ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA

V PRAZE

PROVOZNĚ EKONOMICKÁ FAKULTA

Bakalářská práce

= Útoky na webové stránky =

Vypracoval: Jiří PETLAN

Vedoucí bakalářské práce: RNDr. Dagmar BRECHLEROVÁ

Prohlášení

Čestně prohlašuji, že jsem bakalářskou práci na téma **Útoky na webové stránky** vypracoval samostatně, s využitím jen pramenů uvedených v použité literatuře a po odborných konzultacích s vedoucím bakalářské práce RNDr. Dagmar Brechlerovou.

V Praze dne 10.7.2006

.....

podpis

Poděkování

Mnohokrát děkuji své vedoucí bakalářské práce RNDr. Dagmar Brechlerové za její trpělivost, shovívavost, příkladné a metodické vedení při vypracovávání této práce, cenné rady, podnětné připomínky, pomoc a všechen čas, který mi věnovala.

Útoky na webové stránky

Souhrn

Tato práce se zabývá základními technikami útoků na webové stránky a odpovída na některé bezpečnostní otázky kolem nich. Nejdříve krátce seznámím s historií a vysvětlím původní význam slova „hack“. V další části práce popisuji základní metody průzkumu sítě a následuje přehled nejběžnějších typů útoku. Následující kapitola je praktickou částí této práce a snaží se presentované metody a techniky ukázat v praxi. Nakonec je řešena bezpečnostní problematika. Na problém bezpečnosti je nahlíženo jak ze srtany cílového systému, tak ze strany útočníka

Klíčová slova

Hacker, hackování, Internet, webové stránky, Whois databáze, trasování, skenování portů, inventarizace, odmítnutí služeb, SYN záplava, UDP záplava, ověřování typu Basic, formulářové ověřování, slabá hesla, lámání hesel hrubou silou, firewall, filtry, silná hesla.

Attacking the web pages

Summary

The thesis deals with basic techniques of attacking web pages and answers some security questions. The introducion of history is at the beginning and deffinition of “hack” term is also there. Next part of the thesis describes basic techniques of exploration target net. After that follows most common types of attack overview. The following chapter represents practical part of the thesis and shows presented methods and techniques in practice. End part of thesis solves security questions. This part is divided into two parts, one solves questions of target system and second shows how hacker can keep himself in safe.

Key words

Hacker, hacking, Internet, web pages, Whois database, tracing, port scanning, inventarisation, denial of service, SYN flood, UDP flood, Basic validation, login form, weak passwords, brute force password cracking, firewall, fillters, strong passwords.

Obsah

1. Úvod

Problematika útoků na webové stránky je stará jako webové stránky samotné. Počítačoví hackeři tu byli dříve než Internet, jak ho známe dnes, a jeho masové rozšíření a komercializace jen existujícím útočníkům rozšířily pole působnosti. Přesto musím připustit, že v době, kdy počítač přestal být předmětem vášnivých debat v studentských kavárnách a na základních školách je samozřejmostí vyučovat základy práce s osobním počítačem, se prostředky počítačového zločinu dostávají běžnému člověku přímo na dosah ruky. Příchod webových stránek a masivní nárůst počítačů připojených do sítě Internet jen rozšířil informační základnu a relativní mládí a nedokonalost technologie realizace webových stránek daly živnou půdu pokusům o nabourání se do systémů širokému okruhu lidí. Hackeři přestali být nadšenci a stali se profesionály a z hackingu se stal business. Na jedné straně stojí hackeři neomylně vyhledávající chyby v nových a nových programech a aplikacích a na druhé straně softwarové firmy chrlící ony programy a nepřetržitý proud oprav a záplat na své produkty.

Téma útoky na webové stránky jsem si vybral po dlouhém přemýšlení a vážení všech nabízených možností. Ohromná obsáhlost dané problematiky mi dávala určitý prostor v dodatečné volbě konkrétních témat a technik, kterými se budu v práci zabývat. Tato volnost mi dávala možnost na problematiku nahlédnout z libovolného úhlu a dovolila mi poodhalit problematiku hackování a výrazně rozšířit okruh znalostí a vědomostí v oblasti výpočetních systémů. Rozhodl jsem se zabývat technikami získávání informací a průzkumu sítě a několika základními technikami útoků na webové stránky a počítače, na kterých jsou ony stránky provozovány. Dále jsem se rozhodl prezentovat problematiku spíše z pohledu uživatele operačního systému Windows, i když nejednou zmíním souvislosti a konkrétní programy pro platformu Unix. Volbu tématu jsem prováděl s vědomím, že nově nabyté znalosti z tak mladého a rychle se rozvíjejícího segmentu by mohly být v budoucnu pevnou základnou při specializaci na některé konkrétní odvětví informatiky.

2. Cíl práce a metodika

Cílem této práce je poodhalit roušku tajemství a nahlédnout do zákulisí hackingu webových stránek. Nejdříve krátce seznámím s historií hackingu, kterou si rozdělím na tři etapy oddělené významnými milníky počítačové historie. V této části vysvětlím původní význam slova „hack“ a ukážu, jak se postupem času měnila kromě počítačů i celá povaha a styl „hackerů“. Poté se budu věnoval popisování různých metod a technik, od profilování cíle přes „oťukávání“ až po techniky samotného útoku. představím programy pro obě nejběžnější platformy Unix a Windows NT včetně zdrojů, případně odkazů, kde se dané utility dají stáhnout. Dále pak budu prezentovat jednotlivé typy útoků podle jejich zaměření. Jsou útoky, které mají za cíl změnit obsah stránek a jsou útoky, které mají za cíl vyřadit webový server z provozu a tím i samotné stránky. Po představení těchto technik se budu v další kapitole věnovat praktické ukázce práce s dříve představenými programy s tím, že vzhledem k faktu, že jsem pokus realizoval na počítači s operačním systémem Windows XP, tak se omezím na ukázku programů jen pro Windows. Na konkrétní webové servery budu aplikovat jednotlivé metody a utility s cílem ukázat jejich použití v praxi a představit reálná nebezpečí a náročnost, tedy spíše nenáročnost realizace sběru důležitých informací o cílovém systému a samotného nabourávání se do systémů webových serverů.

V závěru bude následovat kapitola, kde se budu zabývat bezpečností. Na problematiku bezpečnosti se v souvislosti s hackingem podívám ze dvou stran. Na jedné straně stojí útočník, který se snaží zůstat v bezpečí neodhalen, touto problematikou se budu zabývat nejdříve, a na druhé straně stojí webové stránky a servery, na kterých jsou provozovány. Krok za krokem popíši bezpečnostní opatření vhodná k jednotlivým metodám a technikám presentovaným v předchozích kapitolách. Čili, kde naleznu bezpečnostní díru, budu se snažit nalézt i tu pravou záplatu na tuto díru. Tato práce by měla být spíše než návodem, jak útočit na webové stránky, rádcem, jak preventivně bránit stránky a systém zajišťující jejich bezproblémový chod. Přednesu několik obecných rad a pak se budu věnovat jednotlivým konkrétním bezpečnostním problémům.

3. Historie

Je mnoho zdrojů, které se na Internetu zabývají historií počítačového zločinu. Některé dělí čas po desetiletích, některé se snaží o vlastní dělení, jako např. stránka The Hackers Hall Of Fame na serveru Discovery Online (http://tlc.discovery.com/convergence/hackers/bio/bio.html):

1. pravěk (do r. 1969)
2. dřevní doba (1970 - 1979)
3. zlatá éra (1980 - 1991)
4. přichází odveta (1986 - 1994)
5. tolerance nula (1994 - dodnes)

já bych rád toto členění lehce zjednodušil a období rozdělil pouze na 3 etapy:

1. pravěk (od vynálezu telefonu do uvedení prvního PC na trh v roce 1981)
2. středověk (od r. 1981 do r. 1994)
3. novověk (od 1994 dodnes)

3.1. Pravěk

Za první "počítačový" zločin je v literatuře [1] považován případ z Francie r. 1801. Tkadlec Joseph Jacquard sestrojil jednoduché zařízení, které automatizovalo a opakovaně provádělo úkony používané při tkaní speciálních látek. Zaměstnanci Jacquardovy manufaktury ve strachu ze ztráty pracovních míst přiměli Jackquarda pomocí sabotáží od dalšího vývoje upustit.

Vynález telefonu, jakožto prvního prostředku elektronické komunikace, který přenášel hlas na dálku, dal vzniknout fenoménu zvanému elektronické pohraničí (Electronic Frontier). Hranice jsou zde abstraktním pojmem, jedná se o svět mezi telefonními přístroji.A jelikož první komunikace počítačů vedla přes telefonní linku, dal telefon vzniknout tzv. cyberprostoru.

Samotný zrod počítačového věku lze datovat dnem 14. 2. 1946. Tehdy byl na pensylvánské univerzitě sestrojen první elektronický počítač ENIAC (více na http://ftp.arl.mil/~mike/comphist/eniac-story.html). Jeho následovníci na sebe nenechali dlouho čekat. Díky astronomické ceně bývaly počítačové laboratoře nejlépe střeženým místem ve firmě, kam mělo přístup jen pár programátorů. Rozhodně se nedalo mluvit o možnostech kriminálního zneužití. A právě v těchto dobách se zrodilo slovo, jehož sláva měla přijít. Programátoři, kteří na těchto počítačích pracovali, si často potřebovali poradit se špatně fungujícím programem. Technická podpora a masový vývoj software téměř neexistovaly, a tak byli programátoři odkázáni sami na sebe. Zásahy do programů, aby fungovaly lépe, se označovaly slovem "hacks". Odtud pochází slova hacking a hacker. Na tomto místě bych chtěl zdůraznit, že úloha tehdejších hackerů byla pozitivní, činili zásahy do programů, aby mohly být lépe využívány. Zajímavé je, že operační systém pro velké počítače a serverová platforma UNIX přišla na svět právě pomocí takového hacku.

Rok 1971 proslul případem Cap 'n' Crunch. Tak se jmenovaly cereálie, do nichž byla dávána píšťalka. John Draper (osobní stránky: (http://www.webcrunchers.com/crunch) objevil, že píšťalka vydává zvuk o frekvenci 2600 Hz, který při použití v telefonní lince umožní hovory zdarma. Zveřejňováním takových fint se nelegální telefonie rozmohla a dostala název phreaking. Mnohé osobnosti počítačového businessu mají phreakerskou minulost, lze jmenovat např. Steve Jobse, zakladatele firmy Apple Computers [2].

