Univerzita Hradec Králové

Fakulta informatiky a managementu

Katedra informačních technologií

Využití penetračního testování v bezdrátových sítích

Bakalářská práce

Autor: Jan Rydlo
Studijní obor: Aplikovaná informatika

Vedoucí práce: Mgr. Josef Horálek, Ph.D.

Hradec Králové Srpen 2016

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a s použitím uvedených zdrojů.

V Hradci Králové dne 05.08.2016 Jan Rydlo

Poděkování:

Tímto bych rád poděkoval vedoucímu mé bakalářské práce Mgr. Josefu Horálkovi, Ph.D. za metodické vedení práce a odborné rady a názory ohledně zkoumané oblasti penetračního testování a bezdrátových sítí.

Anotace

Název: Využití penetračního testování v bezdrátových sítích

Bakalářská práce se zabývá problematikou penetračního testování a hlavně jeho využití při testování bezdrátových sítí. První oddíl teoretické části se zabývá obecnými principy penetračního testování, jsou zde vymezeny základní pojmy a představeny některé rámce používané při tomto testování. V druhém oddílu jsou představeny nástroje, které je možné k testování zabezpečení sítí využít, a to s důrazem na ty, které budou následně využity v praktické části práce. V posledním oddílu teoretické části je nastíněn princip fungování bezdrátových sítí, dále je zde navrženo využití postupů pro jejich penetrační testování. V praktické části je tento návrh převeden do reálného použití za pomoci nástrojů představených v teoretické části. Bakalářská práce je zakončena shrnutím výsledků praktické části.

Annotation

Title: Usage of penetration testing in wireless networks

The Bachelor thesis covers the issue of penetration testing, mainly its usage in wireless networks. The first section of the theoretical part is overview of general penetration testing methodology, the main concepts and some general frameworks. The second section is an introduction of tools which are used for penetration tests with the main focus on tools planned for use in the practical part. The final section of the theoretical part is twofold, it introduces concepts of wireless networks and suggests framework for practical part. In the practical part is suggested framework used for execution of penetration test using tools introduced in theoretical section. In the final part of the Bachelor thesis are presented the results of the case of study.

Obsah

1 Úvod

Jelikož se v poslední době obrovským způsobem rozmáhá trend využívání bezdrátových sítí jak na veřejných místech, v běžných domácnostech a menších podnicích, tak i ve velkých firemních sítích, je cílem této bakalářské práce představení a důkladné prozkoumání způsobů a typů jejich zabezpečení a poukázání na případná rizika, která při použití tohoto typu sítí reálně hrozí nebo mohou vzniknout při jejich nesprávném nastavení nebo používání. Tato rizika se pokusím prozkoumat z mnoha různých úhlů pohledu, od použitého zabezpečení, přes chyby, které se mohou vyskytnout při nesprávném nebo neúplném nastavení sítě, až po využití klientů, kteří se k dané síti připojují nebo jsou k ní již připojeni. Tato práce by měla sloužit hlavně uživatelům bezdrátových sítí, kteří by si díky ní mohli plně uvědomit, jaké bezpečnostní problémy jim při jejich používání hrozí a jak se jim mohou případně vyhnout. Dalším účelem této práce je představit uživatelům nástroje a postupy, které mohou využít k vlastnímu otestování a případnému zlepšení úrovně bezpečnosti své sítě.

V teoretické části práce budou představeny principy penetračního testování s hlavním zaměřením na postupy, které jsou využívány při testování bezdrátových sítí. Budou zde definovány základní pojmy a představeny některé frameworky využívané v této oblasti. Dále zde budou popsány dostupné nástroje, které jsou nebo mohou být k penetračnímu testování využity. Poslední kapitolou teoretické části by měl být návrh využití dříve představených nástrojů a postupů pro penetrační test, který bude následně převedený do reálného testu v praktické části práce.

Test zabezpečení v praktické části bude proveden proti testovací síti. K aplikaci navržených postupů bude použita specializovaná Linuxová distribuce Kali Linux a nástroje, které jsou v ní dostupné. Výstupem praktické části by mělo být posouzení úrovně bezpečnosti všech momentálně využívaných bezpečnostních protokolů a postupů, jak pro domácí sítě, tak i pro sítě využívající zabezpečení typu enterprise s RADIUS serverem jako bezpečnostní autoritou. V závěru bych poté rád ukázal využití nástrojů Nmaps a Nessus pro získání informací o testované síti a vyhledání existujících bezpečnostních chyb na straně klientů dané sítě. Následně bych se rád pokusil některé z bezpečnostních chyb zneužít, a to za pomoci specializovaného frameworku Metasploit.

2 Literární rešerše

Ke zpracování bakalářské práce byla použita nejen literatura zabývající se výhradně přístupy pro penetrační testování v bezdrátových sítích, ale i literatura pokrývající problematiku bezpečnostního testování a k tomu hojně využívaného etického hackingu v širších souvislostech a spojitostech. To mi pomohlo obsáhnout zkoumanou problematiku z více úhlů pohledu a uchopit danou část počítačové bezpečnosti z co nejširšího spektra bez toho, abych se výrazně odchýlil od hlavního zaměření práce, a to penetračního testování v bezdrátových sítích.

Teoretická část se tedy snaží uchopit problematiku penetračního testování a etického hackingu z co nejširší a nejobecnější perspektivy, a tím zasadit testy prováděné v praktické části práce do co největších souvislostí. Pravděpodobně nejrozsáhlejší knihou použitou v tomto oddílu je Ethical Hacking and Penetration Testing Guide [1] od pakistanského experta pohybujícího se v oblasti počitačové bezpečnosti a jednoho z nejúspěšnějších a nejznámějších etických hackerů Rafaye Balocha. Tato publikace se dané problematice věnuje velmi podrobně a snaží se představit doménu penetračního testování v teoretické rovině z mnoha úhlů pohledu. Konkrétně pro účely této práce mi kniha pomohla s vymezením základních pojmů vyskytující se v zkoumané oblasti a informace v ní osažené mi posloužily k představení postupů běžně doporučovaných a používaných k zajištění bezpečnosti IT systémů metodami penetračních testů. Další knihou použitou pro teoretický základ práce je kniha nesoucí název Kali Linux: Assuring security by penetration testing [2] od tria autorů Leeho Allena, Tediho Heyrianta a Shakeela Aliho. Všichni tři autoři jsou vysoce kvalifikovanými odborníky v oboru počítačové bezpečnosti s hlavním zaměřením na zabezpečení počítačových sítí. Informace načerpané z této knihy byly následně použity převážně k představení a definování rámcových postupů, které je možné využít při obecném penetračním testování v počítačových sítích nebo i ostatních informačních systémech vyskytujících se ve firemním prostředí. V druhé řadě jsem z ní byl schopen načerpat mnoho užitečných faktů a informací ohledně specializované Linuxové distribuce Kali a hlavně o jejím využití při provádění penetračních testů.

Pro informace zaměřující se specificky na bezdrátové sítě a bezpečnostní testy v nich prováděné jsem sáhl po publikaci Kali Linux Wireless Penetration Testing Essentials: Community Experience Distilled [3] od italského autora Marka Alamanniho. Tento autor je vysoce uznávaným a zkušeným administrátorem Linuxových systémů a odborníkem na

zajišťování bezpečnosti v oblasti IT systémů. Konkrétní užití znalostí čerpaných z této knihy je možné vidět v kapitolách věnovaných jak teoretickému, tak následně i praktickému testování prováděnému v bezdrátových sítích. Dále byla tato publikace hlavním zdrojem pro čerpání informací o nástrojích dostupných v distribuci Kali Linux a použitelných pro testování v oblasti bezdrátových sítí. Velmi zajímavým a užitečným zdrojem informací ze zkoumané oblasti pro mě byla kniha Kali Linux Wireless Penetration Testing: Beginner\'s Guide [3] od dua autorů Vivek Ramachandran a Cameron Buchanan. Tato publikace byla pro práci velmi přínosným zdrojem informací o bezpečnostních rizicích vyskytujících se při používání v bezdrátových sítích a o možnostech jejich využití k proniknutí do těchto sítí, tyto znalosti byly následně využity převážně při tvorbě teoretického návrhu penetračního testu. Praktické ukázky z této knihy byly dále zdrojem inspirace pro testy provedené v praktickém oddílu této bakalářské práce. Základní teoretické znalosti a nápady pro praktikování pokročilejších testovacích technik při práci v prostředí bezdrátových sítí a využití vysoce specializovaných nástrojů dostupných pro toto testování byly převzaty hlavně z knihy Mastering Wireless Penetration Testing for Highly Secured Environments[25] od amerického autora Aarona Johnse, který je expertem na poli zajištění bezpečnosti pro klienty převážně z business sféry.

Jako další zdroj informací z daného odvětví dále připadala v úvahu publikace The Basics of Hacking and Penetration Testing, Second Edition: Ethical Hacking and Penetration Testing Made Easy od Patricka Engebretsona, která se opět věnuje penetračnímu testování z obecnějšího a širšího hlediska, avšak z důvodu redundantnosti poskytovaných informací nebo neaktuálnosti některých částí tato publikace nebyla nakonec pro tvorbu práce využita. Pro bližší informace o představených nebo použitých nástrojích byly využity oficiální stránky autorů daného nástroje, případně oficiální dokumentace linuxové distribuce Kali. Z důvodu zachování aktuálnosti práce byly informace z výše uvedených publikací ověřovány v internetových zdrojích.

V praktické i teoretické části práce je podrobně představen soubor nástrojů Metasploit Framework, pro zisk a ucelení informací o tomto frameworku byla z velké části použita publikace Metasploit:Penetration Tester´s Guide [26], za kterou stojí čtveřice autorů David Kennedy, Jim O'Gorman, Devon Kearns, and Mati Aharoni. Tato publikace podrobně představuje možnosti využití Metasploit frameworku jak při testování klasických LAN sítí, tak i při testech sítí bezdrátových a mnoho dalších využití včetně testů webových serverů a aplikací. Důležitým přínosem této práce byly ukázky využití nástrojů třetích stran pro

rozšíření základní funkčnosti daného frameworku. Poslední knihou, kterou bych rád zmínil, je publikace s názvem Learning Nessus For Penetration Testing[27] sepsaná Himanshu Kumarem, která mi velmi ulehčila práci s velmi rozsáhlým nástrojem Nessus Security Scanner, a to hlavně díky jejímu širokému rozsahu, který je ale podáván ve velmi srozumitelné formě.

3 Principy penetračního testování

Cílem této kapitoly je definovat základní pojmy používané při penetračním testování, a to jak v cílové oblasti bezdrátových, tak i v oblasti klasických ethernetových sítí. V druhé části budou představeny metody, které jsou k penetračnímu testování využívány, a základní frameworky do kterých jsou tyto metody uskupovány.

3.1 Vymezení základních pojmů

Etický hacking

Jedná se o soubor postupů, metod a principů hloubkového testování informačních systémů za pomoci simulací útoků na systém, při kterých jsou používány metody, které by mohly být využity potenciálním útočníkem. Toto testování provádí bezpečnostní expert (viz. Etický hacker) při plném vědomí nebo na žádost vlastníka systému, je předem známý čas i rozsah útoku na systém. Toto testování má za účel odhalit závažná i potenciální bezpečnostní rizika systému. Ve většině případů hacker předá vlastníkovi informačního systému zprávu o průběhu testování i s návrhem řešení problému a změn v systému. Baloch Rafay [1, s. 2].

Etický hacker

Rafay Baloch ve své knize Ethical hacking and penetration testing guide [1, s. 2] definuje etického hackera jako člověka, který má povolení nebo je pověřen organizací k provedení útoku na systém za účelem objevení bezpečnostních chyb, které by mohly být využity potenciálním útočníkem k ohrožení daného systému.

Penetrační testování

Dle Allena Leeho [2, s. 51] se jedná o soubor postupů, pomocí nichž je prováděna hloubková bezpečnostní diagnostika informačního systému bez toho, aby vlastník věděl, kdy a v jakém rozsahu testování proběhne. V tomto souboru jsou definována pravidla, postupy a příklady, které jsou v praxi prověřeny a mohou být prováděny během jakéhokoliv bezpečnostního testování, tak aby bylo dosaženo maximálního možného zabezpečení sítě, systému nebo aplikace. Penetrační testování může být prováděno buď jednorázově, nebo jako součást bezpečnostní politiky společnosti. Penetrační testování je nejagresivnější možnost testování systému, musí být prováděno bezpečnostním expertem se zkušenostmi a schopnostmi adaptovat se na jakýkoliv systém. Výstupem tohoto testování bývá obvykle vícestupňový report, kde jsou zaznamenána všechna bezpečnostní rizika daného prostředí a návrh řešení

jednotlivých problémů nebo změn v síti. Baloch ve své knize Ethical hacking and penetration testing guide [1, s. 2] označuje penetrační testování jako podmnožinu nebo jednu z podob etického hackingu.

3.2 Typy testování informačních systémů

Allan Lee [2, s. 52] říká, že penetrační testování se nejčastěji dělí na dva typy uvedené níže.

Black box testing

Ve své knize Kali Linux: Assuring security by penetration testing [2, s. 52] jej Lee definuje jako testování systému, kdy bezpečnostní expert (potenciální útočník) nezná předem vnitřní strukturu systému a technologie používané ve vnitřních částech systému. Je prováděno za pomoci technik používaných reálnými útočníky a pomocí důkladného a organizovaného testování. Při dodržení těchto postupů mohou být nalezeny chyby a potenciálně nebezpečné části systému. Je nutné, aby bezpečnostní expert vyhodnotil možnou nebezpečnost každé chyby. Poté jsou tyto chyby zaznamenány a výstup je předán vedení společnosti. Tento způsob testování může být mnohdy velmi nákladný.

White box testing

Tento typ je ve stejné knize, tedy Kali Linux: Assuring security by penetration testing [2, s. 53] popisován jako princip penetračního testování, při kterém expert (útočník) kompletně zná vnitřní i vnější struktury testovaného systému. Díky tomu je toto testování velmi přesné a zkušenému expertovi umožňuje při co nejmenším úsilí nalézt téměř všechny bezpečnostní trhliny v systému. Tento způsob testování je pro většinu zákazníků mnohem efektivnější, protože umožňuje opravit chyby systému od základů a tím zamezit většině venkovních útoků. Výstup tohoto testování je prakticky stejný jako u Black box testování, ale je jednodušší na provedení, a proto je ho možné provádět v rámci celkových bezpečnostních testů systému. Dle Rafaye Balocha [1, s. 7] jsou nejčastěji podávané informace tyto: typy a verze aplikací, operační systémy, u webových aplikací jsou dodány i zdrojové kódy.

Rafay Baloch [1, s. 7] dodává, že je možno uvažovat i o třetím přístupu, a to přístupu uvedeném níže.

Grey box testing

Ten Baloch [1, s. 7] popisuje jako testování, při kterém jsou hackerovi poskytnuty pouze některé informace. Dle autora [1, s. 7] jsou poskytovány nejčastěji jména aplikací běžících za danou IP adresou. Nejsou však specifikovány verze aplikací. U webových aplikací jsou poskytnuty i některé další informace, jako jsou testovací účty, typ databáze, serveru, atd.

3.3 Rámce pro penetrační testování

Aby bylo možné testovat rozsáhlé a většinou velmi složité systémy, je podle Leeho [2, s. 54] třeba použít některého z ustálených postupů penetračního testování. Autor[2, s. 55] uvádí jako nejběžnější postupy ty níže uvedené.

Open Source Security Testing Methodology Manual (OSSTMM)

Jedná se o mezinárodně uznávaný přístup, který byl sepsán Petem Herzogem a je vyvíjen skupinou ISECOM. A jak sám autor [2, s. 56] uvádí, v dnešní době je používán mnoha společnostmi pro jejich regulérní bezpečnostní testy. Jak se uvádí na oficiálním webu projektu[12], po technické stránce je postup rozdělen na čtyři klíčové části – scope, channel, index a vector. Tyto části jsou popisovány v knize Kali Linux: Assuring security by penetration testing [2, s. 56] následovně. Scope je proces sbírání informací o všech prvcích, které operují v testovaném prostředí. Channel určuje typ komunikace a interakce mezi danými prvky. Na každém z těchto channelů se nacházejí jedinečné bezpečnostní prvky a principy, které je nutné testovat a ověřovat v průběhu testování. Každý z těchto prvků zahrnuje jak fyzické zařízení a média, tak i psychologické pochody jejich uživatelů. Index je ohodnocení typu přístupu a autentizace uživatelů u daného prvku (zda se ověřuje na základě MAC adresy, IP adresy, ověření proti AP…). Vector zkoumá směry, ze kterých může uživatel přistupovat k danému prvku. Tato metoda se dá velmi lehce přizpůsobit pro konkrétní potřeby dané společnosti. Dle pana Leeho [2, s. 56-57] je v současné době uznáváno šest základních podob bezpečnostního testování, které autor popisuje následovně:

- **Blind testing -** není třeba znalost cílového systému, cíl je předem informován o času a velikosti testu. Příkladem je etický hacking nebo war games. Toto testování je také velmi dobře přijímáno z etického pohledu, jelikož cíl vždy předem ví, kdy a k jakému testování dojde.
- **Double Blind testing** - u tohoto testování není třeba znalost cílového systému, na druhou stranu cíl není informován o tom, kdy a jak útok proběhne. Příklady tohoto

testování jsou Black box auditing a Penetration testing. Většina dnešních testů je prováděna pomocí těchto metod, jelikož se nejvíce podobají reálnému nebezpečí.

- **Gray box -** zde jsou předem známy některé informace o cílovém systému a cíl je předem informován a čase a velikosti testu. Příkladem je Vulnerability assesment.
- **Double gray box** - je podobný gray box testu, ale nejsou při něm testovány channels a indexes. Příkladem je White box audit.
- **Tandem** - je předem známo pouze minimum informací o daném systému a majitel je předem informován o průběhu testu. Příkladem je Crystal box a In-house audit.
- **Reversal** - u tohoto testování jsou známy všechny podrobnosti o systému a majitel systému nebude nikdy informován o tom, že test proběhl.

Information Systems Security Assessment Framework (ISSAF)

Jedná se o open source framework pro bezpečnostní testování a analýzu. Dle Leeho [2, s. 58-59] je základem rozdělení testování do několika logických domén. Spojováním výsledků testů jednotlivých domén se dojde k celkovému výsledku testu systému. Takto se docílí logicky uspořádaného a velmi důkladného průběhu testování. Dle webu CiscoPress [17] se dělí na dvě části, na penetrační testování a na personální zabezpečení. Lee [2, s. 58] popisuje tyto části následovně. Penetrační testování se týká testování síťových zařízení, médií, softwaru, protokolů a ostatních technických částí systému. Na druhou stranu personální testování se zaměřuje na dodržování bezpečnostních postupů a pravidel uživateli a správci systému. Podle stejného autora, tedy Allana Leeho [2, s. 59] se v dnešní době tyto metody používají spíše k zavedení a kontrole bezpečnostních pravidel a postupů ve firmách, než k přímému testování bezpečnosti systému, což je způsobeno mnohem větší složitostí implementace testu než u předchozí metody.

