UNIVERZITA HRADEC KRÁLOVÉ
Fakulta informatiky a managementu
Katedra informačních technologií

Problematika penetračního testování

Diplomová práce

Autor: Vladimír Nemanský
Studijní obor: Aplikovaná informatika

Vedoucí práce: Mgr. Josef Horálek, PhD

Hradec Králové listopad 2014

Prohlášení:

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

V Hradci Králové dne 10. 11. 2014 Vladimír Nemanský

Poděkování

Tímto bych rád poděkoval vedoucímu diplomové práce Mgr. Horálkovi, PhD za podporu, spolupráci, náměty a trpělivost při vedení této práce.

Anotace

Cílem této diplomové práce je představení problematiky penetračních testů a případové studie založené na využití nejnovějších metod a technologií využívaných v penetračním testování orientovaných na oblast energetických systémů. Tato práce poskytuje přehled o principech penetračního testování, jednotlivých fázích testování a nástrojích pro ně vhodných. Práce popisuje výhody penetračního testování, strategie a typy a představuje navrženou metodiku pro penetrační testování v oblasti energetiky.

Klíčová slova

Penetrační testování, etický hacking, Nmap, Nessus, OpenVAS, Kali Linux, Metasploit framework

Annotation

This Diploma Thesis introduces the issue of penetration tests and case study based on the latest methods and technologies used in penetration testing focused on the area of energy systems. The thesis provides an overview of the penetration tests, penetration testing phases and testing tools suitable for each phase. The thesis describes the benefits of penetration testing, strategies and types, as well as the methodology for penetration testing in energy systems area.

Keywords

Penetration testing, ethical hacking, Nmap, Nessus, OpenVAS, Kali Linux, Metasploit Framework

Obsah

Úvod

V moderní době, ve které žijeme, jsou všichni závislí na bezvýpadkovém provozu několika hlavních kritických systémů infrastruktury, jako je distribuce energie, hromadná doprava, atd. Operátoři těchto důležitých systémů jsou závislí na online komunikaci mezi jednotlivými částmi používaných počítačových systémů. Tyto systémy musí být adekvátně zabezpečeny před možnými kybernetickými útoky. Jako příklad mohou sloužit vznikající smart grid sítě, kde jsou klasické distribuční transformátory vybaveny monitorovacími a řídicími systémy, které vyžadují komunikaci s centrálním řídicím systémem. V takto rozsáhlém síťovém řešení představuje každý aktivní prvek vstupní bod do kritické infrastruktury energetické soustavy.

Nechvalně proslulý Northeast blackout z roku 2003 demonstruje potenciální nebezpečí a škody, které může představovat kaskádové selhání v elektrické síti způsobené poruchami v několika místech současně a zároveň s poruchou přenosu dat na nadřazený řídicí systém. Tato porucha odstartovala druhý nejrozsáhlejší blackout v historii, který ovlivnil 55 miliónů lidí. V srpnu 2003 nebyla energetická síť vybavena sofistikovanými monitorovacími a řídicími systémy, které jsou k dispozici dnes. Skutečnost, že v případě moderních smart grid sítí, kdy útočník může získat přístup do systému energetické soustavy a může tak vzdáleně ovlivnit chod řídicích systémů kdekoliv v síti, může způsobit výpadek napětí podobný této situaci v Kanadě a osmi státech USA.

Z tohoto důvodu je nutné se také v oblasti energetických systémů zabývat problematikou bezpečnosti informačních technologií, která je dnes jednou z nejvýznamnějších oblastí ve světě IT (Palmar, 2001) a nelze ji podceňovat. Dopady, které by mělo zneužití výpočetních systémů a dat v nich uložených, vzrůstají exponenciálně v závislosti na integraci informačních technologií do jednotlivých oblastí lidské činnosti.

Zajištění bezpečnosti informačních technologií je komplexní problematika vyžadující spolupráci specialistů z jednotlivých oblastí, jelikož dnes již není možné,

aby jedna osoba komplexně obsáhla problematiku všech oblastí a podoblastí IT. I přes veškerou snahu zabezpečit domácí, podnikovou či průmyslovou síť a všechny jejich součásti je nutné jednotlivá pravidla a technologie neustále testovat proti neoprávněnému vniknutí a zneužití, což se nejlépe provádí simulací samotného útoku pomocí tzv. penetračních testů.

Cílem této diplomové práce je tedy představení samotné problematiky penetračních testů a případové studie založené na využití nejnovějších metod a technologií využívaných v penetračním testování orientovaném na oblast energetických systémů.

Penetrační testování je komplexní metoda zahrnující aktivní analýzu systému pro případné zranitelnosti včetně nesprávné konfigurace systému, hardwarových a softwarových chyb a provozních nedostatků a technických protiopatření (BALOCH, 2014). Penetrační testování se zásadně liší od bezpečnostního testování funkčnosti. Funkčnost znázorňuje správné chování bezpečnostních kontrol systému, zatímco penetrační testování zjišťuje, zda je možné proniknout přes bezpečnostní kontroly v organizaci a získat neoprávněný přístup k informacím a informačním systémům. To se provádí simulací útoku na systém neoprávněným uživatelem buď pomocí automatizovaných nástrojů, nebo pomocí manuálních metod, případně kombinací obojího (ANGEL & SARALA, 2011).

Tato práce poskytuje přehled o penetračních testech, popisuje výhody penetračního testování, používané strategie a typy, stejně tak jako metodiku pro penetrační testování. První část práce patří úvodu do problematiky penetračního testování, seznámení s aktuálními dostupnými termíny a metodami, které se v současné době dynamicky a rychle vyvíjejí. V této kapitole je rovněž věnována pozornost dvěma hlavním způsobům testování označovaným jako externí a interní penetrační testování. V následující části je pak představena metodologie penetračních testů. Stejně jako ve všech komplexních oblastech IT je i pro penetrací testování vytvořeno několik celosvětově uznávaných metodik, které jsou představeny ve třetí kapitole, na níž navazuje představení vybraných nástrojů využitelných a vhodných právě pro penetrační testování. V páté kapitole vycházející z představených teoretických principů je pak představena případová studie zaměřená na vytvoření

obecně závazných pravidel pro testování systémů v oblasti energetiky, které budou využity jako základní zadávací dokumentace pro externí subjekty zabývající se bezpečnostním testováním informačních systémů. Tento proces je úmyslně zadáván externím subjektům, aby byla zajištěna maximální objektivita a relevantnost získaných výsledků.

1 Úvod do problematiky penetračního testování

Problematika penetračního testování a s ním spojených metod etického hackingu je dnes velmi aktuální téma, které se dynamicky vyvíjí a mění. Než budou představeny jednotlivé metody, postupy a nástroje pro penetrační testování a etický hacking, je nutné se nejprve podívat, co toto označení znamená a kde jsou jeho počátky. Je zřejmé, že primární vazba vede k dnes často skloňovanému slovu hacker. Lze dohledat celou řadu lepších či horších definic, proto je nejlépe začít slovníkem samotných hackerů (STEELE, et al., 2013), jenž se snaží nastavit pravidla pro používání nejen pojmu hacker, ale i dalších pojmů a žargonu využívaných v této oblasti. Samotného hackera pak lze dle tohoto The Hacker's Dictionary považovat za člověka, který je velmi nadaným počítačovým specialistou nebo programátorem. Jednoznačnou a krátkou definici však v článku neuvádějí. Tu lze naopak nalézt v Microsoft computer dictionary (Microsoft , 2002, p. 243), který má pro hackera dva významy:

„Hacker - a computerphile; A) a person who is totally engrossed in computer technology and computer programming or who likes to examine the code of operating systems and other programs to see how they work.

B) A person, more commonly considered a cracker, who uses computer expertise for illicit ends, such as by gaining access to computer systems without permission and tampering with programs and data.“

Kromě jasně definovaného pozitivního pohledu na hackera je zde zmíněn dnes velice rozšířený pohled na pojem hacker jako na specialistu využívajícího své znalosti v rozporu s dobrými mravy a zákony. Pro potřeby této práce a pohledu etického hackingu vnímáme pojem hacker v prvním z výše definovaných pohledů. Zde je nutno podotknout, že podle (RAYMOND, 2012) lze hackery rozdělit do několika zásadních skupin označovaných jako klasifikace hackerů.

Nejméně schopné hackery lze dle (RAYMOND, 2012) označit jako Script kiddle a v Microsoft computer dictionary (Microsoft , 2002, p. 467) je uvedena tato charakteristika *"Script kiddle - A would-be hacker who does not have the technical skills*

or knowledge needed for traditional hacking methods; one who relies on easy-to-use kiddie scripts."

Grey hat - takto je označován kvalifikovaný a někdy i certifikovaný hacker, který pro některé testy a útoky využívá nezákoné metody, avšak ne pro svůj zisk. Tímto se odlišuje od black hat i white hat, jejichž definice jsou uvedeny níže.

Black hat - někdy nepřesně označován jako cracker, je hacker, který zneužívá své vědomosti o počítačové bezpečnosti ke svému prospěchu při průnicích do informačních systémů. V Microsoft computer dictionary (Microsoft , 2002, p. 63), lze pak nalézt tuto přesnou definici "*Black hat - A hacker who operates with malicious or criminal intent. A black hat will break into a system to alter or damage data or to commit theft."* Zatímco cracker je dle Microsoft computer dictionary (Microsoft , 2002, p. 132) definován následovně: *"Cracker - A person who overcomes the security measures of a computer system and gains unauthorized access. The goal of some crackers is to obtain information illegally from a computer system or use computer resource. However, the goal of the majority is only to break into the system."*

White hat - neboli etický hacker, je obecně specialista na počítačovou bezpečnost, který se zaměřuje na penetrační a další testy počítačových sítí a informačních systémů pro zajištění jejich větší bezpečnosti. V Microsoft computer dictionary (Microsoft , 2002, pp. 566,) lze pak nalézt tuto přesnou definici *"White hat - A hacker who operates without malicious intent. A white hat will not break into a system with the intention of doing damage. White hats may be employed to provide security against other hackers."*, která je zcela v souladu s vnímáním etického hackera dle (RAYMOND, 2012).

Pro etického hackera je charakteristické, že na rozdíl od penetračního testování, které se zaměřuje na konkrétní cílový software nebo systém systematicky, využívá pro plnohodnotný útok i metod sociálního inženýrství a dalších hraničních metod pro získání potřebných informací. Mezi další často využívané metody pak patří i DoS útoky, nástroje jako Nessus či W3af, Metasploit apod. Lze říci, že zde významnou roli hraje i Performance monitor, který je dle Microsoft computer dictionary (Microsoft , 2002, p. 398) definováno jako *"Performance monitor is a process or*

program that appraises and records status information about various system devices and other processes."

Význam etického hackingu v dnešní době prudce narůstá, a z toho důvodu existuje i mezinárodní zkouška Certified Ethical Hacker, kterou garantuje celosvětově uznávaná organizace EC-Council[1] (The International Council of E-Commerce Consultants). Certifikace je zaměřena na komplexní znalosti z bezpečnosti informačních systémů a podnikových procesů, včetně metod sociálního inženýrství kryptografie a webhackingu či na tvorbu vlastních exploitů a malware. Zároveň se zaměřuje na implementaci bezpečnostních opatření v podnikovém prostředí, jako je PKI, Active Directory nebo firewally (GRAVES, 2010).

Význam a aktuálnost problematiky etického hackingu a penetračního testování je zřejmá i z množství dostupné literatury vycházející v několika posledních letech. Komplexní pohled na problematiku etického hackingu a penetračního testování podává novinka z léta roku 2014 Ethical hacking and penetration testing guide (BALOCH, 2014). Tato kniha je významná tím, že ačkoli nevyžaduje žádné předchozí zkušenosti s hackingem ani penetračním testováním, podává čtenáři komplexní pohled na využití operačních systémů pro etický hacking a upozorňuje na slabiny operačních systémů jak z rodiny Windows, tak s jádrem Linux. Dále představuje využití běžně používaných nástrojů pro diagnostiku sítě s důrazem na interpretaci získaných dat pro potřeby etického hackingu a diagnostiku slabých míst jak v operačních systémech, tak v testové počítačové síti. V neposlední řadě pak představuje širokou škálu nástrojů pro penetrační testování a to včetně Kali Linux, Google reconnaissance, MetaGooFil, Nmap, Nessus, Metasploit, Fast Track Autopwn, Hacker Defender rootkit atd. Kniha podává komplexní pohled na danou problematiku, umožní čtenáři pochopit význam jednotlivých nástrojů a technik a také správné prezentace informací pomocí nich získaných.

Praktickým využitím a prací s výše zmíněnými nejpoužívanějšími nástroji pro penetrační testování se na na konkrétních praktických příkladech zabývá The

[1] EC- counsil. [online]. 2014 [cit. 2014-10-16]. Dostupné z: http://www.eccouncil.org/Certification/certified-ethical-hacker

basics of hacking and penetration testing: ethical hacking and penetration testing made easy (ENGEBRETSON, 2013). Obdobný pohled, avšak výrazně podrobnější, pak nabízí (MCCLURE, et al., 2012) ve svém bestseleru Hacking exposed 7: network security secrets, které je již sedmým doplněným vydáním této knihy a s jehož staršími vydáními bylo možné se setkat i v českých překladech. Obě knihy se zabývají využitím běžně dostupných nástrojů, často přímo integrovaných v operačních systémech a aktivních síťových prvcích, pro odhalování slabých a nebezpečných míst na základě správné interpretace získaných dat. Navíc však představují i využití speciálních nástrojů, jako je Cain & Abel nebo distribuce Kali Linux, jež umožňují neoprávněné získávání informací či vstupu do datové komunikace nebo informačního systému. Je však nutné dodat, že všichni výše zmínění autoři důrazně upozorňují čtenáře, aby získané informace nezneužívali a dodržovali pravidla etického hackingu.

Jednotlivým nástrojům se pak autoři podrobně věnují v samostatných publikacích. Problematiku Kali Linuxu, jeho parametrů, využití a interpretaci získaných dat je možné nalézt u (BROAD & BINDNER, 2014). Využitím komplexního penetračního nástroje Nessus se zabývá například (KUMAR, 2014) a Metasploit framework je potom podrobně představen v (BALAPURE, 2013), včetně step-by-step postupů využití tohoto nástroje.

Na závěr je ještě nutné podotknout, že metody etického hackingu využívají i přístupy sociálního inženýrství a přesto, že tato práce věnuje této oblasti jen krátkou kapitolu, pokládá autor za podstatné se o této problematice v úvodu práce zmínit. Jak například uvádí (WATSON, 2014) ve své knize Social engineering penetration testing: executing social engineering pen tests, assessments and defense je sociální inženýrství, které se zaměřuje na nejslabší článek v zabezpečení, jímž jsou samotní lidé v organizaci, jednou z nejúčinnějších metod pro získání neveřejných informací či přímo konkrétních přístupů do informačních systémů a komunikačních sítí. Jak již bylo řečeno, této oblasti se předložená diplomová práce přímo nevěnuje, ale jedná se o nedílnou součást komplexního penetračního testu, kterou není možné v žádném případě zanebat.

O možnostech využití penetračního testování byla napsána celá řada příspěvků v odborných časopisech a vědeckých konferencích. Na závěr úvodu zde budou

představeny vybrané články, které jsou svým zaměřením či orientací podstatné pro téma diplomové práce. O obecných principech, využití a významu interního a externího penetračního testování pro zvýšení bezpečnosti webové aplikace souhrnně pojednává (ANGEL & SARALA, 2011) a obdobně i (SHRAVAN, et al., 2014). Významem výsledků penetračního testování a interpretací získaných dat z penetračního testování ve vztahu ke koncovému uživateli a jeho přístupu ke zvýšení bezpečnosti se ve svém článku Using penetration testing feedback to cultivate an atmosphere of proactive security amongst endusers (STYLES & TRYFONAS, 2009) zabývají Styles a Tryfonas. Jedním z komplexních přehledových článků pojednávajících o výhodách a nevýhodách penetračního testování, jeho pozitivních i negativních stránkách a metodách přístupů k penetračním testům je An overview of penetration testing (BACUDIO, et al., 2011). O výsledcích praktického využití penetračního testování v reálném prostředí pak pojednává příspěvek Results from the Deployment of A Targeted Security Testing Framework for the Testing of Email Systems in Local Government in Western Australia (LIMWIRIYAKUL & VALLI, 2011). Zde autoři představují zajímavé výsledky testů poukazující na slabiny poštovních serverů, na jejichž základě byla přijata doporučení pro zvýšení jejich zabezpečení.

Přístupy a možnosti využití etického hackingu, včetně poukázání na využívané techniky a postupy, jsou představeny v několika článcích, jako je Hacking Spaces: Place as Interface (WALLS, et al., 2009), dále (STROUPE, 2007) či (RAETHER, 2008). Společným aspektem těchto článků je důraz na význam etického hackingu jako jedné z nejspolehlivějších metod odhalení slabin či chyb v datových sítích a informačních systémech.

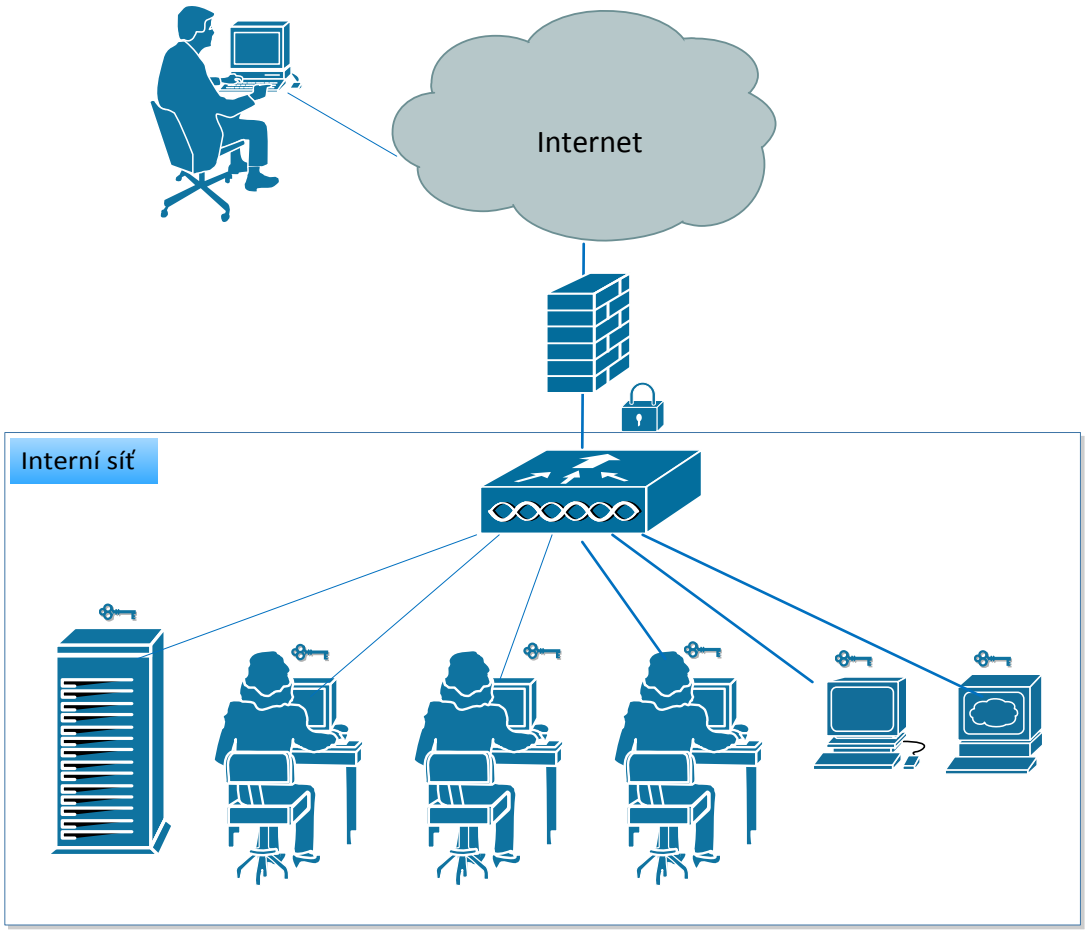
Pro úplnost analýzy dané problematiky je také nutno zmínit, že v databázích lze nalézt celou řadu článků a příspěvků z konferencí, jež se zabývají testováním či nasazením konkrétních nástrojů pro provedení penetračních testů jako součásti etického hackingu. Zde budou přehledově uvedeny jen ty nejvýznamnější a nejnovější jako je Global network security a vulnerability assessment of seven popular outsourcing countries (MALTEGO , 2014), Security vulnerabilities from inside and outside the eucalyptus cloud (GUSEV, et al., 2013), A preliminary cyber-physical security assessment of the robot operating system (MCCLEANA, et al., 2013)

a Penetration depth forecast using BP neural network-based system (Yuan, et al., 2014). Pro všechny výše zmíněné články je charakteristické, že využívají standardní nástroje a přístupy pro odhalení slabin systémů a komunikací.

Následující kapitola věnuje pozornost dvěma hlavním způsobům testování označovaným jako externí a interní penetrační testování.

1.1 Externí a interní penetrační testování

Na problematiku penetračního testování lze z pohledu připojení k síti dle (Selecký, 2012) nahlížet ze dvou směrů, jak ukazuje následující obrázek.



Obrázek 1 Externí a interní pohled na síť (zdroj: autor)

1.1.1 Externí testování

Externí penetrační test slouží k simulaci útoku na interní systémy z vnějšího prostředí. V tomto případě simuluje IT expert počínání hackera, který se snaží prolomit zabezpečení sítě z prostředí internetu. Testy prověří stupeň zabezpečení služeb a prvků z vnějšího prostředí se zaměřením na bezpečné připojení k internetu a ověření správné konfigurace a bezpečného nastavení prvků. Tester využívá informace, které jsou běžně dostupné libovolnému uživateli internetu, případně použije seznam IP adres a další informace, které mu dobrovolně poskytne zákazník. Test by měl proběhnout společně s testy sociálního inženýrství, fyzické bezpečnosti nebo bezdrátových sítí. Test zmapuje stav bezpečnosti systému z pohledu WAN rozhraní do LAN.