3.2. Středověk

Za počátek nové éry můžeme označit uvedení osobního počítače typu IBM PC na trh. Stalo se tak 12. 8. 1981 a změna, která následovala byla obrovská. Příchod PC vytvořil předpoklady pro to, aby se počítač dostal do každého domova a na každý pracovní stůl. Právě v 80. letech došlo k syntéze počítače a telefonní linky. Stále více počítačů se pomocí modemů začalo propojovat do sítí. Díky spojení počítače a telefonu vznikla hackerská legenda Legion of Doom (Legie zkázy, Legie soudného dne, dále jen LoD). Členové skupiny pochopili, že ke slávě je třeba publikační činnosti, a tak se jejich samizdaty začaly jako čísla hackerského časopisu Phreack šířit v podobě tzv. LoD Technical Journal. Zajímavým datem v této éře je 13. 6. 1989, kdy došlo k tzv. floridskému skandálu. Bruce Sterling o tom píše: "13. 6. 1989 zjistili volající do úřadu kurátora Palm Beach County v Delray Beach na Floridě, že kupodivu mluví se sexuální pracovnicí jménem Tina ve státě New York. Každé zavolání do tohoto úřadu kurátora blízko Miami bylo jakýmsi magickým způsobem okamžitě přesměrováno přes hranice státu, bez přirážky pro volajícího, na pornografickou horkou linku vzdálenou stovky mil!" [2] Ne dlouho po tomto skandálu byla LoD rozprášena.

Impulz přinesla operace Grateful Dead. Šlo o ukradený kód společnosti Apple Macintosh, který byl vysoce utajován a sloužil jako ovladač výstupu zobrazení na monitor. Neznámí hackeři kód ukradli a zaslali vybraným osobám z konkurence Apple a významným lidem počítačového businessu. V souvislosti s tím vyšetřovala FBI i textaře skupiny Grateful Dead Johna Barlowa. Ten byl aktivní v hackerské konferenci WELL, které se účastnili hackeři v původním slova smyslu, kteří navíc patřili mezi vlivné lidi. Poté, co byl několika členům konference zabaven počítač za údajný podíl na kolapsu sítě pro dálkové hovory z 15. 1. 1990, začaly diskuse, zda je průnik do počítače, pokud nedojde ke škodě, ilegální. Nedlouho poté John Barlow s Michaelem Kaporem založili Nadaci elektronického pohraničí (EFF), jejímž úkolem bylo lobováním, soudními spory a šířením publicity dosáhnout rozšíření práv zaručených dodatky Ústavy Spojených států i do cyberspace.

Období středověku by nebylo úplné, kdybych se nezmínil o třech hackerech, Kevinu Mitnickovi, Robertu Morrisovi a Kevinu Poulsenovi. Kevin Mitnick se proslavil útokem na počítače společnosti Digital Equipment v roce 1988. Navíc byl prvním pachatelem počítačového trestného činu, který se objevil mezi "elitou" světového zločinu, tedy ve spise FBI Most Wanted. Robert Morris podal téhož roku do světa svůj virus InternetWorm, který upozornil na nový typ počítačové kriminality, průniky do systémů pomocí cílené infekce počítačovým virem. Oba muži byli zatčeni a odsouzeni. Hackerská dráha třetího z nich - Kevina Pouslena (přezdívka Dark Dante) začala již v pubertálním věku. Pak šel cestou nejednoho odhaleného hackera - stal se uznávaným bezpečnostním expertem. Na rozdíl od jiných ho to od pokračování v hackingu

neodradilo a r. 1991 provedl svůj nejslavnější kousek - naboural se do telefonních linek kalifornské rozhlasové stanice, aby mohl vyhrát automobil Porsche v soutěži. Trest byl méně příjemný - 4 roky vězení, 58 000 dolarů pokuty a také tři roky zákazu práce s počítačem, což je oblíbený trest udělovaný americkými soudy hackerům.

3.3. Novověk

Pro novověk je typická masová rozšířenost počítačů, zejména PC s operačním systémem Microsoft Windows. Roste i vývoj softwaru pro PC. Dochází k rozšíření sítě Internet, která se komercializuje. Lidem se dostal Internet až domů. To přitahuje drobné podvodníky a laiky, kteří se velmi rychle zorientují ve světě počítačů, což zásadně změnilo obraz pachatele. V dobách akademické sítě šlo o nadšence, kteří jsou vystřídáni chladnými profesionály, jejichž motivem je dosažení zisku. Aktivity organizovaných skupin se omezují na specifické činnosti, např. podvody s ukradenými kreditními kartami, útoky na systémy bank či jiných finančních institucí, lámání internetových obchodů apod., či na specifické cíle, jako je společnost Microsoft .

3.3.1. Virové hrozby (1999 - 2001)

S masovým rozšířením Internetu nabývá hrozba virů a útoků na webové stránky na nebezpečnosti. Už r. 1988, kdy Robert Morris vypustil svého červa, se mu podařilo infikovat stovky počítačů po celém světě. Nyní dosahuje počet zasažených počítačů milionů. Elektronická pošta se stala novým distribučním kanálem virů, takzvaného MailBombingu. Prvním virem, který se během několika hodin rozšířil prostřednictvím e-mailu po světě, byla Melissa (březen 1999, tvůrce David L. Smith byl zatčen). Melissa nebyla destruktivní, pouze se sama rozesílala prvním 50 uživatelům z adresáře infikovaného počítače. Význam Melissy nebyl v tom, že ničila data na infikovaných počítačích, ale že ukázala, kudy se může ubírat vývoj v oblasti virů. Dalším virem, který obletěl svět byl I Love You (květen 2000) maskující se za milostný vzkaz, jehož autorem byl filipínský občan. Od té doby bylo virů na podobném principu bezpočet, z nichž některé již mazaly soubory či způsobovaly ztrátu dat.

3.3.2. DoS útoky (únor 2000)

Kriminální aktivitou, mající premiéru nedávno, jsou tzv. Denial of Service (Odepření přístupu) útoky. Na rozdíl od viru se pachatel nesnaží počítač infikovat, ale sérií opakovaných požadavků zahltit, příp. vyřadit z provozu. Tyto útoky byly podniknuty proti známým portálům či elektronickým obchodům (např. Yahoo, e-Bay a dalším).

Asi největší obětí útoků na své stránky v poslední době je společnost Microsoft. V posledních letech byly stránky windowsupdate.com několikrát vyřazeny z provozu pomocí distribuovaných DoS útoků. Vzpomeňme medializovanou kauzu, která se dotkla nejednoho nevinného koncového uživatele PC. Virus win32.Blaster se sám šířil Internetem využívajíce bezpečnostní chyby ve Windows a pak zahájil ze všech infikovaných stanic vysílat požadavky na server windowsupdate.com. Tím došlo k naprostému zahlcení serveru a úspěšný útok s cílem vyřadit stránky z provozu byl dokonán.

V relativně krátké době po tomto incidentu došlo ještě k několika podobným útokům na server Windows, které využívaly modifikace viru Blaster.

V souvislosti se stále více vnímanou hrozbou kriminality v elektronickém pohraničí probíhá v mnoha zemích diskuse o monitorování Internetu státními institucemi nejen v jednotlivých případech na základě soudního příkazu (jak je možné dnes), ale plošně (veškerý provoz). K tomu bych chtěl podotknout, že ač je počítačová kriminalita hrozbou, je nutno mít se na pozoru proti snahám státní moci o narušování soukromí, aby se boj proti počítačové kriminalitě nezvrátil v omezování občanských práv, či ve vznik nové "informační" totality.

Od listopadu 1996 operuje na českém a slovenském internetu hackerská skupina CzERT (http://www.strojsnv.sk/linux/misc/czert.htm, 2006). Jedná se o hackery v historickém smyslu, tedy pro které je hacking kromě zábavy i možností, jak si vydobýt uznání, ale při průnicích nic neničí, maximálně změní WWW stránku a přidají na ni svůj emblém. Členové CzERTu nebyli nikdy obviněni z trestného činu, i když byla jejich identita policejním orgánům známá.

Obr. 3.1 – ukázka práce zkušeného hackera

4. Současné metody a techniky

4.1. Příprava půdy

Než je možné zahájit vlastní útok, je potřeba provést několik základních přípravných kroků. Konkrétně hledání stop, skenování a inventarizace. Pomocí těchto tří postupů by mělo být možno dosáhnout dostatečně velké informační základny, na které by na konci měl stát úspěšně provedený útok.

4.1.1. Hledaní stop

Tato kapitola se zabývá prvním z přípravných kroků, a to konkrétně hledáním stop. Vyhledávání stop je jemné umění shromažďování informací o cílovém systému. Je to velmi podobné technice vykrádání bank. Když jdou lupiči vyloupit banku, určitě nevlétnou do banky naslepo a nezačnou hned loudit peníze. Zcela jistě stráví spoustu času shromažďováním informací o videokamerách, počtech bankovních úředníků na jednotlivých směnách, o počtech a rozmístění pracovníků bezpečnostních složek, o nouzových východech, časovém rozvrhu převozu peněz a trasách pancéřových vozů převážejících ony peníze.

Takřka totéž musí udělat úspěšný útočník na informační systém. Musí získat velké množství informací o systému, aby mohl uskutečnit přesný, rychlý, efektivní a pokud možno nenápadný útok. Musí získat co nejvíce informací o všech aspektech bezpečnostní politiky organizace, do jejíhož informačního systému chce proniknout. Výsledkem je unikátní profil organizace, obsahující informace o přítomnosti na Internetu, o možnostech vzdáleného přístupu k výpočetním systémům, o intranetu organizace včetně osobních dat správců systémů. V následující kapitole budu demonstrovat z jakých zdrojů a jakými prostředky lze tyto informace získávat.

Vyhledáváni stop představuje získávání informací o přítomnosti organizace v Internetu. Pomocí kombinace utilit, technik a veřejně přístupných databází může útočník získat konkrétní informace o doménových jménech, přidělených IP adresách a o adresách zařízení připojených přímo do Internetu. Technik získávání informací o cílovém objektu je obrovské množství. Nejčastěji se však používají techniky zaměřené právě na získávání informací o přítomnosti v Internetu, o intranetu o vzdálených přístupech do vnitřních sítí.

Techniky vyhledávání stop v internetu a intranetu se v zásadě shodují, takže by mělo být dostačující, když budou popsány jen v prostředí Internetu.

Je poměrně obtížné vytvořit přesně definovaný postup získávání informací o cizích systémech, protože existuje několik různých cest, kterými lze postupovat. Já se v této části práce budu zabývat základními kroky, které by měly vést k poměrně dobrým výsledkům.

4.1.2. Prohledávání volně přístupných zdrojů

Jako odrazový můstek může posloužit studium webového serveru organizace. V mnoha případech lze na stránkách najít neočekávané množství informací, které

mohou být případnému útočníkovi velmi užitečné. Na serveru nejmenované organizace bylo dokonce možné najít popis konfigurace jejich firewallu [3].

Je také více než vhodné prohlédnout zdrojové texty HTML dokumentů a prostudovat komentáře. Za komentářovými HTML tagy „<“, „!“ a „--“ lze najít mnoho užitečných informací, které původně určitě nebyly určeny pro veřejnost. Mnohdy je rychlejší studovat zdrojový kód offline, takže je výhodné zkopírovat celý webový server organizace na lokální počítač. Prohledávání lze pak automatizovat pomocí vhodného softwaru. Zkopírovat celý webový server na lokální počítač se systémem Windows lze pomocí programu „Teleport Pro“(http://www.tenmax.com/teleport/pro/home.htm), případně pro operační systém Unix pomocí programu „wget“(http://www.gnu.org/software/wget/wget.html). Po prostudování webových stránek lze hledat informace o organizaci ve veřejně přístupných zdrojích. V prostředí českého internetu lze najít mnoho cenných informací například v databázi firem na www.seznam.cz, na severech www.nic.cz, www.underground.cz apod., kde lze narazit na zprávy o bezpečnostních incidentech, které odkrývají slabá místa organizace.