Open Web Application Security Project (OWASP)

Dle stránek CiscoPress[17] vznikla tato metodologie z důvodu obrovského množství velmi špatných zvyklostí při psaní webových aplikací, které vedou k dlouhodobým a velmi závažným bezpečnostním rizikům pro uživatele těchto aplikací. Lee ve své knize Kali Linux: Assuring security by penetration testing [2, s. 60] uvádí, že se jedná o prezentaci deseti nejlepších projektů open source komunity v oblasti bezpečnostního testování, pomocí nichž se snaží komunita upozornit na hrozící bezpečnostní rizika. Tyto projekty se mění každý rok a tím reagují na nejnovější bezpečnostní rizika. V rámci OWASP je také možno najít podrobnou příručku, která je podle autorů[15] použitelná k otestování vlastního projektu.

Allan Lee v knize Kali Linux: Assuring security by penetration testing [2, s. 60] dále uvádí, že OWASP se snaží řadit bezpečnostní rizika podle jejich rizikovosti v závislosti na technologickém a obchodním riziku. U každé hrozby je také přiložen návod, jak tuto hrozbu otestovat, případně jaké bezpečnostní postupy implementovat a tím se proti ní bránit.

Web Application Security Consortium Threat Classification (WASC-TC)

Jedná se o podobný projekt jako je OWASP, ale zaměřený na webové aplikace. Jak říkají sami autoři [18], jedná se o neziskovou organizaci, sdružující experty v oblasti internetové bezpečnosti která se nesnaží ukázat aktuální hrozby, ale spíše se snaží vysvětlit jejich technické a myšlenkové principy a tím odborníky nutí naučit se přemýšlet nad způsoby, jak může nebezpečí vzniknout a jak mu předcházet. Projekt je zaměřen zejména na tři oblasti, při jejich správném pochopení by mělo dojít k výraznému zlepšení webové bezpečnosti webové aplikace. Lee se ve své knize [2, s. 61] snaží tyto postupy řadit do tří kategorií:

- **Enumeration view** – zde je ukázáno několik základních útoků na webové aplikace včetně jejich implementace na různých platformách a jejich nedokonalostí.
- **Development view** – zde je ukázáno, jaké nejčastější bezpečnostní chyby se objevují a v jaké části vývoje softwaru je třeba dávat si na ně pozor.
- **Taxonomy cross-reference view** – zde je ukázáno několik bezpečnostních standardů, které se nejčastěji používají u webových aplikací, a také možnosti jak jeden standard rozšířit, aby pokrýval více oblastí.

Penetration Testing Execution Standard

Jak píší autoři na oficiálním webu standardu [10], jedná se o standardizovaný postup při provádění penetračního testování. Byl vytvořen při spolupráci s největšími profesionály v tomto oboru. Dále je na tomto webu [10] uvedeno, že se tento standard bude skládat se ze sedmi fází, které pokryjí penetrační test od přípravy až po vytvoření finálního reportu. Při jejich dodržení by mělo být zaručeno efektivní a přesné penetrační testování. Tyto fáze jsou na jejich webu [10] uvedeny následovně:

- **Pre-engagement interactions** – příprava před samotným testováním, stanovení předběžných cílů, předpokládané doby testování a očekávaných výsledků.
- **Intelligence gathering** – fáze, ve které dochází ke sběru informací o cílovém systému, které budou později použity jako zdroj k uskutečnění pozdějšího útoku.

Tyto informace mohou být získány pomocí automatizovaných programů, pomocí social engineeringu nebo ve spolupráci s experty v dané oblasti.

- **Threat modeling** – pokus o vymodelování útoku. Jsou analyzovány předem známé vlastnosti systému a možnosti útočníka a dle toho je vybrán jeden ze standardizovaných modelů penetračního testování.
- **Vulnerability analysis** - v této fázi se analyzují veškeré informace, které jsou k dispozici, a je zde snaha o definování potenciálních slabin systému, které by se daly využít k samotné penetraci. Je zde také plánován samotný útok, je vytvořena mapa systému s nejdůležitějšími body, ze kterých je možno získat cenné informace.
- **Exploitation** – pokud byla předchozí fáze dobře provedena, dochází k samotnému proniknutí do systému za pomoci využití některé z chyb objevených v předchozí fázi. Toto proniknutí se většinou provádí za pomoci škodlivého kódu (viru), proces injection, falšování adres atd.
- **Post-exploitation** – snaha o vytvoření trvalého a znovupoužitelného přístupu do systému, ke kterému není potřeba provádět novou penetraci.
- **Reporting** – vytvoření zprávy z průběhu testu, kde budou zaznamenány veškeré provedené úkony, a odevzdání zprávy vedení firmy. Tato zpráva bude později použita k opravení a lepšímu zabezpečení systému.

General penetration testing Framework

Jak píše Lee ve své knize Kali Linux: Assuring security by penetration testing [2, s. 64], tato metoda vznikla jako reakce na velké množství postupů a principů při provádění penetračního testování s použitím specializovaného systému Kali Linux. Snaží se tyto postupy standardizovat a tím předejít chybnému nebo neúplnému průběhu testování a v důsledku toho zlepšit a zpřesnit i jeho výsledky. Proto byl vydán soubor obecných postupů, při jejichž dodržení by mělo jakékoliv penetrační testování vrátit úplné a pravdivé výsledky. Tyto postupy jsou ve stejné knize [2, s. 66-68] popisovány následovně:

- **Target scoping** – předpříprava samotného testování. Je nutno určit si typ cílového systému, způsoby testování, čas testování, rozsah testování, předpokládaný výsledek.
- **Information gathering** – v této fázi se tester snaží najít si co nejvíce informací o cílovém systému, využívá k tomu veřejně dostupné zdroje, jako jsou diskuzní fóra, sociální sítě, webové stránky nebo blogy. Dalšími zdroji dat jsou vyhledávací služby jako je Google, Yahoo! nebo Bing. Pomocí specializovaného softwaru se dá z těchto

vyhledávacích služeb vytáhnout mnoho informací o cílovém systému. Čím více informací se podaří získat během této fáze, tím větší je šance na úspěch celého útoku.

- **Target discovery** – v této fázi přichází snaha o zjištění co nejvíce informací o vnitřní struktuře cílového systému. Zjišťuje se například, jaké prvky se v síti nacházejí, množství prvků, jaké technologie jsou použity, jaký software je nainstalovaný, jaké protokoly jsou zapnuty.
- **Enumerating target** – v této fázi je snaha o nalezení otevřených portů v cílovém systému. Pomocí těchto portů je možné spustit služby, které nám podají podrobnější informace o struktuře systému i přesto, že se systém nachází za bránou firewall nebo bez záznamu v bezpečnostních prvcích systému.
- **Vulnerability mapping** - v této fázi je snaha o zmapování potencionálních bezpečnostních problémů z informací, které jsme zjistili v předchozích fázích, a pomocí některého z těchto problému získat přístup do sítě. Toto mapování je ve většině případů prováděno kombinací manuálního prohledávání s automatizovaným softwarem.
- **Social engineering** - pokud nebylo možné nalézt bezpečnostní chybu v technické části systému, další možností je využití lidských chyb. Například pomocí phishingových e-mailů, falešných telefonátů nebo dalších postupů, které mohou využít lidskou chybu.
- **Target exploitation - v** této fázi dochází k samotnému využití chyb, které jsme předem nalezli. Zde také dochází k proniknutí do systému.
- **Privilege escalation** - pokud je proniknutí do systému úspěšné, je pravděpodobné, že se podařilo pouze v roli řádového uživatele. V této fázi je snaha o nalezení chyb uvnitř systému, pomocí nichž by bylo možnost zvýšit úroveň uživatelských práv a tím převzít úplnou kontrolu nad systémem.
- **Maintaining access** - v této fázi se útočník snaží vytvořit možnost trvalého a opakovatelného přístupu do systému bez nutnosti provedení nového penetračního testování.
- **Documentation and reporting** - Po dokončení testování je třeba vytvořit a zpracovat dokumentaci o provedení penetračního testování, zde by měly být zapsány všechny postupy, které byly provedeny, a výsledky, kterých bylo docíleno. Tato dokumentace je následně odevzdána majiteli systému a měla by být použita pro lokaci a odstranění bezpečnostních rizik.

4 Analýza dostupných nástrojů

Jak již název kapitoly napovídá, cílem této kapitoly je představit nástroje, které jsou nebo mohou být využívány k penetračnímu testování bezdrátových sítí. Hlavní důraz je kladen na nástroje, které budou později využity v praktické části této práce.

4.1 Sledování a odposlech síťového provozu

Wireshark

Dle stránek výrobce [10] se jedná o světového leadera v oblasti analýzy síťových protokolů. Na těchto stránkách se doslova říká, že Wireshark nám dovoluje pozorovat naši síť pod mikroskopem. Tento software se stal definujícím standardem v mnoha průmyslových odvětvích i ve školských institucích. Jako hlavní výhody jsou uvedeny:

- Možnost zkoumat stovky různých síťových protokolů s postupnou aktualizací pro nově se objevující protokoly a standardy.
- Možnost zachytávání a analýza reálného provozu nebo zkoumání síťového provozu zaznamenaného v souboru.
- Multiplattformost: existuje verze pro Windows, Linux, OS X, Solaris, FreeBSD a mnoho dalších.
- Nejlepší možnost třídění síťového provozu na základě filtru, která je na trhu dostupná.
- Podpora pro dešifrování packetů mnoha protokolů včetně IPsec, Kerberos, WEP, WPA/WPA2.

Tshark

Jedná se o verzi softwaru Warshark, která je uzpůsobena pro použití výhradně v příkazové řádce operačního systému Linux.

Kismet

Podle Alammaniho [3, s.35] se jedná o velmi silný nástroj v oblasti pasivního skenování provozu bezdrátových sítí. Kismet má ale i využití jako program pro analýzu rámců nebo jako nástroj pro odhalení průniku do sítě. Nástroj se skládá ze dvou hlavních kompenent – kismet_server a kismet_client. Kismet_server se stará o zachytávání, dekódování a ukládání rámců, na druhou stranu kismet_client se stará o předkládání zachycených informací uživateli.

Wifi Honey

Jak uvádějí autoři na svých stránkách [5], jedná se o nástroj, který ulehčuje zjištění typu zabezpečení sítě, pro niž vysílá klient probe request. Wifi Honey tento proces ulehčuje tím, že nastaví pět virtuálních interface do monitorovacího módu, na nichž spustí virtuální Access Point pro každý typ zabezpečení bezdrátové sítě (tzn. žádné zabezpečení, WEP zabezpečení, WPA a WPA2). Navíc tento nástroj automaticky otevře i okno s nástrojem airodump, což nám umožní odchytit první dva packety z four-way handshake pro případný pokus o slovníkový útok na danou síť. Toto vše v přehledném uživatelském rozhraní.

Nmap

Jedná se o opensource nástroj, který je předinstalovaný v základní verzi operačního systému Kali Linux. Tento nástroje slouží ke kompletnímu prozkoumání sítě a je používaný k zajištění jejího bezpečnostního auditu. Obrovskou výhodou nástroje Nmap je jeho všestrannost, nástroj dokáže získat informace o otevřených portech na mnoha protokolech a také zobrazit služby, které na těchto portech naslouchají. V neposlední řadě je tento nástroj schopný interpretovat informace o operačních systémech na klientských počítačích nebo pomocí svých skriptů vyspat uživatelské účty užívané na sledovaném PC.

4.2 Nástroje pro zneužití slabin sítě

Cowpatty

Podle popisu autorů aplikace [7] se jedná o program, který slouží k provedení slovníkového útoku proti WPA/WPA2-PSK sítím. Hlavní výhodou tohoto programu je implementace zrychleného útoku, pokud je k dispozici předpřipravený PMK soubor pro dané BSSID.

Reaver

Jak se uvádí v dokumentaci ke Kali Linux distribuci [16], jedná se o impelementaci brute-force útoku na zařízení podporující standard WPS. Reaver byl navrhnutý tak, aby fungoval obecně proti většině typů zařízení s WPS, a byl také úspěšně otestován proti většině těchto routerů. V běžném případě je Reaver schopen vrátit WPA-WPA2 heslo v průběhu čtyři až deset hodin.

Bully

Podle dokumentace ke Kali Linux distribuci [6] se jedná o novou implementaci algoritmu pro prolomení WPS zabezpečení za pomoci zneužití dobře známé bezpečnostní chyby. Od

dříve zmíněného Reaveru se liší hlavně sníženou náročností na paměť a procesor, menším počtem závislostí a lepším řešením chybových stavů.

DnsChef

Jak je uvedeno v dokumentaci ke Kali Linux [8], jedná se o vysoce konfigurovatelný DNS proxy server (jinak řečeno falešný DNS server), který umožňuje zasílat upravené DNS requesty a tím pádem například přesměrovat všechen síťový provoz na jednu doménu. Jelikož byl tento program primárně vyvinut pro penetrační testování, nabízí mnohem více možností nastavení než ostatní nástroje v této oblasti. Funkcemi programu jsou například podpora mnoha typů DNS záznamů, spojování domén s jejich wildcards, konfigurace pomocí externího souboru, plná podpora pro IPv6 a mnoho dalšího.

4.3 Sady programů pro kompletní penetrační test

Aircrack-ng suite

Jak se píše v dokumentaci [13], Aircrack-ng suite je kompletní sada nástrojů k testování bezdrátových sítí. Vývoj nástrojů začal v únoru 2006 a v dnešní době se jedná o uznávaný standard v této oblasti. Všechny nástroje jsou uzpůsobeny práci v příkazovém řádku, proto je zde možnost vysoké kustomizace. Pokud by z nějakého důvodu nevyhovovalo rozhraní příkazové řádky, je možné zvolit některou z GUI implementací. Nástroje jsou primárně určeny pro systémy typu Linux, ale je možné je provozovat i na dalších operačních systémech jako je Windows, OSX nebo FreeBSD. Některé z nástrojů tohoto balíku jsou představeny v následujících kapitolách.

Airmon-ng

Skript sloužící k zapnutí nebo vypnutí monitorovacího modu na wi-fi interface síťové karty a případně k řešení problémů s procesy, které ho blokují. Dále si pomocí něho můžeme zobrazit informace o daném rozhraní nebo si dané rozhraní nastavit pouze pro určitý kanál.

Airodump-ng

Nástroj sloužící k zachytávání packetů, který je vhodný ke shromažďování IVS při testování WEP sítí. Pokud je na interface povolená GPS lokace, je program schopný zachytit i GPS souřadnice daného routeru. Navíc dokáže program zachytit a uložit informace o všech klientech a routerech, které jsou mu viditelné. K jeho práci je nutné mít bezdrátové interface v monitorovacím modu.

Airdrop-ng

Program sloužící k odpojení klienta od daného routeru pomocí zaslání deauthentication paketu. Airdrop se může zaměřit na konkrétního klienta dle MAC adresy na klienty dle typu hardware nebo může odpojit všechna zařízení v dosahu.

Aireplay-ng

Nástroj sloužící pro vytváření vlastního síťového provozu, díky němuž je možné získat více informací pro pozdější prolomení WEP nebo WPA klíčů. Program umožňuje několik typů akcí pro získání síťového provozu, příkladem jsou Fragmentation attack, Cafe-Latte attack, ARP request replay, KoreK chopchop attack a další.

Packetforge-ng

Program k vytváření šifrovaných packetů, které mohou být používány k injection útokům. Nástroj umí vytvářet různé typy packetů jako je ARP request, UDP a ICMP pakety nebo dovoluje vytvořit i zcela vlastní typ paketu.

Airbase-ng

Jedná se o všestranný nástroj, který umožňuje provádět různé typy útoků oproti klientům nebo se sám chovat jako access point. Hlavní myšlenkou tohoto nástroje je přinutit klienta, aby se k tomuto falešnému routeru připojil, přičemž by si stále myslel, že se pojí ke správnému routeru. Toto je možné uskutečnit pomocí reakce na probe requesty, které vysílají některá zařízení.

Wifite

Dle dokumentace Kali Linux [21] se jedná o balíček programů pro automatizovaný útok na několik WEP, WPA a WPS sítí v řadě. Tento balíček automatizuje činnost programů z balíku Aircrack-ng suite. Výhody tohoto balíku jsou například: seřazení sítí a útoků dle kvality signálu, možnost nastavení útoků pomocí mnoha filtrů, možnost rychlého zrušení útoku pomocí ctrl+c a další.

FernWifiCracker

Dalším z automatizovaných nástrojů pro penetrační testování bezdrátových sítí je FernWifiCracker. Jak je uvedeno v dokumentaci ke Kali Linux [9], jedná se o balík programů, který je schopný získat klíče k WEP/WPA/WPS sítím a je také schopný provést

některé z útoků proti těmto sítím. Balík je schopný, penetrovat WEP síť za pomoci Fragmentation, Chop-Chop, Caffe-Latte, Hirte a dalších útoků, dále penetrovat WPA/WPA2 pomocí slovníkových útoků, provést Man-in-the-Middle attack a mnoho dalšího.

4.4 Distribuce určené pro penetrační testování

Kali linux

Jak uvádí oficiální dokumentace této distribuce [19], jedná se o Linuxový systém založený na Debianu a jeho účelem je podpora pokročilého penetračního testování a bezpečnostních kontrol. Kali obsahuje několik stovek nástrojů pro podporu bezpečnostních činností v informačních technologiích jako jsou penetrační testy, forenzní testování, reverse engineering a mnoho dalšího. Kali je vyvíjen, financován a udržován skupinou Offensive security a je momentálním leaderem ve své oblasti. První verze této Linuxové distribuce byla vydána již 13.4.2013 jako kompletní předělávka starší distribuce BackTrack Linux přizpůsobená vývojovým standardům pro Debian systémy.

Metasploit framework

Jak uvádí výrobce ve své oficiální dokumentaci [14], jedná se o kompletní platformu určenou k penetračnímu testování. Tato platforma poskytuje nástroje potřebné k nalezení slabin cílové sítě, jejich zneužití a následnému opravení. Tento projekt je open-source a je možné ho používat jak na Linuxu, pro který byl vytvořen, tak i na platformách Windows a OSX. Navíc pro efektivnější práci s tímto frameworkem existuje interaktivní grafické rozhraní vytvořené Raphaelem Mudgem nesoucí název Armitage.

Nessus Security Scanner

Aaron Johns ve své knize Mastering Wireless Penetration Testing for Highly – Secure Enverionment [25] tento nástroj popisuje jako scanner, který poskytuje opravy, nastavení a kompletní bezpečnostní audit. Nástroj podporuje odhalování malwaru a botnetů, identifikaci citlivých údajů a mnoho dalších možností pro testování zabezpečení. Autor také tvrdí, že díky pravidelným aktualizacím, více než 60 000 doplňkům a podpoře pro práci v týmu je to nejlepší bezpečnostní scanner, který kdy použijete. Nessus dokáže odhalit bezpečnostní chyby routeru, otevřené porty a tyto chyby poté přehledně zobrazit společně s linky na bližší informace o bezpečnostním riziku. Navíc jak píše Kumar ve své knize Learning Nessus for Penetration testing[27, s.12] jedná se o jeden z nejpopulárnějších nástrojů, který je v oboru používaný po více než 15 let.

5 Návrh využití principů penetračního testování pro bezdrátové sítě

V této části bude podrobně představen standard 802.11 a ostatní charakteristiky týkající se stavby a hlavně bezpečnostních charakteristik bezdrátových sítí. V druhé části bude představen plán pro otestování bezdrátové sítě dle obecného frameworku používaného při penetračním testování Wi-Fi sítí.

