identifikace verzí softwaru je důležitá, protože umožňuje identifikovat starší nebo zastaralé verze, které mohou obsahovat známé bezpečnostní zranitelnosti snadno zneužitelné útočníky. Úspěšné zjištění verze služby nebo operačního systému umožňuje lépe pochopit potenciální útoky. Nástroj jako Nmap dokáže skenovat verze daných systémů a služeb, pokud se použijí správné přepínače.

Skenování zranitelností identifikuje a analyzuje potenciálně zranitelné místa v systému. Tento proces prochází síťová zařízení a aplikace ve snaze odhalit zranitelnosti, od zastaralého softwaru po špatně nakonfigurované služby. Skenování se zaměřuje na všechny kritické komponenty systému, včetně operačních systémů, databázových serverů, webových aplikací a síťové infrastruktury. Pro efektivní skenování zranitelností lze využít nástroje jako Nessus nebo OpenVAS.

Odposlech komunikace pro zachycení NTLM hashe se využívá tzv. Listener, který funguje jako server, neboť naslouchá na síti za účelem zachytávání a analýzy vstupujících síťových požadavků. Zachycené hashe se uloží do souboru a použijí se pro prolomení hesla. Nástroj, který se může využít jako Listener, je Responder. Při aktivaci tohoto nástroje se stanice v síti může stát cílem pro MITM útoky, kde Responder odposlouchává.

Po dokončení skenování zranitelností a nebo nástrojem, který identifikuje verze a operační systémy, se ověří, zda byly odhaleny nějaké zranitelnosti nebo zastaralé služby. Pokud skenování odhalí nějakou zranitelnost, následuje fáze jejich potenciálního zneužití.

Když sken zranitelností odhalí potenciální zranitelnost, nejprve se zjistí, zda se ze zranitelnosti dají získat oprávnění. Pokud zranitelnost umožňuje získání administrátorských práv, následuje extrakce dat ze systémových komponent Windows, jako jsou LSASS (Local Security Authority Subsystem Service), SAM (Security Account Manager) a LSA (Local Security Authority). V případě nalezených hashů v SAM, se provádí pokus o prolomení těchto hashů k získání hesla v čitelné podobě. Poté se ověří, zda hesla v čitelné podobě patří doménovému administrátorovi nebo běž-

nému uživateli. Následně se provede výčet dostupných dat, služeb jako jsou NFS, SMB, FTP a databází. Vzápětí se zkouší slovníkový útoky na získané přihlašovací údaje. Pro slovníkové útoky je vhodný nástroj Hashcat.

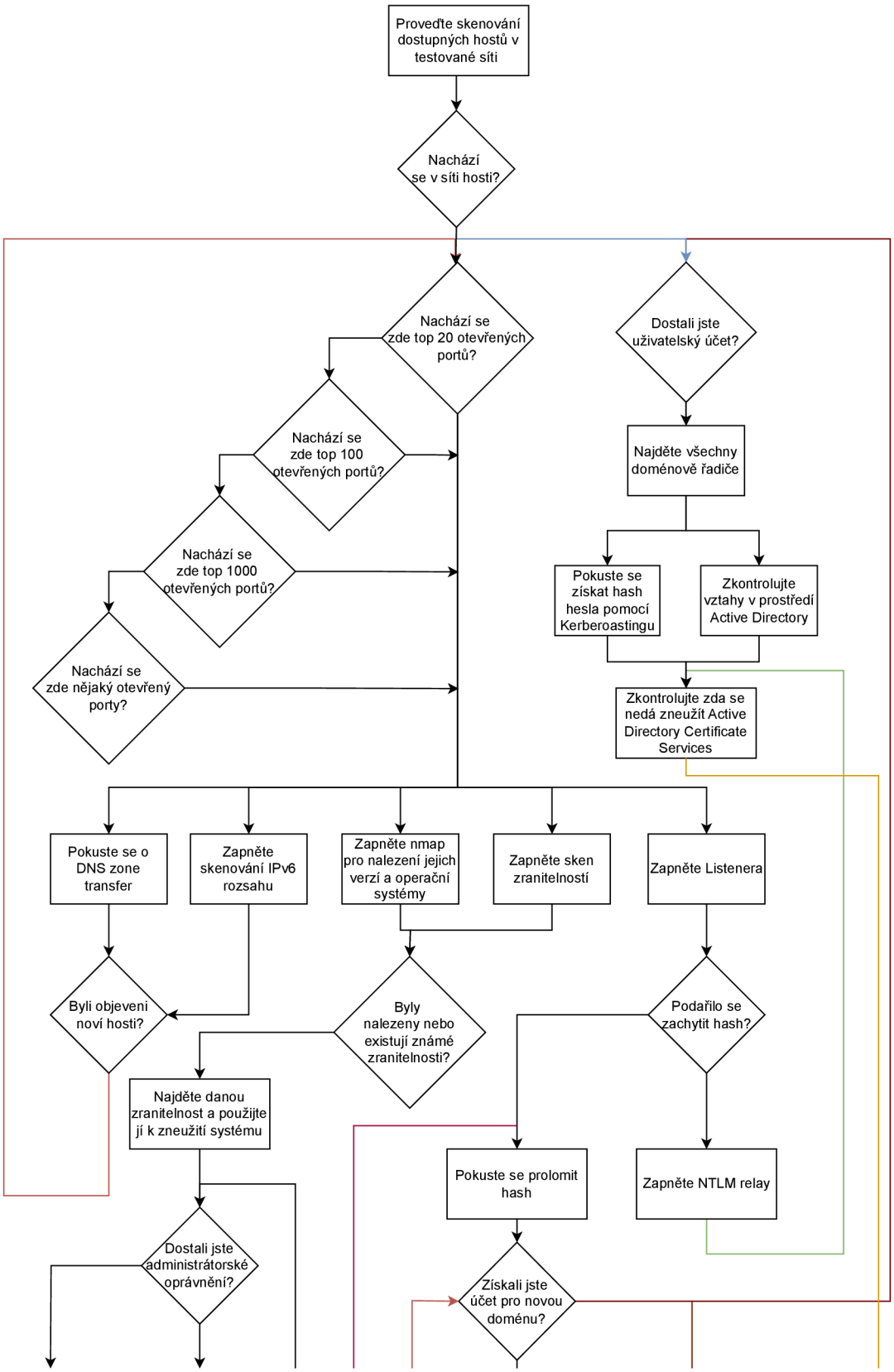
V případě, že se přihlašovací údaje týkají běžného uživatele, nejprve se zkontroluje přítomnost AppLockeru, který umožňuje správcům definovat pravidla, která omezují, jaké aplikace mohou být spuštěny na základě jednotlivých identit uživatelů. Dále se prověří, zda v systému nejsou uloženy v čitelné podobě soubory obsahující hesla. Obdobně pak probíhá výčet dostupných dat, protokoly jako jsou NFS, SMB, FTP, databází a slovníkových útoků.

Když při skenování dojde k odhalení jiného typu zranitelnosti, například webová aplikace bude zranitelná na SQL injekci, je klíčové potvrdit, zda se skutečně jedná o zranitelnost a zjistit, jaké může mít dopady.

Pokud dojde k získání účtu doménového administrátora, lze využít několik metod pro udržení přístupu v systému. Golden ticket umožňuje neomezený přístup ke všem službám a zdrojům v rámci domény tím, že zneužívá funkčnost Kerberos protokolu a vytváří falešné ověřovací tikety pro libovolného uživatele. Silver ticket je podobný, ale omezenější, protože umožňuje přístup pouze k určitým službám. Diamond ticket a Sapphire ticket jsou méně běžné varianty, které nabízí různé úrovně přístupu a perzistence v závislosti na cílených službách nebo systémech.

Dále je důležité ověřit důvěryhodný vztah mezi různými doménovými stromy v rámci lesa Active Directory. Doménové stromy jsou kolekce domén, které sdílejí společnou strukturu a bezpečnostní politiku. Udržování důvěryhodnosti mezi těmito stromy je klíčové pro zabezpečení komunikace a správné fungování síťových operací mezi různými doménami.