Mnoho zajímavých informací lze též získat hledáním článků souvisejících s @doménaorganizace v USENET news konferencích. Můžete objevit příspěvek, ve kterém se administrátor organizace ptá na postup při konfiguraci třeba nového routeru, s nimž dosud nemá zkušenosti. Tyto informace mají pro útočníka cenu zlata[3].

4.1.3. Mapování sítě

Prvním krokem v mapování sítě je identifikace domén a odpovídajících síťových adres náležících organizaci. Doménová jména representují organizaci na Internetu a jsou tedy internetovým ekvivalentem obchodního jména společnosti.

Jména domén, organizací a adresy sítí jsou uvedeny v takzvaných whois databázích. Těchto databází existuje v současnosti několik. Důvodem byla ztráta monopolu Network Solutions jako jediného registrátora doménových jmen v doménách com, net, edu a org. Nyní tedy existuje několik registrátorů a samozřejmě i několik databází. Tento fakt poněkud komplikuje snahu o vyhledávání informací o doméně, protože je potřeba nejdříve nalézt tu správnou databázi, kde je ona potřebná informace uvedena.

Existuje mnoho různých metod, jak se dotazovat do různých whois databází. Tyto metody závisí na typu platformy na které se pracuje, avšak nezávisle na metodě, která je zvolena, výstupem je vždy stejná informace.

V prostředí Unixu se zdá být zdaleka nejlepší práce buď s webovým prohlížečem nebo pomocí whois klienta, který je dostupný na většině verzí Unixu. S whois klientem se pracuje přes příkazovou řádku. Pro práci s webovým prohlížečem zde pro pořádek uvedu servery obsahující státní, vojenské a mezinárodní whois databáze[3]:

Evropa	http://www.ripe.net
Asie a Tichomoří	http://whois.apnic.net
U.S vláda	http://whois.nic.gov
U.S armáda	http://whois.nic.mil

V evropské databázi nebo na serveru http://www.nic.cz jsou informace o doménách, které náleží do domény cz. Dalším zdrojem informací, zvláště o whois serverech mimo Spojené státy, je server www.allwhois.com.

V Evropě udržuje whois databázi registrátor RIPE NCC na adrese whois://whois.ripe.net. Tato společnost pracuje velice profesionálně, a tak je evropská whois databáze považována za nejkompletnější. Na ftp serveru ftp://ftp.ripe.net je možno stáhnout utilitu whois, která podporuje i nejrůznější rozšíření evropské whois databáze [3].

Pro ty, kteří nemají k dispozici program whois a ani se nechtějí zabývat jeho stahováním z internetu, případně vůbec neplánují přechod z Windows NT na Unix, stačí připomenout, že protokol WHOIS je velice jednoduchý. Server protokolu WHOIS zpravidla očekává dotazy na portu 43/TCP. Stačí navázat spojení s příslušným WHOIS serverem například pomocí programu telnet a zadat řetězec, který je požadován, v databázi vyhledat. Takovým řetězcem může být DNS jméno, IP adresa, číslo autonomního systému apod. Například ve Windows XP se navazuje spojeni se serverem whois.rip.net takto:

```
C:\> telnet    whois.ripe.net    43
```

Nyní je třeba zadat řetězec, který se má vyhledat:

```
82.208.37.11
```

Server vrátí:

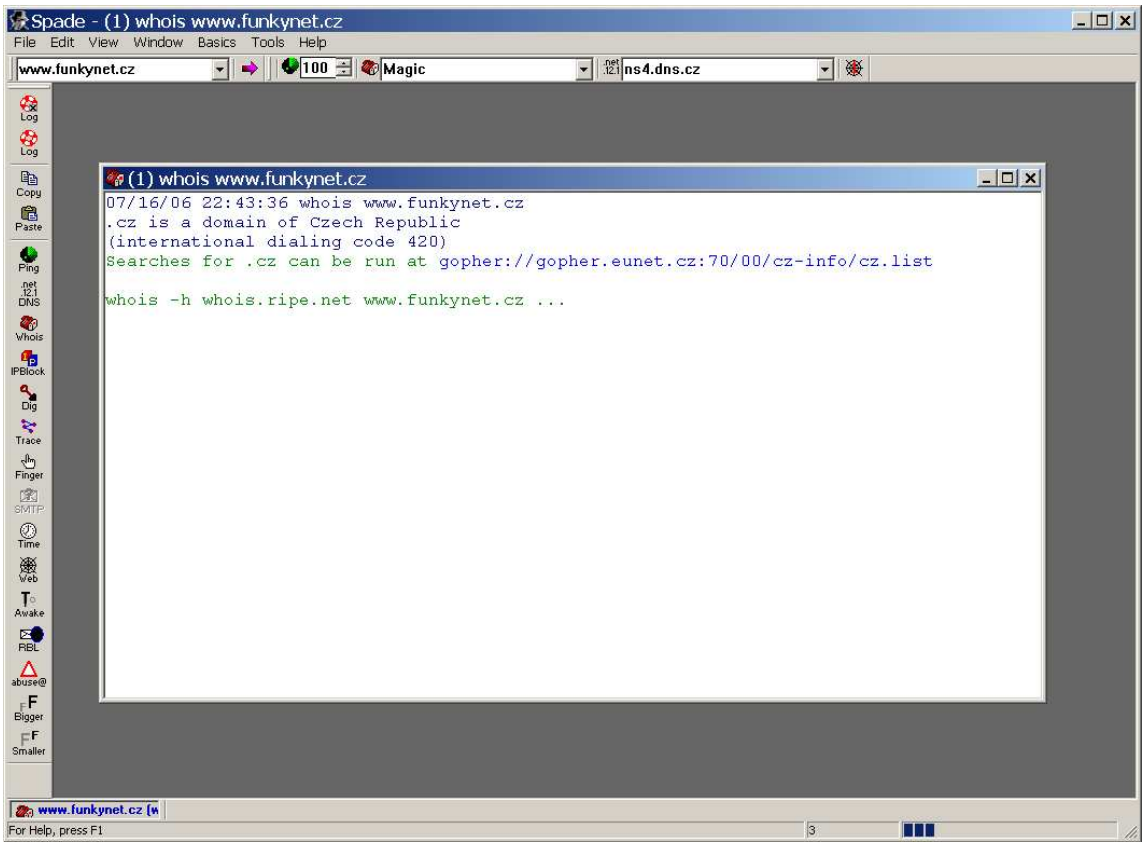
```
% Note: This output has been filtered.
% To receive output for a database update, use the "-B" flag
% Information related to '82.208.0.0 - 82.208.63.255'

……inetnum:          82.208.0.0 - 82.208.63.255
         org:          ORG-CI1-RIPE
     netname:          CZ-CASABLANCA-20030916
       descr:          Casablanca INT
       descr:          PROVIDER LIR
     country:          CZ
     admin-c:          MS29114-RIPE
     admin-c:          LP636-RIPE
      tech-c:          LP636-RIPE
      status:          ALLOCATED PA
      mnt-by:          RIPE-NCC-HM-MNT
   mnt-lower:          MS29114-RIPE-MNT
   mnt-lower:          LP636-RIPE-MNT
  mnt-routes:          MS 29114-RIPE-MNT
  mnt-routes:          LP636-RIPE-MNT
      source:          RIPE # Filtered
organisation:          ORG-CI1-RIPE
    org-name:          Casablanca INT
    org-type:          LIR
     address:          Plzenska 183/181 Prague 5
     address:          15000
     address:          Prague
     address:          Czech Republic
       phone:          +420 2 67132036
      fax-no:          +420 2 67132039
      e-mail:          info@casablanca.cz

     nic-hdl:          MS29114-RIPE
      mnt-by:          MS29114-RIPE-MNT
      source:          RIPE # Filtered
      person:          Lubos Pinkava
     address:          Casablanca INT
```

```
address:          Vinohradska 184, Prague 3 - 130 52
address:          Czech republic
  phone:          +420 2 67132035
 fax-no:          +420 2 67132039
 e-mail:          lubos.pinkava@casablanca.cz
nic-hdl:          LP636-RIPE
 source:          RIPE # Filtered
 mnt-by:          LP636-RIPE-MNT

      % Information related to '82.208.0.0/18AS15685'

  route:          82.208.0.0/18
  descr:          Casablanca INT
 origin:          AS15685
member-of:        AS15685:RS-NETWORKS
  mnt-by:         LP636-RIPE-MNT
  source:         RIPE # Filtered
```

server po vypsání odpovědi ukončí spojení:

```
Připojení k hostiteli bylo ztraceno.
```

Nakonec bych se rád zmínil o programu pod Windows, který v sobě spojuje většinu zde uvedených utilit a nástrojů. Je jím Sam Spade (http://www.samspade.org/ssw/download.html) od Steva Atkinse. Tento šikovný program má v sobě zabudované takové funkce, jako je whois klient, ping, trasování a mnoho dalších. A to vše ve velmi zdařilém a přehledném grafickém uživatelském rozhraní.



Obr. 4.1 – Sam Spade

4.2. Průzkum sítě

Nyní, když jsou známé základní informace o síti, je dalším krokem pokus určit její topologii a potenciální přístupové cesty.

4.2.1. Trasování

K trasování lze použít program „traceroute“ (ftp://ftp.ee.lbl.gov/traceroute.tar.gz), který je součástí mnoha implementací Unixu a nachází se i v systému Windows NT, kde se ovšem jmenuje „tracert“. V systému Windows 2000 a XP pak přibyl ještě program „pathping“.