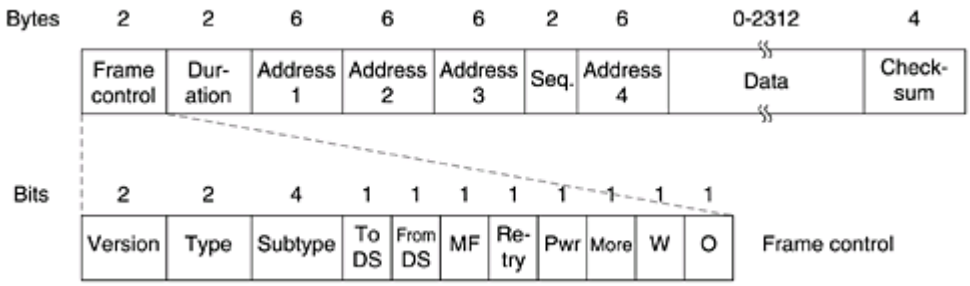
5.1 Bezdrátové sítě a jejich charakteristiky

Standard 802.11

Jedná se o standard pro bezdrátové sítě pracující na druhé vrstvě OSI modelu pro počítačové sítě. Zařízení a sítě používající tento standard se označují jako Wi-Fi, což je ochranná známka registrovaná skupinou Wi-Fi Alliance. V dnešní době je standard 802.11 postupně rozšiřován tak, aby vyhovoval požadavkům na moderní počítačové sítě a hlavně jejich přenosové rychlosti. Proto dnes rozlišujeme typy 802.11.a, 802.11b, 802.11g a 802.11n. Tyto typy se odlišují hlavně frekvencemi, na kterých jsou schopny vysílat signál. Dnes nejpoužívanější typ 802.11b/g vysílá na frekvenci 2,4GHz. Vysílací frekvence jsou u bezdrátových sítí dále děleny na čtrnáct kanálů. Ve většině případů však není povoleno používat všechny tyto kanály, vysílání je omezeno pouze na specifický výběr. Jelikož se jedná o vysílání signálu, je dosah Wi-Fi sítě značně omezen prostředím ve kterém se síť nachází. Typicky udávanými hodnotami pro dosah těchto sítí je 20-25 metrů v budovách a až 100 metrů na volném prostranství. Rychlosti Wi-Fi jsou ovlivněny stejnými faktory jako jejich dosah. U nejnovějšího typu 802.11.n je udávána maximální rychlost až 600 Mb/s. Alamanni [3, s. 25-26].

Rámce v standardu 802.11

Jako u dalších standardů na druhé vrstvě OSI modelu jsou i data u 802.11 přenášena v podobě rámců. Rámce u tohoto standardu jsou složeny z těla obsahujícího samotná přenášená data, která jsou většinou šifrována dle některého z bezpečnostních standardů, a kontrolního součtu pro ověření celistvosti obdrženého rámce. V neposlední řadě obsahuje rámec hlavičku s mnoha údaji, pro penetrační testování je asi nejdůležitějším údajem typ rámce. Alamanni [3, s. 26-27].



Obrázek 1 - rámec standardu 802.11 zdroj: [22]

Typy rámců

- **Servisní rámce –** tyto rámce slouží k řízení komunikace mezi AP a klientem.
    - **Beacon -** rámce používané k oznámení přítomnosti AP včetně jeho základní konfigurace.
    - **Probe request -** rámce zasílané klientem pro zjištění přítomnosti jakéhokoliv AP v dosahu nebo zjištění informací o konkrétním AP.
    - **Probe response -** rámce zasílané jako odpověď na předchozí rámec obsahující informace o dané síti.
    - **Authentication request -** rámce posílané klientem pro zahájení autentizační fáze s AP.
    - **Authentication response -** kladná nebo záporná odpověď na předchozí rámec.
    - **Association request -** rámec zasílaný klientem pro zahájení asociační fáze s AP.
    - **Association response -** kladná nebo záporná odpověď na předchozí rámec.
- **Kontrolní rámce –** rámce řídící provoz v síti. Typy kontrolních rámců jsou **Request-to-Send (RTS)** a **Clear-To-Send (CTS),** které se snaží redukovat kolize mezi rámci, a **Acknowledgment(ACK)** sloužící jako potvrzení o správném přijetí rámců.
- **Datové rámce –** tyto rámce obsahují samotná data přenášená po Wi-Fi síti zabalená do packetů vyšších vrstev OSI. Alamanni [3, s. 25-26].

5.2 Návrh postupu penetračního testování

5.2.1 Fáze plánování

Tato fáze penetračního testování je dle Alamanniho [3, s.2] klíčová pro celý průběh penetračního testování. Jak ale autor uvádí, v mnoha případech jí není věnováno dostatečně velké množství pozornosti. Stejný autor dále říká, že v této fázi by měl být definován rozsah celého testu a domluveny podmínky pro testování s vlastníkem dané sítě. Dále se v této fázi stanoví cíle, které by měly být splněny po skončení celého procesu penetračního testování. Jako příklady prováděných akcí v této části uvádí Alamanni [3, s.3] například vymezení oblasti, ve které se budou hledat bezdrátové sítě, vymezení oblasti, která je pokryta signálem sítě, a zjištění počtu klientů, kteří se připojují k dané síti. Dále se zde definují oblasti, na které by měl být test zaměřen, jako jsou konkrétní bezpečnostní problémy a jejich priority. V neposlední řadě zde určíme hranice testu. Těmito hranicemi je myšlena nutnost oznamování výskytu rouge skrytých access-pointů, případně jsou-li povoleny útoky proti klientům, připojovaným do testované sítě. Dále by zde měla být stanovena všechna ostatní pravidla pro testování včetně ohlášení data a času provádění testů a sepsání kontraktu o mlčenlivosti.

5.2.2 Fáze analýzy prostředí

Druhou fází testování je celková analýza zkoumaného prostředí. Jak se píše v Alamanniho knize [3, s.3], hlavním cílem této fáze penetračního testování je snaha o sběr co možná největšího množství informací o sítích, které byly stanoveny jako předměty testu v první fázi. Dle tohoto principu autor tuto fázi přejmenovává na fázi sběru informací, jelikož dle jeho názoru je sběr informací nejdůležitějším počinem této fáze, a to hlavně z důvodu, že pokud je sesbíráno dostatečné množství informací o dané síti, je možné odhalit její potenciální slabiny ještě před začátkem samotného testování. Tyto odhalené slabiny nám později pomohou se správným směřováním samotného útoku a mnohonásobně tuto fázi usnadní a zrychlí.

Nalezení sítí včetně skrytých sítí a rouge access-points

Nalezení bezdrátových sítí je poměrně jednoduchý proces, a to hlavně z důvodu, že rámce jsou přenášeny volně v prostředí a mohou být odchyceny prakticky kýmkoliv. Proces zachytávání těchto rámců se nazývá wirless scanning, a jak říká Alamanni [3, s.29], tento

proces se stal poměrně oblíbeným i u lidí bez technického vzdělání. Stejný autor popisuje dva druhy tohoto skenování.

Aktivní scanning

Tento druh odchytávání zahrnuje vyslání probe request rámce do broadcastu v dané oblasti a čekání na probe response od všech AP, které jsou v oblasti aktivní. Tento postup má však velkou nevýhodu v tom, že neumožňuje nalézt skryté sítě. Skryté sítě mají totiž nastaveno zahazování probe request rámců a díky tomu nejsou viditelné pro klasická zařízení. Alamanni [3, s. 29].

Pasivní scanning

Jiným druhem je pasivní skenování, které nám poskytuje ve většině případů lepší výsledky. Při tomto skenování nejsou rozesílány probe requesty do broadcastu. Místo toho je adaptér nastaven do monitorovacího módu, takže může odposlouchávat veškeré rámce, které jsou přenášeny v dané oblasti. Poté je možné zachycené pakety analyzovat například pomocí Wireshark a z beacon rámců zjisti informace o všech AP vyskytujících se v oblasti. Díky tomu je zajištěno zachycení informací i o zařízeních, která jsou pro aktivní scanning skryta. Alamanni [3, s. 29].

5.2.3 Určení typu zabezpečení používaného u cílové sítě

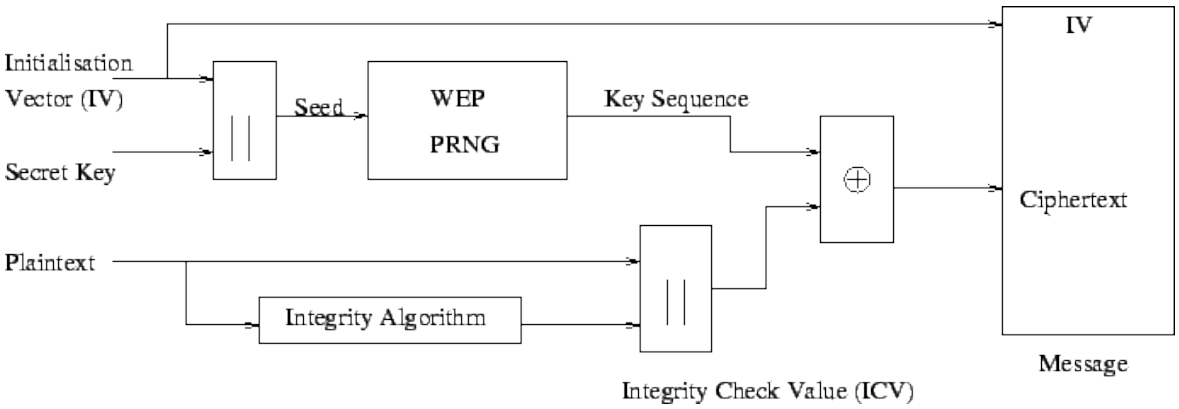
Otevřená síť

Jedná se o bezdrátové sítě používané převážně na veřejných prostranstvích, takovými místy jsou například autobusová nádraží, kina, kavárny a podobně. Tento typ sítě nepoužívá žádný bezpečnostní mechanismus a může se k němu připojit kterékoliv zařízení podporující 802.11 standard. Pakety procházející sítí nejsou žádným způsobem šifrovány a data v nich jsou přenášena jako plain text. Není pravděpodobné, že bychom se s tímto typem sítě setkali během penetračního testování.

Zabezpečení typu WEP

Jak píše Almanni ve své knize [3, s.43], jedná se o zkratku vzniklou ze spojení Wired Equivalent Privacy, a je to protokol, který byl představen s originálním standardem pro bezdrátové sítě 802.11 jako prostředek, který měl poskytnout zabezpečení bezdrátových sítí a šifrování přenášeného obsahu. Protokol je založený na RC4 (Rivest Cipher 4) a používá preshared secret key (PSK) skládající se ze 40 nebo 104 bitů podle konkrétní

implementace. K šifrování obsahu se používá 24 bitový pseudo-náhodný Initialization Vector (IV), který po zřetězení s PSK tvoří RC4 klíč pro zašifrování každého přenášeného packetu. To znamená, že výsledná velikost zašifrovaného řetězce se bude pohybovat mezi 64 a 128 bity.



Obrázek 2 - šifrování WEP zdroj: [23]

V dnešní době je standard WEP označen skupinou Wi-Fi Alliance jako zastaralý a není doporučeno ho nadále využívat jako prostředek pro zabezpečení bezdrátové sítě. Hlavním důvodem je, že tento standard trpí mnoha různými bezpečnostními riziky, ať už ohledně generování řetězců, používání Initialization Vector nebo délky šifrovacího klíče. Problém s použitím IV je, že je pouze 24 bitů dlouhý (tzn. pouze 16 777 2116 možností) a je přenášen jako nešifrovaný text v každém packetu. Z logiky věci vyplývá, že po určité době se vyčerpají všechny možnosti a IV řetězce se začnou opakovat. Toho může útočník zneužít a po sesbírání dostatečného množství dat použít statistický útok k prolomení klíče. Jelikož k provedení tohoto útoku je třeba velké množství zachycených packetů, je třeba zvýšit provoz v síti. Toho může potenciální útočník dosáhnout například pomocí ARP request Replay.

Zabezpečení typu WPA

Alamani ve své knize [3, s.61] říká, že WPA je zkratka pro Wi-Fi Protected Access protocol. Tento standard byl vyvinut skupinou Wi-Fi Alliance jako náhrada za zastaralý protokol WEP. WPA byl poprvé představen v roce 2003 a hned další rok byl nahrazen protokolem WPA2. Protokol podporuje dva typy autentizace – personal a enterprise. U typu personal je používán pre-shared key (PSK) a není zapotřebí využití autentizační autority (nejčastěji v podobě autentizačního serveru). Zde používaný PSK může být složen z 8-63 tisknutelných

znaků ASCII tabulky. Na druhou stranu u typu enterprise je za potřebí využití autentizační autority, dále je zapotřebí zakoupit AP, které má zabudovanou podporu RADIUS protokolu a klienti musí být schopni použít Extensible Authentication Protocol (EAP) pro svoji autentizaci. K autentizaci mezi klientem a AP se používá proces zvaný 4-way handshake.

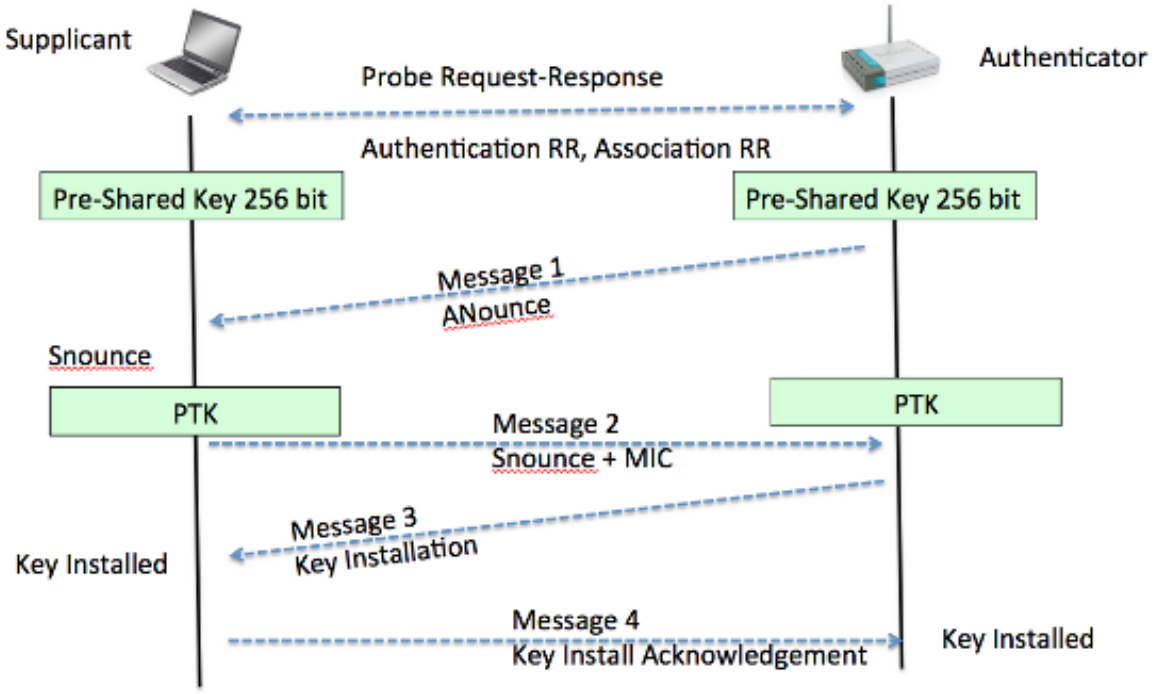
Zabezpečení typu WPA2

Standard WPA je nejmodernějším protokolem pro zabezpečení bezdrátových sítí. Tento standard vylepšuje zastaralé WPA a WEP hlavně po stránce šifrování a autentizace. WPA2 nutně implementuje šifrovací algoritmus Advanced Encryption Standard (AES), který je při dostatečné délce klíče neprolomitelný. K autentizaci se stejně jako u protokolu WPA může použít buď PSK (pre-shared key) nebo protokol EAP společně s nastavením autentizační autority. Pro autentizaci mezi AP a klientem se stejně jako u staršího WPA používá 4-way handshake. [20]

Four-Way Handshake

Pokud se zařízení chce připojit do sítě zabezpečené pomocí WPA/WPA2 protokolu, musí nejdříve proběhnout Four-Way Handshake.

- Na začátku tohoto procesu je na obou stranách (klienta i routeru) nezávisle na sobě nastaven Pairwise Master Key (PMK). Tento klíč je generován za pomocí BSSID a PSK. Poté co je PMK nastaven, zašle AP klientovi náhodné číslo. Toto číslo se nazývá A-Nonce.
- Následuje druhá fáze, v níž pošle klient kontrolní Message Integrity Code společně s náhodným číslem, toto číslo je zvané S-Nonce, zpátky k AP. V mezičase vytváří klient tzv. Pairwise Transient Key (PTK), který je použit na šifrování obsahu přenášeného mezi AP a klientem po dobu jejich spojení. Tento klíč je vytvořen z PMK, A-Nonce, S-Nonce a MAC adres obou zařízení.
- Ve třetí fázi provede stejný proces vytváření PTK i AP a pošle Group Temporal Key (GTK) klientovi. Tento klíč používá klient k dešifrování multicastů, broadcastů a MIC.
- V poslední fázi se už provádí pouze odeslání potvrzovacího packetu od klienta.



Obrázek 3- WPA 4-way-handshake zdroj: [24]

Jak je možné vidět na obrázku číslo 3 nad tímto textem. U WPA/WPA2 je každé PTK unikátní a navíc se neposílá mezi klientem a AP. Z toho vyplývá, že standard WPA/WPA2 je bezpečný a jediná možnost, jak ho prolomit, je uhádnout slabý PSK, například pomocí slovníkového útoku.

Wi-Fi protected setup

Jedná se o bezpečnostní mechanismus pro AP představený Wi-Fi Aliancí v roce 2006, účelem jeho představení bylo ulehčit připojení do Wi-Fi sítí i pro méně technicky zdatné uživatele. Toho mělo být docíleno použitím osmimístného PIN místo PSK. Pokud je PIN zadaný správně, dojde k autentizaci. Dále tato specifikace podporuje i Push-Button-Connect (PBC), díky čemuž stačí pro autentizaci pouze zmáčknutí tlačítka na routeru a na klientovi.

V roce 2011 však vědci Stefan Viehbock a Craig Heffner nezávisle na sobě zjistili, že tento nový standard obsahuje bezpečnostní vadu, která umožní útočníkům zjistit PIN během několika hodin pouze za využití brute-force útoku. Chyba se vyskytuje v ověřování PIN na straně AP. Mezi AP a klientem je totiž zasílána pouze polovina PIN, pokud je tato polovina ověřena, zašle klient druhou část, pokud není ověřena, je zaslán packet s chybovým kódem. Z toho vyplývá, že na AP jsou obě poloviny PINU ověřovány zvlášť. Navíc poslední cifra PIN je pouze kontrolním součtem ostatních sedmi. Z toho vyplývá, že z původních

10 000 000 možností pro osmi místný PIN se stává pouze 11000 možností, což je s dnešním výpočetním výkonem otázka několika minut i za použití prostého brute-force útoku.

5.2.4 Fáze útoku

Jak píše Alamanni [3, s.4], jedná se o fázi, kde se využívá praktických dovedností k zneužití slabin, které byly objeveny v průběhu analytické fáze, a jejich následné využití k získání s udržení přístupu do sítě. Autor tuto fázi dále dělí na dvě pod-fáze.