Pokud byl jakýmkoliv způsobem získán nový uživatelský účet, postup je podobný tomu, který byl popsán výše, závisí pouze na právech, která uživatel má.


Proveďte skenování dostupných hostů v testované síti
Nachází se v síti hosti?
Nachází se zde top 20 otevřených portů?
Nachází se zde top 100 otevřených portů?
Nachází se zde top 1000 otevřených portů?
Nachází se zde nějaký otevřený porty?
Dostali jste uživatelský účet?
Najděte všechny doménově řadiče
Pokuste se získat hash hesla pomocí Kerberoastingu
Zkontrolujte vztahy v prostředí Active Directory
Zkontrolujte zda se nedá zneužít Active Directory Certificate Services
Pokuste se o DNS zone transfer
Zapněte skenování IPv6 rozsahu
Zapněte nmap pro nalezení jejich verzí a operační systémy
Zapněte sken zranitelností
Zapněte Listenera
Byli objeveni noví hosti?
Byly nalezeny nebo existují známé zranitelnosti?
Podařilo se zachytit hash?
Najděte danou zranitelnost a použijte jí k zneužití systému
Pokuste se prolomit hash
Zapněte NTLM relay
Dostali jste administrátorské oprávnění?
Získali jste účet pro novou doménu?

Obr. 4.2: Vlastní návrh diagramu penetračního testování pro IPv4.

5 Vlastní implementace podpůrných penetračních nástrojů

Při implementaci vlastních podpůrných penetračních nástrojů se vychází z návrhu nástroje pro penetrační testování Wi-Fi sítí a IPv4, které byly stanoveny a popsány v předchozí kapitole. Postupy a principy vychází ze šablony od Penterepu, což umožňuje jejich snadnou integraci do platformy Penterep. Tato implementace lze proto přímo vložit do zmíněné platformy, což umožňuje efektivní využití v rámci standardizovaných penetračních testů.

5.1 Vlastní nástroje pro Wi-Fi sítě

V této části jsou popsány nástroje pro skenování a analýzu Wi-Fi sítí, zahrnující zachycení handshaků, detekci skrytých sítí, prolomení WPA/WPA2 hesel, provádění opakování ARP požadavků a chopchop útoku na WEP zabezpečené sítě.

5.1.1 Nástroj scanWifi.py

Nástroj scanWifi slouží ke skenování bezdrátových Wi-Fi sítí. Umožňuje uživatelům využít různé přepínače pro specifické úkoly: `-mm` pro aktivaci promiskuitního režimu, `-sn` pro skenování dostupných Wi-Fi sítí, `-sc` pro sledování určitého přístupového bodu, díky čemuž je umožněno sledovat připojené klienty a `-w` pro detekci WPS. Každý z těchto přepínačů má specifický typ výstupu.

Zapnutí promiskuitního režimu

Přepínač `-mm` slouží k aktivaci promiskuitního režimu na zvoleném síťovém rozhraní pomocí přepínače `-i`. Tato operace zahrnuje sekvenci příkazů jako vypnutí rozhraní, změnu na promiskuitní režim, ukončení všech možných rušivých procesů a opětovné zapnutí rozhraní. Během tohoto procesu může dojít například k chybě o špatném parametru rozhraní. V případě chyby nástroj zachytí a vypíše standardní chybový výstup. Pokud operace proběhne úspěšně, vypíše se MAC adresa rozhraní.

K aktivaci promiskuitního režimu je použita metoda `activate_monitor_mode`, která postupně provede potřebné kroky pro změnu režimu rozhraní. Po úspěšné změně režimu rozhraní následuje získání MAC adresy pomocí metody `get_mac_address`. Tato metoda využívá systémový příkaz `ifconfig` pro získání potřebných informací o rozhraní, včetně MAC adresy. Pokud dojde k jakékoliv chybě v průběhu získávání MAC adresy, Nástroj to detekuje a informuje uživatele o nepodařeném

pokusu, přičemž výstupem je specifická chybová zpráva nebo informace, že MAC adresa nebyla nalezena.

Skenování dostupných Wi-Fi sítí

Přepínač `-sn` je určen pro iniciaci skenování všech viditelných Wi-Fi sítí v okolí. Přepínač spustí příkaz s Airodump-ng, který v reálném čase zaznamenává informace o dostupných sítích do CSV souboru. Po určité době skenování, která je definována v kódu, a kterou si může každý uživatel změnit, je proces ukončen a data jsou následně zanalyzována a formátována pro další použití. Metoda `scan_wifi_networks` řídí tento proces, včetně spuštění, dohledu a ukončení příkazu Airodump-ng. Po ukončení příkazu Airodump-ng nástroj očekává existenci CSV souboru, který obsahuje data o skenovaných sítích.

Výpis 5.1: Metoda pro extrahování dat z CSV souboru do JSON formátu.

```
    def parse_ap_data(self, file):
        reader = csv.reader(file)
        next(reader)
        next(reader)
        aps = []
        for row in reader:
            if len(row) >= 14:
                ap_info = {
                    "name": row[0].strip(),
                    "netType": "wifi",
                    "bssid": row[0].strip(),
                    "wifiChannel": row[3].strip(),
                    "wifiEncryption": row[5].strip(),
                    "wifiCipher": row[6].strip(),
                    "wifiAuth": row[7].strip(),
                    "wifiEssid": row[13].strip() if row[13].strip()
    else "Hidden SSID",
                }
                aps.append(ap_info)
                self.ptjsonlib.add_node(self.ptjsonlib.
    create_node_object("net", properties=ap_info))
        if not self.use_json:
            for ap in aps:
                ptprinthelper.ptprint("AP Info:", bullet_type="
    TITLE", condition=not self.use_json)
                for key, value in ap.items():
                    ptprinthelper.ptprint(f"{key}: {value}",
    bullet_type="INFO", condition=not self.use_json)
                print()
```

Data extrahovaná z CSV souboru jsou následně zpracována metodou `parse_ap_data`, která čte záznamy a extrahuje důležité informace o každé síti, jako jsou BSSID, kanál, šifrování, ESSID a další relevantní atributy. Metoda je navržena tak, aby efektivně zpracovávala pouze validní a kompletní záznamy. Úryvek kódu lze vidět ve výpisu 5.1.

Na závěr nástroj odstraňuje původní CSV soubor, aby se zabránilo nahromadění nepotřebných dat. Tento postup udržuje dostatečné místo na disku a předchází problémům způsobeného hromaděním souborů, které se generují při každém spuštění nástroje.

Skenovaní jednotlivých přístupových bodů

Přepínač `-sc` je specificky navržen pro skenování jednotlivých přístupových bodů, což umožňuje využití konkrétních útoků a vidět připojené klienty k danému AP. Nástroj využívá Airodump-ng k zaznamenávání informací o klientech spojených s cílovým AP do CSV souboru. Tato operace je řízena metodou `capture_data_clients`, která spouští Airodump-ng s příslušnými parametry jako jsou BSSID `-b` a kanál `-ch`, specifikované uživatelem přes argumenty příkazové řádky, viz výpis 5.2.

Výpis 5.2: Metoda pro skenování konkrétního přístupového bodu.

```
    def capture_data_clients(self):
        interface = self.args.interface
        output_file = 'clientsWifi'
        command = ['airodump-ng', '--bssid', self.args.bssid, '--
    channel', str(self.args.channel), interface, '-w', output_file,
    '--output-format', 'csv']
        self.execute_command(command, output_file, self.
    parse_station_data)
```

Data zaznamenaná do CSV souboru jsou po ukončení skenu zpracována metodu `parse_station_data`. Tato metoda čte CSV soubor obdobně, jak již bylo vyobrazeno ve výpisu 5.1 s tím že, identifikuje relevantní záznamy klientů. Pro každého klienta extrahuje důležité informace jako MAC adresu, sílu signálu, počet přijatých paketů. Všechny nerelevantní soubory jsou na konci nástroje odebrány.