Traceroute je diagnostický program původně napsaný van Jacobsonem [5]. Tento program zobrazuje cestu, kterou projde IP packet na své cestě sítí. Využívá TTL (zkratka z anglického Time to live - parametr životnosti paketů) pole v IP paketu. Toto pole je vždy při průchodu směrovačem zmenšeno o 1. Pokud dosáhne nulové hodnoty, vrátí směrovač ICMP [5] zprávu TIME_EXCEEDED. Traceroute vyšle první paket s hodnotou TTL rovnou jedné a hned první směrovač na cestě tohoto paketu zmenší TTL na nulu a vrátí ICMP zprávu TIME_EXCEEDED. Traceroute zobrazí IP adresu tohoto směrovače a vyšle další paket s hodnotou TTL nastavenou na 2. Tento paket se dostane až ke druhému směrovači v pořadí. Paket s hodnotou TTL 3 ke třetímu atd., dokud poslední paket nedojde až k cílovému zařízení. Traceroute tedy zobrazí všechna zařízení, kterými pakety prochází na cestě k cíli. pomocí něho lze však odhalit i zařízení, která průchod paketů blokují, jako jsou například firewally nebo třeba filtry [3].Uvedu následující příklad:

```
C:\>tracert crackcomputers.com

Výpis trasy k crackcomputers.com [82.208.37.11]
s nejvýše 30 směrováními:

  1   15 ms    15 ms    18 ms  boraac.rip [172.20.3.17]
  2    9 ms     8 ms     8 ms  172.20.3.241
  3   27 ms     *        7 ms  brouter.el-cha.cz [212.80.95.34]
  4   31 ms    18 ms    35 ms  csw2-3550--elcha.tgnet.cz [212.80.67.161]
  5   36 ms    17 ms    10 ms  bbsw1-ttc.tgnet.cz [212.80.65.28]
  6   22 ms    24 ms    21 ms  nix2.to.cas.ip-anywhere.net [194.50.100.16]
  7     59 ms       10 ms       10 ms   sitelNE40-ge4-0-1.cas.ip-anywhere.net
[82.208.0.220]
  8   22 ms    21 ms    10 ms  tr-horackova-r2.casablanca.cz [82.208.51.2]
  9   18 ms    14 ms    22 ms  tr-dykova-horackova.casablanca.cz [82.208.30.2]
 10   16 ms    26 ms    42 ms  master.crackcomputers.com [82.208.37.11]

Trasování bylo dokončeno.
```

Je vidět cestu paketů přes několik stanic až k cíli. Paket prochází několika směrovači, aniž by byl blokován. Lze tedy soudit, že cílový počítač je spuštěn a naslouchá na síti. Traceroute vlastně pomůže vytvořit podrobnou mapu zkoumané sítě. Mapa bude obsahovat směrovací zařízení a zařízení, která mají funkce řízení přístupu. Jedná se vlastně o diagram přístupových cest.

Ještě také zmíním, že většina unixových implementací programu traceroute posílá implicitně UDP pakety, a pokud se použije přepínač –I, pak posílá ICMP pakety.

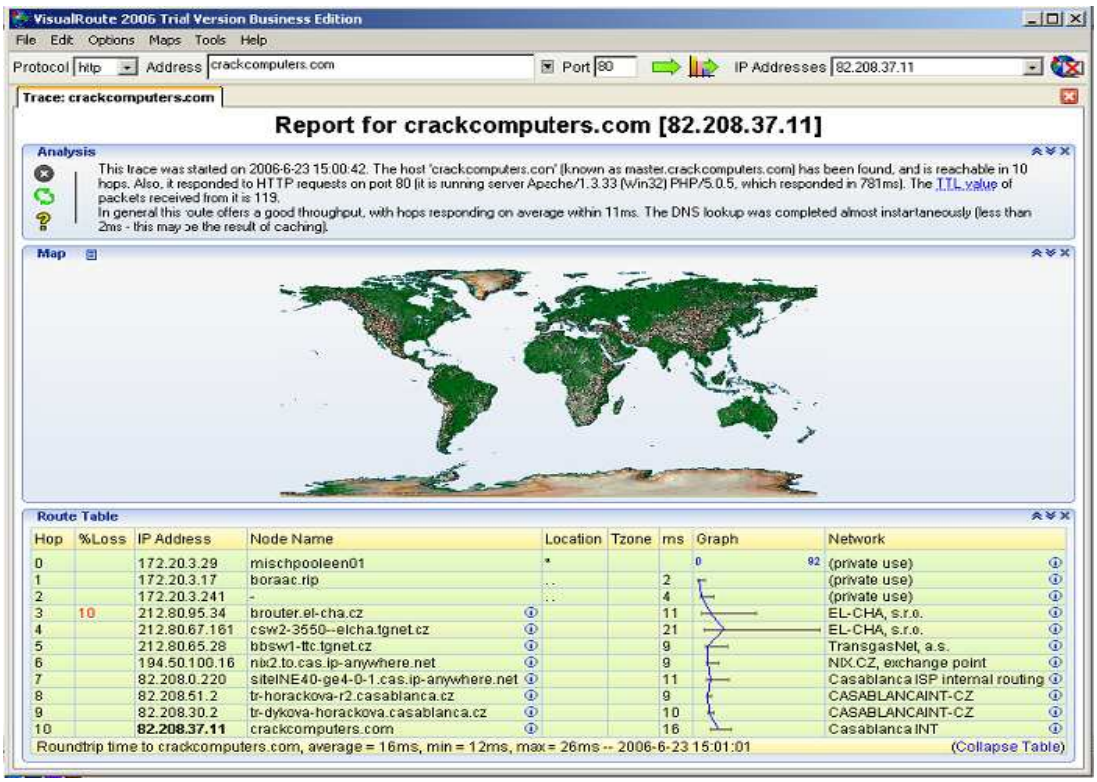
Implementace Windows NT naopak generuje implicitně ICMP pakety. Výsledky průzkumu se potom mohou lišit v závislosti na použitém operačním systému, pokud některá zařízení blokují UDP datagramy a ICMP datagramy propouštějí nebo naopak.

Uvádím přepínače, které jsou k dispozici ve Windows XP implementaci tracert:

```
Použití: tracert [-d] [-h max_počet] [-j sezn_host] [-w doba] cíl_název

Možnosti:
    -d                  Nepřeváděť adresy na názvy hostitelů.
    -h max_počet        Nejvyšší počet přeskoků pro dosažení cíle.
    -j sezn_host        Seznam hostitelů, kterými má paket projít s možností
                        libovolného počtu přeskoků mezi určenými hostiteli
                        (volný režim).
    -w doba             Čeká na odezvu z každé brány po dobu (v ms).
```

Unixové implementace jsou ke smůle všech uživatelů systémů z rodiny Windows NT rozhodně bohatší, co se možností a množství přepínačů týče. Jako velmi užitečná se jeví možnost pevně nastavit port, na který budou odesílány testovací pakety. Zvláště vhodnými se jeví porty třeba pro DNS dotazy (port 53), protože mnoho sítí povoluje dotazování na své jmenné servery, a tak je velmi pravděpodobné, že takovéto pakety budou propuštěny jak přes filtry, tak přes firewally [3].

Kromě programu traceroute, který je orientován čistě na příkazovou řádku, jsou na internetu k dispozici i programy využívající výhod grafického uživatelského rozhraní. Mezi ně patří například VisualRoute (http://www.visualroute.com) nebo NeoTrace(http://www.neotrace.com). VisualRoute navíc používá informace z whois databáze, takže jeho výstup je opravdu působivý.



Obr. 4.2 – výstup programu VisualRoute 2006

4.3. Scanování

Přirovnám-li hledání stop systémů k vyhledávání obydlených míst v pustinách, pak scanování je bušení do zdí za účelem objevení oken a dveří[3].

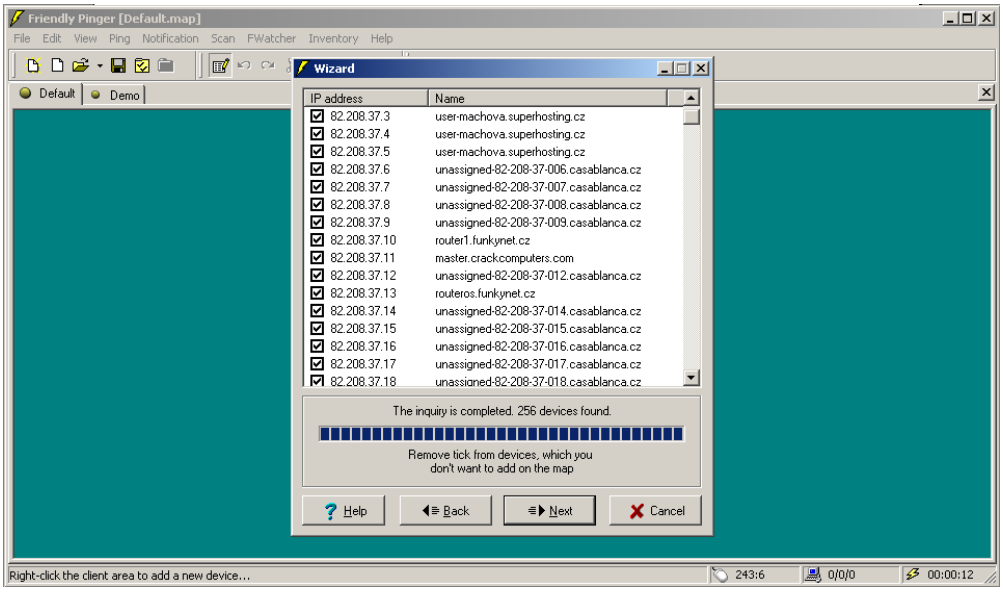
Pomocí hledání stop lze z whois systémů, trasovacích programů a z ostatních v minulé kapitole uvedených zdrojů získat bloky IP adres přidělených zkoumané organizaci, jména administrátorů, telefonní čísla, adresy DNS serverů atd. Nyní se pomocí různých utilit a postupů, jako je ping, scanování portů a automatizované lokalizování serverů, pokusím zjistit, na kterých IP adresách se nacházejí funkční systémy dostupné z internetu.

Jeden ze základních kroků mapování sítě je automatický hromadný ping na interval IP adres, který umožňuje identifikovat v rámci tohoto intervalu fungující (živé) systémy. Program ping zasílá cílovému systému ICMP pakety ECHO (typ 8) (více na http://www.root.cz/clanky/sokety-a-c-icmp-protokol/#ICMPEchoResponse) s tím, že pokud dostane jako odpověď paket ICMP ECHO_REPLY (typ 0), předpokládá, že testovaný systém je funkční. Ping se hodí k použití v malých až středních sítích, ale je značně neefektivní ve velkých podnikových sítích. Skenování velkých sítí muže trvat hodiny až dny.

4.3.1. Hromadný ping

Existuje obrovské množství utilit, jak pro Unix, tak pro Windows, umožňujících hromadný ping. V unixovém světě je nejprověřenější program fping (http://packetstormsecurity.org/Exploit_Code_Archive/fping.tar.gz). Většina těchto tradičních utilit čeká, až dostane odpověď z testované IP adresy. Teprve pak začne testovat další IP adresu v pořadí. Fping odešle na testované adresy více paketů současně, a tak je schopen prověřit velké množství IP adres mnohem rychleji než klasický ping. Dalším programem, který umožňuje hromadný ping, je nmap (http://www.insecure.org/nmap/download.html). O této utilitě budu ještě mluvit později.

Uživatelé Windows mohou použít volně šiřitelný Pinger (http://www.topshareware.com/Friendly-Pinger-transfer-4855.htm), viz obrázek 4.3.



Obr. 4.3 – Friendly Pinger

Je to jeden z nejrychlejších programů tohoto typu. Obdobně jako fping paralelně generuje ICMP pakety, ECHO umožňuje zobrazit jména počítačů a výsledky umí ukládat do souboru. Stejně rychlý jako pinger je komerční Ping Sweep od SolarWinds (http://www.solarwinds.net/Broadband/Engineers/Categories/Ping_Diagnostic.htm).
Ping Sweep umožňuje stejně jako pinger zadat časový interval, který má uplynout mezi odesláním jednotlivých paketů. Jestliže je tento interval nastaven na 0 nebo 1, lze oskenovat síť typu C i se zobrazením jmen počítačů za méně než 7 sekund. Tento postup se však nedoporučuje u pomalých linek (128K ISDN), protože by velmi snadno mohlo dojít k zahlcení linky. A v neposlední řadě je velkou nevýhodou těchto programů s grafickým uživatelským rozhraním fakt, že jejich výstup, ať už je sebevíc přehledný a líbivý, je zcela nepoužitelný ve skriptech a automatizovaných procedurách.