Útoky na infrastrukturu a zabezpečení

Nejčastějším typem útoků v bezdrátových sítích jsou útoky přímo proti AP a jak je popsáno v přechozí kapitole, existuje mnoho chyb, které je možné využít pro získání přístupu do testované sítě. Tyto chyby se nejčastěji vyskytují v koncepci protokolů využívaných pro zabezpečení přístupu. Některé další možnosti využití slabé infrastruktury sítě jsou popsány níže.

Denial of service

Tento typ útoku je častý v mnoha odvětvích systémové bezpečnosti. Jedná se o vytváření velkého množství síťového provozu, který eventuálně zahltí router nebo klienta a znemožní mu využívání služeb poskytovaných danou sítí. DDOS útok je v bezdrátových sítích nejčastěji prováděn jako zahlcení AP obrovským množství deautentizačních paketů. Dalšími možnostmi je rozesíláním ARP paketů pro falešný access point. Dalším možností je vysílání velkého množství pokusů o autentizaci na routeru, tento útok však potřebuje obrovské množství těchto requestů k úspěšnému dokončení.

Rogue access point

Pokud se útočníkovi podaří nějakým způsobem získat přístup do sítě, Rogue access point je jednou z možností, jak si tento přístup udržet. Jedná se o přidání nového AP do existující sítě, který může útočník následně využít jako přístupový bod pro pozdější využití. Dle Alamanniho [3, s.100] může být Rogue access point buď fyzický, který zahrnuje násilné zapojení nového routeru do sítě, nebo softwarový.

Využití slabých přístupových údajů do routeru

Většina routerů, hlavně v domácích sítích, má možnost grafické konfigurace za pomoci webového rozhraní. Tato rozhraní mají nastavena defaultní přístupové údaje a velká část uživatelů nevěnuje dostatečnou pozornost jejich přenastavení. To se děje hlavně z důvodu,

že toto rozhraní je dostupné pouze zevnitř sítě a na první pohled se tedy zdá bezpečné proti venkovním útokům. V posledních letech se však objevily možnosti, které dovolují připojit se k tomuto rozhraní i ze sítě internet. Jednou z těchto možností je například DNS Rebinding. Alamanni tento procese popisuje zde[3, s.105], říká že se jedná o využití DNS serveru, který předá klientovi falešnou stránku se skriptem, který získá kontrolu nad routerem využívajícím defaultní přístupové údaje. K tomuto útoku se dá využít program Rebind, který je standardní součástí Kali Linux.

Útoky na klienty

V mnoha reálných případech nastává při penetračním testování bezdrátových sítí situace, kdy nemá tester přístup k infrastruktuře dané sítě a není možné provést testování představené v minulé kapitole. V této situaci je možné využít k napadení sítě klienty, kteří se do této sítě připojují.

Evil Twin a Honeypot

V minulé kapitole byl představen princip rouge access point, jehož obdoba se dá využít i pro testování na klientech sítě. Jak píše Alamanni [3, s.107], je možné vytvořit falešný access point, který láká klienty k tomu, aby se připojili k němu místo reálného přístupového bodu. Jak autor říká, pokud se tento honeypot skrývá za identitou určitého routeru, je možné ho využít k provedení tzv. Evil Twin útoku. Pokud je naše falešné AP blíže a má silnější signál než pravé AP, je pravděpodobné, že klient, který je nastaven tak, aby se automaticky pojil do dříve využitých sítí, použije náš AP místo pravého AP. Jelikož servisní rámce v bezdrátových sítích nejsou šifrovány nebo jakkoliv kryptograficky podepsány, nemá klient žádnou možnost rozlišení těchto dvou APOD.

Man-in-The-Middle

Jak se píše v knize [3, s.112], jedná se o útok, kdy se útočník vmísí do komunikace mezi dvěma stranami (v našem případě nejčastěji klientem a AP) a přeposílá tuto komunikaci tak, že si obě strany nejsou vědomy jeho účasti.

Pro penetrační testování je nejčastějším přístupem vytvoření Evil Twin a odposlech síťového provozu za pomoci programů jako je Wireshark nebo Ettercap a jeho následné přeposílání reálnému AP. Pokud je nastaven Man-In-The-Middle, je možné provádět DNS poinsoning, který nám dovoluje upravit DNS záznamy, a tím donutit klienta k připojení k jinému serveru

bez jeho vědomí. Další možností je Session Hitjachking, které nám dovoluje zfalšovat certifikáty webových stránek nebo využít chyby v protokolech šifrovaného přenosu.

6 Praktické ověření navrženého řešení

6.1 Fáze plánování

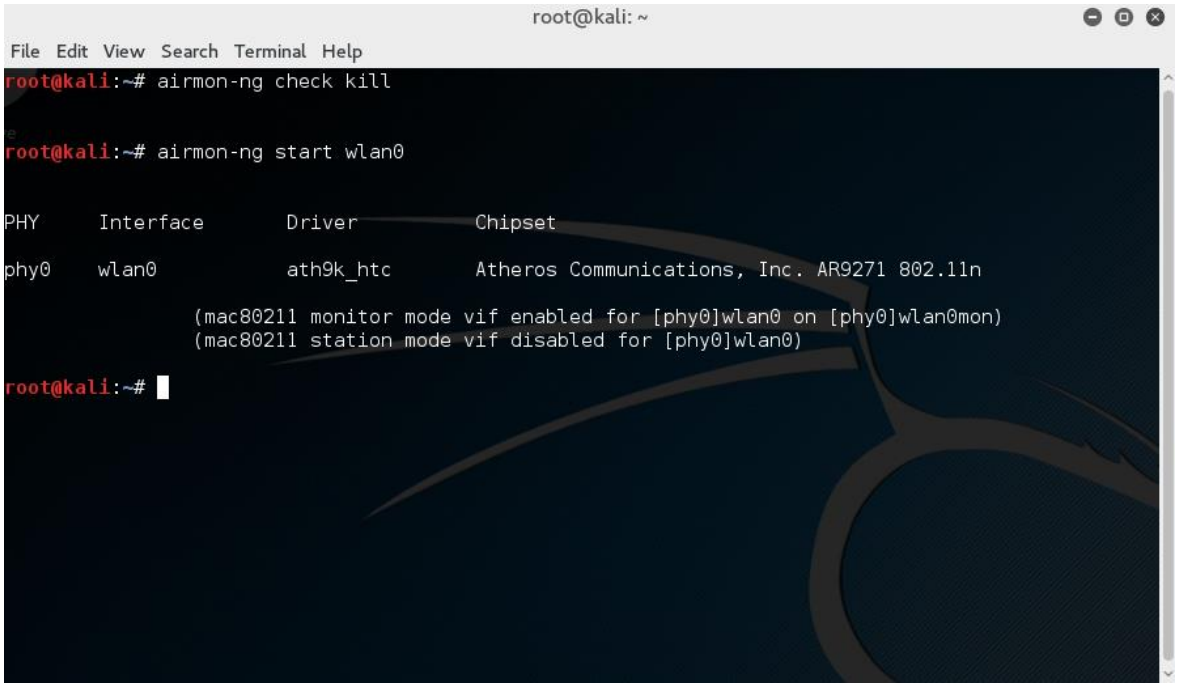
V této kapitole bych rád představil infrastrukturu, kterou jsem použil pro penetrační testování, a také zde představím dva postupy, které se používají prakticky v každém penetračním testu.

Příprava testování

Plánem praktické části této práce je ověřit možnosti testování bezdrátových sítí, které byly nastíněny v části teoretické. Pro toto testování jsem využil Linuxovou distribuci Kali Linux, která byla představena v teoretické části práce. Dále jsem využil síťovou kartu TP-LINK TL-WN722N, která umožňuje jak zachytávání síťového provozu v monitorovacím módu, tak i provádění packet injection testů. Jako testovací síť jsem použil síť vytvořenou na routeru TP-LINK TL-WR841N, která má nastavené SSID „Wireless Lab“ a BSSID C4:E9:84:88:B5:0E. Tento jsem zvolil hlavně z důvodu své vysoké rozšířenosti v domácnostech, což mi umožňuje demonstrovat potenciální díry v zabezpečení běžných domácích bezdrátových sítí.

Přepnutí do monitorovacího modu

Prakticky pro všechny úkony potřebné k provádění penetračního testování bezdrátových sítí je potřeba přepnout síťovou kartu do tzv. monitorovacího modu. Toho lze docílit například pomocí nástroje Airmon-ng obsaženého v balíčku Aircrack, který je součástí základní verze Kali Linux. V novějších verzích této distribuce je před přepnutím do monitorovacího modu potřeba zkontrolovat procesy, které by mohly tomuto přepnutí bránit a pokud se nějaký takový najde, je nutné ho před provedením úkonu korektně ukončit. K tomu má Airmon přímo příkaz `airmong-ng check kill`, který tyto dva kroky provede automaticky. Pokud nastavení již nic nebrání, následuje samotné přepnutí, které je provedeno za pomoci příkazu `airmon-ng start <<interface >>`. Průběh můžete vidět na obrázku číslo 4 níže. Změnou od předchozích verzí je změna názvu interface při přepnutí do monitorovacího modu, jak lze vyčíst z obrázku, k názvu interface se přídá přípona mon.

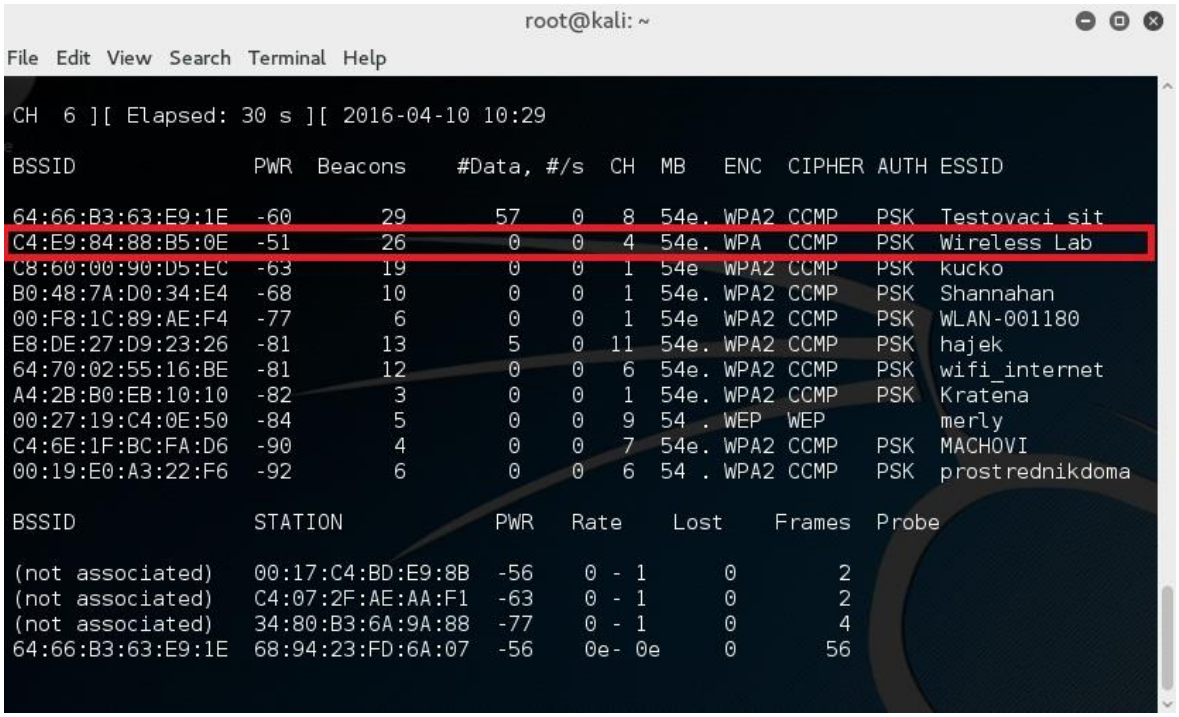


Obrázek 4- přepnutí do monitorovacího modu

Využití nástroje Airodump-ng pro odposlech sítě

Pro zachytávání síťového provozu budu prakticky v každém testu používat nástroj Airodump-ng z balíčku Aircrack obsaženého v Kali Linux. Airodump-ng se pouští příkazem *airodump-ng<<interface >>*. Jak je vidět na obrázku číslo 5 pod tímto odstavcem, Airodump po svém zapnutí nalezl moji testovací síť společně s několika dalšími, které jsou v dosahu síťové karty. Jelikož se budu v průběhu testování zaměřovat pouze na svoji testovací síť a ve většině případů budu potřebovat mít zachycená data uložena v souboru, budu mnohem častěji používat tuto variantu příkazu *airodump-ng -bssid <<BSSID>> -write <soubor>> <<interface >>*. Dalším použitelným parametrem tohoto

nástroje je možnost -c, která definuje kanál, na němž AP vysílá.



Obrázek 5- výpis Airodump

6.2 Analýza prostředí

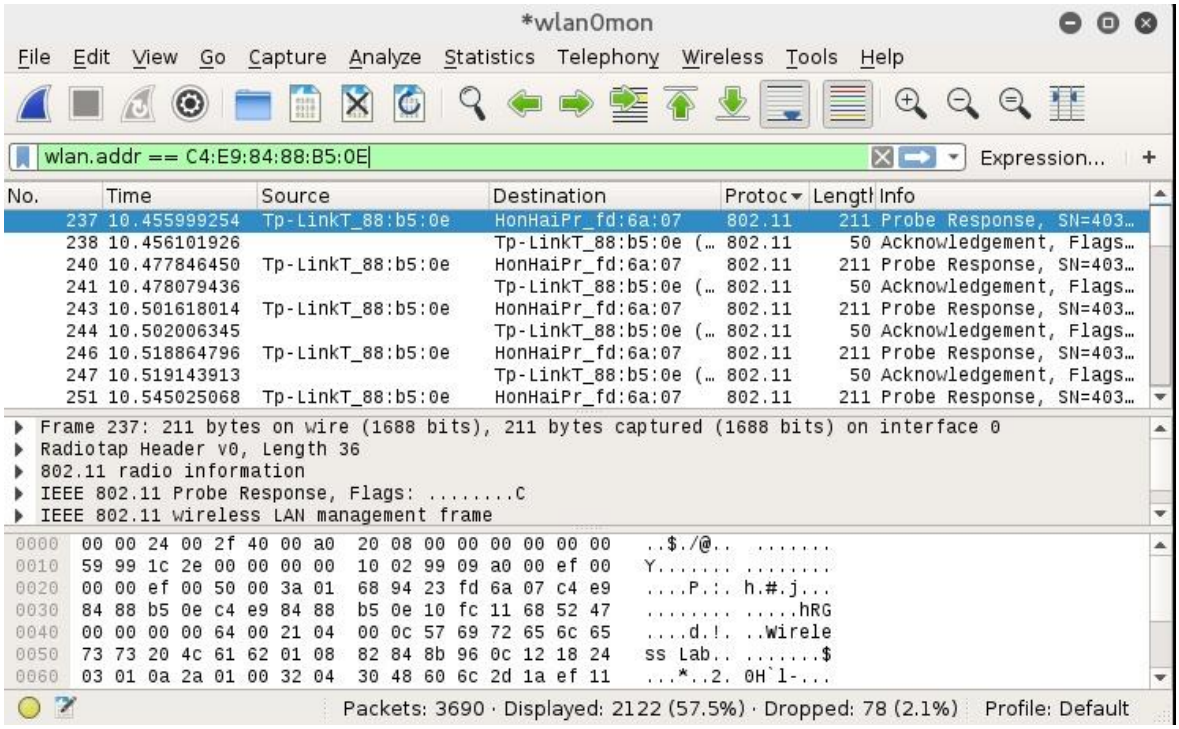
Ve fázi analýzy prostředí je mým cílem ukázat, jaké druhy informací může útočník získat pouhým odposlechem datového provozu v prostředí. Konkrétně bych rád ukázal, jak nalézt routery, které mají nastaveno chování skryté sítě, a také, jak jednoduše obejít filtr MAC adres.

Nalezení skrytých SSID

Jednou z velmi častých chyb při zabezpečnování domací sítě je pocit, že když je síť nastavena jako neviditelná, je pro útočníka nemožné zjistit, že taková síť vůbec existuje. Proto bych rád ukázal, že to nemožné není, naopak, je to poměrně snadné. Jak se píše v teoretické části, lze toho dosáhnout pomocí dvou přístupů, buď pouhým odposlechem sítě a čekáním na probe request od klienta a nebo aktivně odpojením všech okolních zařízení a tím získáním probe requestů od těchto klientů.

Pasivní přístup

Technika pasivního odposlechu je poměrně jednoduchá a jediné, co k ní potřebujeme, je nástroj Wireshark, který je dostupný v základní distribuci Kali Linux. Poté pouze přepneme interface naši síťové karty do monitorovacího modu (jak je popsáno v kapitole 5.1), zapneme Wireshark a nastavíme mu filtr, tak aby zachytával pouze rámce, které obsahují adresu mé sítě. Poté už pouze počkám na nové připojení od klienta. Pokud se klient připojí, vygenerují se mezi ním a AP probe request a probe response rámce, ze kterých je již možné SSID této skryté sítě jednoduše vyčíst, jak je možné vidět na obrázku číslo 6.



Obrázek 6 - Wireshark SSID

Aktivní přístup

Pokud se neobjeví žádný klient, který by se k síti nově připojoval a tím vygeneroval probe response/request, je nutné použít tzv. aktivní přístup. Tento přístup spočívá v rozeslání deatuthentication packetu do broadcastu, čímž jsou již připojení klienti nuceni znovu odeslat probe request, ze kterého můžeme SSID vyčíst stejně jako v případě pasivního přístupu. K rozeslání deauthentikačního packetu použijeme nástroj Aireplay-ng, který je jako ostaní nástroje balíčku Aircrack-ng dostupný v základní verzi Kali Linux. Deauthentikační packet se posílá příkazem `aireplay-ng -0 -5 -a <<adresa AP >>`, kde 0 symbolizuje deauthentikační packet, 5 je počet packetů, které chceme rozeslat.

Filtr MAC adres

Dalším častým omylem uživatelů Wi-Fi je domnění, že pokud mají na svém routeru nastavenou filtraci MAC adres, jsou v totálním bezpečí. Jak ale ukazuje následující příklad, není to zas taková pravda. Obejít filtr MAC adres je jednoduché v případě, že existuje klient, který se k síti připojuje. Pokud takový klient je, pak jednoduše spustíme nástroj Airodump-ng a zaměříme se na svoji testovací síť (viz. kapitola 5.1). Zde již vidíme klienty, kteří jsou k síti momentálně připojeni. Vybereme si tedy jednu z MAC adres těchto klientů a za pomoci nástroje Macchanger, který je součástí Kali, změníme MAC adresu své síťové karty na MAC adresu vybraného klienta. To provedeme příkazem *macchanger -m <<Vybraná MAC>> <<interface>>*.

6.3 Zneužití chyb v použitém zabezpečení

V následující kapitole bych rád demonstroval možné využití bezpečnostních děr, které vznikají při použití různých bezpečnostních protokolů. Otestuji, které zabezpečení se dá prolomit prostým brute-force útokem a které se dá naopak prolomit pouze při špatném nastavení (tedy chybě na straně uživatele), případně zda existuje nějaké, které je stoprocentně bezpečné.