Aktivace WPS

Přepínač `-w` je speciálně navržen pro detekci WPS v dostupných bezdrátových sítích. Tento přepínač používá nástroj Wash, který skenuje a identifikuje sítě, u kterých je WPS povoleno. Metoda `detect_wps` zodpovídá za spuštění tohoto nástroje a následné zpracování výstupních dat, které jsou prezentovány uživateli.

Nástroj zpracovává a formátuje data získaná z Wash, aby bylo snadno identifikovatelné, které sítě mají WPS povoleno a mohou být potenciálně zranitelné. Tato data zahrnují BSSID sítě a stav WPS. Nepotřebná data jsou opět odstraněna.

5.1.2 Nástroj hiddenNetworkHandshakeCapture.py

Nástroj je speciálně navržen pro zachytávání handshaků a odhalování skrytých Wi-Fi sítí. Přepínač `-hn` umožňuje efektivně detekovat sítě se skrytým ESSID, které ve svých tzv. Beacons rámcích neuvádějí své ESSID. Na druhé straně, přepínač `-hc` je zaměřen na zachytávání handshaků, což jsou kritická informace pro prolomení hesla sítě.

Detekování skrytých Wi-Fi sítích

Pro aktivaci režimu k detekování skrytých sítí slouží argument `-hn`, který se povinně zadává při spuštění nástroje. Tento režim spustí nástroj v konfiguraci pro identifikaci sítí, který nevysílá své ESSID v Beacons. Pro zachytávání dat se používá metoda `run_airodump`, která nastaví a spustí Airodump-ng na zvoleném rozhraní `-i`, kanálu `-ch` a BSSID `-b`, ukládající data do formátu CSV.

Klíčovou částí procesu je využití deautentizačního útoku provedené metodou `run_aireplay`, který je definovaný pomocí přepínače `-cm`. Tato metoda spustí Aireplay-ng, aby odpojila klienta od sítě pomocí deautentizačních rámců. Účelem tohoto útoku je vynutit znovupřipojení klienta, což donutí přístupový bod k vyslání tzv. probe response rámce, pro odhalení ESSID.

Jakmile Airodump-ng dokončí sběr dat, aktivuje se metoda `parse_ap_data` k analýze vygenerovaného CSV souboru. Tato metoda prochází záznamy a vyhledává informace o skrytých ESSID. CSV soubory jsou po ukončení nástroje odstraněny.

Zachycení handshaku

K aktivaci zachycení handshaku slouží argument `-hc`, který při spuštění nástroje instruuje, aby provedl nezbytné kroky pro zachycení handshaku.

Deautentizační útok je realizován pomocí metody `run_aireplay`. Tato metoda spustí Aireplay-ng, který donutí odpojit klienta od sítě pomocí deautentizačních rámců. Cílem tohoto útoku je přinutit klienta k opětovnému připojení k síti, což vede k výměně EAPOL (Extensible Authentication Protocol over LAN) rámců, nezbytných pro úplný handshake. Tento moment je zásadní, protože bez opětovného připojení klienta by nebylo možné zachytit potřebné rámce.

Současně s deautentizačním útokem běží Airodump-ng, který monitoruje síťový provoz na specifikovaném kanálu `-ch`, BSSID `-b` a rozhraní `-i`. Tento nástroj sleduje

a zaznamenává všechny relevantní síťové rámce, ve funkci `check_and_extract_-eapol_packets`, tj. EAPOL, Probes a Beacons, viz výpis 5.3, které jsou nezbytné pro kompletní analýzu handshaku a potenciální prolomení šifrované sítě. Jakmile jsou tyto klíčové rámce úspěšně zachyceny, nástroj vytvoří .cap soubor s BSSID názvem sítě. Tento soubor obsahuje všechny důležité informace potřebné k dešifrování šifrované Wi-Fi sítě, přičemž přebytečná data jsou odstraněna, aby se minimalizovala velikost .cap souboru. Následně jsou všechny dočasné soubory, které byly během procesu vytvořeny, smazány.

Výpis 5.3: Úryvek metody k extrahování potřebných rámců pro prolomení hesla.

```
    def check_and_extract_eapol_frames(self, input_file, bssid):
        try:
            frames = rdpcap(input_file)
            eapol_frames = []
            beacon_frames = []
            probe_frames = []

            for frm in frames:
                if frm.haslayer(EAPOL):
                    eapol_frames.append(frm)
                elif frm.haslayer(Dot11Beacon):
                    beacon_frames.append(frm)
                elif frm.haslayer(Dot11ProbeReq) or frm.haslayer(
Dot11ProbeResp):
                    probe_frames.append(frm)

            if len(eapol_frames) >= 4:
                all_needed_frames = eapol_frames + beacon_frames +
probe_frames
                new_file = f"{bssid}_handshake.cap"
                wrpcap(new_file, all_needed_frames)
                os.remove(input_file)

                message = "All 4 EAPOL frames, along with the
beacon and probe frames, were captured and stored."
                ptprinthelper.ptprint(message, bullet_type="INFO",
condition=not self.use_json)

                        ... <snipped> ...
```

5.1.3 Nástroj wpaCracking.py

Nástroj je určený k prolomení WPA/WPA2 hesel ze zachycených EAPOL, Beacons a Probes rámců, používajících kombinaci Hcxpcapngtool a Hashcat. Pro spuštění

nástroje se musí specifikovat přepínač `-cp`, který spustí nástroj. Přepínač `-f` definuje cestu k .cap souboru obsahujícímu EAPOL, Beacons a Probes rámců. Tento soubor se vygeneruje nástrojem hiddenNetworkHandshakeCapture.py. Dalším přepínačem je cesta ke slovníkovému souboru `-w` pro útok slovníkovou metodou.

Zahájení procesu prolomení hesla je využit Hcxpcapngtool, který nejprve převede .cap souboru do formátu .hc22000, což je formát kompatibilní s Hashcatem. Následně Hashcat provádí útok na heslo s použitím specifikovaného slovníku. Úspěšnost útoku je poté ověřena pomocí příkazu `--show` v Hashcat, který zobrazí výsledné heslo, pokud bylo úspěšně prolomeno.

5.1.4 Nástroj arpRequestReplayAttack.py

Nástroj arpRequestReplayAttack.py je určený pro provedení útoku opakováním ARP požadavků na WEP zabezpečené sítě, se zaměřením na zachycení a prolomení síťového klíče.

Nástroj začíná spuštěním procesu Airodump-ng pomocí metody `start_airodump`, která nastaví a spustí proces pro zachycení paketů v síti. Tento proces se spouští s příslušnými argumenty, tj. BSSID cílové sítě `-b`, kanál `-ch` a rozhraní `-i`, které bylo určeno při spuštění nástroje. Zachycené pakety jsou ukládány do souboru ve formátu .cap.

Následně se pomocí metody `start_aireplay` spustí Aireplay-ng proces, který provádí útok opakování ARP požadavků. Proces přijímá argumenty BSSID cílové sítě a MAC adresu zdroje `-sm`.

Pro pokus o prolomení šifrované sítě slouží metoda `start_aircrack`, která spustí Aircrack-ng. Tento proces analyzuje zachycené pakety a hledá šifrovací klíč. Pokud je klíč nalezen, nástroj extrahuje tento klíč a jeho ASCII (American Standard Code For Information Interchange) reprezentaci pomocí metody `extract_key`, která analyzuje výstup z Aircrack-ng a hledá konkrétní vzory textu označující nalezení klíče, viz výpis 5.4.

Metoda `timeout_handler` zajišťuje ukončení procesu po uplynutí nastaveného časového limitu. Pro odstranění dočasných souborů, které byly vytvořeny během útoku, je v kódu implementována metoda `cleanup_files`. Tato metoda prohledává adresáře, identifikuje soubory vytvořené během útoku a následně je odstraní, aby nedocházelo k zanechávání nepotřebných dat.