Otázkou zůstává, co dělat, když cílová síť blokuje pakety ICMP. V dnešní době je totiž běžným jevem, že ICMP pakety jsou blokovány hraničními směrovači nebo firewally. Existují však další utility a metody, které tento problém řeší. Tyto metody však nejsou tak přesné a efektivní jako hromadný ping.

První alternativní metoda, kterou lze použít, je skenování portů. Pokud je na testovaném počítači otevřený port, je zřejmé, že počítač je funkční. Tato metoda je časově náročná a ne vždy zcela přesná. Ke skenování můžeme například použít program nmap. Jak jsem již zmínil dříve, lze pomocí nmapu provádět ICMP pingy. Tento program však poskytuje mnohem více možností. Jednou z nich je takzvaný TCP ping scan. Vyžaduje zadání přepínače „-PT“ a čísla portu, který se má testovat. Často se používá port 80 (HTTP). Protokol http je totiž často propouštěn hraničními směrovači sítě a někdy dokonce i skrze firewall do vnitřní sítě. A v neposlední řadě musí tento port být otevřen na každém WWW serveru.

4.3.2. Skenování portů

Skenování portů je proces připojování se k TCP a UDP portům systému s cílem identifikovat běžící služby. Někdy se také používá termín naslouchající nebo také otevřené porty. Identifikace otevřených portů na cílovém počítači pomůže zjistit typ operačního systému počítače a typ provozovaných aplikací. Pokud jsou běžící aplikace špatně nakonfigurovány nebo osahují programové chyby, je možné jich využít k průniku do systému. Protože je skenování portů velmi rozšířenou technikou, popíšu několik nejčastěji požívaných programů a metod.

Je několik cílů, kterých lze pomocí skenování portů dosáhnout. Toto jsou nejdůležitější z nich:

a. identifikace TCP a UDP služeb na cílovém počítači
b. identifikace typu operačního systému na cílovém zařízení
c. identifikace konkrétních aplikací a jejich verzí

4.3.2.a Identifikace služeb TCP a UDP

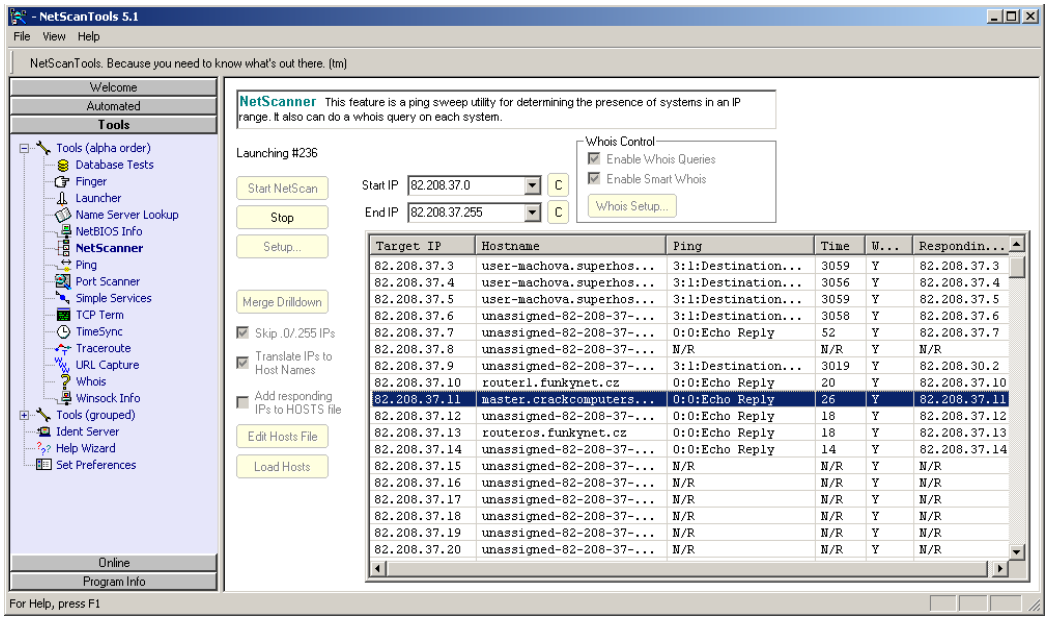
Mezi nejlepší utility pro operační systém Unix patří netcat, naprogramovaný Hobbitem (hobbit@avian.org). Tato utilita umožňuje tolik činností, že bych si jí dovolil přirovnat ke švýcarskému armádnímu noži. Co se skenování týče, umožňuje jak analýzu TCP portů, tak analýzu UDP.

Špičkou ve svém oboru vytvořenou jak pro unixové platformy tak pro systém Windows je již zmíněný program nmap. Kromě základního skenování TCP a UDP portů jednoho systému umožňuje i skenování celé sítě. Navíc je schopen výsledky skenování ukládat do souboru. Další neopomenutelnou výhodou nmapu je schopnost odesílat testovací pakety jako fragmenty. Fragmenty jsou obecně hůře detekovatelné a starší systémy si s nimi neporadí vůbec a bez okolků je propustí dále, tak jak jsou.

Pokud je cílová síť dobře administrovaná, lze skeny snadno identifikovat. Nmap poskytuje možnost takzvaných klamných skenů. Během opravdového skenu proběhne ještě několik dalších (klamných) skenů, které se tváří, jako by probíhaly z jiných počítačů. Je pak velmi těžké mezi nimi odhalit ten pravý.

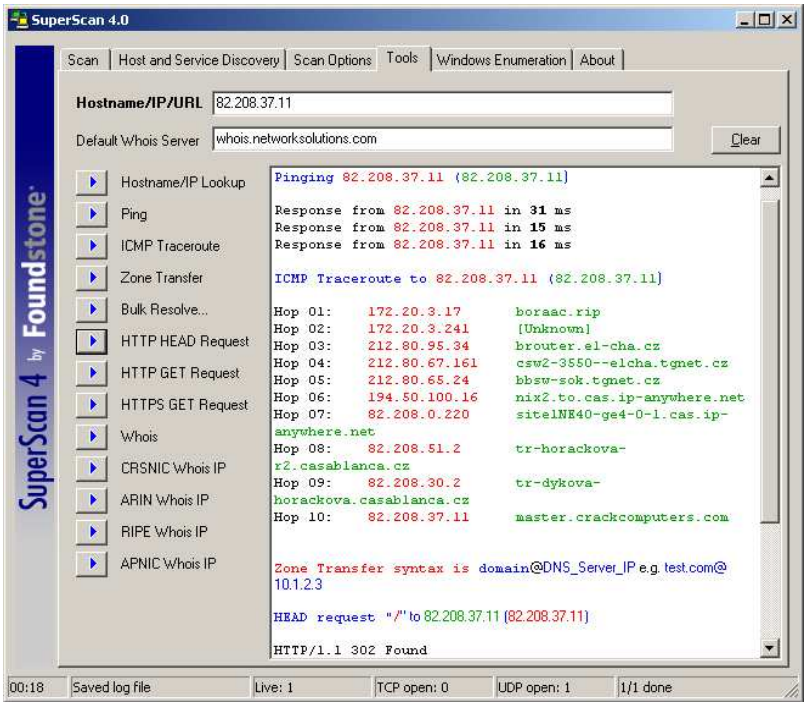
Popsal jsem zde nástroje, které se dají použít ke skenování portů, je ovšem potřeba, abych řekl i několik slov k výstupům z těchto programů. Nezávisle na stroji, který je použit, cílem hledání jsou otevřené porty vypovídající o použitých službách i o operačním systému počítače. Pokud budou například nalezeny otevřené porty 139 a 135, je vysoce pravděpodobné, že na cílovém počítači běží operační systémem Windows z rodiny NT. Naopak na unixovém systému je pravděpodobné, že poběží třeba portmapper na portu 111, nebo NFS (http://www.abclinuxu.cz/slovnik/nfs) na portu 2049.

Výše zmíněné utility jsou doménou spíše operačního systému UNIX. Nyní napíšu několik řádků i o utilitách pro Windows. Jako první budu mluvit o programu NetScanTools(http://www.majorgeeks.com/downloadget.php?id=1297&file=13&evp=de7080e3997b13eca5c4172fab88464d). Tento program obsahuje snad všechny představitelné síťové utility integrované do jednoho uživatelského rozhraní: nslookup, dig, whois, hromadné pingy, skenování jmenného prostoru NetBIOSu a další. Program je navíc schopen multitaskingu. Může najednou skenovat porty v jedné síti a hromadným pingem testovat jinou sít. Skener obsažený v NetScanTools je jeden z nejlepších na platformě Windows [3]. Umožňuje snadnou specifikaci cílového počítače a portů s tím, že seznamy IP adres a portů mohou být importovány z textových souborů. Jsou podporovány jak TCP tak UDP skeny a proces skenování je díky multithreadingu dostatečně rychlý. Nevýhodou je výstup, jako u většiny programů s GUI (Grafic User Interface-grafické uživatelské rozhraní), který se dá jen velice těžko dále zpracovávat pomocí automatizovaných nástrojů a program se díky své grafické podstatě nedá použít ve skriptech. Další nevýhodou je fakt, že výstup jedné funkce nelze poslat na vstup funkce druhé.



Obr. 4.4 – NetScanTools 5.1

Druhou velmi šikovnou utilitou, kterou zde uvedu, je SuperScan (http://www.foundstone.com/index.htm?subnav=resources/navigation.htm&subcontent=/resources/proddesc/superscan.htm) od společnosti Foundstone. SuperScan je rychlý a universální TCP port skener. Stejně jako NetScanTools umožňuje pružnou specifikaci cílových IP adres a portů, které lze načítat z textového souboru. A samozřejmě v sobě integruje množství dalších velmi užitečných funkcí.



Obr. 4.5 – SuperScan 4.0

Na závěr se zmíním ještě o programu ScanLine, který dohromady kombinuje vlastnosti TCP skenu, UDP skenu a inventarizačního prostředku. Praktické použití na konkrétním případu ukážu v kapitole Vlastní pokus o útok.

4.3.2.b Identifikace operačního systému

Pro útočníka je životně důležité určit typ operačního systému, který je provozován na cílovém počítači. S touto informací může podniknout mnohem cílenější útok a může využít nepřeberné množství údajů o chybách v konkrétních operačních systémech a aplikacích. Pokud možno přesné určení operačního systému je tedy jedním z nejdůležitějších úkolů. Metoda užívaná k identifikaci operačního systému je založená na získávání stop TCP/IP implementace, takzvaný stack fingerprinting. Implementace protokolů TCP/IP se operační systém od operačního systému v mnoha detailech liší. Pokud se tedy tyto rozdíly stanou předmětem zkoumání, bude s vysokou mírou pravděpodobnosti možné rozlišit jednotlivé implementace, a tím i jednotlivé operační systémy. Aby byla zaručena maximální spolehlivost, vyžaduje tato metoda na cílovém počítači alespoň jeden otevřený port [3].