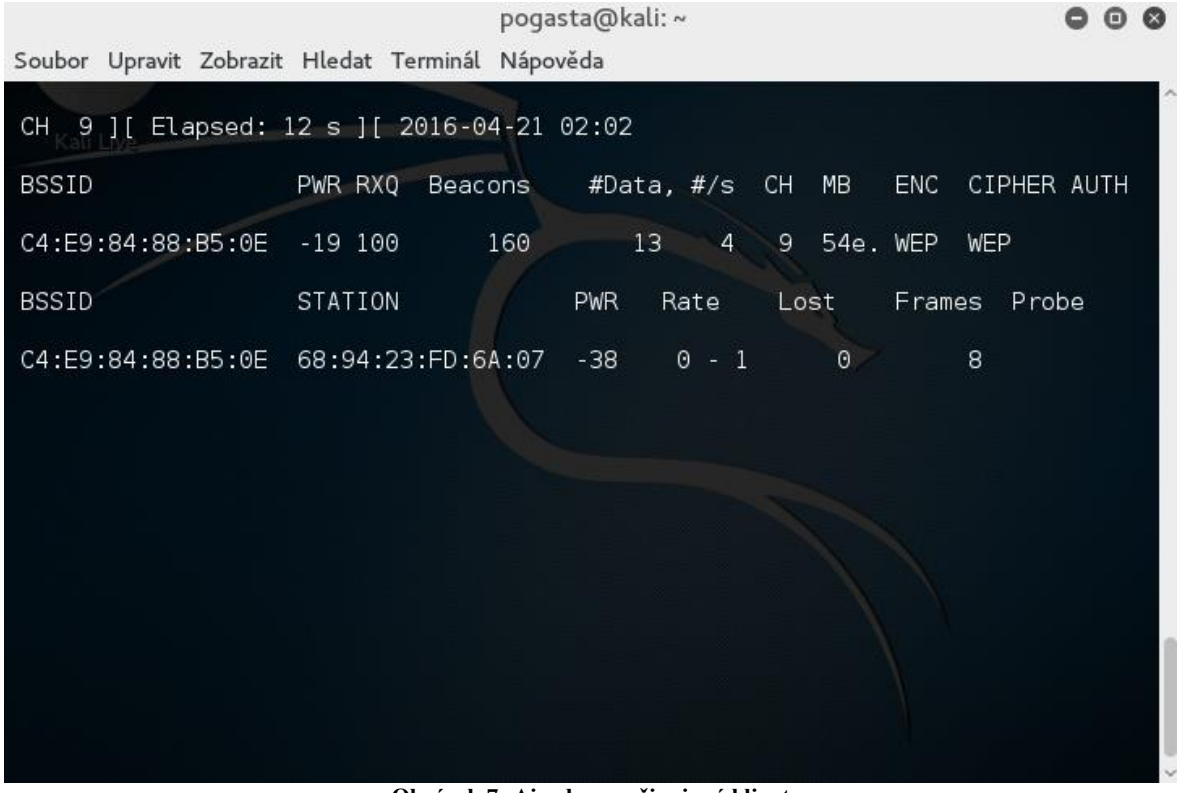
Zabezpečení typu WEP

Jak jsem již popsal v teoretické části, toto zabezpečení je zastaralé a Wi-Fi Alliance doporučuje ho vůbec nevyužívat, stále se ale najde obrovské množství sítí, které toto zabezpečení používají jako svůj bezpečnostní protokol. Proto zde ukážu několik útoků, které dokážou toto zabezpečení poměrně jednoduše obejít.

S připojeným klientem

Pokud existuje klient, který je připojený k naší cílové síti, můžeme ho využít k rychlému získání ARP packetu, tento získaný packet následně využijeme k útoku zvanému ARP Request Replay, který nám dovolí mnohonásobně zvýšit množství dat přenášených mezi klientem a AP.

Nastavíme tedy síťovou kartu do monitorovacího modu (viz. kapitola 5.1) a spustíme Airdoump-ng (jak je popsáno v téže kapitole), který nám zobrazí připojené klienty a zároveň díky němu můžeme zachytit nově vygenerovaný síťový provoz. Na obrázku číslo 7 pod tímto textem je vidět připojený klient, kterého následně využijeme pro provedení útoku. Otevřeme si tedy druhý terminál, ze kterého pomocí nástroje Airbase-ng, konkrétně jeho příkazu *airbase-ng --arpreplay -h <<MAC klienta>> -b <<MAC AP>> <<interface>>* spustíme ARP replay útok. V terminálu, v němž běží Airodump, je možné vidět zvýšené množství síťového provozu. Pokud je zachyceno dostatečné množství packetů (uvádí se okolo 40 000), pokusíme se získat heslo pomocí nástroje Aircrack-ng, a to konkrétně příkazem *aircrack-ng -b <<MAC AP>> <<soubor z Airodump>>*. Pokud je prolomení hesla úspěšné, objeví se na obrazovce hledaný klíč, viz. obrázek 7, pokud ne, je nutno celý proces opakovat a zachytit větší objem dat.

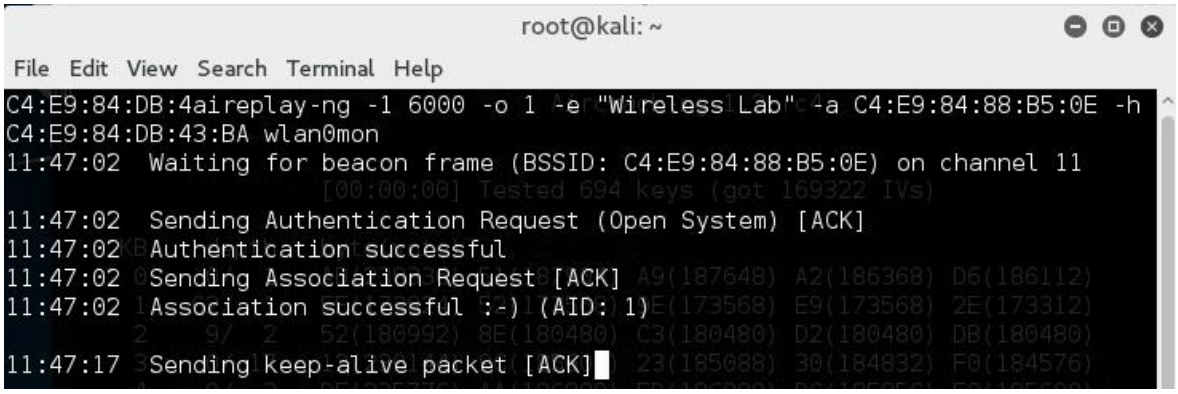


Obrázek 7- Airodump připojený klient

Bez připojeného klienta

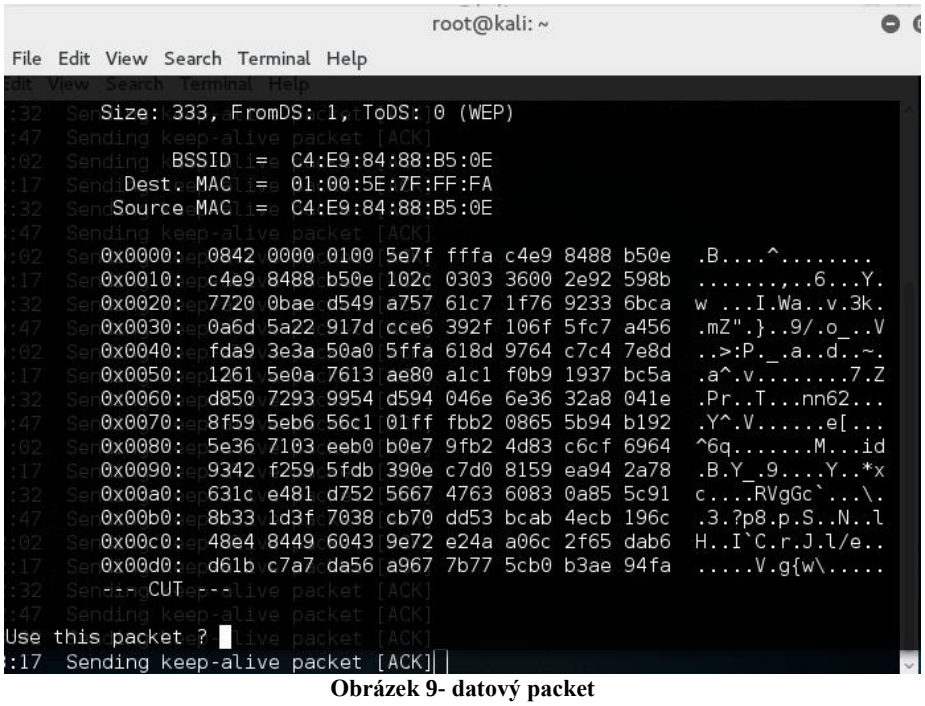
Útok na router zabezpečený pomocí WEP je možné provést i bez existence klienta, který by se k němu přípojoval. Postup je následující: nejdříve přepneme síťovou kartu do monitorovacího modu (jak je předvedeno v části 5.1), dále provedeme falešnou autentizaci na AP pomocí utility Aireplay-ng, a to konkrétně pomocí příkazu *aireplay-ng -1 0 -e <<ESSID AP >> -a <<BSSID >>, -h <<MAC sitove karty >> <<interface >>*,

kde 1 symbolizuje falešnou autentizaci a 0 určuje interval, v němž se bude autentizace automaticky obnovovat. Úspěšně provedenou autentizaci můžete vidět na obrázku číslo 8.



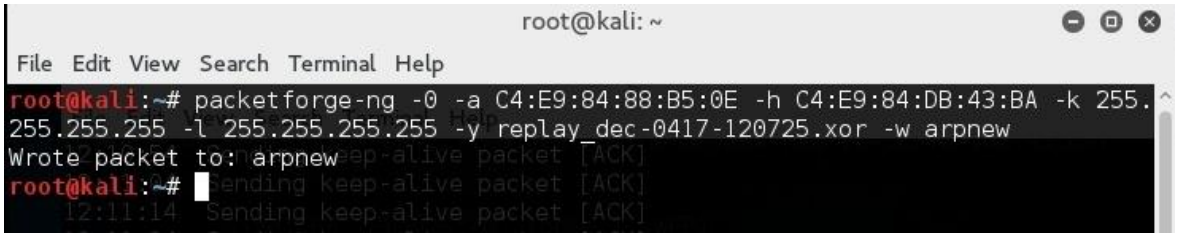
Obrázek 8- falešná autentizace na AP

V dalším kroku si otevřeme druhý terminál, v němž se pokusíme provést tzv. ChopChop attack, abychom získali PRGA a mohli si pomocí něj sestavit vlastní ARP packet. Toho se pokusíme docílit opět za využití Aireplay-ng. Tentokrát použijeme příkaz *aireplay-ng -4 -b <<BSSID >>-h <<MAC síťové karty >> <<interface >>*, parametr -4 zde symbolizuje typ útoku tedy v mém případě Chop-Chop (je možné využit i volbu -5, která docílí toho samého, ale za použití Fragmenatation útoku). Program zachytí datový packet zobrazí se volba, využití tohoto packetu, jak je možno vidět na obrázku číslo 8.



Obrázek 9- datový packet

Nyní máme PRGA a můžeme si pomocí programu Packetforge (který je samozřejmě součástí Kali Linux) vytvořit svůj vlastní ARP packet. To provedeme pomocí *příkazu packetforge-ng -0 -a <<BSSID >> -h <<MAC sitove karty >> -k 255.255.255.255 -l 255.255.255.255 -y <<soubor získaný v minulém kroku >> -w <<soubor, do nějž chceme uložit ARP >>*. Parametr -0 udává typ packetu ARP, parametry -l a -k jsou cílová a zdrojová IP, ale většina routerů reaguje na 255.255.255.255, jak ukazuje obrázek číslo 10.



Obrázek 10- vytvoření ARP packetu

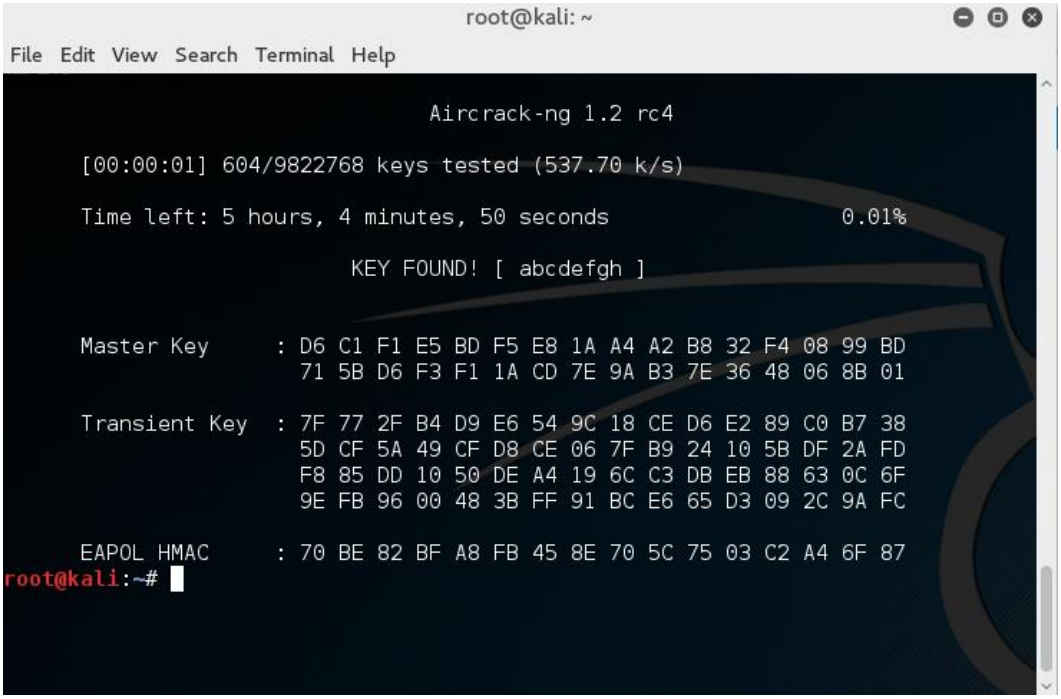
Pokud máme ARP packet, můžeme přejít k samotnému útoku. Otevřeme si terminál, v němž si spustíme Airodump-ng (jak jsem popsal v kapitole), a v druhém terminálu pustíme ARP replay útok pomocí příkazu *aireplay-ng -2 -r <<soubor s ARP packetem >><<interface >>*, kde -2 zastupuje typ útoku ARP replay. Pokud se nám povede nasbírat dostatečně velké množství dat, použijeme stejně jako v případě s připojeným klientem nástroj Aircrack-ng, příkazem *aircrack-ng -b <<BSSID >> <<soubor z airodump>>*. Pokud vše proběhlo v pořádku, je výstupem hledaný WEP key, pokud Aircrack klíč nenašel, je proces nutno opakovat s větším objemem zachycených dat.

Zabezpečení typu WPA/WPA2

Jak již bylo demonstrováno v teoretické části práce, správně nastavené AP s WPA a vyšším zabezpečením je velmi bezpečné a odolné proti útokům typu brute-force, proto jediným funkčním útokem je zde prostý slovníkový útok. Úspěšné provedení tohoto útoku je podmíněno chybou na straně uživatele v podobě volby krátkého nebo lehce uhodnutelného PSK. Rád bych tedy alespoň ukázal, jak tento slovníkový útok provést, a uvedl pár tipů na jeho optimalizaci.

Začneme opět přepnutím síťové karty do monitorovacího modu (postup je možné nalézt v kapitole 5.1), dále použijeme airodump-ng , díky němuž můžeme po připojení klienta zachytit 4-way-handshake. Pokud se žádný nový klient nepřipojí, použijeme Aireplay-ng a příkazem *aireplay-ng --deauth 1 -c <<klient MAC adresa >>, -a <<adresa AP >>, <<interface >>*, pošleme klientovi deauthentication packet, který ho donutí provést opětovné připojení, při němž se vygeneruje hledaný handshake, který můžeme zachytit. Dále budeme potřebovat soubor, který bude obsahovat potenciální PSK, tzv. wordlist (několik

worldlistů je dodáváno přímo v Kali, bohužel ale obsahují klíče, které jsou používané spíše v anglicky mluvících zemích). V posledním kroku pustíme Aircrack-ng příkazem *aircrack-ng -w <<worldlist >> <<soubor s handshake* >>, pokud wordlist obsahuje hledaný klíč měli bychom po nějakém čase (záleží na mnoha faktorech, jako je výpočetní výkon, velikost worldistu, umístění klíče,…) vidět stejný výstup jako na obrázku 11.



Obrázek 11- WPA Aircrack úspěch

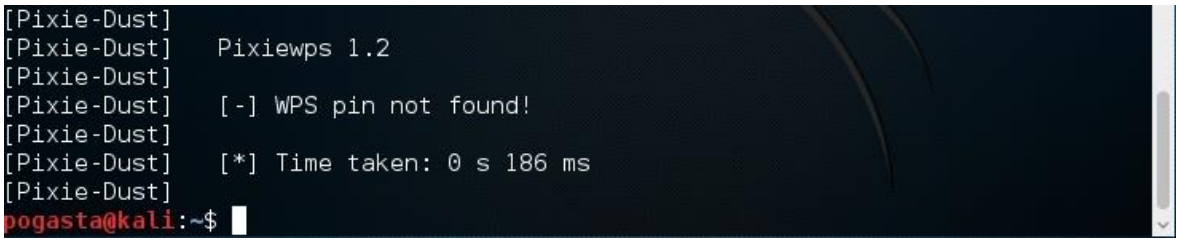
Pro efektivnější provedení slovníkového útoku můžeme proces optimalizovat například pomocí využití databáze s předkompilovanými PSK a 4-way-handshake, tzv. Rainbow table. Tuto tabulku můžeme vytvořit například pomocí nástroje Airlolib-ng. Další možnou optimalizací je využití nástroje, který využívá výpočetní výkon grafické karty namísto obvyklého procesoru, takovými nástroji dostupnými v Kali jsou například Pyrit nebo oclHashacat. Jelikož jsem k testování používal virtualzovaný systém, nebylo možno tento postup možno efektivně využít.

6.4 Útok proti AP a infrastruktuře sítě

Pokud nejsme schopni získat přístup do sítě s využitím chyb v bezpečnostních protokolech, můžeme se pokusit využít chyb v infrastruktuře sítě nebo špatného nastavení samotného AP. Nejčastějším bezpečnostím rizikem v tomto směru je aktivní WPS, jak se pokusím ukázat v této části. Pokud má router povolené WPS, je poměrně jednoduché získat do sítě přístup a to i přes použití bezpečného WPA2 zabezpečení.

AP s aktivním WPS

Princip WPS je detailně popsán v teoretické části, proto bude v této části ukázán pouze postup samotného útoku. Nejdříve pomocí nástroje Wash zjistíme, které sítě mají povolené WPS a to konkrétně příkazem *wash -i <<interface >>*. Výstup je možné vidět na obrázku. Následně použijeme utilitu Reaver, která implementuje útok Pixie Dust. Konkrétním příkladem použití je příkaz *reaver -i <<interface>> -b <<BSSID cílové sítě>>-vvv -K 1*. Bohužel mnou testovaný router je proti tomuto typu útoku zabezpečen, proto se získání klíče nezdařilo (jak je možno vidět na obrázku 12). Pokud by ale router toto zabezpečení neměl, je získání klíče otázkou několika málo sekund.