Výpis 5.4: Metoda k extrahování klíče.

```
    def extract_key(self, line):
        key_match = re.search(r'\[ (.+?) \]', line)
        ascii_match = re.search(r'\(ASCII: (.+?) \)', line)
        key = key_match.group(1) if key_match else "N/A"
```

```
        ascii_key = ascii_match.group(1) if ascii_match else "N/A"
        return key, ascii_key
```

5.1.5 Nástroj fragmentationAttack.py a chopchopAttack.py

Nástroj pro provádění útoku KoreK chopchop a fragmetačního útoku jsou zcela dva odlišné nástroje pro penetrační testování bezpečnostního protokolu WEP. Každý útok má své specifika, ale oba sdílejí podobnou základní strukturu využívající následující metodu pro dosažení svých cílů. Nástroj fragmentationAttack.py se aktivuje pomocí přepínače `-wf`, zatímco chopchopAttack.py prostřednictvím `-wc`.

Metoda `run_airodump` spustí proces Airodump-ng, který monitoruje síť a zaznamenává provoz na specifikované BSSID `-b` a kanálu `-ch`. Tento krok je potřebný pro shromáždění dostatečného množství dat pro další fáze. Po zahájení monitorování následuje metoda `run_fakeauth`, která se pokouší kontinuálně autentizovat na síti s falešnou MAC adresou prostřednictvím Aireplay-ng.

Následuje metoda `run_aireplay`, která provádí buď KoreK chopchop nebo fragmentační útok, závisle na výběru uživatele. V obou případech Aireplay-ng slouží k získání klíčového proudu potřebného pro vytvoření padělaného paketu. Po úspěšném získání klíčového proudu, metoda `run_packetforge` použije tento proud k vytvoření nového ARP paketu, který je následně použit v opakovacím útoku.

Metoda `run_replay_attack` poté využívá padělaný paket k vygenerování síťového provozu, což přiměje síť k uvolnění dalších datových rámců nebo k odhalení šifrovacího klíče. Paralelně s tímto útokem metoda `run_aircrack` analyzuje data získaná během monitorování a útoku s cílem identifikovat a prolomit šifrovací klíč. Po dokončení operace se veškeré soubory, které nejsou již potřeba, smažou.

5.2 Vlastní nástroje pro IPv4

Tato sekce prezentuje nástroje určené pro síťové skenování, jako je detekce otevřených portů a IP adres a identifikace anonymního FTP.

5.2.1 Nástroj ipAndPortScan.py

Nástroj slouží pro skenování IP adres pomocí dostupných síťových portů pro rozsáhle adresní prostory IPv4. Využívá se k efektivnímu provádění rozsáhlých skenů s vysokou rychlostí. Je navržen tak, aby dokázal zjistit otevřené porty na určeném adresním prostoru IPv4, co nejrychleji a nejefektivněji, jelikož se jedná o základní krok pro bezpečnostní audity a identifikaci potenciálních zranitelných míst v síťové infrastruktuře.

Pro spuštění nástroje je zapotřebí použít přepínač `-ps`, který je definovaný s `--rate` určující požadovanou rychlost přenosu paketů, `-p` specifikující porty a rozsah skenovaných IP adres upřesňuje `--range`. Nejprve dojde k inicializaci třídy `ip_and_port_scan` s argumenty z příkazové řádky. Třída zahrnuje metodu `run_masscan`, která spustí Masscan s definovanými parametry a zachytí výstupy do dočasného JSON souboru.

Po dokončení skenu se vezme JSON soubor metodou `read_tmp_masscan_output`, která extrahuje data o skenovaných portech a IP adresách. Následuje metoda `create_json_nodes`, která zpracovává výstupy a převádí je do strukturovaného formátu, přičemž každá IP adresa s příslušnými porty jsou organizovány do uzlů v JSON databázi, viz výpis 5.5.

Výpis 5.5: Metoda k získání jednotlivých uzlů.

```
    def create_json_nodes(self, scan_results):
        ip_nodes = {}
        for result in scan_results:
            ip = result.get("ip")
            if ip not in ip_nodes:
                main_node = self.ptjsonlib.create_node_object("
    device", properties={
                "name": ip,
                "ipAddress": ip
                })
                self.ptjsonlib.add_node(main_node)
                ip_nodes[ip] = main_node.get("key")
            for port in result.get("ports", []):
                sub_node = self.ptjsonlib.create_node_object("
    service", parent=ip_nodes[ip], properties={
                    "name": port["port"],
                    "port": port["port"],
                    "proto": port["proto"],
                    "status": "open",
                    "reason": port["reason"],
                    "ttl": port["ttl"]
                })
                self.ptjsonlib.add_node(sub_node)
                self.ptjsonlib.set_status("finished")
        return self.ptjsonlib.get_result_json()
```

Pokud uživatel nevyžaduje JSON formát, nástroj obsahuje metodu `print_normal_output`, která formátuje a vypisuje výsledky skenování v čitelné podobě na konzoli. Tato metoda zobrazuje informace o každé otevřené službě na dané IP adrese, včetně portu, protokolu, stavu portu a TTL (Time To Live) hodnoty. Nástroj končí odstraněním dočasného souboru a zajišťuje.

5.2.2 Nástroj anonymousFTP.py

Nástroj je navržen k detekci anonymního FTP přístupu. Spouští se prostřednictvím přepínače `-af`, kde se definuje také host pomocí `--host`.

Pro zjištění, zda na specifikovaném serveru je povolen anonymní přístup k FTP se využívá Nmap s nasazením skriptu `ftp-anon`. Toto nastavení může zpřístupnit citlivá data neautorizovaným útočníkům. Inicializace nástroje probíhá předáním argumentů z příkazové řádky do třídy `anonymous_ftp`. Tyto argumenty specifikují cílového hosta a další volby, jako je možnost formátování výstupu do JSON. Třída pak spustí metodu `nmap_scan`, která provádí skenování portu 21, standardního portu pro FTP.

Po dokončení skenování metoda `check_ftp_vulnerability` ověří bezpečnost FTP služby na serveru tím, že zanalyzuje výsledky skenování portu 21. Pokud zjistí, že tento port je uzavřen, znamená to, že FTP služba není na serveru spuštěna a tudíž není přístupná. Naopak, pokud sken ukáže, že anonymní přístup k FTP je povolen, identifikuje toto jako potenciální zranitelnost. V případě, že sken neshledá žádné problémy týkající se anonymního přístupu, metoda uzavře, že žádná zranitelnost v tomto ohledu není přítomna, viz výpis 5.6.

Výpis 5.6: Metoda k detekování zranitelnosti FTP.

```
    def check_ftp_vulnerability(self, output):
        if "21/tcp closed" in output:
            return "FTP port is closed"
        elif "_ftp-anon: Anonymous FTP login allowed (FTP code 230)
    " in output:
            return "It is vulnerable on anonymous FTP"
        return "No anonymous FTP vulnerability found"
```

6 Výsledky a návrhy na další rozšíření nástrojů pro Wi-Fi a IPv4

Vlastní implementace penetračních nástrojů byla inspirována existujícími nástroji, které fungují jako wrappery. Tato implementace odstraňuje nedostatky, které jsou uvedené v tabulce pro Wi-Fi sítě 6.1 a IPv4 6.2.

Funkcionality	Sada nástrojů Aircrack-ng	Wifite2	Airgeddon	Vlastní nástroje
Promiskuitní režim	✓	✓	✓	✓
Automatické ukončení rušivých procesů	×	×	×	✓
Zachytávání paketů	✓	✓	✓	✓
Generování síťového provozu	✓	✓	✓	✓
Lámání hesel	✓	✓	✓	✓
Lámání hesla pomocí GPU	×	✓	✓	✓
Jednoduchost	×	✓	×	✓
Automatizace	×	✓	✓	✓
Uživatelsky přívětivé	×	✓	×	✓
Přenositelnost	✓	✓	✓	✓
Integrace nástrojů	×	✓	✓	✓
Optimalizované ukládání dat na disk	×	×	×	✓
Formátování	×	×	×	✓

Tab. 6.1: Dosažené výsledky pro Wi-Fi sítě.