1.1.2 Interní testování

Dalším možným testem je simulace útoku z vnitřního prostředí, kdy tester simuluje chování interního uživatele s nekalými úmysly, který je připojen do vnitřní sítě a snaží se o neoprávněný přístup k důvěrným informacím společnosti. Takový test prověří v praxi bezpečnostní opatření, která mají interním uživatelům zabránit získat neoprávněně data. V praxi typickou hrozbou může být nespokojený zaměstnanec s cílem poškodit nebo odcizit firemní data. Útok může rovněž simulovat průnik cizí osoby bez znalosti prostředí s přístupem do interní sítě a to včetně vzdáleného přístupu. Účelem testu je zvýšit úroveň zabezpečení stanic a síťových prvků a také informačních systémů organizace z vnitřní strany.

2 Etický hacking

Jak je popsáno v (Palmar, 2001), růst využití internetu přinesl mnoho dobrých věcí, například elektronické obchodování, přístup k široké zásobě referenčních materiálů, sdílení dat, e-mail, nové cesty pro reklamu a distribuci informací apod. Stejně jako u většiny technologických pokroků je tu ovšem také stinná stránka a to nebezpečí napadení a zneužití informací organizace hackery. Vlády, firmy a soukromé osoby po celém světě chtějí být součástí této revoluce, ale zároveň se obávají situace, kdyby některý hacker pronikl do jejich webového serveru a změnil jeho obsah, manipuloval s firemními e-maily, ukradl údaje o kreditních kartách z on-line nákupního místa, nebo vpašoval do firmy software, který bude tajně přenášet obchodní tajemství jejich organizace do otevřené internetové sítě. S vyřešením podobných hrozeb může pomoci etický hacking.

Slovní spojení „etický hacking“ zní na první pohled nelogicky, jakoby se význam těchto slov navzájem vylučoval. Slovo hacking totiž působí často negativně, a proto za ním hledáme automaticky něco škodlivého. Pokusme se tento pojem vysvětlit. Etický hacking a penetrační testování jsou příbuzné pojmy. Na penetrační testy lze pohlížet jako na aplikování etického hackingu na výpočetní infrastrukturu. Literatura často tyto pojmy nerozlišuje.

Etický hacking je dle (Palmar, 2001) činnost prováděná počítačovými experty za účelem odhalení chyby v zabezpečení počítačové sítě nebo operačního systému, kdy se následně zkoumá, zda jsou zabezpečovací mechanismy v souladu s firemní politikou. Jedná se o jedinou bezpečnou simulaci útoku, kdy dojde k analýze výsledků a zjištěných nedostatků. Výsledný report je následně posouzen kompetentními osobami, které zpracují návrh opatření vedoucích k odstranění bezpečnostních hrozeb a ke zlepšení bezpečnostní infrastruktury. Etický hacker jedná v souladu s pravidly organizace, pro kterou se snaží chyby v zabezpečení odhalit a nemá v úmyslu této společnosti škodit, proto jsou testy vedeny nedestruktivním způsobem. Na druhou stranu klasický hacker hledá možnost, jak využít slabiny v systému, za účelem zcizení či změny dat, případně spáchání jiných škod.

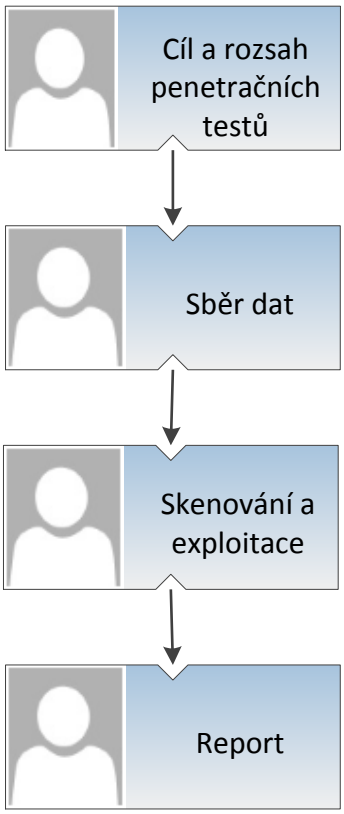
Firmy, jejichž systémy prošly penetračním testováním, běžně neposkytují výsledky testů z důvodu utajení bezpečnostních děr. Sdílení podobných informací představuje vysoké riziko následného zneužití. I přesto byl před časem uveřejněn skupinou testerů způsob, kterým se jim podařilo proniknout do sítě jedné ze zadavatelských společností penetračního testu. Jednalo se o společnost, která se umístila v žebříčku Fortune 500 (Dark reading, 2011), což je každoroční žebříček sestavený a vydaný časopisem Fortune, který řadí 500 amerických soukromých a veřejných korporací podle jejich hrubého obratu. Pro získání důležitých informací byla využita nezabezpečená ústředna, pomocí které získávali hlasové zprávy určené technické podpoře uživatelů. Z uživatele, který zavolal na helpdesk a žádal o radu k připojení do VPN, se jim podařilo za pomoci sociálního inženýrství získat přístupové informace do sítě. Elektronický článek pojednávající detailněji o tomto testu je dostupný ze zdroje (Dark reading, 2011).

Je nutné podotknout, že penetrační trester musí mít před započetím testu výslovný a písemný souhlas majitele nebo správce testovaného informačního systému. V opačném případě se jedná o konflikt se zákonem. Proto by každý, kdo realizuje penetrační test, měl být seznámen s právními aspekty této problematiky. V případě České republiky řeší právní stránku průniku do systému a změny či zcizení informací následující paragrafy zákona č. 40/2009 Sb., trestního zákoníku v aktuálním znění (Trestní zákoník, 2009):

- §230 Neoprávněný přístup k počítačovému systému a nosiči informací,
- §231 Opatření a přechovávání přístupového zařízení a hesla k počítačovému systému a jiných takových dat,
- §232 Poškození záznamu v počítačovém systému a na nosiči informací a zásah do vybavení počítače z nedbalosti.

Z uvedených paragrafů vyplývá, že je třeba vždy před započetím penetračního testu stanovit rozsah testu a hranice, které penetrační tester za žádných okolností nesmí překročit.

Penetrační test se skládá z několika fází, kdy na výstup z jedné fáze navazuje fáze následující. Pro penetrační testování existují různé metodiky, některé z nich jsou popsány ve třetí kapitole. Metodiky se svým pohledem na členění penetračního testu mohou lišit. Jeden z možných pohledů na členění průběhu penetračního testu lze vidět na obrázku 2.



Obrázek 2 Možné fáze penetračního testování (zdroj: zpracováno dle (OWASP, 2013))

Pojmy etický hacking a penetrační testování se navzájem prolínají a jejich definice obsahují mnoho společného. Důležité je si uvědomit, že penetrační testování slouží k simulaci útoku hackera, odhalení nebezpečných trhlin v zabezpečení informačních systémů a zamezení možným škodám nápravnými opatřeními před jejich vznikem.

3 Metodiky penetračního testování

Tato kapitola se věnuje metodikám penetračního testování, popisuje informace, které tyto metodiky obsahují a v závěru porovnává metodiky OSSTMM a NIST SP 800-115. Metodik existuje širší skála.

3.1 NIST SP 800-115

Autorem metodiky NIST Special Publication 800-115 je agentura National Institute of Standards and Technology při Ministerstvu obchodu Spojených států amerických. Dle (NIST, 2014) byla agentura založena v roce 1901 s cílem podpořit inovace a průmyslovou konkurenceschopnost USA v té době před Anglií, Německem a dalšími ekonomickými soupeři. Dnes podporuje NIST měření od nejmenších nano technologií přes zemětřesení odolné mrakodrapy až po globální komunikační sítě.

Metodika je veřejně přístupná z webových stránek organizace NIST (NIST, 2014). Publikace NIST SP 800-115 se skládá z osmi kapitol a příloh A – G. V dalších odstavcích budou prezentovány klíčové informace z této metodiky.

Kategorie posuzování způsobu zabezpečení informačního systému:

- examination (šetření),
- testing (testování),
- interviewing (pohovory se specialisty)

Šetření nepatří mezi invazivní metody a základem je studium interních směrnic organizace, které se týkají bezpečnosti. Dále je nutné prostudovat obsahy logů prvků infrastruktury (např. IDS/IPS systémy, servery apod.) stejně jako analyzovat konfigurační soubory klíčových zařízení infrastruktury, které jsou součástí penetračních testů.

Metoda testování již může být velmi invazivní, a proto je potřeba k ní přistoupit opatrně a rozvážně, aby vysoká invaze testů neměla za následek ovlivnění chodu testovaného systému a nevedla k odstavení systému a vysokým ekonomickým

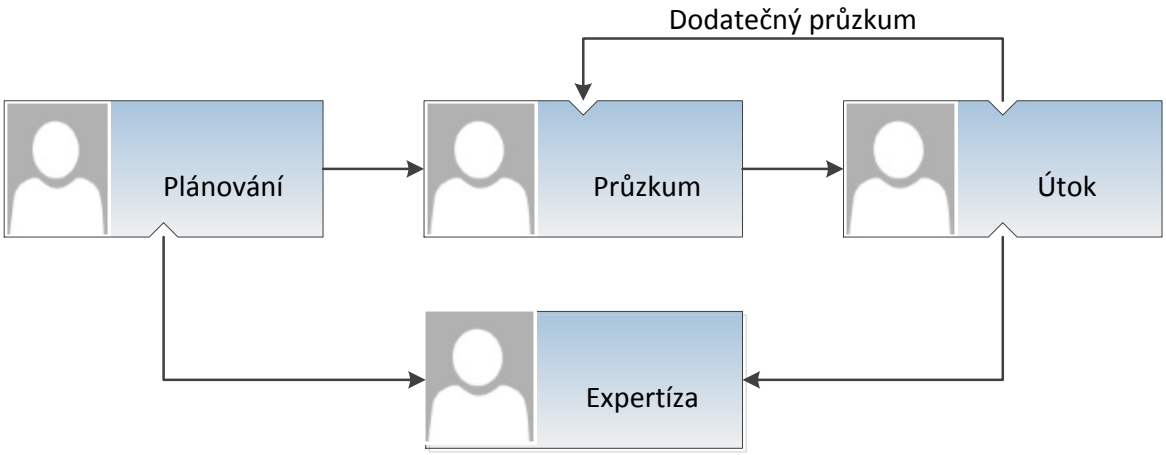
ztrátám. Cílem testování jsou přímo koncové systémy, kde pomocí simulovaných útoků zkoumáme jejich zranitelnost a slabiny.

Metoda pohovorů spočívá ve vedení dialogů se zaměstnanci z různých odělení a různého postavení, které mají za úkol pochopit fungování bezpečnostních mechanismů v organizaci, identifikovat klíčové cíle a odhalit případné problémy.

V dalších částech metodiky jsou popsány různé druhy nástrojů pro identifikaci cílů, programového vybavení, zranitelností takového programového vybavení a doporučený průběh penetračního testu. Je zde zmíněna i sada nástrojů SCAP. Jedním z nástrojů je jazyk pro výměnu informací o bezpečnostních chybách software a nastavení softwarových produktů na bázi a také sada šablon obsahujících doporučené konfigurace běžných OS a bezpečnostně optimalizovaných softwarových produktů, které lze nastavit v souladu s doporučeními NIST.

Tester provádějící penetrační testy musí před každým testováním aktualizovat všechny nástroje tak, aby obsažené signatury a vzorky byly v době testu aktuální. Pokud tak tester neučiní, nemusí být provedený test relevantní. V případě provádění externích a interních testů je výhodnější provést nejprve test externí. Pokud by tester provedl nejprve interní testy, postupoval by z pohledu vnitřní struktury organizace za pomocí informací, které běžný hacker nemá k dispozici. Takové informace samozřejmě poskytují dodatečnou výhodu při provádění externích testů.

Dále metodika poskytuje informace ohledně organizačních záležitostí a fázování během penetračního testu. Jak již bylo zmíněno, každá metodika je specifická a poskytuje jiný pohled na fázování penetračního testu. Metodika NIST vidí v penetračním testu jako jednu z fází procesu také ověření možných zranitelností. Na následujícím obrázku 3 jsou znázorněny fáze penetračního testu dle doporučení NIST.



Obrázek 3 Jednotlivé fáze penetračního testování dle NIST800-115 (zdroj: zpracováno dle (NIST, 2014))

Metodika v závěru popisuje, nad čím je nutné se před započetím penetračního testu zamyslet. Tester by měl brát v potaz následující faktory:

- vymezení časového rozsahu pro penetrační test,
- v případě nedostatečného časového rozsahu je nutné prvkům infrastruktury přidělit prioritu, podle níž bude proveden test,
- je nutné stanovit periodicitu testování
- lze provést test v prostředí firemní infrastruktury, nebo je z důvodu hrozícího rizika rozumnější vytvořit model firemní infrastruktury a ten testovat,
- je známý způsob zajištění ochrany dat během testu včetně uložení a likvidace,
- jsou dostatečně informovány všechny osoby, kterých se testování týká.

Na některé z těchto otázek lze nalézt odpovědi v dalších metodikách od agentury NIST. Definice priorit u testovaných systémů řeší například metodika FIPS 199 (Federal Information Processing Standard Publication 199), která je standardem vlády Spojených států amerických a stanoví bezpečnostní kategorie informačních systémů používaných federální vládou jako jednu složku posouzení rizik. FIPS 199, společně s FIPS 200, jsou povinné bezpečnostní standardy podle požadavků FISMA (Federal Information Security Management Act of 2002) viz (NIST, 2014) .

FIPS 199 vyžaduje, aby federální agentury přistupovali k informačním systémům v každé z kategorií důvěrnosti, integrity a dostupnosti s hodnocením nízkého, středního nebo velkého dopadu v každé kategorii. Nejzávažnější hodnocení z kterékoli kategorie se stane pro informační systém celkovou bezpečnostní kategorizací.

Na metodiku FIPS navazují i další metodiky od agentury NIST, jako je například publikace SP 800-600, která popisuje kategorizaci informačních systémů dle metodiky FIPS 199. Publikace je rovněž volně dostupná na stránkách NIST (NIST, 2014).

Metodika NIST SP 800-115 odpovídá na řadu dalších otázek. Mimo jiné doporučuje vytvoření písemné strategie testování, která bude řešit výše zmíněné otázky a dále upozorňuje na nutnost vyřešit právní rámec testování, který bude definovat oblasti testerovi povolené a naopak zakázané, aby se předem zabránilo možným škodám, které při testování mohou vzniknout.

Jediným výstupem testování je, podle metodiky, závěrečná zpráva obsahující výsledky testování, které jsou určeny pro management organizace. Tester by měl rovněž vypracovat doporučení vedoucí k nápravám nedostatků zjištěných během penetračního testu.

Metodika v závěru uvádí praktické informace. Například v příloze A můžeme nalézt seznam nástrojů určených k penetračnímu testování. U každého nástroje je potom uvedeno, pro jakou část penetračního testování je nástroj vhodný. Příloha B uvádí pravidla pro šablony zakázek, které individuálně pomohou organizacím zahrnout informace do těchto šablon. Příloha C se věnuje zkouškám bezpečnosti a testování aplikací. Testování bezpečnosti a vyšetření aplikací pomůže organizaci určit, zda vlastní aplikační programy, například webové aplikace, jsou zranitelné a mohou být zneužity a zda se software chová a komunikuje bezpečně se svými uživateli, jinými aplikacemi (například databázemi) a jeho spouštěcím prostředím. Příloha D popisuje testování vzdálených přístupů, pokrývá technologie jako VPN, SSH

tunely, aplikace pro vzdálenou plochu, modemy apod. V příloze E lze najít tabulku obsahující další metodiky, normy a doporučení, které se týkají informační bezpečnosti, včetně odkazů na on-line zdroje těchto metodik. Příloha F je slovníček pojmů použitých v metodice a konečně v poslední příloze G můžeme najít použité zkratky.

Další z používaných metodik je OSSTMM (Open Source Security Testing Methodology Manual), kterou popisuje následující kapitola.

3.2 **OSSTMM**

Autorem této metodiky je otevřené společenství a nezisková organizace ISECOM (Institute for Security and Open Methodologies) (ISECOM , 2014), která v lednu 2001 začala vydávat manuál OSSTMM (Open Source Security Testing Methodology Manual). Organizace ISECOM je oficiálně registrována v Katalánsku ve Španělsku a má pobočky v Barceloně a v New Yorku. Financování organizace je založené na partnerství, předplatném, certifikátech, licencích, seminářích a soukromých výzkumných dotacích.

Metodika OSSTMM prošla několika verzemi. Poslední oficiální verze č. 3 pochází ze 14. 12. 2010 a lze ji získat v elektronické podobě na stránkách ISECOM (HERZOG, 2014). Pro registrované partnery je na stejných stránkách k dispozici i pracovní verze č. 4 stejně jako metodika OSSTMM Web App Draft. Oproti předchozí popsané metodice SP800-115 klade OSSTMM větší důraz na teorii a zabývá se problematikou penetračního testování hlouběji.

V úvodu dokument vymezuje pojmy z oblasti bezpečnosti, auditu a penetračních testů. Dále pak představuje filosofii a chápání bezpečnosti a souvisejících termínů, včetně představení příkladu možného penetračního testu.

Metodika OSSTMM vidí míru zabezpečení ve zlepšování kontrolních mechanismů pro zvýšení bezpečnosti. Tyto mechanismy člení metodika do deseti kategorií. Novými pojmy pro další studium metodiky jsou například control (kontrolní mechanismus interakcí), vector (směr interakce), attack vector (cesta

vedení útoku), attack surface (část vektoru, kde mohou selhat kontrolní mechanismy), limitation (omezení funkčnosti kontrolních mechanismů), apod.

Definice testované úrovně zabezpečení a výstupu tohoto testu jsou metodikou podrobně definovány v kapitole 2.1 sedmi body. Výstupem testu je identifikace částí vektorů, ve kterých selhaly kontrolní mechanismy v různých směrech interakce. V kapitole 2.2 jsou úrovně interakce rozděleny do těchto tříd a kanálů:

- fyzická bezpečnost (PHYSSEC)
  - fyzický kanál – interakce lidí s prostředím
  - lidský faktor – interakce fyzická nebo psychologická
- spektrální bezpečnost (SPECSEC)
  - wireless - bezdrátová komunikace,
- komunikační bezpečnost (COMSEC)
  - telekomunikace – telekomunikační sítě
  - datové sítě – metalické datové sítě

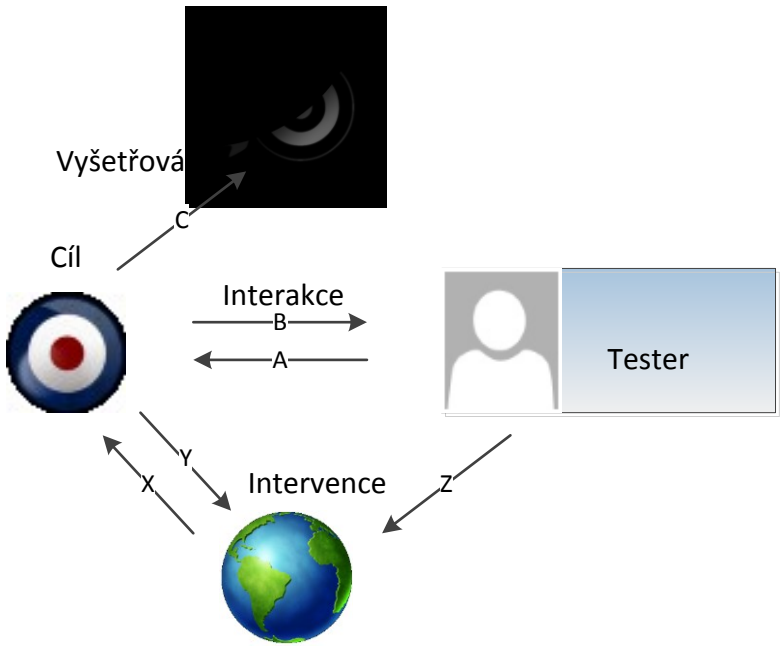
Zatímco kanály a jejich rozdělení mohou být zastoupeny jakkoliv, v tomto manuálu jsou organizovány jako rozeznatelné prostředky komunikace a interakce. Tato organizace je navržena tak, aby usnadnila testovací proces a zároveň minimalizovala neefektivní režii, která je často spojena s přísnou metodikou.

Další kapitoly popisují společné typy testů, jejich rozdělení do šesti kategorií podle znalostí útočníků o cíli a o znalostech cíle o útoku. Některé z testů budou sloužit spíše pro testování znalostí testera než testování bezpečnosti cíle. Kapitola 2.4 definuje pravidla provozních pokynů, přijatelných postupů v oblasti marketingu a prodeji testování, provádění testovacích prací a manipulace s výsledky testů.

Kapitola 2.6 metodiky OSSTMM se zabývá čtyřfázovým prováděním penetračního testu. Four Point Process (4PP) rozkládá test od začátku až po závěrečnou zprávu.

Čtyři fáze penetračího testování podle OSSTMM:

- Induction (Uvedení): **Z**
- Inquest (Vyšetřování): **C**
- Interaction (Interakce): **A/B**
- Intervention (Intervence): **X/Y/Z**



Obrázek 4 Fáze penetračního testování dle OSSTMM (zdroj: zpracováno dle (ISECOM , 2014))

Během fáze uvedení (Z) je třeba definovat časový rámec testu, rozsah a právní normy a rozhodnout se pro typy testů, které mají být použity. Fáze interakce (A/B) slouží ke zjištění cílů, které bude test pokrývat, ke zmapování kontrolních mechanismů chránících tyto cíle a k zaznamenávání informací. Cílem fáze vyšetřování (C) je zjistit maximum informací často i z veřejných zdrojů o cílech spadajících do obsahu testů a vyhodnocení získaných informací. Takto lze například zjistit, kdo má přístup k jakému typu informací. Poslední fáze intervence (X/Y/Z) slouží k ověření funkčnosti kontrolních a poplašných mechanismů.