Obrázek 12- PixieDust neúspěch

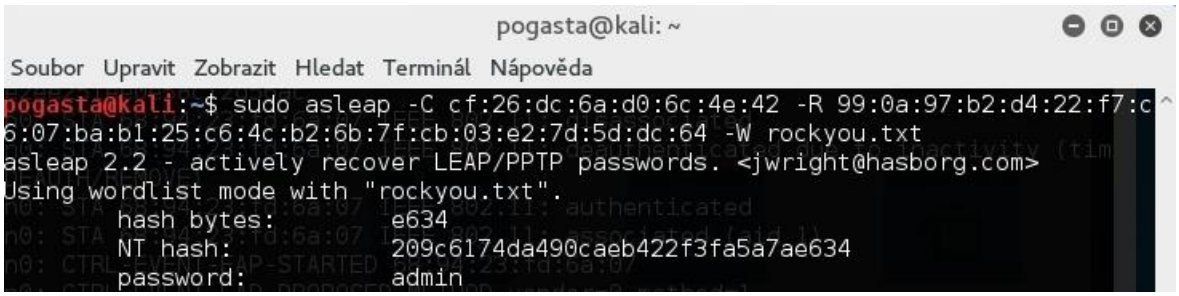
Radius server a WPA2 – Enterprise

Prolomení sítě využívající zabezpečení typu WPA/WPA2-Enterprise je možné pouze za příhodných okolností. Nejjednodušší je to v případě, kdy je pro komunikaci mezi serverem a klientem použit protokol LEAP nebo EAP-MD5, protože tyto dva protokoly jsou velmi slabé a je možné je prolomit pomocí jednoduchého bruteforce útoku. Jelikož nemám k dispozici síť implementující WPA2 – Enterprise použil jsem k testování aplikaci Hostapd a konkrétně její upravenou verzi WPE (Pownage edition), která byla vyvinuta přesně pro účel penetračního testování sítí s tímto zabezpečením.

V první fázi testování WPA2-Enterprise je nutné zachytit EAP handshake, to lze provést klasicky za pomoci Airodump-ng, jak je popsáno v kapitole 5.1. Pokud Airodump zachytí handshake je potřeba zachycený soubor prozkoumat například ve Wiresharku a zjistit, který bezpečnostní protokol je v síti používaný. Pokud je to jeden z prolomitelných protokolů pustíme slovníkový útok pomocí nástroje aesleap. Toto se mi bohužel nepodařilo prakticky ověřit, jelikož jsem disponoval pouze jednou síťovou kartou a nemohl jsem mít na síťové kartě emulovaný radius server a také odposlouchávat síťový provoz.

Dalším napadnutelným protokolem je PEAP a to pouze v případě, že klient nevaliduje na své straně certifikát RADIUS serveru, pokud tomu tak je, je možné použít falešný radius

server, ke kterému se klient přípojí a my získáme challenge a response MSCHAPv2 protokolu, pomocí níž můžeme již jednoduše pustit slovníkový útok pomocí programu Asleap, a to konkrétně příkazem *asleap -C <<Challenge>> -R <<RESPONSE>> -w <<WORDLIST>>*. Úspěšný pokus je možné vidět na obrázku 13 níže.



Obrázek 13- Asleap úspěch

6.5 Útoky proti klientům

Pokud selžou všechny útoky mířené na AP nebo na infrastrukturu sítě, je možné využít k získání přístupu i klienty, kteří jsou přímo nebo nepřímo spojení s testovanou sítí. V této části bych rád ukázal, jak vytvořit falešné AP, tzv. Honeypot, a nalákat na něj klienty a poté proti nim provést některý z útoků.

EvilTwin

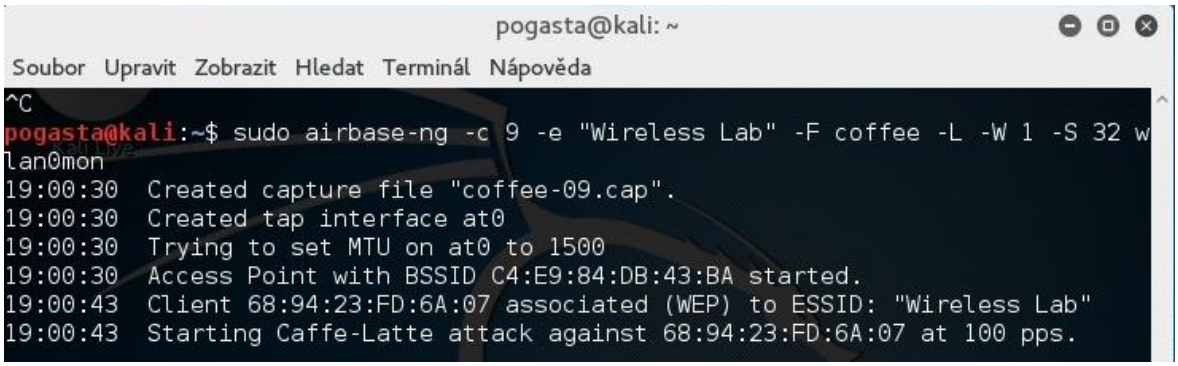
Evil Twin není typ útoku, který by nás sám o sobě dostal do sítě nebo byl jakkoliv efektivní, lze ho ale použít jako přípravu pro provedení složitějších útoků jako je například CaffeLatte nebo Man-in-The-Middle. Jak jsem psal v teoretické části, principem tohoto útoku je vytvořit přesnou kopii již existujícího a funkčního AP a donutit klienty, aby se připojili k našemu falešnému AP namísto původního. Prakticky jediné co k tomu potřebujeme, je umístit naše dvojče (Evil Twin) blíže ke klientovi, aby byla síla našeho signálu vyšší. Pokud se nám to povede, měl by se klient automaticky připojit k našemu falešnému AP namísto pravého AP bez toho, aby poznal rozdíl.

K úspěšnému nasazení evil twin potřebujeme znát BSSID routeru, který chceme napodobit, a kanál, na kterém vysílá. Tyto informace zjistíme jednoduše pomocí nástroje airodump-ng, jak je popsáno v kapitole 5.1. Poté již stačí vytvořit falešné AP pomocí utility Airbase-ng, konkrétně použitím příkazu *airbase-ng --essid <<ESSID cílového AP >> -c <<kanál AP >> <<interface >>*.

Momentálně tedy existují dvě AP se stejnými parametry, naše virtuální AP má však silnější signál, proto by se měl klient, který má nastaveno automatické připojování k dané síti, připojit rovnou na naše dvojče. Pokud se v dosahu neobjeví žádný nový klient, můžeme rozeslat stávajícím klientům deauthentication packet a donutit je k novému připojení, které by už ale mělo být na náš honeypot.

WEP síť a Caffe-Latte attack

Caffe-Latte je dalším důkazem nespolehlivosti WEP zabezpečení, jediné, co nám stačí, je najít klienta, který se automaticky připojuje k testované síti. Pokud takového klienta najdeme, stačí pouze, abychom v jeho blízkosti nastražili falešné dvojče cílové sítě pomocí Airbase-ng a to příkazem *airbase-ng -c <<kanál>> -e <<essid>> -F coffee -L -W <<soubor>> <<interface>>*. Tento příkaz spustí Caffe-Latte attack proti klientovi, který se na honeypot připojí, viz. obrázek 14. To nám dovolí získat dostatečné množství dat, které následně použijeme k prolomení klíče. Pokud máme nasbíráno dostatečné množství dat, stačí použít Aircrack-ng stejně jako u ostatních útoků na toto zabezpečení.



Obrázek 14- Caffe-Latte útok proti klientovi

6.6 Testování sítí s daným přístupem

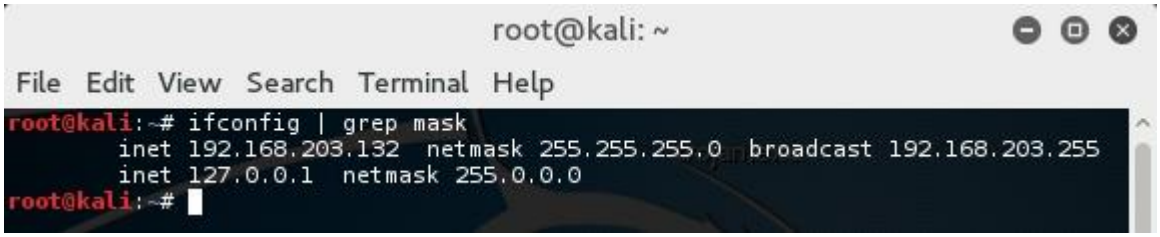
Pokud se nám podaří získat přístup k cílové síti, případně ho máme poskytnutý od majitele sítě již před začátkem testování, můžeme k jejímu otestování použít nástroje známé z penetračních testů prováděných na klasických LAN sítích. V této kapitole si tedy ukážeme, jak zjistit informace o testovaném prostředí a jeho případné bezpečnostní slabiny za použití bezpečnostních scannerů Nmap a Nessus Security Scanner. Nalezené slabiny se poté pokusíme zneužít za pomoci Metasploit frameworku. Pro lepší a názornější provedení testů budou všechny procesy předvedeny na specializované Ubuntu distribuci Metasploitable 2, která je již v základu distribuována s mnoha bezpečnostními chybami a je tedy určena pro předvedení principů penetračního testování hlavně za pomoci frameworku Metasploit.

6.6.1 Zjištění slabin sítě

Jako první si tedy ukážeme získání podrobných informací o testované sítě za pomoci specializovaného scanneru Nmap (pro uživatele, kteří preferují grafické rozhraní, existuje nadstavba Zenmap). V druhé části bude předveden obdobný proces za pomoci komerčního a velmi rozsáhlého nástroje Nessu Security Scanner.

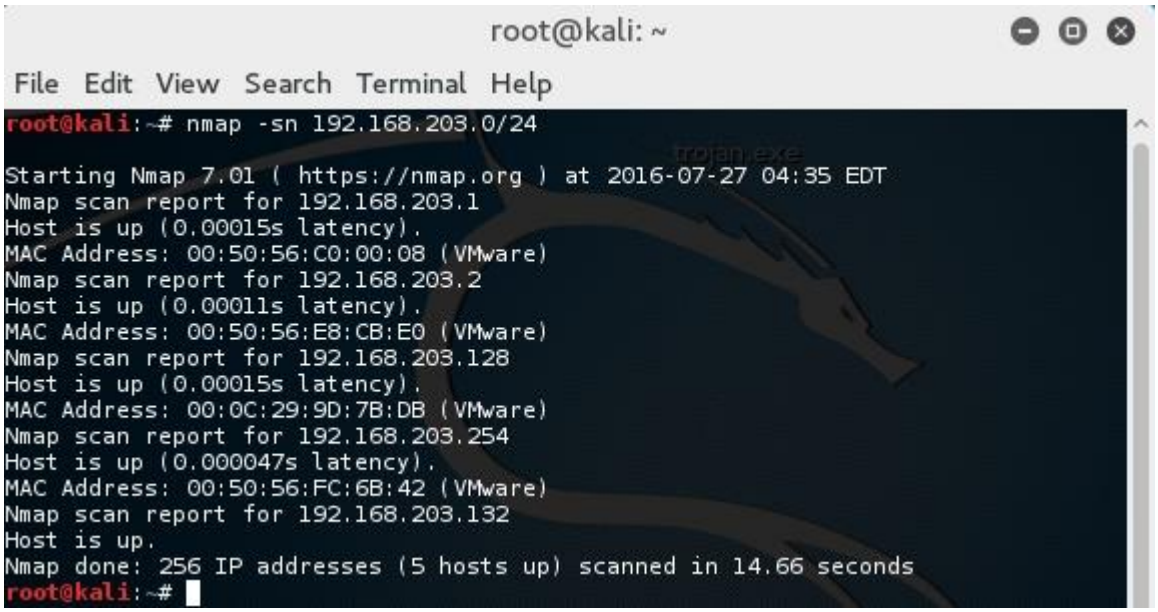
Využití Nmap

V prvním kroku potřebujeme zjistit velikost testované sítě, abychom snížili dobu prováděných skenů na minimum. To můžeme zjistit poměrně jednoduše, postačí nám k tomu IP adresa a maska sítě, tyto informace si můžeme vytáhnout pomocí standardního nástroje Ifconfig, tedy konkrétně příkazem `ifconfig | grep mask` . Jak je vidět na obrázku číslo 15 níže, v této testovací síti se může nacházet 0-254 hostů. Dále tedy potřebujeme zjistit, kteří hosté se v síti reálně vyskytují.



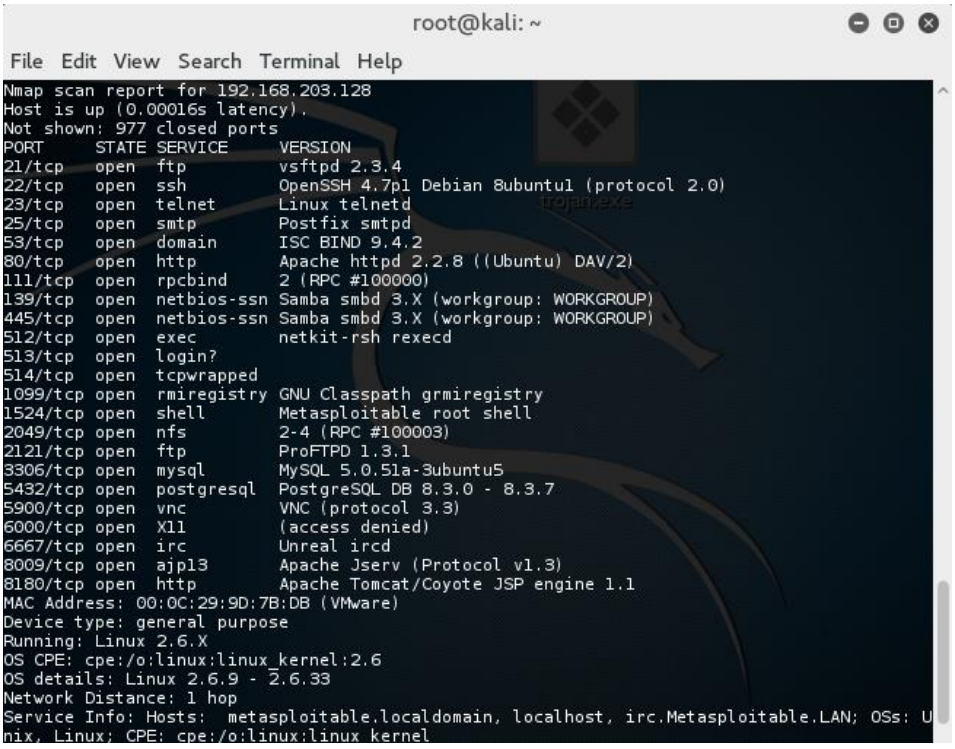
Obrázek 15- Zjištění velikosti sítě

K tomuto účelu využijeme dříve zmíněný nástroj Nmap a příkazem `nmap -sn <<IP adresa sítě>>` proskenujeme celý subnet, díky čemuž získáme informace o všech připojených klientech (parametr -sn zakazuje Nmap zjišťovat o klientovi další informace, jako jsou otevřené porty a běžící služby po jeho objevení, tyto informace budou zjišťovány až v následujících krocích). Jak se ukazuje na obrázku s číslem 16, k této síti je připojeno pět klientů, z nichž jeden je náš nastrčený zranitelný klient. V dalším kroku se tedy o tomto klientovi pokusíme zjistit podrobnější informace, které by nám mohly pomoci v následném pokusu o zneužití tohoto klienta.



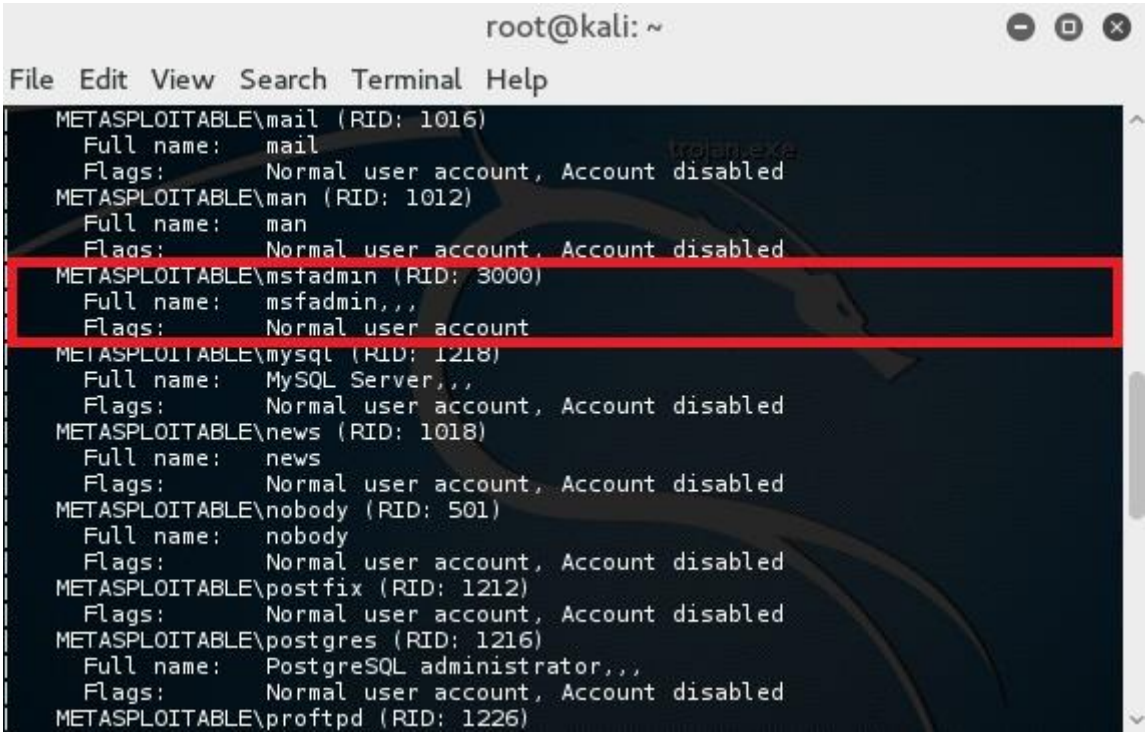
Obrázek 16 - Nmap klienti v síti

K zisku těchto informací nám opět pomůže Nmap, a to konkrétně příkaz *nmap -sS -sV -O <<IP adresa klienta>>*, který nám zjistí o klientovi důležité informace jako jsou otevřené porty, běžící služby, případně i verze a typ operačního systému. Jak je ukázáno na obrázku číslo 17 pod tímto odstavcem, ve výpisu z Nmap se objevuje velké množství otevřených portů, a to z důvodu použití distribuce Metasploitable pro důvody názorného předvedení funkčnosti nástroje.



Obrázek 17- Nmap detail klienta

Další velmi zajímavou a hlavně užitečnou informací, kterou můžeme o našem testovaném klientovi zjistit za pomoci tohoto nástroje, je počet a některé další informace o uživatelích existujících na daném klientovi. K tomuto účelu nám poslouží jeden ze skriptů dostupných v námi používaném Nmap. Tento skript je vyvolán příkazem *nmap -script smb-enum-users.nse -p 445 <<IP adresa klienta>>.* Jak je možné vidět na výstupu z Nmaps na obrázku číslo 18, na našem testovacím stroji se nachází poměrně velké množství různých accountů, asi nejzajímavějším z nich je administrátorský účet zvýrazněný na daném obrázku červeným rámečkem.