Asi nejlepší volbou pro identifikaci operačního systému je již několikrát zmiňovaný program nmap:

```
C:\>nmap 82.208.37.11 -O

Starting Nmap 4.11 ( http://www.insecure.org/nmap ) at 2006-06-27 16:54
St°ednÝ
Evropa (bý×nř Ŕas)
Interesting ports on master.crackcomputers.com (82.208.37.11):
Not shown: 1655 closed ports
PORT     STATE    SERVICE
20/tcp   open     ftp-data
21/tcp   open     ftp
25/tcp   open     smtp
.
.
.
3389/tcp open     ms-term-serv
8080/tcp open     http-proxy

Device type: general purpose

Running: Microsoft Windows 2003/.NET
OS details: Microsoft Windows 2003 Server SP1
```

Zatím jsem popsal pouze metody tzv. aktivní identifikace. Tyto metody jsou označovány jako aktivní, protože při jejich nasazení dochází k aktivnímu odesílání testovacích paketů na porty cílového systému. Toto je činnost, kterou lze poměrně jednoduše detekovat nebo monitorovat na samotném cílovém počítači. Rozhodně tedy nelze mluvit o neviditelné technice. Pokud je žádoucí zůstat v utajení, je třeba použít metodu pasivního získávání stop TCP/IP implementace. V tomto případě nedochází ke generování žádných testovacích paketů, ale pouze se pasivně monitorují toky dat v síti. Programem, který toto umožňuje, je pod Linuxem běžící utilita siphon.

4.4. Inventarizace

Jakmile se podaří pomocí technik uvedených v předešlých kapitolách získat přehled o běžících počítačích a službách, které poskytují, následuje obvykle hlubší průzkum zjištěných služeb. Tento proces pátrání po známých slabinách je nazýván inventarizací.

Zásadní rozdíl mezi předešlými metodami sbírání informací a inventarizací systému spočívá v tom, že inventarizace je mnohem intimnějším způsobem navázání kontaktu s cílovým systémem. Jedná se totiž o aktivní navázání spojení a generování přesné směrovaných dotazů, které mohou být monitorovány a logovány.

Mnohé z informací získaných během inventarizace systému mohou být v rukách útočníka velmi nebezpečné a představa, že se jich útočník opravdu zmocnil, je noční můrou nejednoho administrátora. Útočníci obvykle pátrají po názvech uživatelských účtů, nedbale nakonfigurovaných sdílených prostředcích (například nezabezpečených sdílených adresářích) a starších verzí softwaru, u nichž jsou známé bezpečnostní chyby (kupříkladu webové servery se špatně ošetřenou kontrolou vstupních dat). Jakmile jsou dvířka do systému nalezena, bývá už jen otázkou času, kdy se útočníkovi podaří dostat do systému, když ne úplně, tak alespoň částečně.

Techniky inventarizace jsou většinou závislé na platformě, proto velmi záleží na informacích získaných během skenování portů a detekce operačního systému. Ve skutečnosti jsou funkce skenování portů a následná inventarizace často spojeny do jednoho nástroje, jak jsem ukázal již v předchozí kapitole na programech jako je SuperScan. Ten umožňuje skenovat porty na síti a zároveň shromažďovat bannery z otevřených portů.

4.4.1. Inventarizace bannerů

Zcela základní technikou inventarizace je inventarizace bannerů. Jednoduše jí lze popsat jako navázání spojení se vzdálenou aplikací a sledování výstupu, který může být pro vzdálené útočníky překvapivě bohatý na informace. Přinejmenším mohou rozpoznat typ a stavbu běžící služby, což v mnoha případech stačí k nastartování procesu hledání známých chyb.

Jak jsem již zmínil, mnohé nástroje pro skenování portů mohou kromě své hlavní funkce zajistit i bannery na otevřených portech. Níže bych rád podal stručný přehled o manuálních technikách pro inventarizaci bannerů, které by neměly být opomíjeny.

4.4.2. Inventarizace http na port 80 pomocí netcat

Netcat(http://searchwindowssecurity.techtarget.com/downloadPage/0,295339,sid45_gci1110400,00.html) patří mezi časem prověřené nástroje k inventarizaci. Jeho použití je velmi jednoduché, stačí otevřít spojení na známý port na cílovém počítači, stisknout párkrát „enter“ a analyzovat informaci, kterou program vrátí.

```
C:\jira\hackovani - bakule\netcat>nc -v master.crackcomputers.com 80
master.crackcomputers.com [82.208.37.11] 80 (http) open
```

Tento obecný postup se dá aplikovat na mnoho běžných aplikací, které naslouchají na vyhrazených portech, například pro http je to port 80, pro SMTP port 25 a pro FTP port 21. Aby bylo možné vyprovokovat cílový server k odpovědi, musíme na jeho vstup poslat nějaká data. Podle jednoho tipu z doprovodného textového souboru README k programu netcat lze na vstup přesměrovat soubor s příkazy specifikovanými pro daný server (aplikaci). Uvedené příkazy pak netcat předá testovanému serveru. Příkazy lze zadat i ručně:

```
C:\jira\hackovani - bakule\netcat>nc -v master.crackcomputers.com 80
master.crackcomputers.com [82.208.37.11] 80 (http) open
```

ručně zadaný příkaz z readme souboru:

```
get / http/1.0
```

dvojí stisknutí enter a následný výstup:

```
HTTP/1.1 302 Found
Date: Fri, 30 Jun 2006 21:37:17 GMT
Server: Apache/1.3.33 (Win32) PHP/5.0.5
X-Powered-By: PHP/5.0.5
location: news.php
Connection: close
Content-Type: text/html; charset=windows-1250
```

Program vrátil velmi cennou informaci, a to, že na cílovém počítači běží Apache server. Tato informace může výrazně zaměřit útočníkovo úsilí. Když je nyní znám výrobce a verze softwaru, na kterém je provozována služba, lze se soustředit na techniky specifické pro danou platformu.

4.4.3. Inventarizace NETBIOS relací, port 139

Windows NT a jeho následovníci jsou známy tím, že velmi ochotně vydávají informace zlodějíčkům po síti. Téměř výhradním původcem této pověsti je slabina, kterou popíši nyní.

Prázdné relace a útoky přes anonymní spojení patří mezi bezedné zdroje informací pro hackery. Pokud jste už někdy přistupovali k souborům po síti nebo tiskli na síťové tiskárně připojené k Windows, pravděpodobně jste při tom používali protokol Microsoft Server Message Block (SMB), který je základem sdílení souborů a tiskáren Windows. Přístup k SMB zprostředkovávají API [5], které poskytují skutečné bohatství informací o Windows. Pokud se SMB nezabezpečí, dělá kvalita takto získaných informací ze SMB Achillovu patu systému.

Abych ukázal míru škod způsobených SMB bez zabezpečení, předvedu několik dobře známých technik hackování tohoto protokolu. Prvním krokem při inventarizaci SMB je spojení se službou s pomocí takzvané „prázdné relace“, což ilustruje tento příkaz“

```
C:\>net use \\192.168.202.33\IPC$ "" /u:""
```

Předchozí ukázka způsobí spojení se skrytým „adresářem“ IPC$, používaným pro komunikaci mezi procesy, na adrese 192.168.202.33, přičemž je k tomu využit zabudovaný anonymní uživatelský účet (/u:““) a prázdné heslo (““). V případě, že se příkaz provede, má útočník otevřenou cestu k různým dalším technikám uvedeným dále, díky kterým vysaje z cílové stanice co nejvíce informací, mezi jinými síťové informace, sdílené jednotky, uživatelské účty a skupiny, klíče registru a tak dále. Ať se to nazývá anonymním spojením nebo prázdnou relací, je to zřejmě nejničivější bezpečnostní „díra“, po které útočníci pátrají.

Jedním z nejlepších programů pro získávání informací přes prázdné relace je program DumpSec (http://www.somarsoft.com/cgi-bin/download.pl?DumpAcl). Program je zadarmo a jen málo takových si zaslouží místo mezi nástroji administrátora NT tak jako on. Umožňuje sledovat vše, od přístupových práv k souborům až po služby dostupné na počítači v síti. Pomocí prázdné relace umožňuje získat i základní informace o uživatelích. Program může být spuštěn z příkazové řádky, takže ho lze zabudovat do skriptů. DumpSec je kromě informací o sdílených prostředcích schopen dát pomocí prázdných relací i informace o uživatelských účtech a registrech vzdáleného počítače.

4.4.4. Inventarizace TFTP, port 69

I když vzhledem k závažnosti získaných informací se nejedná o inventarizační techniku v pravém slova smyslu, dá se za praotce všech inventarizačních triků ve světě Unixu považovat únik souboru hesel /tec/passwd. Jedna z technik získávání tohoto souboru souvisí právě s TFTP [5], který běží typicky na UDP portu 69. Špatně zabezpečený soubor /tec/passwd prostřednictvím TFTP lze získat snadno:

```
tftp 192.168.202.34
tftp> connect 192.168.202.34
tftp> get /etc/passwd /tmp/passwd.steal
tftp> quit
```

ve windows se příkaz zpracovává dávkově:

```
TFTP [-i] hostitel [GET | PUT] zdroj [cíl]
```

Když pominu, že útočník má nyní k dispozici i zašifrovaná hesla uživatelů (tedy pokud systém nevyužívá jejich stínovou databázi), která se jistě pokusí rozlousknout, může si pohodlně přečíst jména uživatelů a všechny relevantní informace.