Penetračním testům datových sítí se věnuje jedenáctá kapitola metodiky, která je rozdělena do podkapitol podle problematiky (například audit viditelnosti

v podkapitole 4, verifikace přístupu v podkapitole 5 apod.). Informace uvedené v jedenácté kapitole slouží jako příprava pro testování, řeší se zde i otázky průběhu testu, funkce poplašných mechanismů, jako je IDS (Intrusion Detection System) a správná funkce ochrany dat.

Metodika OSSTMM využívá metriku zranitelnosti RAV (Risk Assessment Values), která pomocí číselné hodnoty vyjadřuje, jaký stav odpovídá testované infrastruktuře a zda je v pořádku, případně jaká je odchylka od normálního stavu. Této problematice se věnuje čtvrtá kapitola metodiky OSSTMM. Na základě hodnot RAV lze odhalit slabá místa. Nástroj pro výpočet RAV v excel formátu, který se jmenuje RAV Calculator, je k dispozici na stránkách organizace ISECOM (ISECOM , 2014). Funkce nástroje spočívá v zadání údajů testerem do tabulkového procesoru, kde poté získáme hodnotu, která vyjadřuje úroveň ochrany sítě v procentech. Na základě výsledku testu je třeba odstranit případné nedostatky a eliminovat možné bezpečnostní hrozby.

Zdroj (HERZOG, 2013) popisuje princip výpočtu RAV. Jako vstupy pro výpočet hodnoty RAV slouží hodnoty, které zobrazuje tabulka 1. Vstupní hodnoty jsou rozděleny do tří kategorií, které tvoří Provozní bezpečnost (OpSec), Omezení (ActSec) a Řízení ztrát (LC). Provozní bezpečnost potom vyjadřují údaje jako Visibility (viditelnost), Access (přístup) a Trust (důvěra). Viditelnost představuje počet testovaných cílů, které jsou dostupné a viditelné z internetu. Přístup vyjadřuje hodnotu závislou na úrovni interakce. Touto hodnotou může být počet otevřených TCP/IP portů, které spadají do testovaného rozsahu. Důvěra představuje počet důvěrných vztahů mezi cíli v testovaném systému.

Tabulka 1 Vstupní údaje pro výpočet RAV (Risk Assessment Values)

Operational Security - Provozní bezpečnost (OpSec)	1. Visibility - viditelnost
	2. Trusts - důvěra
	3. Accesses - přístup
Actual Security - Omezení (ActSec)	1. Vulnerabilities - chyby
	2. Weaknesses – slabé stránky
	3. Concerns - obavy
	4. Exposures - expozice
	5. Anomalies - anomálie
Loss Controls - Řízení ztrát (LC)	1. Authentication - ověření
	2. Repudiation - odstoupení
	3. Confidentiality - důvěrnost
	4. Privacy – ochrana osobních údajů
	5. Indemnification – pojistné plnění
	6. Integrity -integrita
	7. Safety - bezpečnost
	8. Usability - použitelnost
	9. Continuity - kontinuita
	10. Alarm - alarm

Výpočet hodnoty RAV vyžaduje, aby každé z těchto kategorií bylo přiděleno základní číslo, které představuje buď procento rozsahu (Scope) chráněného bezpečnostními opatřeními/řízením ztrát ($OpSec_{base}$ a LC_{base}), nebo mírou rizika, které problémy způsobují ($ActSec_{base}$). Následující rovnice se používají k výpočtu prvních dvou základních čísel se všemi proměnnými Sum, které jsou součty složek vstupů z výše uvedené tabulky:

$$OpSec_{base} = 100 - \frac{OpSec_{Sum}}{(Scope + OpSec_{Sum})}$$

$$LC_{base} = Scope \times \frac{LC_{Sum} \times 0.1}{(Scope + OpSec_{Sum})}$$

Součet vstupů řízení ztrát je násoben hodnotou 0,1. Důvodem je kategorie Řízení ztrát, která má do celého RAV vzorce pouze jeden vstup. Z toho důvodu musí být vynásobena hodnotou 0,1, aby bylo možné normalizovat všech 10 kategorií pouze do jednoho vstupu.

Omezení je vypočteno na základě hodnot, které rozlišují mezi Identifikovaným problémem (problém identifikovaný na základě pohovoru, skenování portů, detekce zranitelnosti nebo předpoklad založený na konfiguraci systému) a Verifikovaným problémem (problém manuálně ověřený auditorem). Následující tabulka se používá pro výpočet proměnné $ActSec_{Sum}$, jako mezistupeň mezi vstupy Omezení a proměnné $ActSec_{base}$, která slouží jako základní vstupní hodnota Omezení pro výpočet RAV.

Tabulka 2 Výpočet hodnoty Omezení dle zdroje (HERZOG, 2013)

Vstup	Verifikovaná hodnota	Identifikovaní hodnota
Chyby	$\left(\frac{100 - OpSec_{base}}{LC_{base}}\right)\frac{1}{100}$	(Verifikovaná chybová hodnota)/2
Slabé stránky	(Verifikovaná chybová hodnota)/2	(Verifikovaná chybová hodnota)/3
Obavy	(Verifikovaná chybová hodnota)/3	(Verifikovaná chybová hodnota)/4
Expozice	(Verifikovaná chybová hodnota)/4	(Verifikovaná chybová hodnota)/5
Anomálie	(Verifikovaná chybová hodnota)/5	(Verifikovaná chybová hodnota)/6

$ActSec_{base}$je potom vypočítána jako průměrná hodnota každého vstupu vynásobená korespondující hodnotou. Výpočet potom vypadá následovně:

$$ActSec_{base} = \frac{ActSec_{Sum}}{Scope}$$

Korespondující výpočet RAV:

$$RAV = OpSec_{base} - \left(\frac{OpSec_{base} \times ActSec_{base}}{100}\right) + \left(\frac{OpSec_{Sum}}{(Scope + OpSec_{Sum})} \times \frac{LC_{base}}{100}\right)$$

Je důležité rozeznat rozdíl mezi hodnotami base a sum a vždy se ujistit, že jsou použity správné hodnoty.

Příklad použití RAV Calculatoru ze zdroje (ISECOM , 2014), do které autor zadal hodnoty analytik, ukazuje obrázek 5.

Závěrem je k této metodice nutné uvést, že OSSTMM se rovněž zabývá tvorbou výsledné zprávy, reportingem. S tvorbou reportu souvisí zkratka STAR (Security Test Audit Report). Jedná se o nástroj pro tvorbu reportů vypovídajících o stavu zabezpečení testované infrastruktury. Tato šablona je vhodná jak pro management, tak i pro skupinu techniků pro zkoumání cílů testování, hodnot hodnocení rizik a výstupu z každé fáze testování. Šablona STAR je k dispozici ke stažení v pdf formátu ze stejného zdroje jako RAV Calculator, tedy (ISECOM , 2014).

Attack Surface Security Metrics

OSSTMM version 3.0

Fill in the white number fields for OPSEC, Controls, and Limitations with the results of the security test. Refer to OSSTMM 3 (www.osstmm.org) for more information.

OPSEC	
Visibility	10
Access	230
Trust	5
Total (Porosity)	245

CONTROLS		
Class A		Missing
Authentication	210	35
Indemnification	2	243
Resilience	2	243
Subjugation	2	243
Continuity	2	243
Total Class A	218	1007

Class B		Missing
Non-Repudiation	1	244
Confidentiality	5	240
Privacy	2	243
Integrity	3	242
Alarm	51	194
Total Class B	62	1163

		True Missing
All Controls Total	280	2170
Whole Coverage	11,43%	88,57%

LIMITATIONS		Item Value	Total Value
Vulnerabilities	4	9,857143	39,428571
Weaknesses	12	5,110204	61,322449
Concerns	5	5,746939	28,734694
Exposures	6	0,953353	5,720117
Anomalies	2	0,103790	0,207580
Total # Limitations	29		135,4134

ISECOM
INSTITUTE FOR SECURITY AND OPEN METHODOLOGIES

OPSEC	19,264935
True Controls	11,883968
Full Controls	11,883968
True Coverage A	17,80%
True Coverage B	5,06%
Total True Coverage	11,43%
Limitations	17,070893
Security Δ	-24,45
True Protection	75,55

OSSTMM Open Source Security Testing Methodology Manual
OSSTMM
www.osstmm.org

Actual Security: 74,5187 ravs

Obrázek 5 RAV Calculator pro výpočet metriky RAV

3.3 **OWASP**

Tato metodika je speciálně zaměřena na penetrační testování webových aplikací a byla vytvořena organizací OWASP (Open Web Application Security Project). Nadace OWASP byla založena 21. dubna 2004 ve Spojených státech jako nezisková charitativní organizace. OWASP je otevřená komunita, která umožňuje organizacím navrhovat, vyvíjet, získávat, provozovat a udržovat aplikace, které mohou být považovány za důvěryhodné. Všechny OWASP nástroje, dokumenty, fóra a kapitoly jsou volně dostupné každému, kdo se zajímá o zlepšení bezpečnosti aplikací.

Dle (OWASP, 2013) se OWASP projekty dají rozdělit do těchto dvou oblastí:

- vývojové projekty
- dokumentační projekty

Příklad dokumentačních projektů OWASP:

- OWASP Appliacation Security Verification Standard (ASVS)
- The Guide – poměrně podrobné pokyny pro zabezpečení webových aplikací
- OWASP Top Ten (Top Ten Most Critical Web Application Vulnerabilities) – zaměření na nejkritičtější problémy webových aplikací
- Metrics – definuje metriky zabezpečení webových aplikací
- Legal – pomáhá prodávajícím i kupujícím sjednat odpovídající zabezpečení ve smlouvách
- Testing Guide – průvodce testováním zabezpečení webových aplikací
- ISO 17799 – podklady pro organizaci realizující ISO 17799
- AppSec FAQ – často kladené otázky

Příklady vývojářských projektů:

- WebScarab – nástroj pro testování zranitelností webových aplikací
- WebGoat – děravá aplikace, na které si můžete v bezpečném právním prostředí zkoušet bezpečnostní nedostatky

- Validation Filters – filtry
- DotNet – různé nástroje pro zabezpečení .NET aplikací

Organizace OWASP vydává každoročně dokument, který se nazývá OWASP Top Ten a slouží pro zabezpečení webových aplikací. OWASP Top Ten představuje nejkritičtější bezpečností rizika webových aplikací na základě posouzení a shody odborníků pro zabezpečení aplikací z celého světa, kteří se dělí o své odborné znalosti při tvorbě tohoto seznamu. OWASP Top Ten je vhodným měřítkem pro organizace k měření zabezpečení svých webových aplikací. OWASP Top Ten nejen zvyšují povědomí o bezpečnostních hrozbách, ale také umožňují vývojářům a bezpečnostním profesionálům prioritizovat a následně řešit tyto hrozby.

Tabulka 3 TOP Ten hrozby pro rok 2013 dle (OWASP-TOP10, 2013):

A1	Injection
A2	Broken Authentication and Session Management
A3	Cross-Site Scripting (XSS)
A4	Insecure Direct Object References
A5	Security Misconfiguration
A6	Sensitive Data Exposure
A7	Missing Function Level Access Control
A8	Cross-Site Request Forgery (CSFR)
A9	Using Known Vulnerable Components
A10	Unvalidated Redirects and Forwards

Dodržování zásad bezpečnosti při tvorbě webových aplikací podle OWASP Top Ten, který se stal standardem pro zabezpečení webových aplikací, je prvním krokem k identifikaci a zmírnění rizik webových aplikací.

Kromě žebříčku hrozeb pro webové aplikace vydává OWASP i Top Ten pro oblast mobilních aplikací. V roce 2013 byly shromážděny údaje pro statistiky zranitelnosti také v této oblasti. V následující tabulce můžeme vidět výsledek zpracování těchto statistik pro rok 2014 v oblasti hrozeb pro mobilní aplikace.

Tabulka 4 Top 10 Mobile Risks - Final List 2014 (OWASP, 2013)

M1	Weak Server Side Controls
M2	Insecure Data Storage
M3	Insufficient Transport Layer Protection
M4	Unintended Data Leakage
M5	Poor Authorization and Authentication
M6	Broken Cryptography
M7	Client Side Injection
M8	Security Decisions Via Untrusted Inputs
M9	Improper Session Handling
M10	Lack of Binary Protections

Pro vývojáře a testery webových aplikací je na stránkách organizace OWASP (Projects - OWASP, 2014) volně přístupný OWASP Testing Guide v4, který je cenným dokumentem jak pro návrh, tak i pro penetrační testování webových aplikací.

3.4 **Shrnutí**

Při porovnání jednotlivých metodik nalezneme rozdíly. Metodiky NIST a OSSTMM se zabývají síťovou bezpečností organizací, zatímco metodika OWASP

je úzce specializovaná na bezpečnost webových a mobilních aplikací. Metodika NIST je v porovnání s OSSTMM méně obsáhlá, nepokrývá například fyzickou bezpečnost ani vliv lidského faktoru. Metodika OSSTMM je rozsahem obsáhlejší s větším obsahem definic, pojmů a detailů, navíc obsahuje metriky na posouzení zabezpečení informačních systémů. Svým zpracováním je vhodná pro širokou škálu penetračních testerů, najde uplatnění u začátečníků i zkušených testerů včetně zpracování rozsáhlých penetračních testů. Metodika NIST je vhodnější spíše pro začínající testery, pokrývá pouze zabezpečení informačních systémů bez ohledu na fyzické zabezpečení a další interakce oproti OSSTMM. Projekty organizace OWASP slouží vývojářům webových aplikací ke správnému kódování s ohledem na zabezpečení webových aplikací proti napadení hackery a stejně dobře slouží i k penetračnímu testování. Přidanou hodnotu tvoří všeobecně uznávaný žebříček OWASP Top Ten, který obsahuje deset nejobávanějších bezpečnostních hrozeb a je vydávaný pro vývojáře a bezpečnostní experty na roční bázi. Pro zájemce o problematiku penetračního testování se doporučuje nastudovat širší škálu metodik, kde si podle vlastního uvážení mohou vybrat konkrétní informace pro test, který mají za úkol zpracovat. Z dalších metodik lze uvést například ISSAF (Information Systems Security Assesment Framework), což je framework zabývající se ohodnocením informační bezpečnosti rozdělený na technickou a manažerskou část. Technická část obsahuje základní soubor pravidel a procedur a vytváří odpovídající proces hodnocení organizace, zatímco manažerská část předepisuje odpovědnosti managementu a best practices, kterých je nutno se držet po celou dobu testování. Tato metodika je oproti ostatním a zavedeným nová a stále nevyspělá, ale dá se dobře kombinovat s jinými např. OSSTMM. Poslední metodikou, která stojí za zmínku je WASCO-TC (Web Application Security Consortium Threat Classification), což je konsorcium, které se opět zabývá klasifikací hrozeb bezpečnosti webových aplikací. Jedná se o průmyslově akceptovanou metodiku, kterou můžeme najít v produktech hodnocení zranitelnosti a v manažerských produktech. Je dobře zařaditelná ke standardu, jako je OWASP.

4 Nejpoužívanější nástroje pro penetrační testování

Tato kapitola představí nástroje, které se používají k penetračnímu testování s rozdělením podle fází, ve kterých jsou jejich výstupy použité. Tyto fáze, které kategorizují jednotlivé nástroje, vycházejí z metodik NIST 800-115 a OSSTMM

4.1 Linuxové distribuce určené pro penetrační testování

V současné době existuje mnoho použitelných linuxových distribucí, které jsou zaměřené na penetrační testování. Může se stát, že narazíme na distribuci, která není delší dobu udržovaná, pak její použitelnost časem klesá. Příkladem podobné distribuce může být Whax nebo Knoppix-STD. Následující distribuce jsou neustále aktualizovány a slouží jako základ pro penetrační testování informační bezpečnosti.

4.1.1 Kali Linux

První věc, která vás na stránkách projektu Kali Linux (Kali Linux, 2014) zaujme, je motto této distribuce zaměřené na penetrační testování: „the quiter you become, the more you are able to hear“, což lze volně přeložit jako „čím tišší budeš, tím více budeš schopen naslouchat“. Autorem této distribuce založené na Debian, která vychází z původního oblíbeného BackTrack Linuxu, je společnost Ofensive Security. Kali Linux obsahuje více než 300 penetračních nástrojů, které jsou bezplatně použitelné na bázi open source. Namátkou lze uvést aplikace jako Wireshark pro analyzování packetů, Nmap pro skenování portů, sqlmap pro penetrační testování SQL databází, Hydra nebo John the Ripper pro lámání hesel, Aircrack nebo Wifite – oba vhodné pro penetrační testy Wi-Fi sítí a Metasploit framework pro testování zranitelnosti aplikací. Některým z těchto a dalším nástrojům se budou věnovat další kapitoly. Kali Linux lze instalovat na pevný disk počítače, existují verze pro WMware v 32 i 64 bitových verzích a dokonce i pro ARM procesory, které umožňují provozovat tuto distribuci na stále populárnějších RaspberryPi. Všechny tyto verze jsou dosažitelné z oficiálního webu (Kali, 2014).

Během testování Kali Linuxu a v něm obsažených nástrojů pro penetrační testování Wi-Fi sítí se podařilo pomocí programu Wifite prolomit WEP zabezpečení

zhruba během 4 minut. Pokud se rozhodnete zaútočit na WAP zabepečení, lze z různých nástrojů použít například Fern Wi-Fi Cracker. K tomuto účelu je zapotřebí použít soubor se slovníkem hesel, které aplikace využívá pro prolamování WAP zabezpečení. Pomocí metody MitM (Man in the Middle) lze nepozorovaně odposlouchat identitu právě přihlašovaného uživatele na poštovní server některého webového portálu a zjistit jeho přihlašovací údaje. Tato metoda pracuje na principu komunikace oběti, která prochází přes útočníka, který takto zachytí nešifrovanou komunikaci např. serveru Seznam.cz, protože jeho úvodní stránka není šifrovaná. Podobným způsobem lze zjistit, jaké stránky oběť právě otevírá. Pomocí zachytávače paketů Ettercap, který je v Kali k dispozici mezi nástroji Network spoofing, lze otevírat ve svém prohlížeči adresy, které právě otevírá oběť. Další z použitelných metod může být DNS phishing. Takto lze oběti, která otevírá ve svém browseru určitou webovou stránku, podvrhnout předložením falešné odpovědi DNS jinou stránku. V tomto případě lze přesměrovat oběť na falešnou kopii například internetového bankovnictví a tím od oběti nepozorovaně zjistit přihlašovací údaje.

4.1.2 Blackbuntu

Na rozdíl od Kali Linuxu je Blackbuntu založen na linuxové distribuci Ubuntu. Jedná se o alternativu ke zmíněnému Kali Linuxu či původní distribuci Backtrack speciálně vytvořenou pro studenty či praktiky testující informační bezpečnost. Nejaktuálnější verze, která je dnes k dispozici, je postavena na Ubuntu 10.10, Linux 2.6.39 a Gnome 2.32.0. Oficiální stránka projektu je sice dosažitelná, neobsahuje však žádné aktuální informace. Zřejmě to svědčí o větší populárnosti Kali Linuxu.

4.2 **Nástroje použitelné pro fázi průzkumu**

Tato fáze slouží k získání informací o cíli penetračního testování. Tyto informace potom slouží jako základ pro další fáze testu. Jmenujme některé z klíčových dat, které v této fázi získáváme: rozsah testovaných IP adres, e-mailové adresy, adresy serverů, dokumenty obsahující důležité informace, které mohou být terčem útoku. Jelikož se jedná o širokou problematiku, je složité porovnat jednotlivé nástroje, každý z nich totiž slouží k poskytování informací jiného druhu a některé z nich

poskytují informace jako mezičlánek a vstup pro test jiného druhu. Tato kapitola se proto bude věnovat nástrojům pro fázi průzkumu.

4.2.1 Shodan

Shodan je vyhledávač umožňující najít konkrétní počítače (routery, servery, etc.) pomocí různých filtrů. Funkce je podobná internetovým vyhledávačům, kdy ovšem nejsou sbírána data z webových stránek, ale z bannerů (metadata, která server odešle zpět klientovi) zařízení, která jsou připojena do internetu. Může se jednat o webové či IP kamery, směrovače, servery, tiskárny, či nezabezpečené SCADA systémy a jiné. Výsledky těchto vyhledávání ukládá Shodan do databáze a po přihlášení umožní svým uživatelům po správně zvolených dotazech vyhledávat.

Banner naskenovaného FTP serveru připojeného k internetu může vypadat následovně:

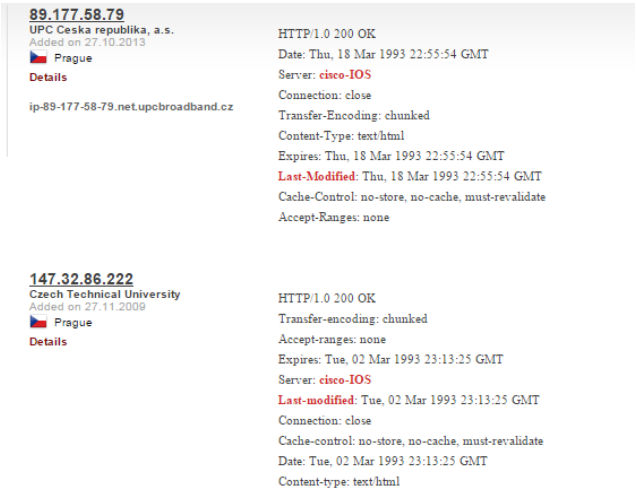
```
220 kcg.cz FTP server (Version 6.00LS) ready.
```

Z této informace vyplývá, že název zjištěného přístupného serveru je: kcg.cz, typ FTP serveru je: Solaris ftpd a jeho verze: 6.00LS.