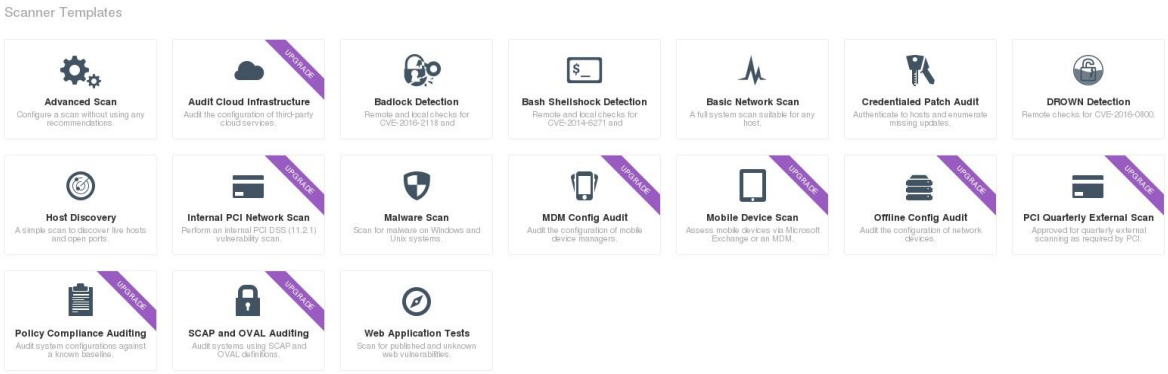
Obrázek 18- Uživatelé na testovaném klientovi

Díky využití Nmap jsme tedy byli schopní zjistit, že k naší testované síti je připojených pět klientu z 254 možných. Jedním z těchto klientů je objekt našeho testu s předem připravenými bezpečnostními riziky. Tento klient má přidělenou IP adresu 192.168.203.128. Provedli jsme tedy podrobnější test daného PC, při němž byla zjištěna přítomnost 35 uživatelských účtů včetně administrátorského s názvem msfadmin. Dále jsme byli schopni vyhledat otevřené porty a služby, které na nich momentálně naslouchají. Jako potencionálně rizikové faktory se zatím jeví běžící SQL a webserver.

Využití Nessus

Velmi silným nástroje pro oskenování sítě pro zjištění jejího rozsahu, připojených hostů a následné zjištění a ohodnocení bezpečnostních slabin a chyb sítě je komerčně hodně

využívaný nástroj Nessus Security Scanner. Tento nástroj umožňuje provést prakticky jakékoliv skenování potřebné pro zjištění informací o testované síti. Možnosti tohoto nástroje jdou ovšem daleko za pouhé testování počítačových sítí, v dnešní době jsou jeho pluginy používány pro testování v prakticky všech existujících odvětvích počítačové bezpečnosti. Je ho možné využít pro otestování zabezpečení webových aplikací, mobilních zařízení nebo třeba cloudových řešení. Možnosti, které nabízí verze používaná pro účely testu, jsou zachyceny na obrázku s číslem 19 níže. Pokud by pro účely testu dostupné moduly nevyhovovaly, existuje možnost stažení některého z přídavných modulů nebo vlastnosti existujících modulů zkombinovat a vytvořit si test vlastní, vyhovující přesně potřebám daného testování. Obrovskou výhodou tohoto nástroje je velmi propracované a uživatelsky přivětivé prostředí. Navíc existuje obrovská řada nastavení, které je možné při vytváření každého testu použít. Testy je možné naplánovat na určitou dobu, je možné nastavit rozsah scanu nebo dokonce i jaké porty a protokoly bude moci nástroj používat, aby neporušoval podmínky nastavené před začátkem testu. Po skončení scanu je možné si nechat výsledky vyexportovat do jednoho z mnoha formátů včetně PDF nebo HTML a tento soubor si společně s notifikací o skončení nechat odeslat na email.



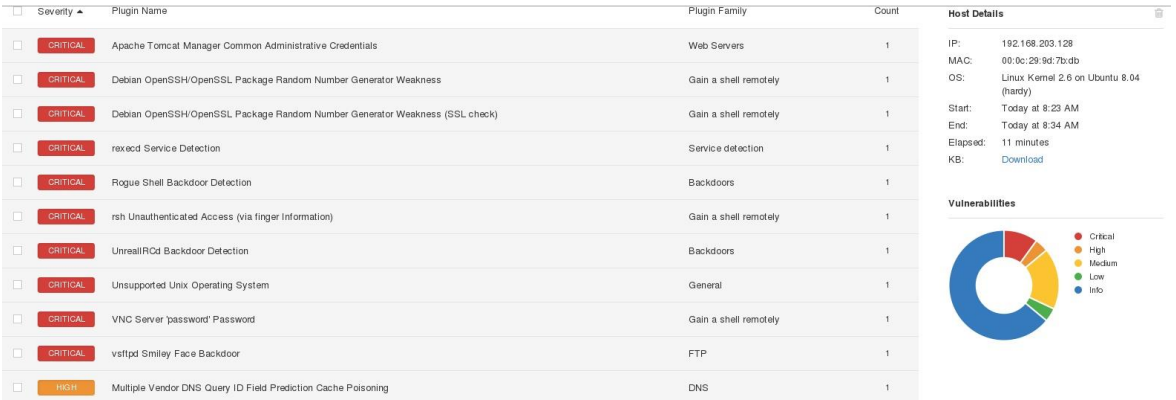
Obrázek 19- Nessus typy bezpečnostních scanů

Pro účely testů prováděných v praktické části této práce byla použita verze Home, která je zdarma a poskytuje všechny součásti potřebné pro otestování bezpečnosti počítačové sítě. V prvním kroku tedy opět potřebujeme vědět, kolik klientských PC se v cílové síti reálně nachází. Velikost sítě zjistíme stejným způsobem jako při použití Nmap, a to za pomoci běžného Linuxového nástroje Ifconfig, stejně jak je to ukázané v přechozí kapitole a na obrázku číslo 15. Poté co jsme určili velikost sítě, potřebujeme zjistit počet klientů, kteří se v síti reálně nacházejí. Toho docílíme provedením nové skenu užívajícího jeden z modulů dostupných v Nessus a nesoucí název Host Discovery. A jako cílovou oblast zvolíme celý

zkoumaný subnet stejně jako v případě použití Nmap v předchozí části. Poté co je sken dokončen, Nessus nám poskytne velmi přehledné výstupy z jeho průběhu. Na obrázku 20 je možné vidět výstup dokončeného testu na naší testovací síti. Jak je možné z obrázku vysledovat, test dopadl stejně jako v předchozím případě s použitím Nmap a bylo potvrzeno, že k síti je v době testu připojeno právě pět klientských PC.

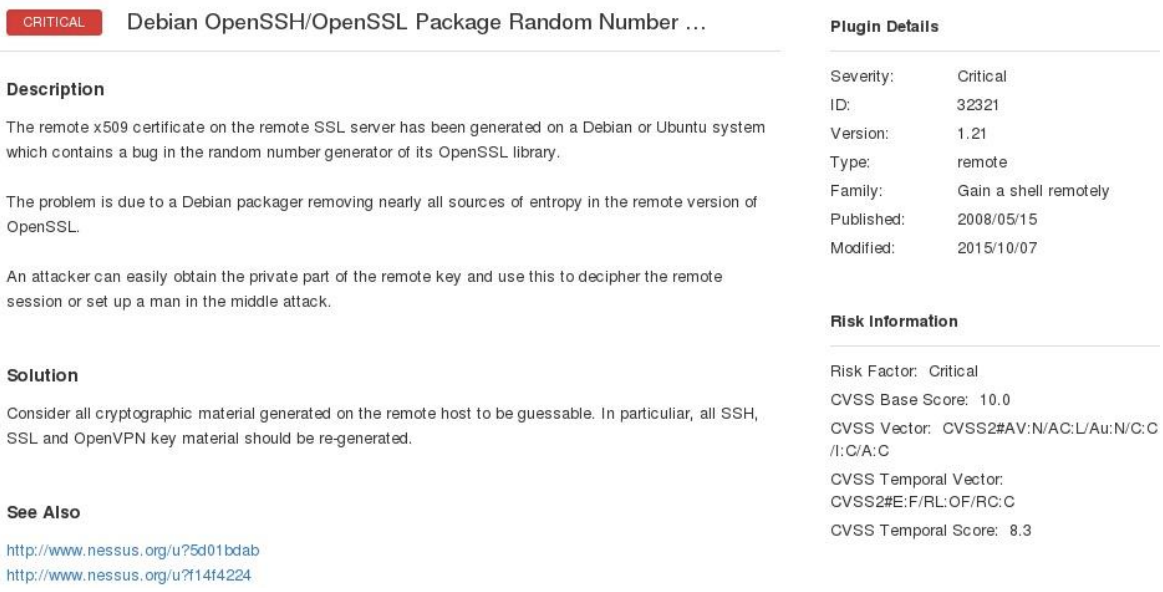
Obrázek 20- Nessus host discovery

V další fázi následuje zjišťování slabin pro jednotlivé nalezené klienty. Pro účely tohoto testu budeme však toto zjišťování provádět pouze pro klienta, na němž je nainstalovaný Metasploitable 2. Vytvoříme tedy nový sken, pro nějž použijeme modul s názvem Basic Network Scan, a použijeme ho proti námi zvolenému klientovi. Tento sken by měl odhalit většinu známých bezpečnostních chyb, které jsou na cílovém klientském počítači dostupné. Jak je vidět na výstupu skenu na obrázku číslo 21, Nessus tyto slabiny přehledně řadí dle jejich nebezpečnosti postupně od těch, které pouze poskytují potenciálnímu útočníkovi informace o daném PC, až po ty, které jsou kritické a dovolují útočníkovi převzít kompletní kontrolu nad cílovým systémem. Jako kritické navíc vyzdvihuje chyby, které jsou způsobeny nebezpečnými praktikami, jako je používání defaultních hesel pro přístup nebo administraci.



Obrázek 21- Výsledek testu Nessus

Každá takto nalezená slabina se dá rozkliknout na detail, kde lze o dané bezpečnostní hrozbě zjistit základní množství informací a jsou zde poskytnuty odkazy na zdroje pro zisk podrobnějších faktů. Každý tento detail navíc poskytuje řešení, díky kterému se dá daná bezpečnostní hrozba eliminovat, případně pomáhá jejímu předcházení v budoucnosti. Z pohledu bezpečnostního testera jsou v detailu důležité informace o tom, zdali existuje pro danou hrozbu možnost jejího zneužití, případně informace o tom, zda a kdy byla vydána bezpečnostní záplata. V neposlední řadě jsou velmi užitečné poskytované odkazy na záznamy v databázích bezpečnostních problémů. U některých známějších chyb existují dokonce odkazy na moduly Metasploit frameworku, které dokážou daný problém zneužít. Ukázka detailu bezpečnostního problému je zobrazena na obrázku číslo 22 níže.

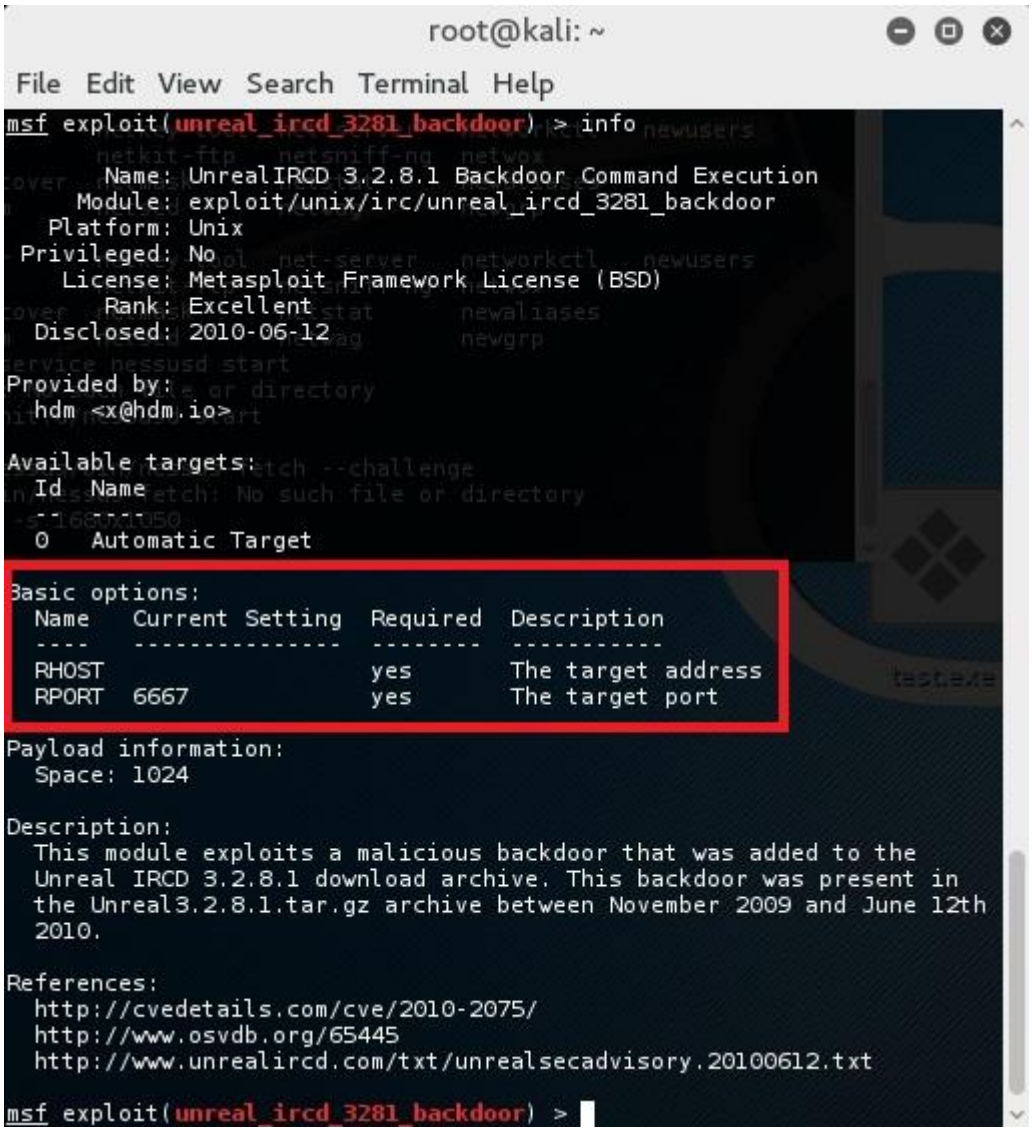


Obrázek 22- Detail bezpečnostního problému Nessus

6.6.2 Využití Metasploit framework

Poté co jsme dokončili zisk informací o zkoumané síti a povedlo se nám nalézt některé bezpečnostní problémy, je na čase zkusit tyto problémy zneužít a zhodnotit tak jejich opravdovou míru nebezpečnosti pro testovanou infrastrukturu. K tomu, abychom mohli využít některé z bezpečnostních chyb nalezených při prováděných skenech v předchozích kapitolách, využijeme pravděpodobně nejrozsáhlejší nástroj používaný pro tyto účely – Metasploit Framework. Tento nástroj je součástí základní distribuce Kali a dá se spustit přímo v terminálu za pomoci příkazu `msfconsole`. Metasploit obsahuje databázi bezpečnostních chyb, které umí využívat, v této databázi se dá vyhledávat použitím příkazu `search <<nazev exploitu>>`. Pokud chceme danou chybu použít, provedeme to přes

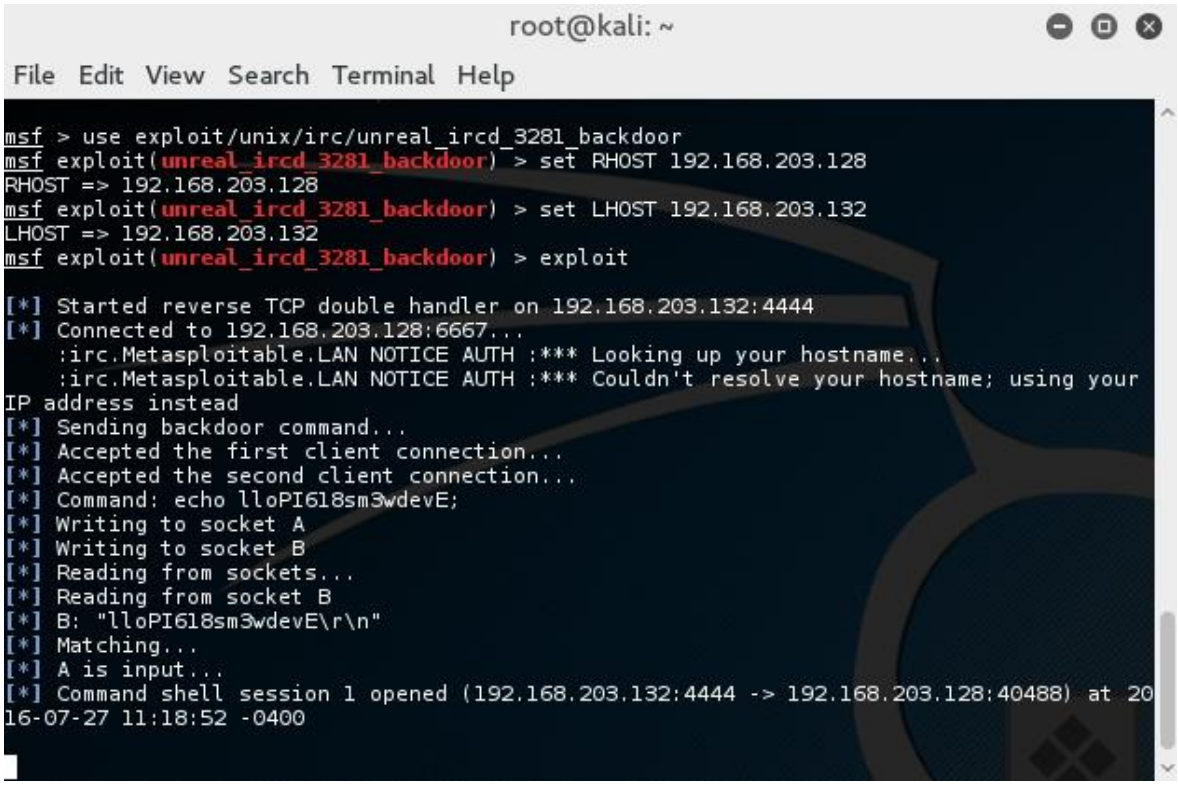
příkaz *use <<umístění exploitu>>*. Pro zjištění informací o používané chybě je nutné dotázat se na informační výpis, a to přes příkaz *info*. Před finálním zneužitím většiny těchto chyb je potřeba nastavit parametry, které jsou povinné (možno zjistit v informačním výpisu viz. Obrázek 23). Nastavení těchto parametrů provedeme pomocí příkazu *set <<název parametru>> <<hodnota parametru>>*. Pokud je vše správně nastaveno a přípraveno k použití, samotné spuštění provedeme přes příkaz *exploit*.