Funkcionality	Nmap	Masscan	Naabu	Vlastní nástroje
Podpora adresního prostoru IPv4	✓	✓	✓	✓
Podpora adresního prostoru IPv6	✓	✓	✓	✓
Rychlost skenování dostupných hostů a portů	×	✓	×	✓
Podpora skriptování	✓	×	×	✓
Použitelnost v příkazové řádce	✓	✓	✓	✓
Formátování	×	×	×	✓
Překlad DNS	✓	×	✓	✓

Tab. 6.2: Dosažené výsledky pro skenování IPv4.

Všechny nástroje jsou připraveny k integraci do platformy Penterep pro budoucí penetrační testování Wi-Fi sítí, skenování otevřených portů a kontrolování anonymních FTP přístupů.

Pro rozšíření nových podpůrných Wi-Fi nástrojů pro platformu Penterep by bylo možné implementovat následující penetrační techniky: útoky pomocí nástroje Reaver, jako jsou útoky WPS Null Pin, útoky Pixie Dust a útoky hrubou silou. Dále k rozšíření možností by patřily i útoky na autentizační protokoly, konkrétně

na WPA2-EAP a WPA3-EAP, což zahrnuje zaměření na zranitelnosti spojené s těmito autentizačními metodami využívajícími šifrování AES. Nástroj Wash může být použit k detekci, zda jsou na WPS implementovány ochrany proti útokům hrubou silou, jelikož dokáže detekovat, zda došlo k uzamčení směrovače a na jak dlouho. Rozšíření zahrnuje také útoky na WPA3 využití Dragonblood útoků. Také by bylo vhodné zkontrolovat zda není implementována MAC filtrace. V neposlední řadě, by se zaměřilo na útoky směrovačů, kde by se testovali různé útoky, například bezpečnost přihlašovacích stránek pomocí slovníkových útoků. V případě nových návrhů pro Wi-Fi sítě lze do diagramu přidávat nové prvky, které by doplňovaly nebo rozšiřovaly stávající metodologii.

Pro rozvoj nových nástrojů IPv4 pro platformu Penterep je možné přidávat nové funkce, které jsou již začleněny v existujícím diagramu pro IPv4. Navíc je možné do tohoto diagramu zahrnout další prvky, které by rozšířily nebo doplnily stávající metodologii. Tento diagram by měl přehledně ukazovat základní metodologii. V současném diagramu také chybí tzv. pivoting, což představuje zajímavou oblast pro další rozvoj této metodologie.

Závěr

V této práci bylo podrobně popsáno penetrační testování a bezpečnost bezdrátových Wi-Fi sítí a IPv4, přičemž se zdůraznil význam těchto technologií v současném digitálním věku. Hlavním cílem bylo navrhnout a implementovat podpůrné nástroje pro bezpečnostní penetrační testování bezdrátových sítí Wi-Fi a síťové infrastruktury IPv4. V rámci analýzy byly identifikovány klíčové zabezpečení a útoky, které hrozí Wi-Fi sítím a protokolu IPv4 a byly navrženy nástroje, které mohou pomoci v detekci a prevenci těchto hrozeb.

Nejprve se práce zaměřila na teoretické pochopení penetračního testování, vývoj Wi-Fi sítí, neboli standardu IEEE 802.11, kde byla vysvětlena historie, bezpečnostní protokoly, existující nástroje a možné útoky. Dále byl poskytnut základní přehled o IPv4 a nástrojích souvisejících s touto technologií.

Následně bylo přistoupeno k testování vyvinutých nástrojů. Tyto nástroje zahrnovaly skenery sítí, zachytávání handshaku, odhalení skrytých sítí a prolamování WPA klíčů, stejně jako nástroje pro zkoumání zranitelností IPv4 a mnoho dalšího.

Práce systematicky prozkoumala existující nástroje a zdůraznila, že jejich omezený výkon a flexibilita vedly k potřebě vytvoření nových nástrojů pro zabezpečení Wi-Fi a IPv4 sítí. Na základě těchto poznatků byly navrhnuty diagramy a vyvinuty nástroje, které odpovídají vlastním potřebám.

Poté bylo demonstrováno praktické použití těchto nástrojů na simulovaných síťových prostředích, aby bylo možné ověřit jejich efektivitu a identifikovat případné zranitelnosti. Výsledky testování ukázaly, že tyto nástroje jsou schopné účinně identifikovat Wi-Fi sítě a připojené klienty. Nástroje dokážou detekovat aktivované WPS, odhalit živé hosty pro IPv4 a otevřené porty a zjistit, zda je povolený anonymní režim FTP. Nástroje také zvládly útoky na Wi-Fi sítě, včetně odchycení handshaků, prolomení hesel, odhalení skrytých sítí a útoky na WEP. Tato zjištění potvrzují jejich potenciál jako cenný přínos k oblasti síťové bezpečnosti. Všechny požadavky byly v souladu s touto prací splněny.

Vzhledem k dynamické povaze síťové bezpečnosti a neustálému vývoji nových útočných vektorů je možné dále navazovat na tuto práci. Budoucí práce by mohly pokračovat v rozšiřování diagramu, nástrojů a optimalizaci existujících nástrojů.

Literatura

[1] MITCHELL, Bradley, HEINE JR, Michael Barton (ed.). *What Is the Range of a Typical Wi-Fi Network?* Online. <Https://www.lifewire.com/. Listopad 5, 2020. Dostupné z: <https://www.lifewire.com/range-of-typical-wifi-network-816564. [cit. 2023-10-30].

[2] RAJA MALIK, Ramiz. *What is Wi-Fi and what will its future look like ?* Online. 2022. Dostupné z: <https://timesofindia.indiatimes.com/readersblog/ramiz-raja-malik/what-is-wi-fi-and-what-will-its-future-look-like-47913/.[cit. 2023-10-30]>. [cit. 2023-10-30].

[3] SHRAVAN, Kumar; NEHA, Bansal a PAWAN, Bhadana. *Penetration Testing: A Review.* Online. Penetration Testing: A Review. 2014, roč. 4, č. 3, s. 752-757. ISSN ISSN:2320-0790. Dostupné z <https://d1wqtxts1xzle7.cloudfront.net/42046583/COMPUSOFT__34__752-757-libre.pdf?1454595834=&response-content-disposition=inline%3B+filename%3DCOMPUSOFT_3_4_752_757.pdf&Expires=1713949910&Signature=XHokkcrJ8XYq-hhY4S9bwhiCVn7MQZYb7UWrQv0s9k7ne6QobMhrG2sifj2qh55-eM4CBhb98E-zQPCKd~Oq-Ggd4U-KaqVRCna5VVN-JiFm1d7vekVznkKKj-4aojkejAoyB4ihVdYXTjOXcUvqpfszjJgsWcD9wF89B7TAq4jy6rsWEh5hJ4hFK-SAWYseud7N13vY0hqp0G~T4pozQ99d7AvirE1RpDQ7eiOSeBhry2nwSXw57eFlb3-FzzSngvOgm4aOHV9rz5vMvOL7wyXuUWOQvFs86FGMRchhDCXDBxbd7gGZqua-qegERWu~NXc744oe0VrPC1K3TxB99HvvA__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA>. [cit. 2024-04-24].

[4] FIRCH, Jason, SWANAGAN, Michael (ed.). *Types-penetration-testing.* Online. 2022, 25.02.2023. Dostupné z: <https://purplesec.us/types-penetration-testing/>. [cit. 2023-11-01].

[5] M. Denis, C. Zena and T. Hayajneh, *Penetration testing: Concepts, attack methods, and defense strategies.* Online. 2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT), Farmingdale, NY, USA, 2016, pp. 1-6, doi: 10.1109/LISAT.2016.7494156 [cit. 2024-03-19].

[6] *The Hacker News. Before and After a Pen Test: Steps to Get Through It.* Online. 2021. Dostupné z: <https://thehackernews.com/2021/10/before-and-after-pen-test-steps-to-get.html>. [cit. 2023-11-12].

[7] *Black box vs white box testing.* Online. Dostupné z: <https://www.practitest.com/resource-center/article/black-box-vs-white-box-testing/>. [cit. 2023-11-01].