HTTP banner vypadá takto:

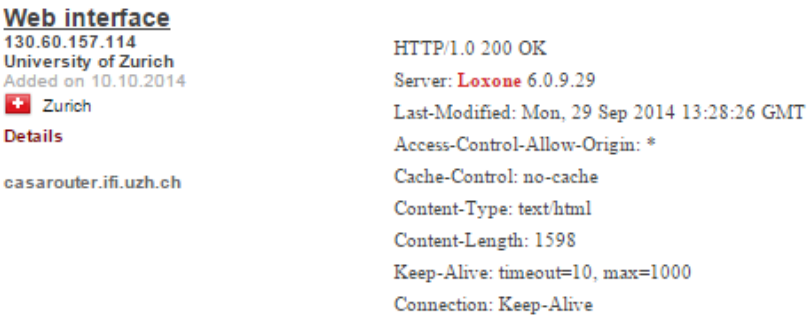
```
HTTP/1.0 200 OK
Date: Tue, 16 Feb 2010 10:03:04 GMT
Server: Apache/1.3.26 (Unix) AuthMySQL/2.20 PHP/4.1.2 mod_gzip/1.3.19.1a
mod_ssl/2.8.9 OpenSSL/0.9.6g
Last-Modified: Wed, 01 Jul 1998 08:51:04 GMT
ETag: "135074-61-3599f878"
Accept-Ranges: bytes
Content-Length: 97
Content-Type: text/html
```

Na základě informací z bannerů lze využít Shodan k cíleným filtrovaným dotazům. Na základě dotazu *"cisco-ios" "last-modified" country:CZ*, zobrazí Shodan údaje viz následující obrázek:



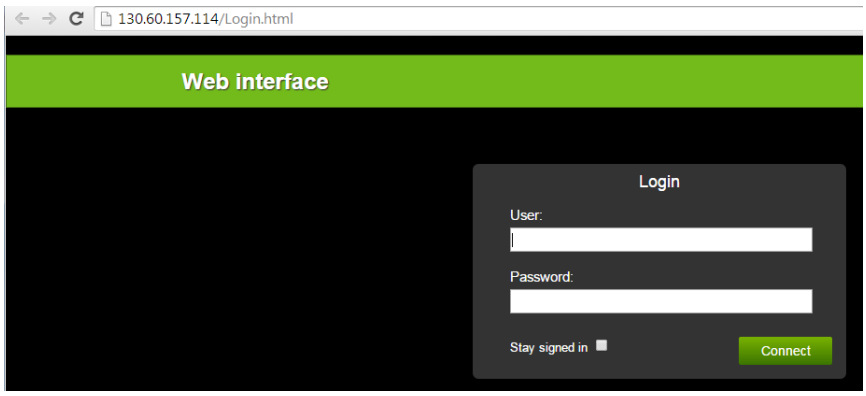
Obrázek 6 Použití filtru ve webové aplikaci Shodan

Podobně lze využít filtrů ke zjištění IP adres volně přístupných aktivních zařízení různého druhu kdekoliv ve světě. Například při použití klíčového slova „loxone“, což je aplikace webového serveru pro inteligentní domy, Shodan zobrazil 7932 položek. Na obrázku 7 lze vidět údaj o tomto serveru na univerzitě v Curychu:



Obrázek 7 Dotaz na klíčové slovo „loxone“ v Shodan

Při testu, zda je IP adresa 130.60.157.114 dosažitelná, lze otevřít přihlašovací stránku serveru:



Obrázek 8 Ověření správnosti informace získané ze Shodan

Shodan je v základní verzi přístupný zdarma. K přihlášení lze použít účet na Facebooku, Google nebo Twitteru. V této verzi existují omezení, která by při rozsáhlejších penetračních testech mohla testera limitovat. Výsledek vyhledávání například nezahrnuje služby https nebo telnet. Pro registrované uživatele neplatí žádná omezení a tester má možnost využít API rozhraní k této aplikaci. Ceník je k dispozici na stránkách projektu (Shodan, 2013).

4.2.2 Maltego

Maltego je extrémně silný nástroj z kategorie OSINT (Open Source Intelligence), který zahrnuje strukturní průzkum a osobní průzkum. Infrastrukturní složka Maltego umožňuje pomocí sad technik a nástrojů shromažďování citlivých údajů z veřejně dostupných zdrojů o cílové organizaci, emailových adresách zaměstnanců, důvěrných souborech, se kterými je manipulováno neopatrným způsobem, interních telefonních číslech, DNS záznamech, informace o IP adresách, geografickém umístění sítí, MX serverech a podobně. Proces shromažďování těchto údajů, v Maltego známý jako transformace, musí být pro dosažení co nejlepších výsledků navržen promyšleně a kreativně. Na druhé straně, Maltego v oblasti osobního průzkumu pomáhá při získávání informací o určité osobě, jako je činnost v sociálních sítích, e-mailové adresy, internetové stránky související s hledanou osobou, telefonní čísla atd. Tato činnost je založena na internetových vyhledávačích, se kterými Maltego účinně komunikuje a ze kterých shromážďuje všechny tyto

informace. Tento nástroj šetří čas penetračnímu testerovi v případě, že by podobné informace musel získávat manuálně.

Za aplikací Maltego stojí organizace Paterva (Paterva, 2014), odkud lze aplikaci získat. Aplikace je také standardně součástí Kali Linuxu. K používání komunitní volné verze je třeba se registrovat na stránkách organizace Paterva a stát se členem komunity.

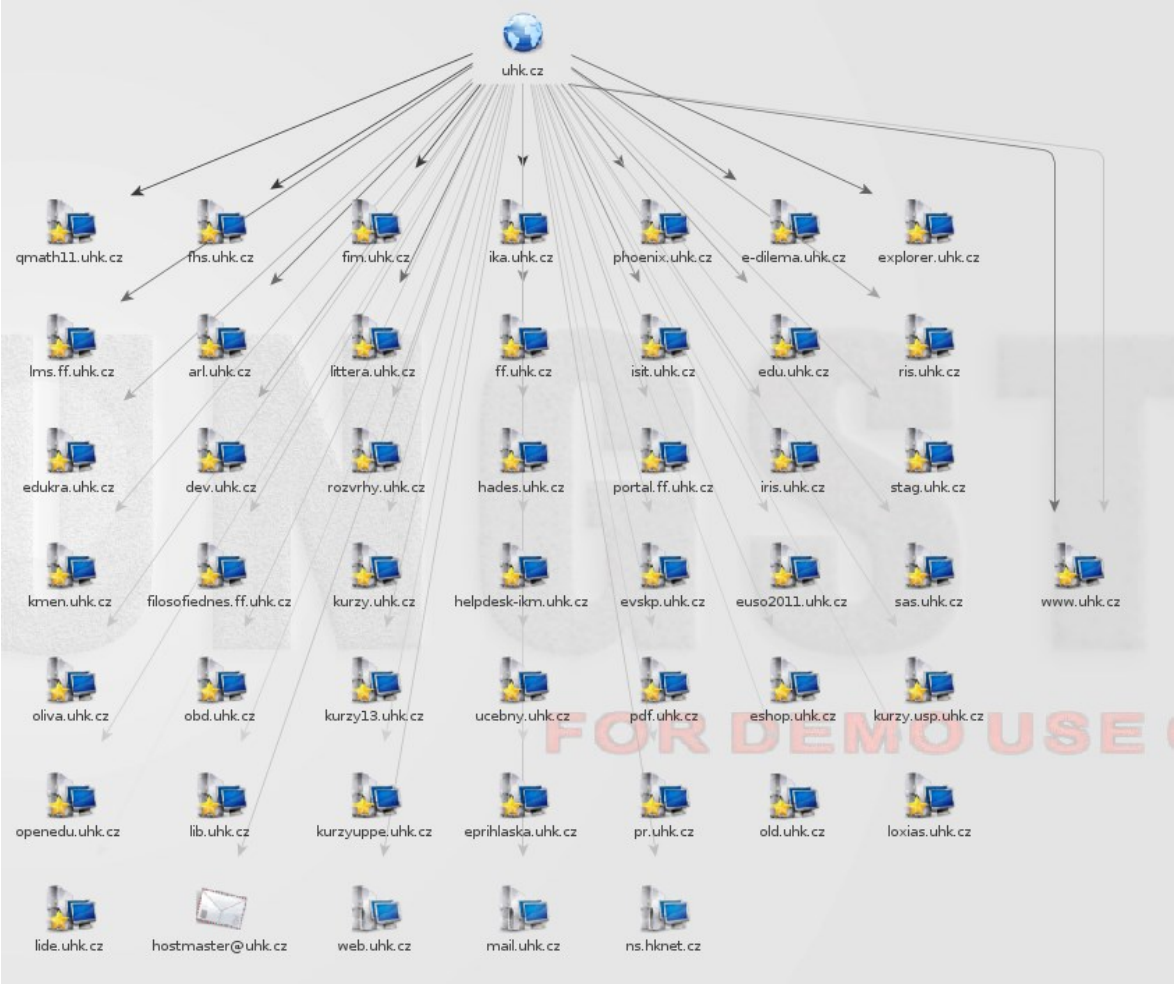
Tester používající Maltego má k dispozici grafické prostředí, což zjednodušuje práci s touto aplikací. Vstupem k zadání pro získání hledaných informací může být tzv. entita, což jsou informace poskytnuté testerem. Komunitní verze má omezení 12 entit. Druhým ze základních kamenů aplikace jsou tzv. transforms. Každý z těchto modulů slouží k vyhledávání jiných typů informací. Takto lze najít transform pro zjištění adresního rozsahu IP adresy, která do rozsahu spadá včetně dalších údajů jako jsou kontakty zodpovědných osob apod.

Podobně lze pomocí Maltego získat seznam e-mailových adres používaných organizací, čímž se dostane do rukou testerovi potencionální seznam přihlašovacích jmen jako základ pro další útoky. Tuto funkci demonstruje následující zkrácený seznam e-mailových adres při dotazu na doménu uhk.cz:

maltego.EmailAddress#katerina.chaloupkova@uhk.cz
maltego.EmailAddress#vladimir.bures@uhk.cz
maltego.EmailAddress#martin.bilek@uhk.cz
maltego.EmailAddress#pavel.jedlicka@uhk.cz
maltego.EmailAddress#petra.maresova@uhk.cz
maltego.EmailAddress#jindra.novotna@uhk.cz
maltego.EmailAddress#petr.skalnik@uhk.cz
maltego.EmailAddress#alena.pozdilkova@uhk.cz
maltego.EmailAddress#antonin.slaby@uhk.cz
maltego.EmailAddress#blanka.klimova@uhk.cz
maltego.EmailAddress#eva.simkova@uhk.cz
maltego.EmailAddress#hana.rohrova@uhk.cz
maltego.EmailAddress#jan.hladena@uhk.cz
maltego.EmailAddress#petr.havelka@uhk.cz
maltego.EmailAddress#martina.manenova@uhk.cz
maltego.EmailAddress#jan.kriz@uhk.cz
maltego.EmailAddress#marek.palatinus@uhk.cz
maltego.EmailAddress#marcel.pikhart@uhk.cz

Nejjednodušším způsobem používání Maltego je využít tzv. Maltego Machines, což je kolekce předdefinovaných tranforms. Oficiální dokumentace k práci s Maltego skriptovacím jazykem je k dispozici ze zdroje (Paterva, 2014), (Paterva, 2012).

Na obrázku 9 je demonstrován grafický výstup základního a rychlého Footprintu L1 domény uhk.cz:



Obrázek 9 Maltego grafický výstup Footprint L1

Uživatelé komerční verze nejsou omezeni nemožností ukládání výsledků nebo počtem provedených transformů během určité doby. K této aplikaci existují různé plug-iny. Jedním z těchto rozšíření může být Sploitego, které rozšiřuje možnosti Maltego o SNMP scannery, Nmap scanner nebo scannery zranitelnosti. Dalším užitečným doplňkem může být již zmíněný Shodan, který existuje také ve verzi plug-inu pro Maltego.

4.2.3 Nslookup, dig

Nslookup je příkaz, který slouží jako nástroj příkazového řádku pro testování a odstraňování problémů se servery DNS. Je instalován spolu s protokolem TCP/IP a je součástí operačního systému již od dob MS-DOS. Během penetračního testování slouží k získání informací, které souvisí s překladem adres, dají se s jeho pomocí zjistit mailové servery, které náleží k dané doméně a taky nám poskytne informace od DNS serveru. Pokud jsou DNS servery konfigurované nekorektně z hlediska zabezpečení, kdy umožní přenos zón k tazateli, nslookup pomůže tuto skutečnost odhalit.

Následuje ukázka použití dnes již zastaralého nslookup příkazu, jak lze s jeho pomocí zjistit z DNS serveru IP adresy poštovního serveru z autorovy domény vlne.eu, kterou hostuje u poskytovatele wedos.cz:

```
C:\ >nslookup
Default Server:  ns.hknet.cz
Address:  195.113.115.171

> type=mx
> vlne.eu
Server:  ns.hknet.cz
Address:  195.113.115.171

Non-authoritative answer:
vlne.eu MX preference = 1, mail exchanger = mx-67681.m81.wedos.net

vlne.eu nameserver = ns.wedos.com
vlne.eu nameserver = ns.wedos.net
vlne.eu nameserver = ns.wedos.cz
vlne.eu nameserver = ns.wedos.eu
ns.wedos.eu     internet address = 164.138.27.146
ns.wedos.eu     AAAA IPv6 address = 2a02:2770::21a:4aff:fe46:ddc9
ns.wedos.com    internet address = 37.157.192.2
ns.wedos.com    AAAA IPv6 address = 2a02:2b88:2:1::14ee:1

> set type=A
> mx-67681.m81.wedos.net
Server:  ns.hknet.cz
Address:  195.113.115.171

Non-authoritative answer:
Name:    we2-mx.wedos.net
Address:  46.28.105.74
Aliases:  mx-67681.m81.wedos.net
```

V příkladu je vidět dotaz na MX záznam, který je druhem položky v DNS obsahující informaci o poštovních serverech obsluhujících e-mailové adresy v doméně vlne.eu. Výsledkem dotazu je adresa poštovního serveru

mx-67681.m81.wedos.net. Pomocí příkazu nslookup lze zjistit informace, které mohou být následně použité v dalších fázích penetračního testu.

Nslookup nepatří mezi udržované nástroje a je nahrazen jinými populárními nástroji jako dig nebo host. Příkaz dig slouží např. k vyhledání IP adresy k doménovému jménu nebo doménového jména k IP adrese, navíc obsahuje podporu scriptování a další řadu funkcí, jež jsou popsány v manuálové stránce tohoto příkazu.

Následující příklad pomocí atributu –t a dotazu na MX záznam zobrazí zkráceně stejnou informaci o mailovém serveru jako předchozí nslookup:

```
root@kali:~# dig -t MX vlne.eu +noall +answer

; <<>> DiG 9.8.4-rpz2+rl005.12-P1 <<>> -t MX vlne.eu +noall +answer
;; global options: +cmd
vlne.eu.          5     IN    MX    1 mx-67681.m81.wedos.net.
```

4.2.4 Whois

Zatímco u unixových distribucí je tento příkaz součástí příkazové řádky, u OS Windows narozdíl od předchozího nslookup součástí systému není. Lze jej ale doinstalovat jako komponentu od firmy Sysinternals. Whois získá registrační záznam pro název domény nebo IP adresy, které určíte, a je závislý na činnosti whois serverů, které tyto informace obsahují. Informace z těchto serverů je možné získat jako kompletní databázi na internetu i v offline verzi. Pro uživatele, kteří nemají v oblibě používání příkazové řádky, poslouží lépe Maltego zmíněné v předchozí kapitole, které poskytuje stejné informace a jeho použití je jednodušší.

V další ukázce je předveden dotaz Whois na univerzitní doménu uhk.cz ve tvaru příkazu Whois s názvem domény za použití příkazu v prostředí Windows od firmy Sysinternals jako bylo uvedeno výše:

```
C:\>whois uhk.cz

Whois v1.12 - Domain information lookup utility
Sysinternals - www.sysinternals.com
Copyright (C) 2005-2014 Mark Russinovich

Connecting to CZ.whois-servers.net...

domain:       uhk.cz
registrant:   SB:UHK
admin-c:      JFLEK
nsset:        NSS:HKNET_UHK:1
```

```
registrar:    REG-GENREG
registered:   14.06.2000 14:32:00
changed:      04.04.2012 00:52:58
expire:       16.06.2015

contact:      SB:UHK
org:          University of Hradec Kr├ílov├ę
name:         University of Hradec Kr├ílov├ę
address:      Rokitansk├ęho 62
address:      Hradec Kr├ílov├ę
address:      500 03
address:      CZ
phone:        +420.493331111
e-mail:       michal.zamecnik@uhk.cz
registrar:    REG-GENREG
created:      10.08.2001 22:13:00
changed:      18.06.2014 13:45:03

contact:      JFLEK
name:         Jan Flek
address:      Rokitanskeho 62
address:      Hradec Kralove
address:      50003
address:      CZ
registrar:    REG-GENREG
created:      17.07.2008 10:48:00

nsset:        NSS:HKNET_UHK:1
nserver:      ns.hknet.cz
nserver:      ns2.hknet.cz
nserver:      ns.ces.net
tech-c:       JA11-RIPE
registrar:    REG-GENREG
created:      01.06.2009 10:39:10

contact:      JA11-RIPE
name:         Jindrich Andrs
address:      CZ
phone:        +420.495518123
fax-no:       +420.495518123
e-mail:       andrs@hknet.cz
registrar:    REG-GENREG
created:      10.08.2001 22:13:00
changed:      22.10.2004 13:25:00
```

Z výpisu je například patrné, že doména byla registrována 14. 6. 2000 ve 14:32:00 a její platnost vyprší 16. 6. 2015. Jsou tu další kontaktní údaje apod.

Další ukázka zobrazuje, jaké informace vrátí příkaz Whois ve tvaru s IP adresou, která patří autorovi, v prostředí Kali Linuxu:

```
root@kali:~# whois 188.120.212.190
% This is the RIPE Database query service.
% The objects are in RPSL format.
%
% The RIPE Database is subject to Terms and Conditions.
% See http://www.ripe.net/db/support/db-terms-conditions.pdf

% Note: this output has been filtered.
%       To receive output for a database update, use the "-B" flag.

% Information related to '188.120.212.0 - 188.120.215.255'
```

```
% Abuse contact for '188.120.212.0 - 188.120.215.255' is
'info@ip4isp.net'

inetnum:        188.120.212.0 - 188.120.215.255
netname:        IP4ISP-DUPETO-NET
descr:          Dupeto s.r.o main block from IP4ISP z.s.p.o
country:        CZ
admin-c:        TS3539-RIPE
admin-c:        BH2040-RIPE
tech-c:         TS3539-RIPE
tech-c:         BH2040-RIPE
status:         ASSIGNED PA
mnt-by:         MNT-NECOSS
source:         RIPE # Filtered

person:         Bohumil Holubec
address:        Dupeto s.r.o.
address:        Svazita 165
address:        Trutnov
address:        54101
address:        Czech Republic
phone:          +420777258172
nic-hdl:        BH2040-RIPE
mnt-by:         bozek-mnt
source:         RIPE # Filtered

person:         Tomas SIMEK
address:        NECOSS s.r.o
address:        U Malse 20
address:        Ceske Budejovice
address:        370 01
address:        Czech Republic
phone:          +420 608 259 701
nic-hdl:        TS3539-RIPE
mnt-by:         MNT-NECOSS
source:         RIPE # Filtered

% Information related to '188.120.192.0/19AS49985'

route:          188.120.192.0/19
descr:          IP4ISP z.s.p.o. network route
origin:         AS49985
mnt-by:         MNT-NECOSS
mnt-by:         MNT-KAORA
mnt-by:         MNT-ARAKIS
source:         RIPE # Filtered

% This query was served by the RIPE Database Query Service version 1.75
(DB-2)
```

Z výpisu získáme cenné informace jako je adresní rozsah, ve kterém se nachází dotazovaná IP adresa, dále je tu informace o původu adresy v CZ, o registraci RIPE, kontakty na odpovědné osoby apod.

4.2.5 Whatismyip?

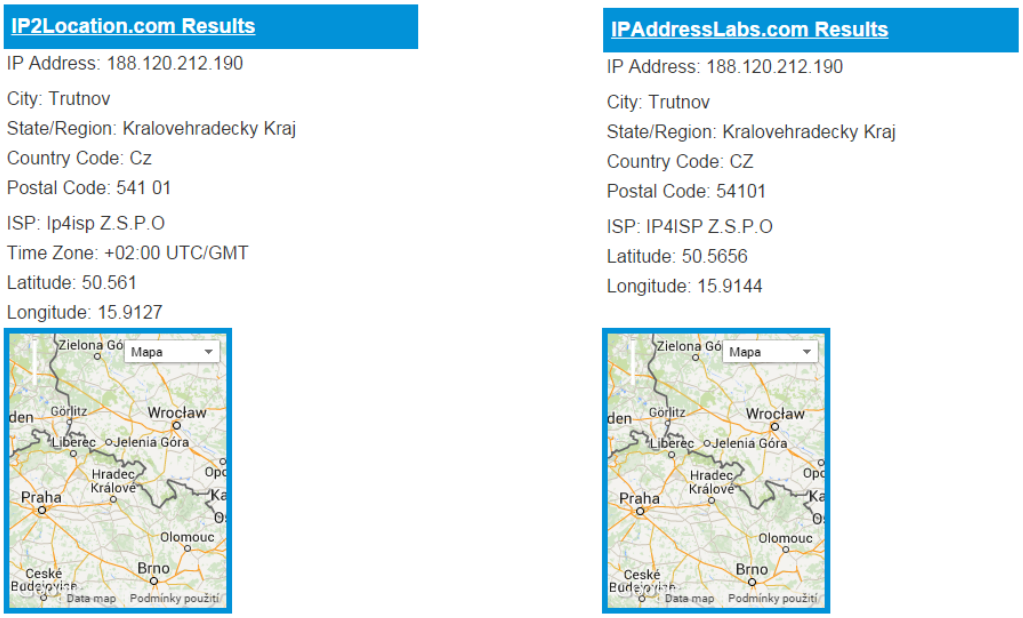
V praxi mohou nastat případy, kdy je nutné znát fyzickou lokaci dané IP adresy. Tuto informaci lze získat například na webových stránkách http://www.whatismyip.com.

Po otevření této webové stránky lze zjistit IP adresu, ze které je uživatel k internetu připojen. Autorova veřejná IP adresa je dle sdělení serveru 188.120.212.190.