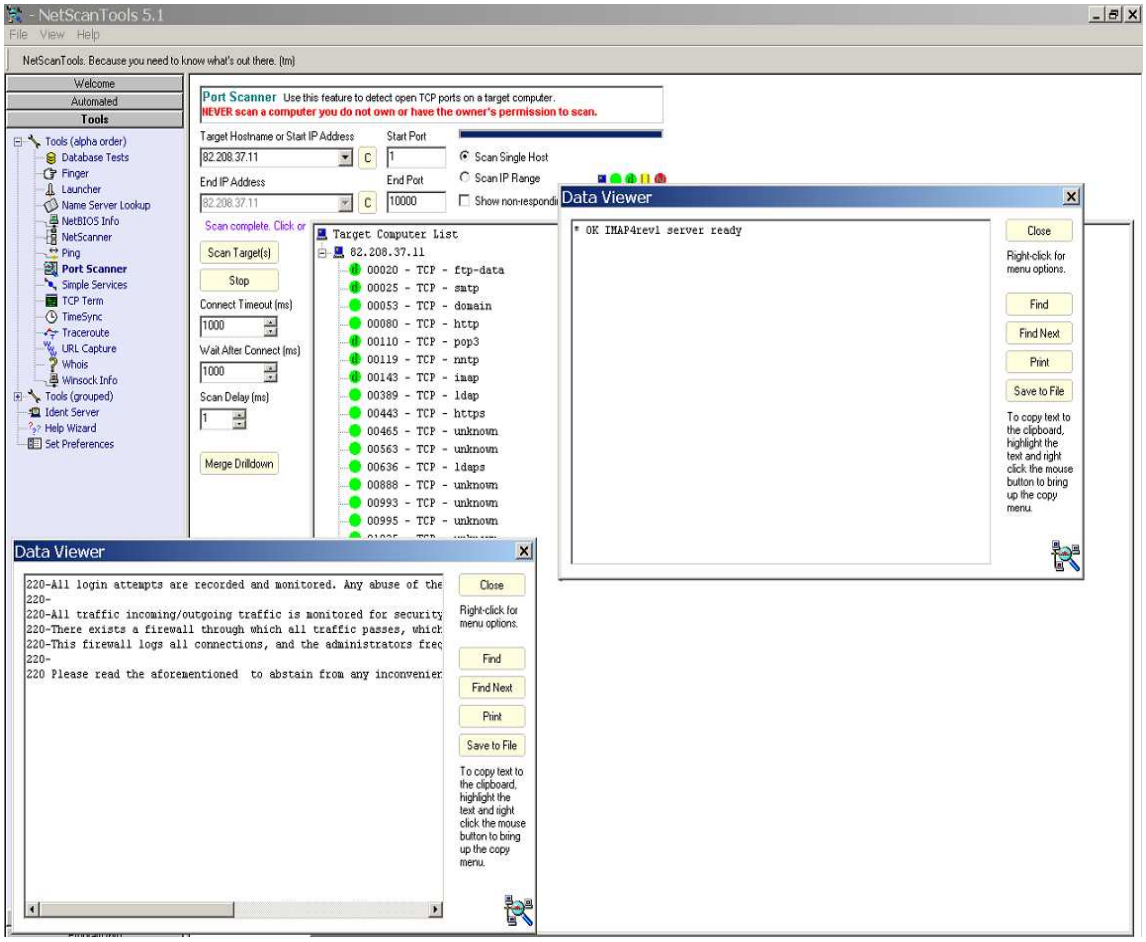
4.4.5. Inventarizace SMTP, TCP port 25

Jednou z nejklasičtějších technik inventarizace je využití SMTP (http://www.abclinuxu.cz/slovnik/smtp) příkazů VRFY a EXPN. VRFY složí k verifikaci poštovních adres (ale i lokálních uživatelů) a EXPN vypíše skutečné cílové adresy aliasů. Ačkoli mnoho organizací nedělá v dnešní době s e-mailovými adresami žádné tajnosti, může použití výše uvedených příkazů pomoci při odhalování lokálních uživatelských kont na poštovním serveru nebo při zneužívání cizích e-mailových adres. V následujícím příkladu to předvedu s pomocí programu telnet:

```
telnet   192.168.202.34   25
```

```
trying 192.168.202.34…
connected to 192.168.202.34
220 mail.organizace.com ESMTP Sendmail 8.8.7/8.8.7; Sun, 9 July 2006 11:08:49
-0700
vrfy root
250 root  <root@organizace.com>
Expn adm
250 adm  <adm@organizace.com>
Quit
221 mail.organizace.com closing connection
```

Na závěr uvedu jednu konkrétní utilitu na automatickou inventarizaci, a to již několikrát zmiňovaný balíček NetScanTools. Během procesu skenování portů cílového počítače zároveň sbírá bannery. „Hlášky“, které cílový systém vrací při kladné odezvě na otevřených portech. Tyto bannery mohou mnoho napovědět o konfiguraci cílového počítače. Skenování portů nám napoví, jaké služby jsou na serveru dostupné, ale při inventarizaci můžeme odhalit, jaké porty jsou filtrovány firewallem a jaký software tyto služby zajišťuje.



Obr. 4.6 – inventarizace pomocí programu NetScanTools

.

4.5. Útoky typu DoS

Útok typu denial of service(DoS), "odmítnutí služby", je útokem, který je namířený proti webovému serveru (resp. celé síti) připojenému k Internetu. Jeho cílem je ochromení nebo jiné narušení provozu tohoto serveru na základě zvýšení počtu přicházejících požadavků na obsloužení. Příkladem takového útoku, velmi laicky řečeno je útok, při kterém hacker spustí program generující nějaká nesmyslná data, která jsou posílána na server. Server je pak zahlcen těmito přicházejícími daty a již není schopen reagovat na požadavky ostatních řádných uživatelů. V horším případě může dojít až k úplnému havarování a zhroucení webového serveru. V praxi je často k vidění stav, kdy primárním cílem není vyřazení serveru z provozu, ale že DoS útok je použit hackerem pouze jako doplňková akce sloužící například pro zametení stop, restartování vzdáleného počítače apod. Podle způsobu provedení lze DoS útoky obecně rozdělit do čtyř skupin:

1. Přeplnění vyrovnávací paměti

2. Útoky na tabulku procesů

3. Útoky, využívající nedokonalostí a nedostatků ve specifikaci TCP/IP(5), - útoky na síťovou vrstvu

4. Útoky na síťovou konektivitu - útoky hrubou silou

4.5.1. Přeplnění vyrovnávací paměti

Chyby typu přetečení vyrovnávací paměti jsou symbolem hackování. Za příznivých okolností mají za následek havarování systému nebo spuštění libovolných příkazů na cílovém počítači, typicky s velmi vysokými oprávněními.

Jednoduše řečeno spočívá přeplnění vyrovnávací paměti v naplnění proměnné větší hodnotou, než je očekávána, což vede při vhodně zvolených datech k neoprávněnému vykonání příkazu na cílovém počítači nebo k zablokování všech systémových prostředků [3]. Problém je téměř vždy způsoben špatně napsaným programem, kdy aplikace přijme a uloží data do vyrovnávací paměti, aniž by zkontrolovala jejich formát.

U těchto útoků se rozlišuje, zda se jedná o útok na počítač, kde jsou stránky (to jsou útoky využívající chyby v implementaci TCP/IP), či o útok na webový server.

4.5.1.1. Útoky, které využívají chyb v implementaci TCP/IP

Jedná se o jedny z nejstarších metod útoku DoS, které využívají nedokonalostí vzniklých při programování.

4.5.1.1.a Ping of Death

Příkaz ping se běžně používá pro zjištění, zda vzdálený server pracuje. Hacker použije příkaz ping pro vytvoření IP paketu, který je větší než maximum povolené ve specifikaci IP protokolu (65 536 bajtů). Tento abnormálně velký paket je pak poslán na cílový server. To může způsobit havárii, zamrznutí nebo restart celého systému

4.5.1.1.b Teardrops Attacks

Novější typ útoku, který využívá slabosti při opětovném sestavování fragmentů IP paketu. Během své cesty po Internetu může být IP datagram rozdělen do menších kusů, takzvaně fragmentován. Každý kus vypadá jako původní IP paket až na to, že obsahuje položku offset, která říká, že packet je pouze fragmentem a kterou pozici v celku má zaujmout. Teardrop program vytváří sérii IP fragmentů s překrývajícími se položkami ofsetů. Když jsou tyto fragmenty opětovně sestavovány na cílovém počítači, výsledný paket má větší velikost, než je velikost bufferu a dojde k přetečení, kdy některé systémy havarují, zamrznou nebo se restartují.

4.5.1.2. Útoky na webové servery

Většina útoků typu přeplnění vyrovnávací paměti má za cíl umožnit útočníkovi vykonat neoprávněný kód, který pak vede k ovládnutí cílového systému.

4.5.1.2.a IIShack

Asi nejproslulejší chybou byla chyba v Microsoft IIS, která byla objevena skupinou eEye security. Je způsobena nedostatečnou kontrolou jmen souborů URL a umožňuje útočníkovi přenést na server kód, který lze poté spustit s právy administrátora. Program, který využívá této chyby, se jmenuje iishack (http://www.megasecurity.org/trojans/iishack/IisHack.htm) a odesílá URL a jméno trojského koně, kterého chce útočník na cílovém systému spustit.

4.5.1.2.b Přeplnění vstupního pole pro heslo webového serveru

Tento útok je jasným důkazem toho, že lze dosáhnout zhroucení webového serveru pomocí prohlížeče. Tvůrci WWW serverů často preferují funkčnost před bezpečností a nic nedemonstruje tuto skutečnost lépe než chyba poprvé objevená na serveru ColdFusion 4.0 od společnosti Allaire. Problém spočívá ve způsobu, jakým jsou ošetřena vstupní pole pro zadání administrátorského hesla serveru. Pomocí například Netscape Navigatoru (http://browser.netscape.com) je možné editovat stránky a pak ukládat na lokální počítač. Pokud však v poli ACTION zůstane zachováno URL přihlašovací stránky na internetu, tak stránka bude stále funkční. Pomocí Navigatoru lze editovat atributy polí uživatelského jména a hesla. Pokud dojde ke změně jejich atributů tak, aby šlo vložit heslo dlouhé například 1000000 znaků a stránku bude uložena na lokální počítač, bude možno jí pak vyvolat standardním způsobem libovolným prohlížečem. Je třeba vygenerovat 1000000 znaků dlouhý řetězec, který pak přijde do upraveného pole pro řetězec znaků hesla a bude odeslán na server. Příliš dlouhý řetězec znaků odeslaný na server způsobí přetečení vyrovnávací paměti a tím i zhroucení sytému.

4.5.2. Útoky na tabulku procesů

Tyto útoky jsou prováděny vytvořením velkého množství paralelních spojení na napadený server — za účelem vyčerpání volných položek tabulky procesů, jehož důsledkem je nemožnost obsluhy legitimních požadavků ostatních klientů. Tento typ útoků je velmi účinný vůči webovému serveru Apache (http://httpd.apache.org/), který má navíc pro obsluhu požadavků vyhrazeno obvykle jen 256 „slotů“ (procesů).

4.5.3. Útoky na síťovou vrstvu

Tento typ útoků je zaměřen na vyčerpání prostředků umožňujících komunikaci po síti. cílem je síťová vrstva napadeného serveru — zpravidla tabulka spojení, kterou se útočník snaží zaplnit výlučně svou aktivitou.

4.5.3.1. SYN flood

Tento útok využívá principu, jakým se pomocí protokolu TCP/IP navazuje spojení. Proces navazování spojení se označuje jako tzv. "třífázový handshaking" (three way handshake). Aplikace, která inicializuje session (tj. chce posílat data, komunikovat) posílá příjemci synchronizační paket SYN. Příjemce posílá zpátky potvrzovací paket TCP SYN-ACK, na který iniciátor odpovídá potvrzovacím paketem ACK. Po takovémto navázání komunikace jsou aplikace připraveny posílat a přijímat data.
Při SYN útoku zaplavuje hacker cílový systém sériemi TCP SYN paketů. Každý paket způsobuje to, že cílový systém pošle odpověď SYN ACK. Zatímco cílový systém čeká na ACK, které následují za SYN-ACK, zařadí všechny nevyřízené SYN-ACK odpovědi do fronty (tzv. backlog queue). Tato fronta má omezenou délku, obvykle jen 128 míst. Jakmile je fronta plná, systém začne ignorovat všechny příchozí SYN požadavky. Pakety SYN-ACK jsou vyřazeny z fronty pouze, když přijde zpět ACK nebo když interní časovač (který je nastaven na relativně dlouhé intervaly) ukončí celý proces navazování komunikace. Celý útok je "vylepšen“ tím, že v záhlaví příchozích SYN paketů je uvedena špatná nebo neplatná IP adresa. Všechny odpovědi SYN-ACK jsou posílány na tuto adresu, což zaručuje , že odpověď na SYN-ACK nikdy zpátky nepřijde. Tak se tedy vytvoří fronta objednávek, která je vždycky plná, což téměř znemožní se legitimním TCP SYN požadavkům dostat do systému. [6]

Příkladem utilit, snadno dostupných na Internetu, mohou být třeba program Spastic [7] nebo Hgod [7].