[8] Mitnick Security. *Understanding the Main Types of Penetration Testing.* Online. 2023. Dostupné z: <https://www.mitnicksecurity.com/blog/understanding-the-main-types-of-penetration-testing>. [cit. 2023-11-02].

[9] *PENETRAČNÍ TESTOVÁNÍ — ÚVOD DO PROBLEMATIKY.* Online. 2022. Dostupné z: <https://nukib.cz/download/publikace/podpurne_materialy/2022-03-07_Penetracni-testovani_v1.1.pdf. [cit. 2023-12-11].

[10] ANDREW, Daniel. *What is an external pentest?* Online. 2023. Dostupné z: <https://www.intruder.io/blog/what-is-an-external-pentest>. [cit. 2023-12-11].

[11] Altulaihan, E.A.; Alismail, A.; Frikha, M. *A Survey on Web Application Penetration Testing.* Electronics 2023, 12, 1229. https://doi.org/10.3390/electronics12051229

[12] Salahdine, F.; Kaabouch, N. *Social Engineering Attacks: A Survey.* Future Internet 2019, 11, 89. https://doi.org/10.3390/fi11040089

[13] MORADOV, Oliver. *9 Penetration Testing Types.* Online. 2022. Dostupné z: <https://brightsec.com/blog/penetration-testing-types/>. [cit. 2023-12-11].

[14] WANG, Shao-Long; WANG, Jian; FENG, Chao a PAN, Zhi-Peng. Wireless Network Penetration Testing and Security Auditing. Online. Wireless network penetration testing and security auditing. 2016, roč. 7, č. -, s. 1-5. Dostupné z: https://doi.org/10.1051/itmconf/20160703001. [cit. 2024-04-24].

[15] BASU, Saumick. *7 Penetration Testing Phases: Your One-Stop Guide.* Online. 2023. Dostupné z: <https://www.getastra.com/blog/security-audit/penetration-testing-phases/. [cit. 2024-04-24].

[16] Penetration Testing. *Understanding the Five Phases of the Penetration Testing Process.* Online. 2022. Dostupné z: <https://www.eccouncil.org/cybersecurity-exchange/penetration-testing/penetration-testing-phases/>. [cit. 2023-11-06].

[17] N. Singh, V. Meherhomji and B. R. Chandavarkar, *Automated versus Manual Approach of Web Application Penetration Testing* 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2020, pp. 1-6, doi: 10.1109/ICCCNT49239.2020.9225385 [cit. 2024-03-19].

[18] admin. *The History of WiFi: 1971 to Today.* Online. 2017. Dostupné z: <https://www.cablefree.net/wireless-technology/history-of-wifi-technology/>. [cit. 2023-10-31].

[19] LINKS, Cees. *The Evolution of Wi-Fi networks: from IEEE 802.11 to Wi-Fi 6E.* Online. 2022. Dostupné z: <https://www.wevolver.com/article/the-evolution-of-wi-fi-networks-from-ieee-80211-to-wi-fi-6e>. [cit. 2023-10-31].

[20] SPADAFORA, Anthony. *Wi-Fi 6E vs Wi-Fi 7: What's the difference?* Online. 2023. Dostupné z: <https://www.tomsguide.com/face-off/wi-fi-6e-vs-wi-fi-7-whats-the-difference. [cit. 2024-05-11].

[21] DUGAND, Franz. *Wi-Fi 7 (IEEE 802.11be) & MLO vs. Wi-Fi 6/6E (IEEE 802.11ax): What to Ask for Optimal Design Considerations.* Online. 2023. Dostupné z: <https://www.ceva-ip.com/ourblog/wi-fi-7-ieee-802-11be-mlo-vs-wi-fi-6-6e-ieee-802-11ax-what-to-ask-for-optimal-design-considerations/. [cit. 2024-05-11].

[22] *Wired Equivalent Privacy (WEP): Definition & Risks.* Online. 2022. Dostupné z: <https://www.okta.com/identity-101/wep/>. [cit. 2023-11-08].

[23] ALAMANNI, Marco a , Packt. *An Introduction to WEP.* Online. 2015. Dostupné z: <https://hub.packtpub.com/introduction-wep/>. [cit. 2024-04-25].

[24] GILLIS, Alexander S. *DEFINITION WPA3.* Online. Dostupné z: <https://www.techtarget.com/searchsecurity/definition/WPA3>. [cit. 2023-11-08].

[25] M. Vanhoef and E. Ronen, *Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd*, 2020 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 2020, pp. 517-533, doi: 10.1109/SP40000.2020.00031. [cit. 2024-04-25]

[26] A. Halbouni, L. -Y. Ong and M. -C. Leow, *Wireless Security Protocols WPA3: A Systematic Literature Review*, in IEEE Access, vol. 11, pp. 112438-112450, 2023, doi: 10.1109/ACCESS.2023.3322931. [cit. 2024-04-25]

[27] Aircrack-ng. Online. 2007, 2023. Dostupné z: <https://www.aircrack-ng.org/>. [cit. 2023-11-29].

[28] DEVITO, Andrew. *How to Use Aircrack-ng: A Guide to Network Compromise*. Online. 2023. Dostupné z: <https://www.stationx.net/how-to-use-aircrack-ng-tutorial/>. [cit. 2023-11-29].

[29] *About airodump-ng*. Online. -. Dostupné z: <https://www.javatpoint.com/airodump-ng>. [cit. 2023-11-29].

[30] admin. *Airodump-ng – Capture packets and display information about wireless networks (Command Examples)*. Online. Dostupné z: <https://www.thegeekdiary.com/airodump-ng-capture-packets-and-display-information-about-wire\less-networks-command-examples>. [cit. 2023-11-29].

[31] *Airodump-ng*. Online. 2022. Dostupné z: <https://wiki.aircrack-ng.org/doku.php?id=airodump-ng>. [cit. 2023-11-29].

[32] mister_x. *Airmon-ng*. Online. -, 09.02.2022. Dostupné z: <https://www.aircrack-ng.org/doku.php?id=airmon-ng>. [cit. 2023-11-30].

[33] AZAD, Usama. *Aireplay-ng*. Online. 2020. Dostupné z: <https://linuxhint.com/aireplay_ng/>. [cit. 2023-11-29].

[34] mister_x. *Aireplay-ng*. Online. Dostupné z: <https://www.aircrack-ng.org/doku.php?id=aireplay-ng>. [cit. 2023-11-29].

[35] *WiFite2 Automated WiFi hacking tool*. Online. 2021. Dostupné z: <https://systemweakness.com/wifite2-automated-wifi-hacking-tool-30773c20cac5>. [cit. 2023-11-29].

[36] derv82. *Wifite2*. Online. 2018. Dostupné z: <https://github.com/derv82/wifite2>. [cit. 2023-11-29].

[37] *Wifite: A step-by-step guide for Kali Linux users*. Online. -. Dostupné z: <https://infosecscout.com/wifite-a-step-by-step-guide-for-kali-linux-users/>. [cit. 2023-11-29].

[38] *Airgeddon*. Online. -. Dostupné z: <https://linuxsecurity.expert/tools/airgeddon/>. [cit. 2023-11-29].

[39] *Airgeddon*. Online. -. Dostupné z: <https://hackingforbabies.gitbook.io/en/wifi-cracking/airgeddon>. [cit. 2023-11-29].

[40] savio-code. *Fern Wifi Cracker*. Online. 2022. Dostupné z: https://github.com/savio-code/fern-wifi-cracker. [cit. 2024-04-25].

[41] *Passive Capture*. Online. 2022. Dostupné z: <https://www.kismetwireless.net/docs/readme/intro/passive_capture>/. [cit. 2024-04-25].

[42] *About Wireshark*. Online. -. Dostupné z: <https://www.wireshark.org/about>. [cit. 2023-11-30].

[43] KAMATHE, Gaurav. *Use Wireshark at the Linux command line with TShark*. Online. 2020. Dostupné z: <https://opensource.com/article/20/1/wireshark-linux-tshark>. [cit. 2023-11-30].