Pod zmíněnou IP adresou následují tyto informace:

```
Proxy: No Proxy Detected
City: Trutnov
State/Region: Kralovehradecky Kraj
Country: Cz - cz flag
ISP: Ip4isp Z.S.P.O
```

Pod informacemi se nachází ikona „MORE INFORMATION“, která po kliknutí otevře novou stránku s detailnějšími informacemi následovanými mapou lokace:



Obrázek 10 Informace o hledané IP adrese

Na této stránce lze nalézt i informace o jiných IP adresách dle zadání uživatele. Podobnou funkci poskytuje i server http://whatismyipaddress.com.

4.2.6 DNSDict6

Tento nástroj, který je opět součástí Kali Linuxu, je ovladatelný přes příkazovou řádku. Dnsdict6 je nástrojem pro shromažďování informací, které se používají k získávání informací z webových stránek. Dnsdict6 umožňuje skenovat webové stránky a zobrazit kolik sub-domén nebo domén je k dispozici. Umožňuje skenování IPv6 i IPv4 adres. Tento nástroj je velmi mocný, protože zjistí i ty subdomény, které jsou pro uživatele z DNS serverů zakázané nebo neviditelné. V následující ukázce je použit příkaz ve formátu dnsdict6 -d -4, kde argument -d slouží k zobrazení informací o name serverech a MX záznamech a -4 slouží k výpisu IPv4 adresy:

```
root@kali:~# dnsdict6 -4 -d -x uhk.cz
Starting DNS enumeration work on uhk.cz. ...
Gathering NS and MX information...
NS of uhk.cz. is ns.hknet.cz. => 195.113.115.171
NS of uhk.cz. is ns.hknet.cz. => 2001:718:1203:2::aa
NS of uhk.cz. is nsa.ces.net. => 195.113.144.205
NS of uhk.cz. is nsa.ces.net. => 2001:718:1:1::144:205
NS of uhk.cz. is ns2.hknet.cz. => 195.113.115.174
NS of uhk.cz. is ns2.hknet.cz. => 2001:718:1203:2::ab
MX of uhk.cz. is uhk-cz.mail.eo.outlook.com. => 213.199.154.87
MX of uhk.cz. is uhk-cz.mail.eo.outlook.com. => 213.199.154.23
No IPv6 address for MX entries found in DNS for domain uhk.cz.

Starting enumerating uhk.cz. - creating 8 threads for 5886 words...
Estimated time to completion: 3 to 9 minutes
av.uhk.cz. => 93.99.58.124
athena.uhk.cz. => 195.113.119.2
autodiscover.uhk.cz. => 195.113.118.11
cg.uhk.cz. => 195.113.118.8
cisco.uhk.cz. => 195.113.118.12
dev.uhk.cz. => 195.113.118.138
dione.uhk.cz. => 195.113.118.15
edu.uhk.cz. => 195.113.118.22
ff.uhk.cz. => 195.113.118.23
fs.uhk.cz. => 195.113.118.135
gw1.uhk.cz. => 195.113.118.5
gw1.uhk.cz. => 2001:718:1202:201::2
hades.uhk.cz. => 195.113.120.222
helpdesk.uhk.cz. => 195.113.118.127
hera.uhk.cz. => 195.113.118.18
idp.uhk.cz. => 195.113.118.123
idp.uhk.cz. => 2001:718:1202:240::155
iris.uhk.cz. => 195.113.118.17
legacy.uhk.cz. => 195.113.118.40
ipv6.uhk.cz. => 2001:718:1202:240::600
m2.uhk.cz. => 195.113.120.240
lib.uhk.cz. => 195.113.119.3
m1.uhk.cz. => 195.113.120.239
mail.uhk.cz. => 195.113.118.11
medusa.uhk.cz. => 195.113.118.41
lync.uhk.cz. => 93.99.58.123
meet.uhk.cz. => 93.99.58.123
mis.uhk.cz. => 195.113.118.133
morpheus.uhk.cz. => 195.113.120.226
old.uhk.cz. => 195.113.118.30
```

```
neptun.uhk.cz. => 195.113.118.39
ns.uhk.cz. => 2001:718:1202:240::aa
orion.uhk.cz. => 195.113.120.236
people.uhk.cz. => 195.113.118.29
octopus.uhk.cz. => 195.113.118.42
owa.uhk.cz. => 132.245.6.139
owa.uhk.cz. => 132.245.6.235
owa.uhk.cz. => 132.245.12.203
owa.uhk.cz. => 132.245.89.171
owa.uhk.cz. => 157.56.233.251
owa.uhk.cz. => 157.56.242.123
owa.uhk.cz. => 157.56.243.11
owa.uhk.cz. => 132.245.3.187
pr.uhk.cz. => 195.113.120.238
prometheus.uhk.cz. => 195.113.118.240
prometheus.uhk.cz. => 2001:718:1202:240::200
portal.uhk.cz. => 195.113.118.240
portal.uhk.cz. => 2001:718:1202:240::200
posta.uhk.cz. => 195.113.118.11
phoenix.uhk.cz. => 195.113.118.30
radius2.uhk.cz. => 195.113.118.14
ris.uhk.cz. => 195.113.118.128
poseidon.uhk.cz. => 195.113.165.130
postman.uhk.cz. => 195.113.118.44
radius1.uhk.cz. => 195.113.118.38
security.uhk.cz. => 195.113.118.8
sftp.uhk.cz. => 195.113.118.18
sip.uhk.cz. => 132.245.193.35
sip.uhk.cz. => 2a01:111:f404:8003::3d
web.uhk.cz. => 195.113.165.130
vr.uhk.cz. => 195.113.118.137
vpn.uhk.cz. => 195.113.165.254
webct.uhk.cz. => 195.113.118.79
www.uhk.cz. => 195.113.118.240
www.uhk.cz. => 2001:718:1202:240::200

Found 52 domain names, 48 unique ipv4 and 6 unique ipv6 addresses for
uhk.cz.
```

4.2.7 Google hacking database

Tato metoda využívá podstaty oblíbeného Google vyhledávače. Google vyhledávač prochází a indexuje obsah webových stránek. Vyhledávač poskytuje informace právě na bázi indexů o navštívených stránkách včetně těch, které nikdy neměly být předmětem indexování. Z toho důvodu lze pomocí Google vyhledávače nalézt servery, které jsou špatně zabezpečené, nebo které obsahují verzi software s určitou zranitelností. Takto lze vyhledat konfigurační soubory, přístupy do administračních nástrojů k databázím, soubory obsahující nešifrovaná hesla, xls soubory s citlivými informacemi, ovládací rozhraní pro webové kamery a další citlivé informace

Informace o Johnovi Longovi, autorovi Google hacking database, lze najít na stránkách organizace Hackers for charity testers (Anon., 2013). Tato organizace shromažďuje výpočetní a kancelářskou techniku, kterou daruje málo rozvinutým

zemím. Spolu s koordinací charitní činnosti, pomáhá John osobně v Africe s nastavováním počítačových sítí a budováním vesnické infrastruktury. Kromě toho je John Long autorem knihy Google hacking for penetration testers (Long, et al., 2007).

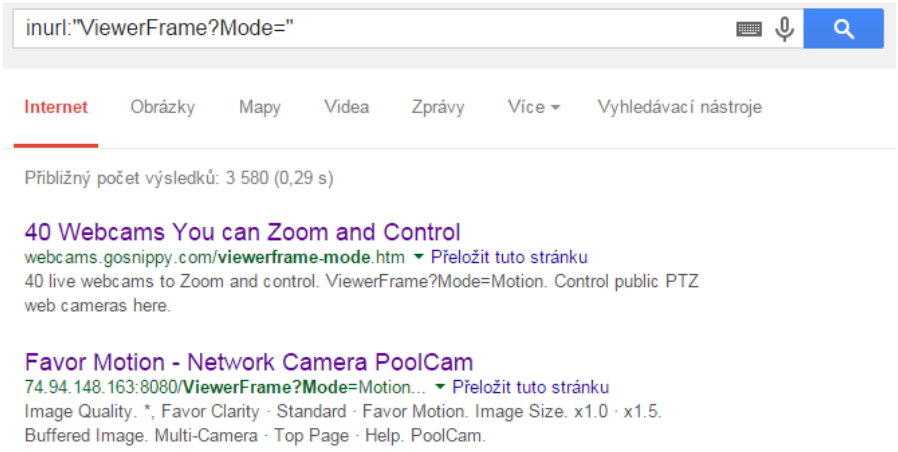
Na stránkách charitativní organizace (Anon., 2013), kterou John Long založil, najdete databázi Google hacking database (GHDB), která má základy z let 2002, kdy John začal sbírat zajímavé Google vyhledávací dotazy a zjistil tak zranitelnost systémů či citlivé informace, které nazýval googleDorks. Jeho databáze s googleDorks obsahuje značné množství dotazů pro Google vyhledávač včetně popisu, jakou informaci dotaz poskytne. Je zřejmé, že takto prakticky bez speciálních technik a prostředků lze přijít k mnoha důvěrným a zajímavým informacím.

Následující dotaz v Google vyhledávači zobrazí odkazy na takřka 7 000 000 různých serverů obsahujících mp3 soubory:

```
intitle:index.of mp3
```

Výsledek dalšího vloženého řetězce do vyhledávače Google najde odkazy na veřejné webové kamery:

```
inurl:"ViewerFrame?Mode="
```



Obrázek 11 Výsledek dotazu na veřejné webové kamery

Aktuálnost odkazů je otestována viz další obrázek, kde je použita druhá adresa poskytnutá vyhledávačem. Lze vidět fukční obraz kamery včetně ovládacích prvků, kterými lze kameru ovládat:

Obrázek 12 Test druhého z odkazů s funkční kamerou včetně ovládání

Pokud si přeje tester zjistit, kde se tato webová kamera nachází, lze použít například nástroj dostupný na webové adrese http://www.whatismyip.com, kterému je věnována kapitola 4.2.5.

Následující adresa v internetovém prohlížeči poskytne údaje o lokalitě, kde se přístupná webová kamera nachází:
http://whatismyipaddress.com/ip/74.94.148.163

Tabulka 5 Informace o lokaci hledané IP adresy

IP:	74.94.148.163
Decimal:	1247712419
Hostname:	74-94-148-163-newengland.hfc.comcastbusiness.net
ISP:	Comcast Business Communications
Organization:	Bassrocks Inn
Services:	None detected
Type:	Corporate

Assignment:	Static IP
Country:	United States us flag
State/Region:	Massachusetts
City:	Quincy
Latitude:	42.2529 (42° 15′ 10.44″ N)
Longitude:	-71.0023 (71° 0′ 8.28″ W)
Area Code:	617

Pomocí Google hacking lze penetračním testerem odhalit mnoho slabin a informací, které neměly být nikdy zveřejněny. Jako poslední příklad slouží řetězec k odhalení webových serverů na bázi EasyPHP, které nemají zabezpečený administrátorský přístup ke správě databáze:

```
intitle:"[EasyPHP] - Administration"
```

Na tento dotaz lze získat více než 100 indexovaných odkazů. V oblasti penetračního testování lze tedy použít i pokročilé vyhledávání pomocí Google.

4.3 Nástroje pro fázi skenování

Další fází po fázi průzkumu, kdy jsou zjištěna zařízení v síti, je pro penetračního testera fáze skenování, kde dochází ke zjišťování zranitelností. Důvodem skenování, které je rovněž popsáno v metodikách OSSTMM a NIST, je získání podrobnějších informací o infrastruktuře sítě a připojených zařízeních, které budou důležité v dalších fázích penetračního testu.

Tato kapitola se bude věnovat detailněji jedné z komplexnějších a rozšířených utilit Nmap, protože právě Nmap je považován za nejlepší skenovací nástroj, který obsahuje prakticky všechny funkce poskytované dalšími aplikacemi jako například AutoScan nebo Unicornscan a dále je vybaven dalšími specifickými funkcemi, kterými

předčil jiné konkurenční nástroje. Jako příklad lze uvést multiplatformnost a možnost skriptování v jazyce Lua, což umožňuje automatizaci skenování pomocí Nmap. Další vlastnosti, pro které je Nmap považován za nejvyspělejší skenovací nástroj, budou popsány v následující kapitole.

4.3.1 Nmap

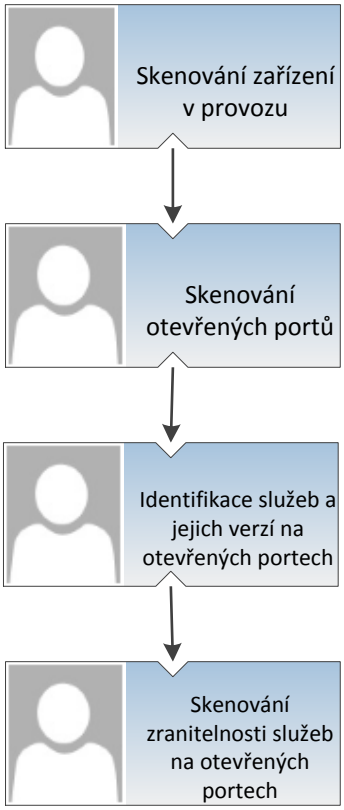
Nmap „Network Mapper“ je multiplatformní open source nástroj pro síťové skenování a bezpečnostní audity. Původně se jednalo o jednoduchý nástroj určený ke skenování portů a od druhé poloviny devadesátých let, kdy vznikl, jeho popularita rostla spolu s počtem funkcí, které tento nástroj v dnešní době poskytuje. Jde především o automatizaci úloh, skriptování a Nmap kromě příkazové řádky obsahuje grafické rozhraní Zenmap, pomocí kterého je přístupnější pro uživatele, kteří preferují grafické prostředí před příkazovou řádkou. Řada systémových inženýrů a administrátorů používá Nmap rovněž pro inventarizaci, řízení servisních plánů pro upgrade a sledování provozuschopnosti zařízení nebo služeb. Je schopen zjistit, které počítače a služby jsou k dispozici včetně názvů aplikace a verze, rozezná běžící operační systémy i typy paketových filtrů nebo firewallů včetně desítek dalších charakteristik. Nmap byl navržen pro skenování rozsáhlých sítí, ale je funkční i vůči jednomu hostiteli. Oficiální instalační balíčky existují pro Windows, Linux a Mac OS X, Nmap je rovněž součástí Kali Linux.

Podle oficiálních webových stránek projektu (Lyon, 2006), ze kterých lze získat instalační soubory i dokumentaci, byl Nmap jmenován "Bezpečnostním produktem roku" časopisem Linux Journal, Info World, LinuxQuestions.Org a Codetalker Digest. I když Nmap obsahuje rozsáhlé manuálové stránky, vydal Gordon Lyon (autor aplikace) knihu (Lyon, 2006), která slouží jako oficiální příručka pro síťové skenování pomocí Nmap.

Nejvýznamnější vlastnosti nástroje Nmap:

- pokoročilé způsoby definic skenovaných adres
- zjištění dostupnosti zařízení a služeb na základě různých typů skenování: TCP SYN/ACK, UDP, SCPT, ICMP (echo, timestamp, netmask), arp, null, xmas, idle scan a jiné
- detekce OS a verzí programů, které poskytují služby na otevřených portech
- definovatelný typ výstupů v podobě XML, txt, výstup do databáze apod.
- možnost použití skriptů pro různé úlohy
- možnost definice agresivity skenu podle počtu odeslaných paketů
- použití metod pro oklamání Firewallů a IDS
- integrace sad nástrojů pro benchmarking sítě a porovnávání výsledků s dalšími provedenými skeny

Podle Kimberly Graves (Graves, 2007) je potřeba skenování pečlivě naplánovat a rozdělit do jednotlivých fází. V každé fázi jsou pak prováděny úkony pro získání co nejvěrnějšího obrazu zařízení v síti a služeb pracujících na těchto zařízeních. Jednotlivé fáze skenování sítě, o kterých ve své knize Certified ethical hacker pojednává Kimberly Graves (Graves, 2007), jsou představeny na následujícím obrázku:



Obrázek 13 Fáze skenování sítě

První fází penetračního testu je najít v síti zařízení, která jsou aktivní a reagují na síťovou komunikaci. Tato zařízení jsou z pohledu testera zajímavá, protože mohou být zranitelná. Pro tento typ skenování obsahuje Nmap volby, pomocí nichž lze získat detailní přehled o připojených zařízeních v daném segmentu.

Příklad příkazu Nmap v prostředí Kali Linux na specifikaci cíle pro konkrétní server, rozsah IP adres nebo textový soubor se seznamem adres:

```
$ nmap vlne.eu
$ nmap 192.168.1.24-248
$ nmap seznamstanic.txt
```

Mezi klasické metody hledání hostů patří ping scan, který zjistí stav IP adresy pomocí ICMP echo dotazu. Mezi další sofistikovanější metody hledání hostů ve stanoveném rozsahu je např. TP SYN/ACK, UDP, SCPT INIT apod., pomocí kterých zjistíme, zda je skenované zařízení aktivní. Nmap pro skenování aktivních zařízení nabízí řadu parametrů. V následujícím příkladu bude naznačeno použití příkazu

s použitím ARP pingu, pro ACK ping slouží parametr –PA, protokol ping je volán volbou –PO apod.

```
$ nmap -PU 192.168.1.24-152
```

Po zjištění, která zařízení jsou v daném síťovém segmentu aktivní, následuje hledání otevřených portů na těchto zařízeních. I pro tento typ skenování nabízí Nmap širokou škálu voleb. Jedním typem pro skenování portů je TCP connect scan, který využívá princip funkce TCP protokolu, kdy se na každém portu pokusí navázat spojení a v závislosti na typu odpovědi protistrany určí, zda je skenovaný port otevřen či uzavřen. Dalším typem skenu pro ověření otevřenosti portů je TCP SYN, který pracuje na stejném principu jako předchozí sken, ale v závěru nedokončí spojení s otevřeným portem, čímž se jeví jako méně nápadný pro systémy chránící síť. Mezi další patří UDP scan, idle scan, null scan a další typy skenů pracující na různých principech. Následující příklad demonstruje skenování UDP portů 53, 111 a TCP portů 21-25, 80, 8080:

```
$ nmap –sU –sT –p U:53,111,T:21-25,80,8080 188.120.212.190
```

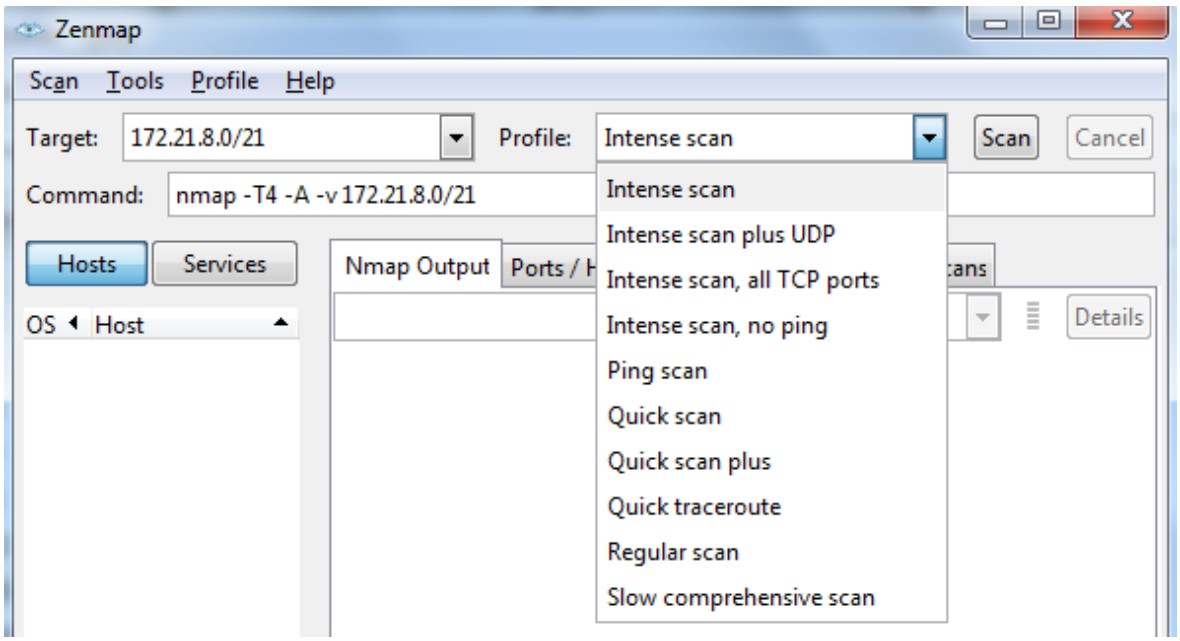
Služby a jejich verze na otevřených portech se dají opět zjistit s použitím vhodných parametrů při dotazu. Podobně je to se zjištěním verze operačního systému, kde jako parametr slouží přepínač –O. V případě, kdy Nmap nenajde v databázi přesnou verzi testovaného OS, je pomocí příkazu –osscan-guess zjištěna verze, která se nejvíce přibližuje skenované verzi, viz následující příklad:

```
$ nmap –O –osscan-guess vlne.eu
```

Jednou ze silných funkcí Nmap je možnost tvorby vlastních skriptů pro zefektivnění práce. Tato funkce se nazývá NSE (Nmap Scripting Engine) a je založená na skriptovacim jazyku Lua (LUA, 2013). Při používání skriptů se nedoporučuje z bezpečnostních důvodů stahovat z internetu skripty třetích stran, pokud se nejedná o ověřený zdroj.