Obrázek 23- Metasploit exploit info

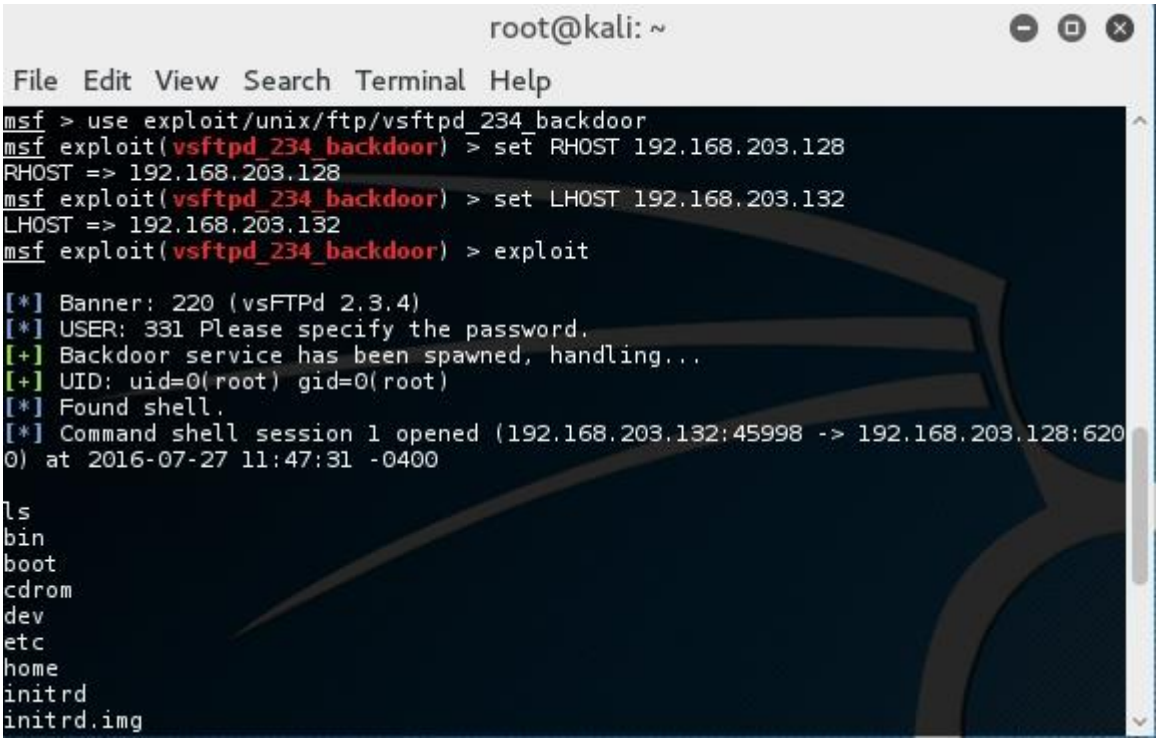
V přechozích fázích bylo nalezeno několik bezpečnostních chyb v podobě užití slabých nebo defaultních přihlašovacích údajů do různých služeb. Tyto chyby jsou samozřejmě poměrně jednoduché k zneužití, jelikož jediné, co stačí, je přihlásit se k dané službě s těmito údaji, proto slabiny tohoto typu nebudou v této práci dále zkoumány. První chybou, kterou se pokusíme zneužít, je UnreallRCDBackdoor. Jedná se o zkompromitovaný balíček IRC

serveru, který v sobě obsahuje backdoor, tato chyba dovoluje vzdáleně vyvolat jakýkoliv příkaz se stejnými právy, jako jsou práva, který má uživatel, který spustil IRC server. K tomu použijeme exploit Unreal IRCD 3.2.8.1, jak je doporučeno v detailu chyby ve výpisu z Nessusu. Dále nastavíme adresu vzdáleného hosta na IP adresu cílového počítače a adresu lokálního hosta na IP adresu našeho počítače. Poté již pouze spustíme exploit. Pokud je vše provedeno správně, měli bychom se dostat do vzdálené console. Celé provedení tohoto postupu je možné vidět na obrázku číslo 24 níže.



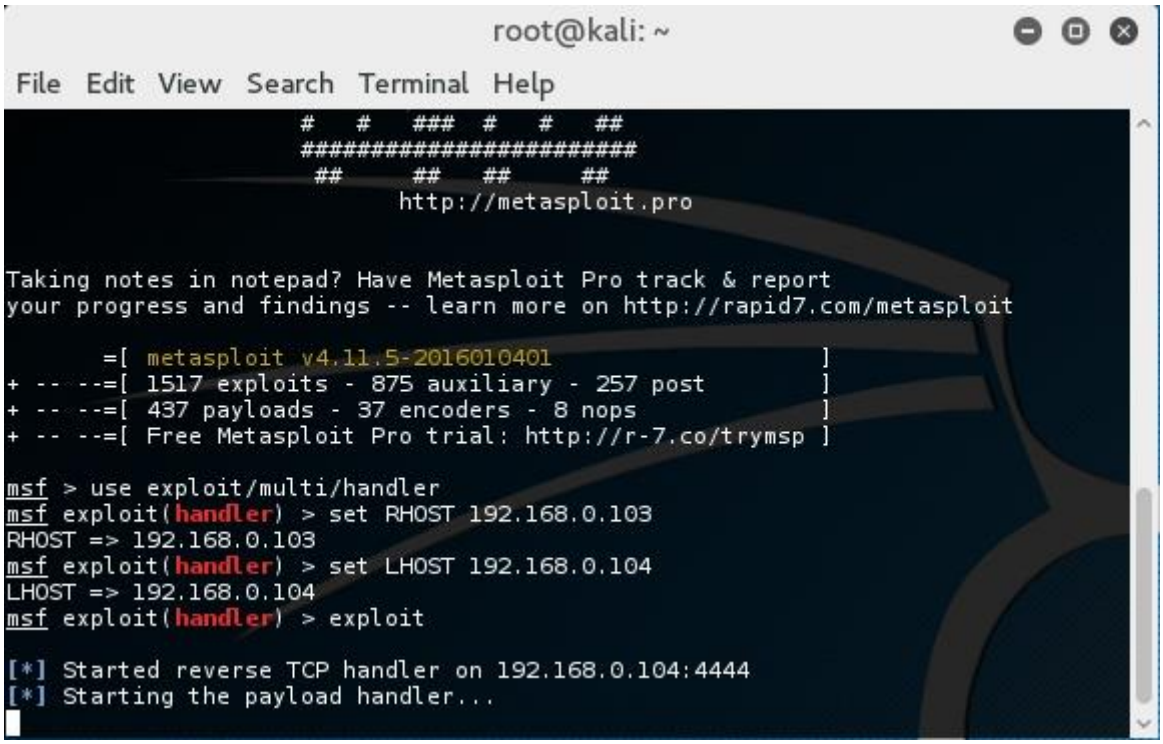
Obrázek 24- Zneužití Unreal IRCD

Druhou chybou, kterou jsme nalezli při skenování, je zkompromitovaný balíček vsftpd, který v sobě nese backdoor. Pokud se tedy útočník pokusí přihlásit na FTP server za pomoci uživatelského jména, které obsahuje ☺ , příkazová řádka začne naslouchat na TPC portu 6200 a umožní útočníkovi spouštět příkazy s root právy. Tento exploit můžeme opět poměrně jednoduše využít za pomoci existujícího modulu v Metasploit frameworku. Postup je velmi podobný tomu použitému v minulém příkladu, jedinou změnou je zde použití modulu VSFTPD 2.3.4, zbytek je stejný, nastavíme vzdáleného hosta na IP adresu vzdáleného PC, lokálního hosta na IP adresu lokálního počítače a spustíme exploit. Poté již bychom měli být připojeni a měli mít možnost zadávat příkazy do konzole. Celý proces a výsledek je možné vidět na obrázku číslo 25 na začátku následující strany.



Obrázek 25- Metasploit využití VSFTPD

Na závěr bych rád demonstroval široké možnosti využití Metasploit frameworku na příkladu, který ukazuje možnost vytvoření souboru, který dokáže na napadeném počítači spustit škodlivý kód a díky němuž je útočník schopen se na tento počítač připojit a ovládnout ho. Tento postup se sice netýká přímo sítě, kterou jsme testovali do této chvíle, nicméně tato ukázka by zde určitě neměla chybět. V první fázi si pomocí příkazu *msfvenom -p windows/meterpreter/reverse_tcp -LHOST <<IP adresa lokálního PC>> -LPORT <<Port lokálního PC>> -f <<koncovka souboru>> -e <<architektura souboru>>* vytvoříme spustitelný soubor, který nám v případě otevření na cílovém PC dovolí tento počítač ovládnout. Pustíme si Metasploit, zvolíme chybu, kterou chceme využít, nastavíme IP adresu lokálního PC, nastavíme adresu cílového PC a spustíme provádění exploitu. Celý tento proces je zachycen na obrázku 26. Pokud se soubor dostane na cílové PC a uživatel ho otevře, terminál přepne do nového zobrazení a my můžeme ovládat napadený počítač.



Obrázek 26- Užití multi handleru

7 Závěr

Z výsledků získaných při praktickém testu bezdrátových sítí je možné potvrdit jejich velmi vysokou nebezpečnost, a to hlavně z důvodu samotného funkčního principu standardu 802.11, který vyžaduje přenášení datových rámců volným prostorem, což dává potencionálnímu útočníkovi možnost tato data bez problému zachytit a vyčíst z nich důležité informace, které mu mohou posloužit k následnému úspěšnému provedení útoku. Dalším rizikem je obrovský prostor pro chybu při jejich nastavování nebo i samotném využívání.

Testy v praktické části ukázaly, že používání filtru MAC adres nebo možnost skrytí AP jsou z pohledu bezpečnosti naprosto nedostačující, jelikož se informace k jejich obejití dají vyčíst pouze za pomoci odposlechu dat z prostředí. Proto je doporučováno používat tyto funkce pouze jako doplňkové, v žádném případě jako jediný prostředek pro zabezpečení Wi-Fi sítě.

V teoretické části bylo řečeno, že zabezpečení typu WEP je zastaralé a Wi-Fi Alliance doporučuje ho nevyužívat. Jak se potvrdilo v praktickém testu, sítě využívající toto zabezpečení je možné poměrně jednoduše penetrovat za prakticky jakékoliv situace, a to dokonce i pokud neexistuje žádný klient, který by byl k síti připojen nebo se k ní nově připojoval. Sítě využívající zabezpečení WPA nebo WPA2 již používají bezpečnou šifru a jedinou možností jejich úspěšné penetrace je chyba na straně vlastníka nebo uživatele, tedy nejčastěji použití slabého PSK, který je možné uhodnout za pomoci jednoduchého slovníkového útoku. Pokud je síť nastavena pro použití zabezpečení WPA/WPA2 Enterprise, je možné ji penetrovat pouze při špatně provedeném nastavení sítě, tedy pouze v případě, kdy je nastaveno použití slabého protokolu. Dalším příkladem chyby při nastavování je povolení připojit se klientům, kteří neověřují identitu RADIUS serveru, což umožňuje jeho záměnu za falešný.

Nakonec jsem předvedl postupy penetračního testování, které je možné využít v případě, že máme před začátkem testování přístup do sítě, případně se nám ho povede získat v důsledku některého z prováděných testů.

V případě, že není možné obejít zabezpečení sítě pomocí chyby v použitém zabezpečovacím protokolu, je nutné hledat další chyby. V praktickém testu bylo ukázáno poměrně jednoduché proniknutí do sítě, která má zapnutou technologii WPS. Proto bych doporučil toto nastavení vůbec nepoužívat nebo ho používat na routeru, jehož software počítá s možností tohoto typu útoku.

8 Seznam použité literatury

[1] BALOCH, Rafay. *Ethical hacking and penetration testing guide*. Boca Raton: CRC Press, Taylor & Francis Group, 2015. ISBN 1482231611.

[2] ALLEN, Lee, Shakeel ALI a Tedi HERIYANTO. *Kali Linux: assuring security by penetration testing*. Birmingham: Packt Publishing, 2014. Community experience distilled. ISBN 978-1-84951-949-6.

[3]ALAMANNI, Marco. *Kali Linux Wireless Penetration Testing Essentials: Community Experience Distilled*. 1. Livery Place 35 Livery Street Birmingham B3 2PB, UK: Packt Publishing Ltd, 2015. ISBN 1785280856.

[4]VIVEK Ramachandran, Cameron Buchanan. *Kali Linux Wireless Penetration Testing: Beginner\'s Guide.* : Packt Publishing Ltd, 2015. ISBN 1783280425, 9781783280421.

[5]About. *Wireshark.* [online]. 22.1.2016 [cit. 2016-01-22]. Dostupné z: *https://www.wireshark.org*

[6]Bully. *Penetration Testing Tool.* [online]. 16.2.2014 [cit. 2016-01-22]. Dostupné z: *http://tools.kali.org/wireless-attacks/bully*

[7]Cowpatty. *Will hack for sushi.* [online]. 2009 [cit. 2016-01-22]. Dostupné z: *http://www.willhackforsushi.com/?page_id=50*

[8]DnsChef. *Penetration Testing Tool.* [online]. 18.2.2014 [cit. 2016-01-22]. Dostupné z: *http://tools.kali.org/sniffingspoofing/dnschef*

[9]FernWifiCracker. *Penetration Testing Tool.* [online]. 18.2.2014 [cit. 2016-01-22]. Dostupné z: *http://tools.kali.org/wireless-attacks/fern-wifi-cracker*

[10] High Level Organization of the Standard. *The Penetration Testing Execution Standard.* [online]. 16.8.2014 [cit. 2015-06-22]. Dostupné z: *http://www.pentest-standard.org/index.php/Main_Page*

[11] IEEE 802.11i. *Wikipedie.* [online]. 19.3.2016 [cit. 2016-04-20]. Dostupné z: *https://cs.wikipedia.org/wiki/IEEE_802.11i*

[12] ISECOM – Open Source Security Testing Manual (OSSTMM). *ISECOM – Open Source Security Testing Manual (OSSTMM).* [online].2015 [cit. 2015-06-22]. Dostupné z: *http://www.isecom.org/research/osstmm.html*

[13]Main. *[Aircrack-ng]*. [online]. 2016 [cit. 2016-01-22]. Dostupné z: *http://www.aircrack-ng.org/doku.php?id=Main*

[14]Metasploit pro documentation. *Metasploit documentation.* [online]. 24.1.2016 [cit. 2016-01-24]. Dostupné z: *https://help.rapid7.com/metasploit/index.html*

[15] OWASP Testing Project. *OWASP*. [online]. 19.4.2016 [cit. 2015-04-21]. Dostupné z:*https://www.owasp.org/index.php/Category:OWASP_Testing_Project#tab=Project_About*

[16]Reaver. *Penetration Testing Tool.* [online]. 18.2.2014 [cit. 2016-01-22]. Dostupné z: *http://tools.kali.org/wireless-attacks/reaver*

[17]Security Testing Frameworks. *Cisco Press.* [online]. [cit. 2015-06-25]. Dostupné z: *http://www.ciscopress.com/articles/article.asp?p=1606900&seqNum=4* [7] Web

[18]Application Security Consortium . *Web Application Security Consortium* . [online]. [cit. 2015-06-24]. Dostupné z:*http://www.webappsec.org/*

[19]What is Kali Linux ?. *Kali Linux.* [online]. 22.1.2016 [cit. 2016-01-22]. Dostupné z: *http://docs.kali.org/introduction/what-is-kali-linux*

[20]Wifi Honey. *DigNinja.* [online]. 22.1.2016 [cit. 2016-01-22]. Dostupné z: *https://digi.ninja/projects/wifi_honey.php*

[21]Wifite. *Penetration Testing Tool.* [online]. 2016 [cit. 2016-01-22]. Dostupné z: *http://tools.kali.org/wireless-attacks/wifite*

[22]AUTOR NEUVEDEN. *CS405 -- Project* [online]. [cit. 21.4.2016]. Dostupný na WWW: http://www4.ncsu.edu/~aliu3/802.bmp

[23]LEE, Micheal; BORCHERT, Otto; KAWAMURA, Satoshi. *A New Wireless Networking Security Scheme* [online]. [cit. 21.4.2016]. Dostupný na WWW: http://www.cs.ndsu.nodak.edu/~oborcher/network_security/fig1.gif

[24]RAMACHANDRAN, Vivek. BackTrack 5: Attacking the Client | PACKT Books [online]. [cit. 21.4.2016]. Dostupný na WWW: https://www.packtpub.com/sites/default/files/Article-Images/5580_06_33.png

[25]JOHNS, Aaron. Mastering Wireless Penetration Testing for Highly Secured Environments. 1. Livery Place 35, Livery Street, Birmingham B3 2PB, UK.: Packt Publishing Ltd., 2015. ISBN 978-1-78216-318-3.

[26]KENNEDY, David. Metasploit: the penetration tester's guide. San Francisco: No Starch Press, c2011. ISBN 978-1-59327-288-3.

[27]KUMAR, Himanshu. Learning Nessus for penetration testing. Birmingham, England: Packt Publishing, 2014. ISBN 978-1-78355-100-2.

9 Seznam obrázků

Univerzita Hradec Králové
Fakulta informatiky a managementu
Akademický rok: 2014/2015

Studijní program: Aplikovaná informatika
Forma: Prezenční
Obor/komb.: Aplikovaná informatika (ai3-p)

Podklad pro zadání BAKALÁŘSKÉ práce studenta

PŘEDKLÁDÁ:	ADRESA	OSOBNÍ ČÍSLO
Rydlo Jan	Nad Stadionem 1323, Nové Město nad Metují	I1300888

TÉMA ČESKY:

Využití penetračního testování v bezdrátových sítích

TÉMA ANGLICKY:

Usage of penetration testing in wireless networks.

VEDOUCÍ PRÁCE:

Mgr. Josef Horálek, Ph.D. - KIT

ZÁSADY PRO VYPRACOVÁNÍ:

Cílem práce je představit možnosti využití principů a postupů penetračního testování pro ověření a zajištění bezpečnosti bezdrátových sítí. V teoretické části práce autor představí základní principy a přístupy penetračního testování a etického hackingu s důrazem na ověření zabezpečení bezdrátových sítí s využitím specializovaných Linuxových distribucí. V praktické části pak autor prakticky otestuje a podrobně popíše metody odposlechu bezdrátových sítí a využije metody etického hackingu pro jejich prolomení. Autor navrhne, zrealizuje a otestuje využití nasazení Radius serveru jako autorizační autority.

Osnova práce:
Úvod
Principy penetračního testování
Analýza dostupných nástrojů
Návrh využití principů penetračního testování pro bezdrátové sítě
Praktické ověření navrženého řešení
Závěr

SEZNAM DOPORUČENÉ LITERATURY:

BALOCH, Rafay. Ethical hacking and penetration testing guide. xxvii, 503 pages. ISBN 14-822-3161-1.

ENGEBRETSON, Pat. The basics of hacking and penetration testing: ethical hacking and penetration testing made easy. Second Edition. xviii, 204 pages. ISBN 978-012-4116-443.

ENGEBRETSON, Daniel W. Basic security testing with Kali Linux: ethical hacking and penetration testing made easy. Second Edition. [S.l: s.n.], 2013, xviii, 204 pages. ISBN 978-149-4861-278.

DIETERLE, Daniel W. Kali linux: assuring security by penetration testing. Second Edition. S.l.: Packt Publishing Limited, 2014, xviii, 204 pages. ISBN 978-184-9519-489.

Podpis studenta: ..... Datum: 12.8.2016

Podpis vedoucího práce: ..... Datum: .....