[44] CARRANZA, Aparicio; MAGALLANES, Josue; DECUSATIS, Casimer a ESPINAL, Javier. *Automated Wireless Network Penetration Testing Using Wifite and Reaver*. Online. Global Partnerships for Development and Engineering Education: Proceedings of the 15th LACCEI International Multi-Conference for Engineering, Education and Technology. 2017, s. 64. ISSN 2414-6390. Dostupné z: <https://dx.doi.org/10.18687/LACCEI2017.1.1.64>. [cit. 2024-04-25].

[45] feitoi a rofl0r. *Reaver-wps-fork-t6x*. Online. Dostupné z: <https://github.com/t6x/reaver-wps-fork-t6x>. [cit. 2024-04-25].

[46] ZerBea. *Hcxdumptool. Online*. 2024. Dostupné z: <https://github.com/ZerBea/hcxdumptool>. [cit. 2024-04-25].

[47] ZerBea. *Hcxtools*. Online. 2024. Dostupné z: <https://github.com/ZerBea/hcxtools>. [cit. 2024-04-25].

[48] jsteube. *Hashcat*. Online. 2022. Dostupné z: <https://github.com/hashcat/hashcat>. [cit. 2024-04-25].

[49] ALI, Khalda; ALZAIDI, Maram; AL-FRAIHAT, Dimah a ELAMIR, Amir M. *Artificial Intelligence: benefits, application, ethical issues, and organizational responses*. Online. Intelligent Sustainable Systems. 2022, roč. 1, s. 685–702. ISSN 2367-3389. Dostupné z: <https://doi.org/10.1007/978-981-19-7660-5>. [cit. 2024-04-25].

[50] VANHOEF, Mathy. *Key Reinstallation Attacks Breaking WPA2 by forcing nonce reuse*. Online. 2017. Dostupné z: <https://www.krackattacks.com/>. [cit. 2024-04-25].

[51] vanhoefm. *Krackattacks-scripts*. Online. 2021. Dostupné z: <https://github.com/vanhoefm/krackattacks-scripts>. [cit. 2024-04-25].

[52] VANHOEF, Mathy. *DRAGONBLOOD Analysing WPA3's Dragonfly Handshake*. Online. -. Dostupné z: <https://wpa3.mathyvanhoef.com/>. [cit. 2024-04-25].

[53] PACKT. *What we can learn from attacks on the WEP Protocol*. Online. 2015. Dostupné z: <https://hub.packtpub.com/what-we-can-learn-attacks-wep-protocol/>. [cit. 2023-11-09].

[54] CROIX, Alexandre. *How does WPA/WPA2 WiFi security work, and how to crack it?* Online. 2019. Dostupné z: <https://cylab.be/blog/32/how-does-wpawpa2-wifi-security-work-and-how-to-crack-it>. [cit. 2023-11-09].

[55] Cybersecurity News. *WPS Cracking with Reaver*. Online. 2020. Dostupné z: <https://outpost24.com/blog/wps-cracking-with-reaver/>. [cit. 2023-11-10].

[56] SPALTER, Michael. *WPA3 - What is it, and what's new?* Online. Březen 2023. Dostupné z: <https://www.draytek.co.uk/information/blog/wpa3-what-is-it-and-what-is-new>. [cit. 2023-11-09].

[57] GRIMMICK, Robert. *What is Wireless Sniffing?* Online. 2023. Dostupné z: <https://www.easytechjunkie.com/what-is-wireless-sniffing.htm>. [cit. 2023-11-11].

[58] *What Is a Replay Attack?* Online. 2023. Dostupné z: <https://www.kaspersky.com/resource-center/definitions/replay-attack>. [cit. 2023-11-11].

[59] *WEP crack*. Online. -. Dostupné z: <https://nordvpn.com/cybersecurity/glossary/wep-crack/>. [cit. 2023-11-30].

[60] mister_x. *KoreK chopchop*. Online. -, 02.06.2009. Dostupné z: <https://www.aircrack-ng.org/doku.php?id=korek_chopchop>. [cit. 2023-11-30].

[61] mister_x. *Fragmentation Attack*. Online. -, 05.09.2009. Dostupné z: <https://www.aircrack-ng.org/doku.php?id=fragmentation>. [cit. 2023-11-30].

[62] C., Ruth. *How a deauthentication attack works*. Online. 2022. Dostupné z: <https://atlasvpn.com/blog/what-is-a-deauthentication-attack>. [cit. 2023-11-11].

[63] VYAS, Radhika. *Wi-Fi Spoofing: A Major Threat to Network Security*. Online. 2023, 2023. Dostupné z: <https://www.cloudradius.com/wi-fi-spoofing-a-major-threat-to-network-security/>. [cit. 2023-11-11].

[64] DAVIES, Vikki. *What is network spoofing and how do you prevent it?* Online. 2021. Dostupné z: <https://cybermagazine.com/network-security/what-network-spoofing-and-how-do-you-prevent-it>. [cit. 2023-11-11].

[65] cybertrust-it. *WiFi Spoofing & Staying Safe On Public Networks*. Online. 2022. Dostupné z: <https://www.cybertrust-it.com/2021/09/wifi-spoofing/>. [cit. 2023-11-11].

[66] *Evil twin attacks and how to prevent them*. Online. 2023. Dostupné z: <https://www.kaspersky.com/resource-center/preemptive-safety/evil-twin-attacks>. [cit. 2023-11-11].

[67] GHIMIRAY, Deepan. *What Is an Evil Twin Attack and How Does It Work?* Online. 2022. Dostupné z: <https://www.avast.com/c-evil-twin-attack>. [cit. 2023-11-11].

[68] ROUSE, Margaret. *Rogue Access Point*. Online. 2023. Dostupné z: <https://www.techopedia.com/definition/4082/rogue-access-point-rogue-ap>. [cit. 2023-11-11].

[69] D. J. Fehér and B. Sándor, *Effects of the WPA2 KRACK Attack in Real Environment*, 2018 IEEE 16th International Symposium on Intelligent Systems and Informatics (SISY), Subotica, Serbia, 2018, pp. 000239-000242, doi: 10.1109/SISY.2018.8524769

[70] M. Alhamry and W. Elmedany, *Exploring Wi-Fi WPA2 KRACK Vulnerability: A Review Paper*, 2022 International Conference on Data Analytics for Business and Industry (ICDABI), Sakhir, Bahrain, 2022, pp. 766-772, doi: 10.1109/ICDABI56818.2022.10041548.

[71] F. T. Sheldon, J. M. Weber, S. -M. Yoo and W. D. Pan, *The Insecurity of Wireless Networks*, in IEEE Security & Privacy, vol. 10, no. 4, pp. 54-61, July-Aug. 2012, doi: 10.1109/MSP.2012.60.

[72] ZOLA, Andrew. *KoreK chopchop*. Online. 2021. Dostupné z: <https://www.techtarget.com/searchstorage/definition/address-space>. [cit. 2024-04-25].

[73] TERRELL, Katie. *Network Address Translation (NAT)*. Online. 2021. Dostupné z: <https://www.techtarget.com/searchnetworking/definition/Network-Address-Translation-NAT>. [cit. 2024-04-25].

[74] BURKE, John. *CIDR (Classless Inter-Domain Routing or supernetting)*. Online. 2022. Dostupné z: <https://www.techtarget.com/searchnetworking/definition/CIDR>. [cit. 2024-04-25].

[75] FERGUSON, Kevin. *Subnet (subnetwork)*. Online. 2021. Dostupné z: <https://www.techtarget.com/searchnetworking/definition/subnet>. [cit. 2024-04-25].

[76] ROSENCRANCEGEORGE, Linda a MOOZAKIS, LawtonChuck, MOOZAKIS, Chuck (ed.). *User Datagram Protocol (UDP)*. Online. 2023. Dostupné z: <https://www.techtarget.com/searchnetworking/definition/UDP-User-Datagram-Protocol>. [cit. 2024-04-25].