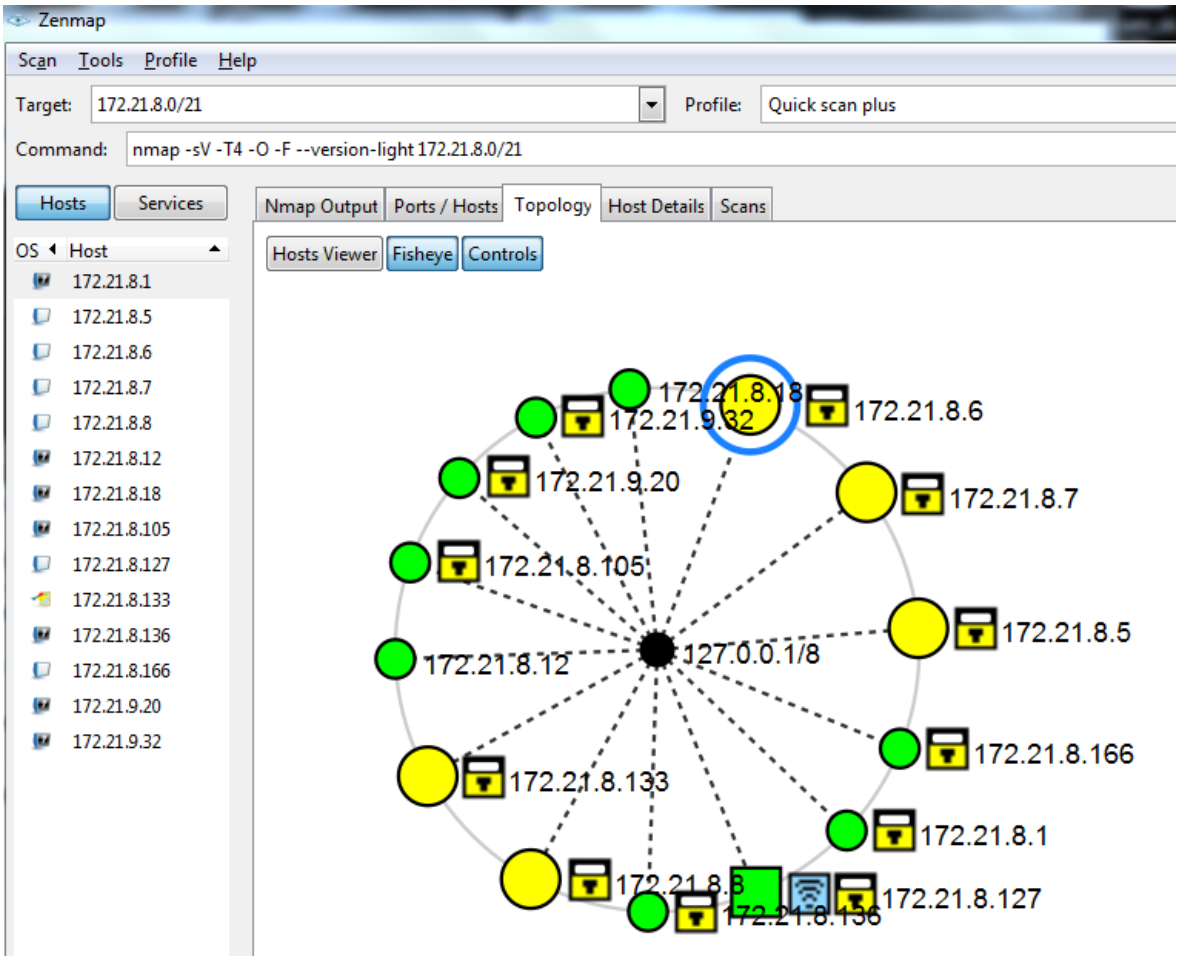
Jak bylo zmíněno, kromě ovládání Nmap pomocí příkazové řádky, existuje grafická nadstavba Zenmap, která je součástí Kali Linux, a obsahuje kromě zmíněného

grafického rozhraní i několik sad připravených skenovacích profilů, které znázorňuje následující obrázek.



Obrázek 14 Skenovací profily v Zenmap

Na dalším obrázku je možné vidět výstup provedeného skriptu Quick scan plus pomocí příkazu „nmap -sV -T4 -O -F --version-light 172.21.8.0/21“ z obrázku 14.



Obrázek 15 Grafický výstup aplikace Zenmap

4.4 Nástroje pro fázi zjišťování zranitelnosti

Ke skenování cílových systémů a zjišťování jejich zranitelnosti slouží například nástroje Nessus či OpenVAS.

4.4.1 Nessus

Nessus je jedním z nejpopulárnějších skenerů zranitelnosti zvláště pro UNIX systémy. Podle společnosti Tenable Network Security, která je autorem tohoto skeneru, je používán více než 75 000 organizacemi po celém světě. Tento proprietární komplexní skener zranitelnosti (Anon., 2014) je k dipsozici v trial verzi zdarma pro prvních sedm dnů pro osobní použití v nekomerčním prostředí. Zpočátku byl k dispozici zdarma, ale v roce 2005 byl uzavřen zdrojový kód. Jeho současná cena je 1200 USD/rok. Nessus je soustavně aktualizován, existuje kolem 46 000 pluginů.

Hlavními vlastnostmi jsou vzdálené a místní ověření bezpečnostních zranitelností a trhlin.

Nessus při běžných operacích začíná skenováním portů s jedním ze svých čtyř vnitřních skenerů (nebo je možné volitelně použít AMAP nebo zmíněný Nmap), aby zjistil, které porty jsou otevřené směrem k cíli, a následně použije na otevřené porty různé exploity. Testy zranitelnosti jsou k dispozici jako placená služba, jsou psány v NASL jazyce (Nessus Attack Scripting Language), což je optimalizovaný skriptovací jazyk.

Tenable Network Security produkuje týdně několik desítek nových pluginů pro testování zranitelnosti, obvykle na denní bázi. Tyto kontroly jsou k dispozici zdarma široké veřejnosti kromě komerčních zákazníků. Profesionální zpoplatněná verze nabízí přístup jak k podpoře, tak i k další skriptům jako například auditorským souborům, testům shody, dalším pluginům pro detekci zranitelnosti apod.

Volitelně mohou být výsledky testu uvedeny v různých formátech, jako je prostý text, XML, HTML a LaTeX. Výsledky lze také uložit ve znalostní základně pro ladění. V systému UNIX lze skenování automatizovat pomocí příkazového řádku klienta.

Kromě testování na známé zranitelnosti sítě poskytuje Nessus další funkce jako například přezkoumání úrovně záplat na počítačích s operačním systémem Windows a může provádět audit hesel pomocí slovníku a brute force metody. Nessus 3 a novější může sloužit pro audit systémů, pro kontrolu nastavení konkrétních politik, jako je NSA průvodce pro hardening Windows serverů.

4.4.2 OpenVAS

OpenVAS (The Open Vulnerability Assessment System) je framework složený z nástrojů a služeb a pro výkonné skenování zranitelnosti. Představuje open source alternativu ke komerčnímu nástroji Nessus, se kterým sdílí základ verze 2.2, což byla

poslední verze Nessus před ohlášením, že se v další verzi stane komerčním produktem.

Na webových stránkách projektu (OpenVas, 2014) se může zájemce dozvědět více informací o OpenVAS a získat instalační soubory a dokumentaci. Prostředí je opět součástí linuxové distribuce Kali Linux, podobně jako mnoho dalších nástrojů zmíněných v této práci. OpenVAS zajišťuje pro své uživatele veřejný zdroj NVT (Network Vulnerability Tests), který obsahuje více než 35.000 NVT testů a je průběžně aktualizován.

4.5 Nástroje pro fázi vedení útoku

Tato kapitola slouží k popisu nástrojů pro vedení útoku. Největší prostor bude věnován rozšířenému a ceněnému nástroji pro penetrační testování Metasploit framework.

4.5.1 Irpas

IRPAS je nástroj určený k útokům na síťovou infrastrukturu od německé společnosti Pheonelit. Jeho funkce spočívá ve zneužití bezpečnostních slabin síťových protokolů využívaných routery, kde bývá problém v chybějící autentifikaci. Router v takovém případě důvěřuje příchozím zprávám od protokolů, kde není nastavena autentifikace, jako jsou CDP, EIGRP, HSRP, OSPF, RIP a další. V případě, že se útočníkovi podaří podvrhnutými pakety zbavit funkce hlavní router, může směřovat veškerý provoz přes své zařízení. Pomocí manipulace CDP tabulek síťových zařízení na lokálním segmentu lze upravit strukturu informací CDP paketu tak, že způsobí nestabilní chování staších CISCO zařízení.

S pomocí tohoto nástroje lze prověřit míru zabezpečení síťové infrastruktury a použitých protokolů, pro zabezpečenou infrastrukturu však nepředstavuje vážné ohrožení.

4.5.2 Ettercap

Tento nástroj umožňuje manipulaci síťového provozu a následný odposlech. Obsahuje množství pluginů a grafické uživatelské rozhraní. Pomocí Ettercap lze síť

používající přepínače donutit ke změně chování. Technika využívá ARP spoofing, který spočívá v rozesílání podvržených nevyžádaných rámců protokolu ARP, které příjemce zahrnou mylnými kombinacemi IP a MAC adres. Postižený počítač následně odesílá rámce na jiné IP adresy.

Použitím pluginů lze rozšířit vlastnosti Ettercapu. Existují moduly, které umožňují podnikat útoky na transportní vrstvě ISO/OSI modelu a rovněž jsou k dispozici moduly určené k prohledávání sítí a spojení, která v této síti existují.

4.5.3 Social Engineering Toolkit

SET (Social Engineering Toolkit) obsahuje sadu nástrojů cílených na lidský faktor zneužívající bezpečnostní zranitelnost aplikací, které jsou nainstalované na koncových uživatelských stanicích. SET je součástí Kali Linuxu a jeho obsluha probíhá pomocí příkazové řádky formou dialogů na základě typu zvoleného útoku.

K napadení koncových stanic dochází zpravidla za účasti jejich uživatelů, kteří buď nejsou dostatečně opatrní, nebo ignorují bezpečnostní směrnice a pravidla organizace. Jeden z možných způsobů použití SET je spuštění webového serveru na počítači útočníka, který bude simulovat přihlašovací stránku, na které oběť zadává své přihlašovací údaje. Tyto údaje jsou potom pro útočníka čitelné v aplikaci SET přes meterpreter shell, který představuje spojitost s MTS (Metasploit framework). Tomuto nástroji je věnována další kapitola.

4.5.4 Metasploit framework

Tato kapitola pojednává o možnostech využití frameworku pro penetrační testování a poukazuje na možné problémy, které framework za penetračního testera řeší. Dále bude představen jeden z nejpoužívanějších zástupců frameworků - Metasploit framework.

4.5.4.1 Využití frameworku při penetračním testování

Díky dynamickému rozvoji informačních technologií je dnes prakticky využívána celá řada druhů a typů informačních, komunikačních a počítačových systémů. Při provozu každého systému jsou postupně odhalovány jeho nedostatky

či neúmyslné chyby vzniklé při jeho tvorbě, kdy celá řada těchto chyb negativně ovlivňuje bezpečnost daného systému. Tyto chyby narůstají v závislosti s někdy až příliš dynamickým vývojem nových platforem a systémů, čímž se používání informačních, komunikačních a počítačových systémů stává více či méně rizikové. Je zřejmé, že zmapovat všechny chyby a nedostatky dnes běžně využívaných systémů není z pohledu lidské mysli reálné, proto jsou pro tento účel vytvářeny frameworky určené pro penetrační testování. Jejich používání může testerovi výrazně usnadnit a zefektivnit jeho práci, jelikož jejich úkolem je otestovat aktuálně známé bezpečnostní nedostatky a chyby v daném systému. Mezi hlavní úkoly frameworku logicky patří aktualizace informací o bezpečnostních chybách, získávání exploitů pro zabránění znovuobjevování již objeveného, usnadnění práce s exploity, organizace payloadů a usnadnění vedení útoku proti konkrétnímu cíli.

V praxi se pak nabízí celá řada frameworků, které jsou specializované na testování webových aplikací, jako je Samurai Web Testing Framework, přes fremaworky použitelné pro komplexní testy jako je Penetration Testing Framework 0.59, Hcon Security Testing Framework, OWASP až po asi nejznámější a nejpoužívanější Metasploit framework. Metasploit framework byl publikován v roce 2003 díky programátorovi jménem HD Moore, který framework naprogramoval nejprve v jazyce Perl a poté byl komunitou programátorů přepsán do jazyka Ruby. V roce 2009 byl Projekt Metasploit v rámci akvizice převzat společností Rapid7, která stále financuje jeho vývoj, ale ponechala jej dostupným pod licencí BSD.

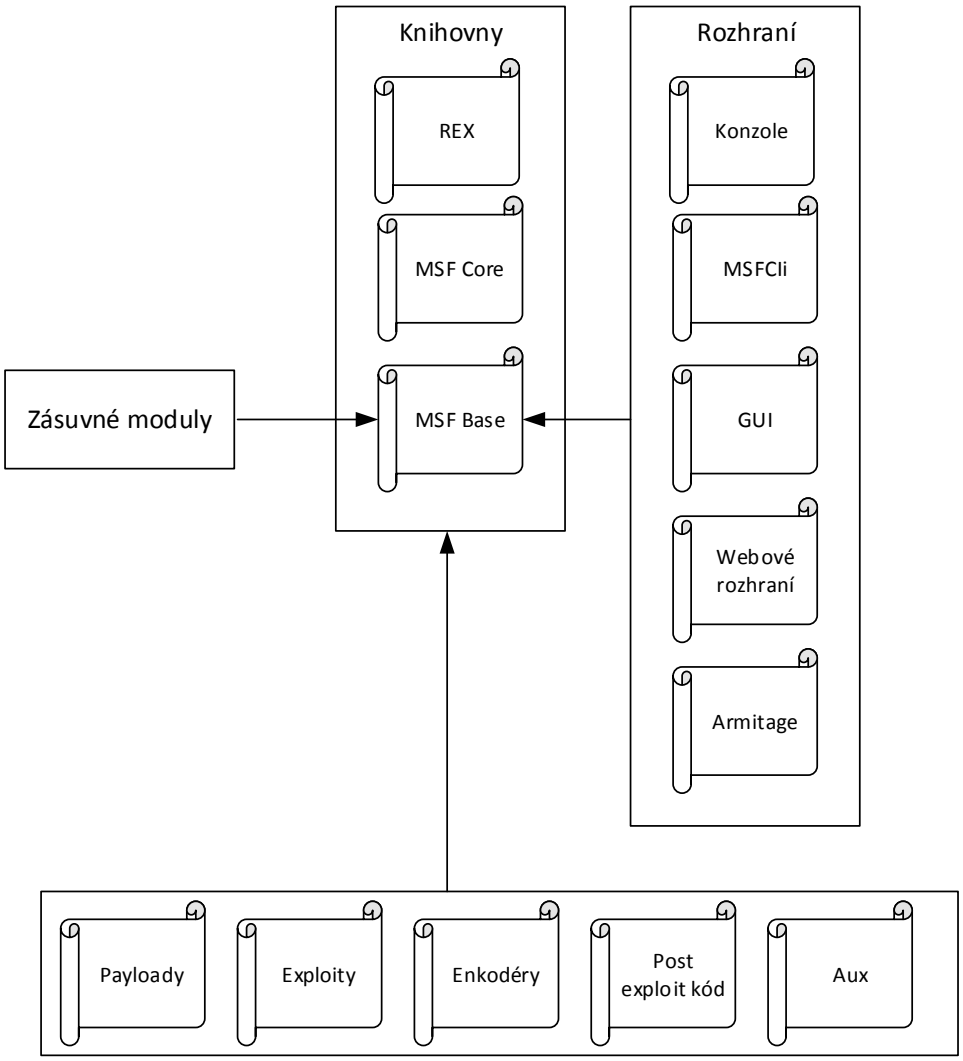
Používáním frameworků pro účely penetračního testování se zabývá například kniha Certified ethical hacker (Defino a Greenblatt, 2012), která se v důvodech jejich používání shoduje s předchozími odstavci a jako nejvhodnější a nejvíce propracovaný framework pro penetrační testování odkazuje právě na Metasploit framework. V knize lze nalézt celou řadu praktických příkladů pro ukázkové použití tohoto nástroje, obdobná řešení a doporučení lze nalézt i v knize Metasploit penetration testing cookbook (2012), která navíc čtenáři poskytuje celou sadu podrobných návodů, jak prakticky využít Metasploit framework. V souladu s úvodní částí diplomové práce a další výše zmíněnou literaturou lze konstatovat, že Metasploit framework (MFS) je mezi odborníky na penetrační testování velice rozšířen

a používán pro jeho funkcionality. Z tohoto důvodu budou MSF věnovány následující kapitoly práce.

MSF existuje ve třech dostupných verzích. Lze zvolit Community edition určenou primárně pro studijní účely nebo Express edition, která je oproti Community edition rozšířena o možnost provádění auditování hesel, generování přehledných reportů a provádění testování zranitelností podle předem připravených scénářů. Ještě vyšší počet funkcionalit nabízí Pro edition, která ke výše zmíněným schopnostem přidává možnost testovat webové aplikace, obsahuje moduly pro sociální inženýrství a nabízí průvodce pro penetrační testování. Jednotlivé ceny pak na žádost sděluje obchodní konzultant společnosti Rapid7.

4.5.4.2 Základní architektura MFS

Architektura Metasploit frameworku je i díky naprogramování v jazyce Ruby, silně modulární. Architekturu lze rozdělit do čtyř základních bloků, jak ukazuje následující schéma MFS.



Obrázek 16 Architektura metasploit frameworku (Autor - zpracováno dle (Jaswal, 2014))

Rozhraní zajišťují interakci mezi uživatelem a samotným jádrem frameworku tvořeným knihovnami. Mezi nejpoužívanější rozhraní pak patří:

- Msfcli - je dostupné přímo z příkazového řádku Linuxové konzole a je vhodné pro začátečníky, protože jim usnadní pochopení podstaty práce s MSF. Podrobnosti o tomto rozhraní získáme v příkazové řádce zadáním příkazu msfcli –h nebo v knize Mastering Metasploit (Jaswal, 2014), kde je uvedena i celá řada praktických příkladů.
- Msfconsole - je oproti Msfcli robustnější, škálovatelnější a umožňuje vyhledávat mezi jednotlivými exploity. Umožňuje používat globální proměnné tak, aby se při obměně používaných exploitů nemusely ke každému exploitu nastavovat znovu. Také lze z jedné Msfconsole

používat více relací, což je vhodné zejména při rozsáhlém penetračním testu, kdy je třeba přepínat mezi různými relacemi napadených systémů.

- Armitage - představuje grafickou nadstavbu pro Metasploit framework. Přidává také některé důležité funkcionality a jeho autorem je Rafael Mudge. Spuštění tohoto GUI obecně patří k těm jednodušším úlohám spadajícím do problematiky penetračního testování. Do příkazové řádky pak stačí napsat příkaz armitage a počkat, než se GUI spustí. Oproti Msfconsoli má hned několik výhod, mezi které patří práce v tzv. multiplayer režimu, kdy jeden z testerů má svůj stroj jako teamserver a ostatní se na něj připojují. Po úspěšné exploitaci se pak jeden tester může zabývat důležitými soubory na disku, jiný pak může zkoumat databázi uživatelů a třetí řešit odchytávání stisknutých kláves pomocí MSF. Další důležitou vlastností je podpora skriptování v jazyce Cortana, pomocí něhož lze automatizovat činnosti prováděné v průběhu penetračního testu.

Další částí architektury jsou knihovny, které poskytují svému okolí základní služby, díky nimž nemusí programátor řešit rutinní úlohy, jako je například přístup k síti. Vše je předpřipraveno v podobě knihoven, které stačí zavolat. Mezi základní knihovny dle Metasploit penetration testing cookbook (2012) patří:

- REX - poskytuje frameworku určitou míru abstrakce nad hardware, neboť jeho úkolem je práce se síťovými sockety, dále poskytuje frameworku logovací rozhraní a obdobné základní operace nutné pro fungování komplexního systému, kterým MSF bezpochyby je.
- MSF core - jedná se o knihovnu plnící základní úlohy ve frameworku, jako je manipulace s relacemi meterpreteru, kódování payloadů, aby ochranné mechanismy exploitovaného systému nic nezpozorovaly a obdobné činnosti.
- MSF base - vytváří rozhraní k MSF core. Poskytuje programátorovi přívětivé wrappery ke komponentám v části MSF core.

Pluginy (česky označované jako zásuvné moduly) slouží pro rozšíření funkcionalit a automatizaci vybraných činností. Díky modulům je MSF univerzální a poskytuje tak široké funkcionality. Zásuvné moduly umožňují MSF rozšiřovat své schopnosti o nové funkce, protože MSF byl vytvořen jako modulární s důrazem na neomezování se na funkcionalitu od vybraných tvůrců. Pro Metasploit framework lze takto vytvářet moduly rozšiřující jeho funkcionality. Několik modulů je implicitně dodáváno s instalací metasploit frameworku a lze je nalézt výpisem adresáře /opt/metasploit/msf3/plugins/. Moduly třetích stran jsou pak do MSF zaváděny pomocí příkazu load.

4.5.4.3 Meterpreter

Po úspěšné exploitaci (zneužití zranitelnosti a následného přístupu ke zdrojům) testovaného systému logicky vzniká potřeba interakce s napadeným systémem. Je zřejmé, že jednou z možností jak zajistit interakci se systémem je pomocí vlastní příkazové řádky, tedy shellu. Tento přístup má však dle dokumentace Metasploit's Meterpreter (2004) několik úskalí:

- byl by vytvořen nový proces,
- závislost shellu na platformě,
- rutinní post-exploitation úlohy by bylo třeba řešit při každé exploitaci stále dokola.

První bod je nebezpečný v tom, že je viditelný pro antivirový software či HIPS. Proto je nutné nově spouštěné procesy nastavit jako skryté, čímž se o něco snižuje pravděpodobnost prozrazení napadení systému. Druhý a třetí bod jsou poměrně problematické, protože exploity existují pro různé typy operačních systémů a každý operační systém má jiný shell a stejné činnosti je v každém OS třeba provádět jinak. Tento problém byl tvůrci MSF vyřešen tak, že vytvořili shell, jež je zcela univerzální a tedy nezávislý na platformě OS. Shell je rozšiřitelný za běhu a je schopen fungovat uvnitř již existujícího procesu bez interakce s pevným diskem. Těchto vlastností bylo docíleno využitím techniky DLL injection, která umožňuje donutit proces, aby zavedl DLL knihovnu do svého virtuálního adresového prostoru, vytvořil v něm nové vlákno a zde prováděl kód mající na starosti obsluhu meterpreter shellu.

Architektura Meterpreteru vychází z obecně znám ho modelu client-server, kdy proces, jenž je využit pro DLL injection (jinými slovy - v jehož adresovém prostoru meterpreter běží), funguje jako server a přijímá příkazy od klienta, zpravidla msfconsole ovládané penetračním testerem a tyto příkazy pak zpracovává ve vlákně, které pracuje s DLL obsahující obslužné rutiny meterpreteru.

Další důležitou vlastností je rozšiřitelnost za běhu. Meterpreter totiž obsahuje API, které lze využívat v modulech, které je možné nahrávat na server za běhu a je tedy umístěno na cílovém exploitovaném systému. Příklady rozšíření jsou:

- Fs - slouží pro nahrávání a stahování souborů na systém s meterpreter shellem.
- Net - umožňuje nastavení sítě na systému, kde běží meterpreter.

Škála možností, jenž meterpreter nabízí, je díky možnostem dynamického rozšiřování prakticky neomezena.

Meterpreter ovšem nezůstává pouze u skriptů využívajících předpřipravených funkcionalit a tvorby těchto skriptů. Například lze využít rozšíření Railgun, které na exploitovaném výpočetním systému s OS Windows umožní spustit libovolný kód, který lze distribuovat ve formě DLL knihoven. Tyto knihovny jsou v systému přítomny již od instalace Windows a umožňují tak například zrušit uživatele, nebo je možno na cílový systém vlastní DLL knihovny nahrát a z nich pak volat libovolný kód.

Komunikační protokol mezi klientem a serverem architektury meterpreteru je postaven na modelu TLV, který jednoduše umožňuje rozšíření stávající komunikace o nové typy zpráv. Pokud je třeba přidat do komunikačního protokolu novou funkcionalitu, definuje se pouze nový typ zprávy bez nutnosti předělávání zdrojového kódu zpracovávajícího stávající TLV. Tento princip také například využívá směrovací protokol EIGRP.

4.5.4.4 Další funkcionality MSF

MSF nabízí kromě vlastního shellu popsaného v předchozích odstavcích i další funkcionality. Jednou z nich je pivoting, který lze využít u sítí s vyšší mírou zabezpečení používajících síťový segment označovaný jako demilitarizovaná zóna. Do této zóny se zpravidla umisťují servery, které mají být přímo dostupné z internetu.

Princip pivotingu spočívá v tom, že po úspěšném spuštění meterpreteru na serveru v DMZ se jménem tohoto serveru, začíná komunikovat s vnitřní sítí. Server poskytující tuto službu se pak nazývá pivot. Z pivota tak lze scanovat porty počítačů ve vnitřní síti, nebo nahrávat a spouštět meterpreter. V MSF je pak potřeba nastavit přes jakou relaci meterpreteru jsou adresy vnitřní sítě dosažitelné.

Jelikož je ve vnitřní síti často třeba udělat nejprve průzkum a cesta do vnitřní sítě je nastavena přímo v MSF, externí nástroje (například Nmap) by nebyly schopny dosáhnout vnitřní sítě. Existuje tedy možnost tyto nástroje nahradit pomocí modulů. Jedním z těchto modulů může být již zmíněný Nmap, který nabízí funkcionality k otestování dostupnosti počítačů v zadaném adresním rozsahu nebo skenování zadaného adresního rozsahu. MSF nabízí i ARP scanner, který testuje, zda daná IP adresa reaguje, a několik dalších technik skenů, které zjistí na reagujících IP adresách otevřené porty a služby na nich běžící.

5 Případová studie

Cílem této kapitoly je zpracovat pracovní postup pro přípravu a realizaci penetračních testů v rámci energetické společnosti včetně doporučení konkrétních metod a nástrojů pro použití ve specifickém prostředí datových sítí společnosti. Tento pracovní postup bude dále podstoupen externím specializovaným partnerům, kteří budou samotné testy realizovat z důvodů zachování maximální míry objektivity a relevantnosti.

5.1 Penetrační testy pro datové sítě v energetice

Navržený postup penetračních testů má za úkol prověřit stav bezpečnosti infrastruktury nebo její části v prostředí energetické společnosti. Metodiky penetračního testování a vhodné typy jednotlivých testů budou navrženy tak, aby pokryly technologie používané v přenosové soustavě. Jedná se zejména o externí testy, které jsou ze své podstaty zaměřené na prověření zabezpečení prvků z prostředí internetu, penetrační testy webových aplikací, jež jsou v systémech využívány pro zobrazování informací z jednotlivých systémů. Dále se jedná o testy, které slouží k prověření zabezpečení speciální aplikace používané k monitorování a ovládání systému pro vzdálený přístup k řídicím systémům společnosti a v neposlední řadě penetrační test využívající metody sociálního inženýrství. Tento poslední test ověří, zda pracovníci společnosti dodržují interní bezpečnostní směrnice a zásady při poskytování citlivých dat.

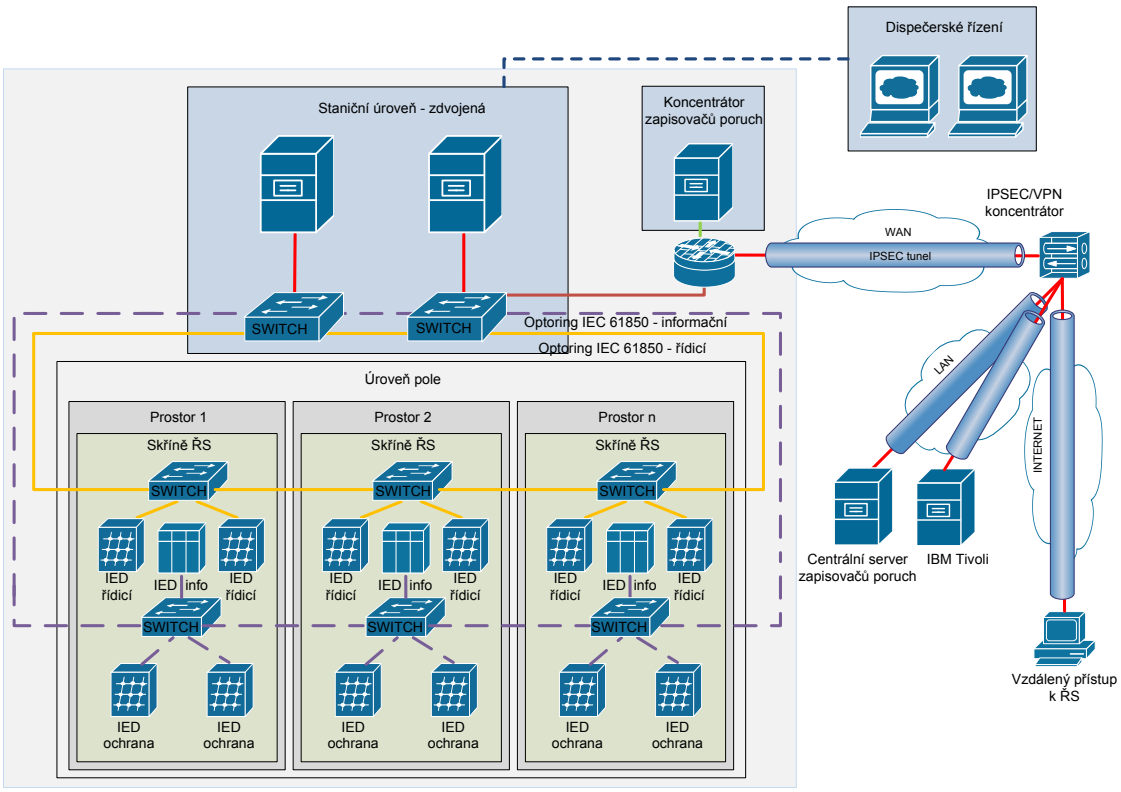
Ke správnému pochopení síťové infrastruktury energetické společnosti je nutné se seznámit s topologií řídicích systémů již před započetím penetračního testování. Tyto informace jsou důležité pro představu testera o rámci testovaných zařízení. První informací bude popis hlavních funkcí řídicích systémů.

Hlavní funkce řídicích systémů ve stanicích energetické společnosti jsou:

- ovládání silových prvků (místně z operátorského pracoviště na rozvodně, dálkově z dispečinku a místní ovládání na úrovni pole);
- blokovací podmínky zabraňující nepovoleným manipulacím;

- synchronizované spínání vývodů;
- hladinová regulace napětí v polích transformátorů
- získávání dat z procesu (události, alarmy, měřené hodnoty), sdružování signalizace na dispečink.

Topologie řídicího systému, který je v zásadě předmětem penetračního testování je znázorněna na následujícím obrázku.



Obrázek 17 Topologie testovaného řídicího systému energetické společnosti

Typický řídicí systém tvoří staniční úroveň a úroveň pole. Staniční úroveň se zpravidla skládá ze SCADA serverů, komunikačních serverů, frontendů, jednotky časové synchronizace (na bázi přijímače GPS), směrovačů, tiskáren a HMI (Human Machine Interface) sloužící pro místní ovládání silových prvků v rozvodně. Od doby, kdy jsou stanice vybaveny řídicími systémy na bázi všeobecně uznávaného standardu IEC 61850, komunikují prakticky všechna zařízení na obou úrovních po ethernetové sběrnici. Komunikační servery, které jsou součástí staniční úrovně, zajišťují komunikaci s dispečinky a s dalšími zařízeními jako jsou automatické regulace napětí na elektrárnách, řídicí systémy distribučních soustav, technická dohledová centra

či jiné monitorovací systémy. Všechny tyto komunikace s externími subjekty jsou sériové na bázi standardu IEC 60870 5-101.

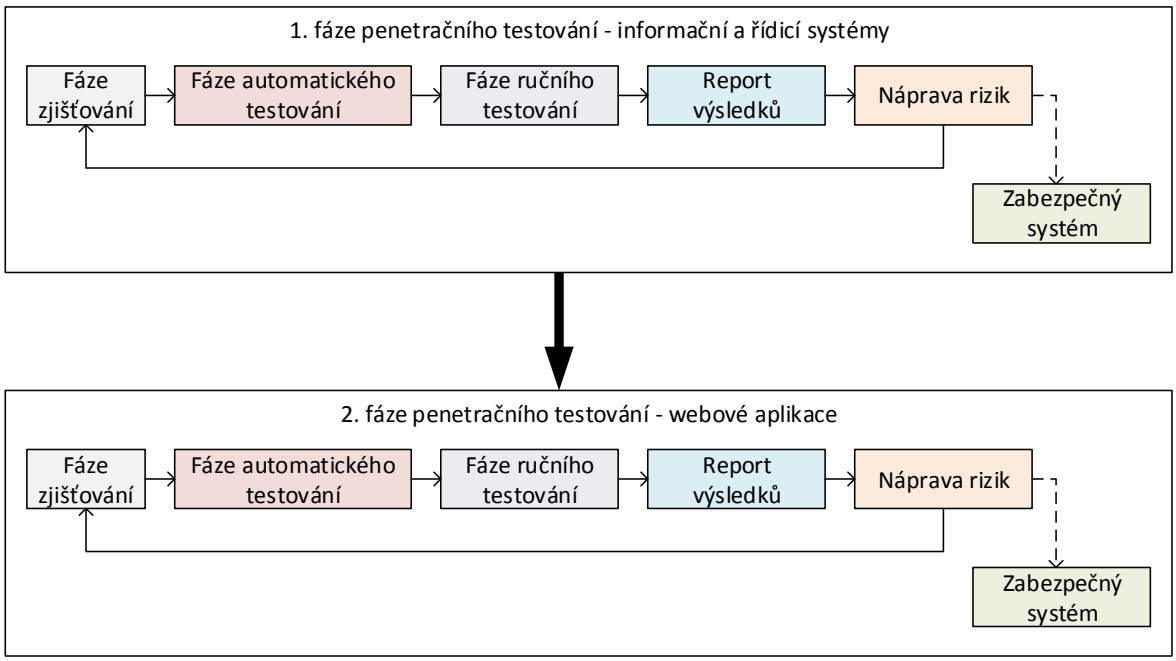
Úroveň pole tvoří inteligentní jednotky IED (Inteligent Electronic Device), které jsou metalicky napojeny na proces rozvodny pomocí binárních vstupů a výstupů a umožňují rovněž přímé analogové měření (napětí, proud, činný a jalový výkon, frekvence) a také měření pomocí externích převodníků (např. teplota oleje transformátoru). Tyto jednotky spolu komunikují horizontálně z důvodu výměny blokovacích podmínek pro manipulaci a zároveň komunikují vertikálně se staniční úrovní rozvodny po dvou nezávislých optických okruzích. Jeden z kruhů slouží pro komunikaci IED pro řízení polí a druhý kruh je použit pro komunikaci IED zařízení, která slouží k přenosu informací z procesu rozvodny do staniční úrovně, jako jsou jednotky RTU, hladinové regulátory napětí nebo ochrany. Jak již bylo řečeno, v současnosti je z hlediska jednotnosti výlučně podporován protokol na bázi standardu IEC 61850 provozovaný na 10/100/1000 Mbps Ethernet. Tento moderní mezinárodní komunikační standard, speciálně vyvinutý pro aplikace v energetice, umožňuje integrovat ochranné, řídicí, měřící a monitorovací funkce rozvoden a poskytuje vysokorychlostní procesní komunikace pro funkci blokovacích podmínek a vypínání ochran pomocí IED zařízení.

Každá rozvodna je vybavena staničním routerem, který zproztředkovává komunikaci zapisovačů poruch na bázi IED úrovně pole a koncentrátoru zapisovačů poruch a potom přenos poruchových záznamů ze všech staničních koncentrátorů přes VPN na centrální server, který pomocí webové aplikace poskytuje informace specialistům ochran. Z obrázku je patrný i systém vzdálených přístupů k zařízením v rozvodně pomocí VPN na bázi IPsec tunelu. Konfigurace VPN koncentrátoru probíhá pomocí GUI v aplikaci Tivoli na IBM serveru.

Jelikož jsou všechna aktivní zařízení, která jsou součástí řídicího systému v rozvodně, dostupná přes Ethernet, přináší toto řešení výhody, jako je vysokorychlostní komunikace nebo dostupnost k zařízením IED přes webový server, který je zpravidla jejich součástí. Na druhé straně výhody, mezi něž patří i vzdálený přístup, mohou, z pohledu tolik diskutované kybernetické bezpečnosti, představovat vysoká rizika zneužití provozované technologie. Proto je třeba z hlediska strategie

a důležitosti řídicích systémů v přenosové soustavě mít na zřeteli jak samotnou funkci, tak co nejvyšší bezpečnost řešení.

V případě úspěšného testu, kdy se podaří testerovi proniknout do infrastrutury společnosti, se jedná o bezpečnostní slabiny v testovaném systému. Prostředí energetiky je specifické v několikaúrovňovém přístupu k jednotlivým systémům a jejich významu pro bezproblémový chod energetické sítě. Proto je bezpodmínečně nutné jednotlivé úspěšné testy podrobněji analyzovat a vyhodnotit, zda se jedná o průnik do čistě informačních systémů, nebo do řídicích systémů ovládajících konkrétní technologii pro přenos elektrické energie. Pokud je test neúspěšný, znamená to, že se testerovi nepodařilo proniknout do infrastruktury. Zde je však nutné si uvědomit, že tento výsledek nezaručuje neexistenci bezpečnostních slabin. Tato skutečnost je přímo závislá na zkušenostech, znalostech testerů a času, který měl tester pro odhalení slabin systému k dispozici.



Obrázek 18 Znázornění logického propojení jednotlivých fází penetračního testování

5.2 **Vhodné metodiky a postupy**

Pro provedení penetračních testů specifických pro energetickou společnost lze jednoznačně doporučit využití uznávaných metodik a standardů, mezi které lze jistě zařadit:

- OSSTMM (Open Source Security Testing Manual) - vhodná a využívaná pro externí penetrační testy.
- OWASP (The open Web Application Security) - metodika využívaná pro realizaci testů webových aplikací.

Aby se test svým průběhem co nejblíže přiblížil reálným útokům, je třeba rozdělit jednotlivé činnosti do několika fází, jež lze jednoznačně identifikovat s doporučenými postupy penetračních testů, jak byly představeny ve třetí kapitole:

- Fáze průzkumu – jedná se o sběr informací z dostupných zdrojů, skenování aktivních prvků v síti, detekce operačních systémů a verze aplikací. V této fázi jsou aktivně využívány i metody sociálního inženýrství.
- Automatizované testy – tyto testy používají automatizované skripty a metodu "Brute force", jež je založena na pokusech o prolomení hesel z dostupných seznamů využívaných hesel. Tyto testy slouží převážně jako výchozí a získané informace jsou využity při dalším rozšířeném testování.
- Manuální testy – jsou využívány pro podrobnější ověření výstupů získaných pomocí automatizovaných testů a následnému provedení specifických testů, mezi které patří útoky na webové aplikace pomocí různých manuálních metod průniků. Tyto testy bývají zpravidla dobrým způsobem rozšíření základních automatizovaných testů.
- Report – jedná se o zpracování závěrečné zprávy, která musí obsahovat popisy průběhů testů a jejich jednotlivých výsledků. Plnohodnotný report také obsahuje návrhy a doporučení testera pro zlepšení bezpečnostních opatření v souladu s pravidly etického hackingu.

Závěrečná zpráva je pak pro společnost, jež zadala provedení penetračních testů, interním materiálem, jehož výsledky nesmí být veřejně dostupné a to ani v rámci firemního prostředí, aby bylo minimalizováno nebezpečení zneužití a to jak úmyslného, tak neúmyslného v rámci metod sociáního inženýrství. Na základě závěrečné zprávy pak musí společnost vyhodnotit výsledek testů vzhledem k interním bezpečnostním směrnicím a zákonnému rámci, v němž se energetické společnosti pohybují. Po podrobné analýze jednotlivých nálezů a vyhodnocení jejich míry rizika a dopadů na společnost, musí být provedena nápravná opatření v čase odpovídajícímu danému riziku pro snížení bezpečnostních rizik a posléze k jejich finální eliminaci. Zde je opět nutné zohlednit míru rizika ohrožení samotné funkčnosti přenosových systémů a jedná-li se o bezpečností rizika související přímo přenosem či distribucí elektrické energie, je nutné provést opravná opatření bezodkladně tak, aby nebyl ohrožen samotný přenos elektrické energie. Finální úpravy, jež mohou být časově náročnější, pak lze řešit v následujícím časovém období. Každý systém, jenž byl na základě závěrečné zprávy inovován a to nejen na úrovni zabezpečení, musí být následně opět otestován penetračními testy.

Specifikem penetračních testů v energetické společnosti je, že během testů nesmí být v žádném případě ohrožen provoz jakéhokoliv zařízení soustavy energetické sítě, dále nesmí dojít k modifikaci či kopírovaní jakýchkoliv dat ze systémů. Pokud by při úspěšném průniku mohlo dojít k nenávratným zásahům či negativnímu ovlivnění funkčnosti testovaného systému, musí být neprodleně kontaktována pověřená kontaktní osoba společnosti a další postup a činnosti testera s ní musí být konzultovány.

5.3 Doporučené nástroje pro penetrační testování

V první fázi průzkumu dojde ke zjištění aktivních zařízení v síti, jak podrobně uvádí třetí kapitola. Mezi vhodné nástroje pro tento typ testu patří aplikace Nmap, s jejíž pomocí tester např. zmapuje otevřené TCP/UDP porty, verzi operačního systému, ale stejně dobře poslouží i pro detekci provozovaných služeb. Výstup tohoto testu je nedílnou součástí závěrečné zprávy. Pro strukturní a osobní průzkum údajů o společnosti se doporučuje použit nástroj Maltego.

V automatizované a manuální fázi penetračního testu lze např. použít aplikaci Nessus, jež slouží pro odhalení globální zranitelnosti sítě. Pro specifické testy pak lze doporučit využití nástrojů Hydra a John the Ripper pro online a offline prolamování hesel tzv. hrubou silou. Nástroje pro testování webových aplikací, které slouží jako dashboard pro správu vzdálených přístupů a informací z řídicích systémů nejsou v této fázi penetračního testování kritické, protože se jedná pouze o monitorovací a zobrazovací funkce konsolidovaných informací, které nejsou z hlediska infrastruktury a případného průniku do systému kritické. Jejich testování je plánováno až v následné sekundární časti.

5.4 Využití externích penetračních testů

Externí penetrační test, jak je popsáno v první kapitole práce, slouží k prokázání kvality správného nastavení prvků, konfigurace a zabezpečení infrastruktury energetické společnosti z vnějšího prostředí internetu. Tester provede simulaci externího útoku a pokusí se o průnik do systému z internetu.

Pro externí penetrační test jsou charakteristické následující vlastnosti. V první řadě tester disponuje pouze omezenou bází znalostí o prostředí společnosti a vychází z běžně dostupných informací, případně má od zadavatele k dispozici seznam IP adres a také může použít informace od odborného pracovníka či informace získané prostřednictvím metod sociáního inženýrství. Tyto informace tester získává v přípravné fázi externího penetračního testu. Jako zdroj obecně slouží veřejně dostupné registry a informace z databáze RIPE, informace poskytnuté službou DNS a získané využitím aplikace Maltego. Takto je zpravidla simulován náhodný útok hackerem nedisponujícím citlivými interními informacemi o struktuře společnosti.

Testování je provedeno formou skenování a zjištení otevřených TCP a UDP portů, které je vedeno na všechna zařízení, jež jsou přístupná z internetu. U otevřených portů, které jsou testem odhaleny, je detekován protokol a nad ním provozované služby, stejně jako verze software těchto služeb. Rovněž dochází k detekci provozovaného operačního systému. Test je prováděn z různých IP adres, aby se jednoduše nedalo odhalit, odkud je útok veden a tyto adresy nebyly v rámci IDS/IPS systému okamžitě zablokovány. O probíhajícím testování musí být

informován pouze omezený okruh zaměstnanců společnosti a to převážně z oblasti bezpečnosti a správy IT infrastruktury. Tato strategie mimo jiné prověří účinnost a dynamiku monitorovacích mechanismů infrastruktury energetické společnosti.