[77] YASAR, Kinza, LUTKEVICH, Ben (ed.). *User Datagram Protocol (UDP)*. Online. 2023. Dostupné z: <https://www.techtarget.com/searchnetworking/definition/TCP>. [cit. 2024-04-25].

[78] YASAR, Kinza a CHAI, Wesley, IREI, Alissa (ed.). *Network protocol*. Online. 2023. Dostupné z: <https://www.techtarget.com/searchnetworking/definition/protocol>. [cit. 2024-04-25].

[79] *Nmap: Discover your network*. Online. Dostupné z: <https://nmap.org/>. [cit. 2024-05-10].

[80] robertdavidgraham. *MASSCAN: Mass IP port scanner*. Online. Dostupné z: <https://github.com/robertdavidgraham/masscan>. [cit. 2024-05-10].

[81] *Naabu*. Online. Dostupné z: <https://www.kali.org/tools/naabu/>. [cit. 2024-05-10].

Seznam symbolů a zkratek

AD Active Directory

AES Advanced Encryption Standard

API Application Programming Interface

AP Access Point

ARP Address Resolution Protocol

ASCII American Standard Code for Information Interchange

AS Autonomous System

BGP Border Gateway Protocol

BSSID Basic Service Set Identifier

CAPTCHA Completely Automated Public Turing test to tell Computers and Humans Apart

CBC Cipher Block Chaining

CCMP Counter Mode Cipher Block Chaining Message Authentication Code Protocol

CCK Complementary Code Keying

CIDR Classless Inter-Domain Routing

CLI Command Line Interface

CSMA/CA Carrier-sense multiple access with collision avoidance

CSV Comma-Separated Values

CVE Common Vulnerabilities and Exposures

CVSS Common Vulnerability Scoring System

CRC32 32-bit Cyclic Redundancy Check

DNS Domain Name System

DSSS Direct Sequence Spread Spectrum

EAP Extensible Authentication Protocol

EAPOL	Extensible Authentication Protocol over LAN
EHT	Extremely High Throughput
ESSID	Extended Service Set Identifier
FCC	Federal Communications Commission
FMS	Fluhrer, Mantin, and Shamir attack
FTP	File Transfer Protocol
GCMP	Galois/Counter Mode Protocol
GH	Gigahertz
GTK	Group Temporal Key
GUI	Graphical User Interface
GPU	Graphics Processing Unit
HW	Hardware
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ICMP	Internet Control Message Protocol
ICV	Integrity Check Value
ID	Identifier
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
ISM	Industrial, Scientific, and Medical
IV	Initialization Vector
IT	Information Technology
JSON	JavaScript Object Notation
KRACK	Key Reinstallation Attack
LAN	Local Area Network

LLC	Logical Link Control
LSA	Local Security Authority
LSASS	Local Security Authority Subsystem Service
MAC	Media Access Control
MIMO	Multiple Input, Multiple Output
MIC	Message Integrity Code
MITM	Man In The Middle
MLO	Multi-Link Operation
MU	Multi-User
NAT	Network Address Translation
NFS	Network File System
NSE	Nmap Scripting Engine
NVD	National Vulnerability Database
NTLM	NT LAN Manager
OFDMA	Orthogonal Frequency-Division Multiple Access
OFDM	Orthogonal Frequency-Division Multiplexing
OS	Operating System
OSI	Open Systems Interconnection
OSINT	Open Source Intelligence
OSPF	Open Shortest Path First
PAT	Port Address Translation
PIN	Personal Identification Number
PMK	Pairwise Master Key
PRGA	Pseudo Random Generation Algorithm
PSK	Pre-shared Key

PTK	Pairwise Transient Key
PTW	Pyshkin, Tews, Weinmann attack
RC4	Rivest Cipher 4
RF	Radio Frequency
SAE	Simultaneous Authentication of Equals
SAM	Security Accounts Manager
SHA	Secure Hash Algorithm
SMB	Server Message Block
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SQL	Structured Query Language
SSH	Secure Shell
SSL	Secure Sockets Layer
SYN	Synchronize Sequence Numbers
TCP	Transmission Control Protocol
TKIP	Temporal Key Integrity Protocol
TTL	Time To Live
TXT	Text
UDP	User Datagram Protocol
VLSM	Variable Length Subnet Mask
WEP	Wired Equivalent Privacy
Wi-Fi	Wireless Fidelity
WPA	Wi-Fi Protected Access
WPA2	Wi-Fi Protected Access 2
WPA3	Wi-Fi Protected Access 3

WPS Wi-Fi Protected Setup

XOR Exclusive OR

XSS Cross-Site Scripting

A Spuštění jednotlivých nástrojů

Nástroje byly testovány ve virtuálním prostředí Kali Linux se síťovým adaptérem, který podporuje pouze pásmo 2.4 GHz.

1. Doporučená verze Pythonu je 3.6 nebo vyšší.
2. Pro testování je nutné nainstalovat potřebné knihovny:
   - pip3 install pexpect scapy ptlibs
3. Pro správnou funkčnost skriptů je potřeba nainstalovat následující nástroje:
   - sudo apt-get install aircrack-ng reaver hashcat masscan nmap
4. Pro instalaci Hcxpcapngtool a ostatních nástrojů z balíčku:
   - git clone https://github.com/ZerBea/hcxtools.git
   - cd hcxtools
   - make
   - sudo make install
5. Jednotlivé nástroje obsahují pomocí přepínače -h --help podrobný návod ke spuštění a uvedení hodnot k různým přepínáčům
6. Příklad spuštění jednotlivých nástrojů, kde se za {} nahradí patřičný parametr:
   - sudo python3 scanWifi.py -i {interface} -mm
   - sudo python3 scanWifi.py -i {interface} -sn
   - sudo python3 scanWifi.py -i {interface} -w
   - sudo python3 scanWifi.py -i {interface} -b {apBssid} -ch {channel} -sc
   - sudo python3 hiddenNetworkHandshakeCapture.py -cm {clientMac} -b {apBssid} -ch {channel} –interface {interface} -hc
   - sudo python3 hiddenNetworkHandshakeCapture.py -cm {clientMac} -b {apBssid} -ch {channel} –interface {interface} -hn
   - sudo python3 wpaCracking.py -f {capFile} -w {wordListFile} -cp
   - sudo python3 fragmentationAttack.py -sm {sourceMac} -b {apBssid} -ch {channel} -i {interface} -wf
   - sudo python3 chopchopAttack.py -sm {sourceMac} -b {apBssid} -ch {channel} -i {interface} -wc
   - sudo python3 arpRequestReplayAttack.py -sm {sourceMac} -b {apBssid} -ch {channel} -i {interface} -wa
   - sudo python3 ipAndPortScan.py --rate {rate} -p {ports} --ranges {ranges} -ps
   - sudo python3 anonymousFTP.py --host {host} -af

Uvedení nejčastějších chyb při spuštění nástrojů:

1. Příkaz není spuštěn pomocí sudo, tedy jako superuživatel.

2. Síťový adaptér podporuje pouze pásmo 2.4 GHz. Přístupové body, které podporují pouze 5 GHz, nejsou odchyceny síťovým adaptérem.

¨

B Obsah elektronické přílohy

V elektronické příloze jsou obsaženy zdrojové kódy této diplomové práce.

- `/` ..... kořenový adresář přiloženého archivu
- `vlastniNastroje` ..... adresář s jednotlivými nástroji
  - `anonymousFTP` ..... adresář pro identifikování anonymního FTP
  - `arpRequestReplayAttack` ..... adresář pro útok přehráním ARP požadavků
  - `chopchopAttack` ..... adresář pro útok Korek chopchop
  - `fragmentationAttack` ..... adresář pro fragmentační útok
  - `hiddenNetworkHandshakeCapture` ..... adresář pro odchycení WPA handshaku a odhalení skrytých sítí
  - `ipAndPortScan` ..... adresář pro skenování adresního prostoru a portů
  - `scanWifi` ... adresář pro promiskuitní režim, skenování Wi-Fi sítě, klientů a WPS režimu
  - `wpaCracking` ..... adresář pro prolomení hesla