Po automatickém skenování portů budou vyhodnoceny výsledky testů a proběhne test jednotlivých služeb přístupných na serverech či aktivních síťových prvcích a to jak automatizovaně, tak i manuálně. Zde jsou ve velké míře využívány specifické znalosti a zkušenosti testera, včetně jeho nejnovějších poznatků v oblasti zabezpečení poskytovaných služeb. Tím je možné odhalit případné slabiny, kterými jsou zpravidla chyby v aplikacích, staré verze software, chybějící instalace opravných balíčků či nedostatky v konfiguracích použitých aktivních síťových prvků.

Externí penetrační testování bude probíhat pomocí vzdáleného přístupu z internetu bez nutnosti přítomnosti testera v budově energetické společnosti v následujícím rozsahu:

- Průzkum oblasti testu – detekce aktivních IP adres skenováním nejpoužívanějších TCP portů, průzkum DNS zón.
- Dostupnost zařízení – skenování TCP portů v rozsahu 1-65535 a skenování UDP portů na aktivních IP adresách, detekce OS a typu provozovaných služeb.
- Fingerprint – detekce poskytovaných služeb a verzí software.
- Síťová topologie – průzkum síťové topologie a chování protokolů TCP/IP.
- Automatizované testy – testy nalezených aktivních prvků pomocí programu Nessus.
- Manuální testy – manuální ověření výsledků automatizovaných testů a ověření typických slabin služeb s realizací průniků.
- Prolamování hesel – testování přednastavených a jednoduchých hesel.
- Ostatní testy- další testování specifických zranitelností.

Získané výsledky, odhalené nedostatky a doporučení pro odstranění zjištěných problémů slouží jako nedílná součást závěrečné zprávy.

5.5 Testování webových aplikací

Testování webových aplikací není z důvodu obsahu a zaměření penetračních testů v první fázi testování vyžadováno, protože webové aplikace nejsou z pohledu zajištění služeb poskytovaných energetickou společností kritické. Některé aplikace nelze testovat externě z důvodu nutnosti využití VPN přístupu, který není testerovi k dispozici během externího penetračního testování. Po dohodě s odborným zástupcem společnosti proběhne v sekundární fázi testování, které bude simulovat počínání externího útočníka z vnějšího prostředí internetu. Během testování webových aplikací je nutné si uvědomit, že jelikož webové aplikace jsou výlučně dílem na zakázku, jedná se o testování specifických a jedinečných neopakovatelných chyb závislých na lidském faktoru a zkušenostech v oblasti bezpečnosti webových aplikací programátora, který aplikaci vytváří. Z toho důvodu málokdy odhalí bezpečnostní slabiny automatizovaný nástroj a je třeba využít znalostí a zkušeností penetračního testera.

Jako základ pro testování je doporučeno využít osvědčenou metodiku OWASP (The Open Web Application Security Project). Pro testování konkrétních oblastí testovaných webových aplikací lze jednoznačně doporučit využití dokumentu OWASP Testing Guide. Okruhy nejvýznamnějších problémů webových aplikací jsou popsány ve známém dokumentu (OWASP-TOP10, 2013).

5.6 Využití získaných výsledků

Na závěr penetračního testování je samozřejmostí zpracování a předání závěrečné zprávy, která musí obsahovat následující části:

- Souhrn určený pro manažery – stručný přehled průběhu testu spolu s výsledky pro účel seznámení manažerů společnosti s výsledky testů.
- Popis testu – bude obsahovat použité metodiky testování včetně přehledu všech provedených činností.
- Zjištěné skutečnosti – detailní popis výsledků provedených testů pro jednotlivá zařízení.

- Závěr – přehled všech doporučení pro odstranění zjištěných závad na základě výsledku testu.

Na základě výsledků plynoucích ze závěrečné zprávy je nutné následně přijmout odpovídající opatření a zároveň je nezbytné poučit se z odhalených nedostatků tak, aby se v budoucnosti neopakovaly. Výsledky zprávy budou škálované dle výše rizika v následujících kategoriích:

- Pozorování - takto označené nálezy nemají zásadní dopad na bezpečnost.
- Připomínka - tímto způsobem jsou označeny hrozby, které nemají zásadní nebo pravděpodobný dopad, nebo nelze míru závažnosti odhadnout. Obvykle se doporučuje problémy tohoto typu odstranit.
- Nedostatek - hrozby tohoto typu mají zpravidla nezanedbatelný dopad na bezpečnost testovaného systému. Ve všech případech následuje doporučení k odstranění nedostatku, které by mělo být společností realizováno v nejbližší době.
- Slabina - tyto hrozby mají významný dopad a vždy následuje doporučení testera k odstranění zjištěných slabin. Doporučené opatření by mělo neodkladně následovat pro dosažení bezpečnosti.
- Průnik - toto označení znamená nejhorší možnost, kterou je úplný nebo částečný průnik testera do systému. Může se jednat i o scénář možného útoku, který tester v závěru testování nemusel realizovat. Nález tohoto typu bývá doprovázen nálezy slabin či nedostatků, jejichž zneužití vedlo k úspěšnému průniku. Následná realizace doporučených opatření je bezpodmínečně a bezodkladně nutná k zajištění bezpečnosti.

Informace obsažené v závěrečné zprávě jsou klíčovým podkladovým materiálem pro zvýšení bezpečnosti informačních a řídicích systémů energetické společnosti. Vzhledem k vysokému riziku zneužití obsažených informací je bezpodmínečně nutné zajistit omezený přístup k této zprávě. Seznam osob, kterým je umožněn přístup k důvěrným informacím obsažených ve zprávě, je určen interní

bezpečnostní směrnicí společnosti. Pracovníci zodpovědní za realizaci nápravných opatření musí na základě doporučení vyplývajících ze závěrečné zprávy zajistit provedení nápravných opatření v době, která odpovídá míře rizika. Po provedení nápravných opatření je důležité realizovat navazující penetrační test, který prokáže účinnost realizovaných nápravných opatření.

6 Závěr

Cílem této diplomové práce bylo představit problematiku penetračních testů a zpracovat případovou studii využívající nejnovější metody a technologie využívané v penetračním testování orientovaném na oblast energetických systémů.

V samotném úvodu práce byly za pomoci vybraných relevantních zrojů definovány základní pojmy z oblasti bezpečnosti informačních technologií a penetračního testování. Dále zde byly představeny nejpodstatnější oblasti zabývající se etickým hackingem a penetračním testováním tak, aby byl ukázán narůstající význam této oblasti IT.

Teoretická část práce je pak logicky rozdělena do čtyř kapitol, které popisují základní pohledy a techniky penetračního testování, kam jistě patří rozdělní na externí a interní peneračí testování. Druhá kapitola byla věnována problematice etického hackingu, kde byly představeny nejen základní přístupy a pohledy na tuto oblast testování informační bezpečnosti, ale také její vztah a zakotvení v právních předpisech.

Penetrační testování, jež lze dnes považovat za specializovanou oblast ve světě informačních technologií vyžadující řadu specifických kompetencí od testera, je v dnešní době definováno pomocí specializovaných metodik, které pro testery představují komplexní a přesně definový rámec v němž se mohou pohybovat. Nejvýznamnějším metodikách penetračního testování byla věnována třetí kapitola, ve které byly představeny jejich charakteristiky, silné a slabé stránky a především oblast jejich využití a efektivního nasazení. Volba vhodné metodiky může výrazně ovlivnit efektivitu a relevantnost penetračních testů, a proto byla tomuto problému věnována dostatečná pozornost i při výběru metodiky pro případovou studii v závěru práce.

Kromě vhodné metodiky musí tester umět efektivně využít i jednotlivé nástroje pro samotné testování s ohledem na konkrétní fázi. Představení, a u vybraných nástrojů i praktickým ukázkám, byla věnována čtvrtá kapitola. Nástoje zde byly rozděleny dle jejich využitelnosti v rámci fází penetračního testování a byl

také kladen důraz na správnou interpretaci získaných informací. Tato kapitola také uzavírá teoretickou část práce z jejichž závěrů vycházel návrh případové studie penetračního testování v prostředí energetických systémů.

Případová studie měla za cíl definovat oblasti významné pro testování, sloužící k ověření bezpečnosti využívaných bezpečnostních systémů ve specifickém prostředí energetické společnosti. Jak již bylo zmíněno v samotné práci, penetrační testy v této oblasti jsou specifické zejména tím, že musí probíhat na živých systémech, jelikož není možné vytvořit komplexní model energetického systému v uzavřeném laboratorním prostředí a to především pro jeho rozsáhlost a provázanost jednolivých systémů. Dalším významným parametrem ovlivňujícím rozsah a dopad penetračních testů v energetice je, že nesmí mít vliv na funkčnost systémů, jenž by mohl znamenat výpadek systémů či částečné omezení jejich dohledu nebo možnosti jejich správy. Z toho důvodu bylo nutné u představené metodiky vyjít nejen z obecně uznávaných a popsaných metodik a postupů, ale také uplatnit úroveň znalosti daného prostředí, aby testy neměly negativní vliv na funkčnost celé energetické soustavy. Představené řešení bylo vytvořeno jako jeden z materiálů, který definuje oblast testování, vhodné nástroje pro testování a mantinely, za které není při testování možné zajít a je dodáváno externím subjektům, jež pro zajištění maximální možné míry nezávislosti a objektivity penetrační testy provádí.

Představená metodika také obsahuje orientační dělení zjistěných chyb a událostí s předem definovanými reakcemi, které zavisí na úrovni zjistěných nedostatků a obsahují i časové limity pro jejich nápravu. I zde musely být využity mezioborové vztahy a kompetence, jelikož na energetické systémy není možné 100% nasadit postupy a reakce z čistého světa informačních technologií.

7 Bibliografie

ANGEL, S. & SARALA, S., 2011. A study on Penetration Testing. *International journal of advanced research in computer science IJARCS ; a bimonthly scientific journal of computer science*, Svazek 2, pp. 396-398.

Anon., 2008. *Guide for mapping types of information and information systems to security categories.* [Online]
Available at: http://csrc.nist.gov/publications/nistpubs/800-60-rev1/SP800-60_Vol1-Rev1.pdf
[Přístup získán 15. 8. 2014].

Anon., 2008. *Technical Guide to Information Security Testing and Assessment.* [Online]
Available at: http://csrc.nist.gov/publications/nistpubs/800-115/SP800-115.pdf
[Přístup získán 9. 8. 2014].

Anon., 2013. *http://www.hackersforcharity.org.* [Online]
Available at: http://www.hackersforcharity.org
[Přístup získán 10. 10. 2014].

Anon., 2013. *Policies Tolls.* [Online]
Available at:
http://trygstad.rice.iit.edu:8000/Policies%20&%20Tools/RiskAssessmentValues.pdf
[Přístup získán 8. 8. 2014].

Anon., 2013. *Testing Guide 4.0 - Release.* [Online]
Available at:
https://www.owasp.org/images/5/52/OWASP_Testing_Guide_v4.pdf
[Přístup získán 10 8 2014].

Anon., 2013. *The world's most advanced Open Source vulnerability scanner and manager.* [Online]
Available at: http://www.openvas.org/
[Přístup získán 8. 6. 2014].

Anon., 2014. *FEDERAL INFORMATION SECURITY MANAGEMENT ACT (FISMA) IMPLEMENTATION PROJECT.* [Online]
Available at: http://csrc.nist.gov/groups/SMA/fisma/
[Přístup získán 10. 9. 2014].

Anon., 2014. *http://www.tenable.com/products/nessus/nessus.* [Online]
Available at: http://www.tenable.com/products/nessus/nessus
[Přístup získán 10. 10. 2014].

Anon., 2014. *Security Metrics - Attack Surface Metrics.* [Online]
Available at: http://www.isecom.org/research/ravs.html
[Přístup získán 10. 10. 2014].

BACUDIO, A. G., YUAN, X., CHU, B.-T. B. & JONES, M., 2011. N OVERVIEW OF PENETRATION TESTING. *International Journal of Network Security & Its Applications (IJNSA)*, Svazek 6, pp. 19-38.

BALAPURE, A., 2013. *Learning Metasploit exploitation and development.* Birmingham: Packt Pub..

BALAPURE, A., 2013. *Learning Metasploit exploitation and development.* Birmingham: Packt Pub..

BALOCH, R., 2014. *Ethical hacking and penetration testing guide.* Oakville: Apple Academic Press Inc..

BROAD, J. & BINDNER, A., 2014. *Hacking with Kali: practical penetration testing techniques.* Rockland: Syngress Media.

BROAD, J. & BINDNER, A., 2014. *Kali linux: assuring security by penetration testing.* místo neznámé:Packt Publishing Limited.

Dark reading, 2011. *Dark reading. Security dark reading.* [Online]
Available at: http://www.darkreading.com/database-security/167901020/security/attacksbreaches/231901051/strange-but-true-

penetration-testing-stories.html
[Přístup získán 10. 10. 2014].

DEFINO, S. & GREENBLATT, L., 2012. *Official certified ethical hacker review.* Boston: Course Technology.

ENGEBRETSON, P., 2013. *The basics of hacking and penetration testing: ethical hacking and penetration testing made easy.* Rockland: yngress Media.

GRAVES, K., 2010. *CEH: certified ethical hacker study guide.* Indianapolis: Wiley Pub..

GUSEV, M., RISTOV, S. & DONEVSKI, A., 2013. Security vulnerabilities from inside and outside the eucalyptus cloud. *ACM International Conference Proceeding Series*, pp. 95-101.

HERZOG, P., 2013. *RAV FAQ.* [Online]
Available at:
http://trygstad.rice.iit.edu:8000/Policies%20&%20Tools/RiskAssessmentValues.pdf
[Přístup získán 13. 10. 2014].

HERZOG, P., 2014. *Open Source Security Testing Methodology Manual (OSSTMM).* [Online]
Available at: http://www.isecom.org/research/osstmm.html
[Přístup získán 10. 10. 2014].

ISECOM , 2014. *ISECOM - Making Sense of Security.* [Online]
Available at: http://www.isecom.org/
[Přístup získán 9. 10. 2014].

JASWAL, N., 2014. *Mastering Metasploit.* Birmingham: Packt Publishing Limited.

Kali Linux, 2014. *Kali Linux: Ukradli jsme heslo do Seznamu a útočili na Wi-Fi Více na: http://www.zive.cz/clanky/kali-linux-ukradli-jsme-heslo-do-seznamu-a-utocili-na-wi-fi/sc-3-a-173671#utm_medium=selfpromo&utm_source=zive&utm_campaign=copylink.*

[Online]
Available at: http://www.zive.cz/clanky/kali-linux-ukradli-jsme-heslo-do-seznamu-a-utocili-na-wi-fi/sc-3-a-173671
[Přístup získán 10. 10. 2014].

Kali, 2014. *Kali.* [Online]
Available at: http://www.kali.org/
[Přístup získán 8. 8. 2014].

KUMAR, H., 2014. *Learning nessus for penetration testing.* místo neznámé:ackt Publishing Limited.

LIMWIRIYAKUL, S. & VALLI, C., 2011. Results from the Deployment of A Targeted Security Testing Framework for the Testing of Email Systems in Local Government in Western Australia. *International Journal of Information and Electronics Engineering*, pp. 16-22.

Long, J., Temmingh, R., Petkov, P. & Stewart, J., 2007. *Google Hacking for Penetration Testers.* místo neznámé:autor neznámý

LUA, 2013. *LUA.* [Online]
Available at: http://www.lua.org/about.html
[Přístup získán 30. 6. 2014].

Lyon, G., 2006. *Nmap Network Scanning : Official Nmap Project Guide to Network Discovery and Security Scanning.* místo neznámé:autor neznámý

MALTEGO , 2014. *MALTEGO SCRIPTING LANGUAGE.* [Online]
Available at: http://www.paterva.com/MSL.pdf
[Přístup získán 6. 6. 2014].

MCCLEANA, J., STULL, C., FARRAR, C. & MASCARENAS, D., 2013. A preliminary cyber-physical security assessment of the robot operating system. *Proceedings of SPIE - The International Society for Optical Engineering*, Svazek 8741.

MCCLEANA, J., STULL, C., FARRAR, D. & MASCARENAS, D., 2013. A preliminary cyber-physical security assessment of the robot operating system. *Proceedings of SPIE - The International Society for Optical Engineering*, Svazek 8741.

MCCLURE, S., SCAMBRAY, J. & KURTZ, G., 2012. *Hacking exposed 7: network security secrets.* New York: McGraw-Hill Education - Europe.

Metasploit's Meterpreter, 2004. *Metasploit framework.* [Online]
Available at: http://dev.metasploit.com/documents/meterpreter.pdf
[Přístup získán 10. 8. 2014].

Metasploit, 2012. *Metasploit penetration testing cookbook.* Birmingham: Packt Publishing ltd.

Microsoft , 2002. *Microsoft computer dictionary.* místo neznámé:autor neznámý

NIST, 2014. *Nation Institute of Standards and Technology.* [Online]
Available at: http://www.nist.gov/
[Přístup získán 9. 10. 2014].

OpenVas, 2014. *OpenVas.* [Online]
Available at: http://www.openvas.org/
[Přístup získán 23. 6. 2014].

OWASP, 2013. *Projects/OWASP Mobile Security Project - Top Ten Mobile Risks.* [Online]
Available at:
https://www.owasp.org/index.php/Projects/OWASP_Mobile_Security_Project_-_Top_Ten_Mobile_Risks
[Přístup získán 8. 9. 2014].

OWASP, 2014. *Welcome to OWASP.* [Online]
Available at: https://www.owasp.org
[Přístup získán 8. 8. 2014].

OWASP-TOP10, 2013. *Top 10 2013-Top 10.* [Online]
Available at: https://www.owasp.org/index.php/Top_10_2013-Top_10
[Přístup získán 8. 8. 2014].

Palmar, C. C., 2001. Ethical Hacking. *IBM Systems Journal*, Svazek 3, pp. 769-780.

Paterva, 2012. *Paterva.* [Online]
Available at: https://www.paterva.com/web6/products/download.php
[Přístup získán 10. 6. 2014].

Paterva, 2014. *https://www.paterva.com/web6/.* [Online]
Available at: https://www.paterva.com/web6/
[Přístup získán 7. 7. 2014].

Projects - OWASP, 2014. *Projects/OWASP Mobile Security Project - Top Ten Mobile Risks.* [Online]
Available at:
https://www.owasp.org/index.php/Projects/OWASP_Mobile_Security_Project_-_Top_Ten_Mobile_Risks
[Přístup získán 10. 10. 2014].

RAETHER, R. I., 2008. DATA SECURITY AND ETHICAL HACKING: Points to Consider for Eliminating Avoidable Exposure. *Business Law Today*, Svazek 1, pp. 55-58.

RAYMOND, E. S., 2012. *New hacker's dictionary version 4.2.2..* místo neznámé:Emereo Pty Limited.

Selecký, M., 2012. *Penetrační testy a exploitace.* Brno: COMPUTER PRESS.

Shodan, 2013. *Shodan.* [Online]
Available at: http://www.shodanhq.com/account/login
[Přístup získán 6. 7. 2014].

SHRAVAN, K., NEHA, B. & PAWAN, B., 2014. Penetration Testing: A Review. *COMPUSOFT International Journal of Advanced Computer Technology*, Svazek 4, pp. 752-757.

STEELE, G. L. a další, 2013. *The Hacker's Dictionary: A Guide to the World of Computer Wizards.* [Online]
Available at: http://jargon-file.org/archive/jargon-1.5.0.dos.txt
[Přístup získán 2014].

STROUPE, C., 2007. Hacking the cool: The shape of writing culture in the space of New Media. *Computers and Composition*, 24(4), pp. 421-442.

STYLES, M. & TRYFONAS, T., 2009. Using penetration testing feedback to cultivate an atmosphere of proactive security amongst endusers. *Information Management & Computer Security*, 17(1), pp. 44-52.

Trestní zákoník, 2009. *Zákon č. 40/2009 Sb., trestní zákoník.* [Online]
Available at: http://business.center.cz/business/pravo/zakony/trestni-zakonik/
[Přístup získán 10. 10. 2014].

WALLS, D. M., SCHOPIERAY, S. & DEVOSS, D. N., 2009. Hacking Spaces: Place as Interface. *Computers and Composition*, 26(4), pp. 269-287.

WATSON, G., 2014. *Social engineering penetration testing: executing social engineering pen tests, assessments and defense.* Rockland: Syngress Media.

Yuan, D., Li, X. & Zhu, N., 2014. Penetration depth forecast using BP neural network-based system. *Journal of Computational Information Systems*, 10(12), pp. 5001-5008.

8 Seznam obrázků

9 Seznam tabulek

10 Seznam zkratek

API	Application Programing Interface
ARP	Address Resolution Protocol
CDP	Cisco Discovery Protocol
SCADA	Supervisory Control And Data Acquisition
DMZ	Demilitarized Zone
DNS	Domain Name System
EIGRP	Enhanced Interiror Gateway Protocol
FIPS	Federal Information Processing Standard
FTP	File Transfer Protocol
GHDB	Google Hacking Database
GUI	Graphic User Interface
GPS	Global Positioning System
MSF	Metasploit Framework
HIPS	Host Intrusion Prevention Systém
HMI	Human Machine Interface
HTML	HyperText Markup Language
HTTPS	Hyper Text Transfer Protocol Secure
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IPS	Intrusion Prevention Systém
IPSEC	Internet Protocol Security
NASL	Nessus Attack Scripting Language
NIST	National Institute of Standards and Technology
OS	Operating System (Operační systém)
OSINT	Open Source Inteligence
OSSTMM	Open Source Security Testing Methodology Manual
OSPF	Open Shortest Path First
OWASP	Open Web Application Security Project
RAV	Risk Assessment Value
REX	Ruby Extension Library

RIPE	Réseaux IP Européens
SCADA	Supervisory Control and Data Acquisition
SET	Social Engineering Toolkit
SCAP	Security Content Automation Protocol
SNMP	Simple network management protocol
TCP/IP	Transmission control protocol/Internet Protocol
TLV	Type-length-value
UDP	User Datagram Protocol
VPN	Virtual Private Network
WWW	World Wide Web
XML	Extended Markup Language