Spastic je klasická TCP SYNFlood attack utilita pro příkazovou řádku, vhodná k použití do skriptů. Využívá chyb v implementaci three-way handshake ve windows XP a 2000 a odesílá pakety s náhodnými zdrojovými IP adresami.

```
C:\jira\hackovani - bakule\sid>spastic 82.208.37.11
Spastic.exe by cys of NewNet
Packeting 82.208.37.11...
```

Druhou utilitou, o které jsem se zmínil, je program Hgod. Tato malá utilita v sobě integruje kromě SYN flood útoku ještě další útoky z rodiny DoS.

```
C:\jira\...\sid>hgod
============================== HUC DoS Tool V0.5 ==============================
================ By Lion, Welcome to http://www.cnhonker.com ==================
[Usage:]
  hgod <Target> <StartPort[-EndPort]|Port1,Port2,Port3...> [Option]
  <Target>       Flooding Host IP|Hostname.
  <StartPort>    Flooding Host Port. Port Num must <100.

[Option:]
  -a:AttackTime  The Time(minute) of Attack. Set 0 for Always. Default is 0.
  -b:Packsize    The Size of Packet, for UDP/ICMP/IGMP Mode. Default is 1000.
  -d:Delay         Delay of Send Packet, for UDP/ICMP/IGMP Mode. Default is
10ms.
  -l:Speed       Your Network Link Speed(?M). Default is 100M
  -m:Mode        Attack Mode, Use SYN/DrDoS/UDP/ICMP/IGMP. Default is SYN.
  -n:Num         Only for SYN/DrDoS Mode, Change SourceIP, Set Num to 1-65535.
  -p:SourcePort  Set SourcePort, Default is Random. DrDoS Mode mast be set.
  -s:SourceIP    Set SourceIP, Default is Random. DrDoS Mode mast be set.
  -t:Thread      The Threads Num for Flooding, Max is 100, Default is 5.
```

4.5.4. Útoky na síťovou konektivitu

Velmi běžné útoky, které se snaží enormním množstvím požadavků a nebo jiným mechanismem „ucpat“ linku spojující napadený server se zbytkem Internetu.

4.5.4.1. Smurf útok

Jedná se o útok hrubou silou zaměřený na vlastnost v IP specifikaci, která je známá jako všeobecné adresování ("direct broadcast addressing"). Hacker zaplaví router speciálními pakety tzv. "pingy". Poněvadž cílové IP adresy každého paketu jsou broadcast adresou sítě, router bude šířit "pingy" všem počítačům sítě. Pokud je na síti mnoho stanic, dojde k velkému nárůstu zatížení sítě. Celý útok se dá znásobit ještě tím, že hacker může zfalšovat zdrojovou IP adresu pingů, a odpovědi na tyto pingy mohou zahltit síť odkud pochází předstíraná zdrojová adresa. Síť se pak stane pouze prostředníkem útoku.

Typickým představitelem programu pro útok typu Smurf je utilita Smurf [7].

```
C:\...\smurff>smurf
usage: smurf.exe <victim> <broadcast file> [options]

Options:
  -p:     Comma separated list of dest ports          (default 7) (0 is
random)
  -s:     Source port                                      (0 is random
(default))
  -P:    Protocols to use. Either icmp, udp or both (default icmp)
  -S:    Data size in bytes                         (default 64)
  -f:    Filename containg extra data               (not needed)
  -n:     Num of packets to send                        (0 is continuous
(default))
  -d:    Delay (in ms)                              (default 0)
```

4.5.4.2. UDP Flood

Při tomto typu útoku hacker pomocí falšování zapne službu, která pro testovací účely generuje sérii znaků pro každý paket, který přijme, spolu s UDP echo službou jiného systému, která také odpovídá na jakýkoliv znak který přijme. Výsledkem je nonstop proud nesmyslných dat mezi dvěma systémy. Programy pro realizaci UDP flood útoku jsou běžně ke stažení na Internetu. Příkladem může být utilita pjam [7]:

```
C:\...\sid>pjam
pjam.exe, a udp flooder by cys / Team HaVoX
Usage: pjam target port[0=rand] ptime(minutes)[0=flood] threads[(ie=1)].
```

4.5.5. DDoS útok

Zkratka DDoS znamená "Distributed Denial of Service" a označuje variantu DoS útoku vedeného ne z jednoho počítače, ale souběžně z velkého množství stanic. Slovo velké množství znamená v této souvislosti desítky, stovky a dokonce i tisíce stanic. Co se týká praktické realizace takového útoku, neznamená to, že v danou chvíli sedí u těchto stanic hackeři, nýbrž že na těchto stanicích je nainstalován hackerem nějaký program (tzv. zombie), který se na pokyn hackera aktivuje a zasype oběť přívalem dat.

Existuje několik různých druhů DoS útoků, které mohou být hackery použity pro ochromení provozu serveru nebo celé sítě. Některé z těchto útoků jsou již poměrně dlouho známé a lze se proti nim účinně bránit např. aplikováním záplaty do příslušného operačního systému, proti jiným je obrana složitější. A v tom spatřuji také velký problém. Ačkoliv obrana proti těmto útokům je poměrně složitá (nastavování routerů, firewallů apod.) samotný útok může být proveden prakticky kýmkoliv a neklade na hackera v podstatě žádné zvláštní požadavky co se znalostí samotného hackingu týká. Člověk provádějící tento útok může být začátečníkem ve světě počítačového záškodnictví, stačí jen když si stáhne příslušný program pro generování provozu třeba na stránkách http://www.antiserver.it/Denial-Of-Service/index.html, nebo http://packetstormsecurity.org/DoS/index.html.

4.6. Hádání hesel

Ačkoliv nepatří hádání hesel k nejpřitažlivějším a nejelegantnějším způsobům překonávání webového ověřování, rozhodně patří mezi ty nejúčinnější. Za předpokladu, že ve výběru přihlašovacího protokolu nebo v jeho implementaci do webového systému není žádná trhlina, stává se nejzranitelnějším aspektem většiny ověřovacích systémů volba uživatelského hesla. Dosavadní praxe ukázala, že hádání hesel je nejefektivnější, když se využívá letitých omylů uživatelů při výběru svých hesel. Tyto omyly lze shrnout do několika bodů zhruba takto:

1. uživatelé mají tendenci vybírat si ta nejjednodušší možná hesla, tedy pokud možno prázdné heslo.
2. pokud systém nepovoluje prázdná hesla, vybírají uživatelé něco, co se dobře pamatuje. Příkladem takovýchto slabých hesel může být vlastní uživatelské jméno nebo křestní jméno nebo nějaký jiný, snadno odvoditelný výraz, například příjmení, jméno firmy, slova jako host, test, heslo atd.

V následující tabulce seznámím s příklady těch nejběžnějších kombinačních párů uživatelské_jméno/heslo. Tyto dvojice jsou nazývány „kombinace s vysokou pravděpodobností“.

Hádaná uživatelská jména	Hádaná hesla
root, administrator, admin	[prázdné], root, administrator, admin, password, heslo, [jméno_firmy]
operator, webmaster, backup	[prázdné], operator, webmaster, backup, heslo
guest, demo, test, trial, pokus	[prázdné], guest, demo, test, trial, pokus
member, private	[prázdné], member, private, [jméno]
[jméno_firmy]	[prázdné], [jméno_firmy], password, heslo
[známé_uživatelské_jméno]	[prázdné], [známé_uživatelské_jméno], [jméno_firmy], heslo

Tab. 1 – kombinace s vysokou pravděpodobností

Jak je vidět, je to velmi omezený seznam. Ale zkušené odhadování hesel využívající uvedené tipy má překvapivě vysoký stupeň úspěšnosti. Přitom jen málo správců věnuje svůj drahocenný čas kontrole síly uživateli zvolených hesel.

Útoky hádáním hesel mohou být vedeny ručně nebo pomocí automatizovaných procedur. Ruční hádání hesel je zdlouhavé a nedovoluje testování tak velkého množství kombinací jako hádání pomocí automatických prostředků. Přesto někdy lidská intuice předčí i ony automatické nástroje, zejména jsou-li jako reakce na neúspěšný pokus o přihlášení použity zcela obecné chybové stránky.

4.6.1. Ověřovací metody

Ještě před tím, než ukážu konkrétní postupy, musím říct několik základních informací o ověřovacích mechanismech ve specifikaci HTTP 1.1 [8]. Nejdříve popíšu dvě základní techniky pro webové ověřování: Basic a Digest. Potom povím ještě něco o formulářovém ověřování, které je pro administrátorské přihlašování ke správě stránek nejběžnější.

4.6.1.1. Ověřování HTTP: Basic

Ověřování typu Basic, jak název napovídá, je nejzákladnější způsob ověřování pro webové aplikace. Poprvé bylo definováno ve specifikaci HTTP [8] samotné a i když není příliš elegantní, plní svůj účel. Ověřování Basic začíná tak, že klient pošle webovému serveru požadavek na chráněný prostředek, a to bez jakýchkoliv přihlašovacích údajů. Server odpoví zprávou „přístup odepřen“ s hlavičkou *WWW-Autenthicate* požadující přihlašovací údaje. Tento dialog probíhá v samostatném okně operačního systému vytvořeném prohlížečem a ne přes HTML formulář. Jakmile uživatel zadá heslo, prohlížeč znovu vyšle požadavek, tentokrát s přihlašovacími údaji. Celý proces vypadá zhruba takto:

1. Klient vyšle běžný požadavek:

```
GET /dokumenty/tajne.html HTTP/1.1
```

2. Server vrátí odpověď:

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Basic realm="pocitac1"
```

3. Klient zobrazí uživatelovi dialogové okno pro zadání jména a hesla. Dialogové okno obsahuje i název oblasti (realm) z hlavičky v příkladu řetězec pocitac1, což je pouze řetězec přiřazený serverem. Většina implementací typicky nastavuje oblast na jméno počítače nebo IP adresu webového serveru. Po zadání jména a hesla opakuje klient původní dotaz, do kterého přidá hlavičku Authorization.

```
GET /soukrome/text.html HTTP/1.1
Authorization: Basic QWxhZGRpbjG
```

4. Pokud je jméno a heslo správné vrátí server odpověď

```
HTTP/1.1 200 OK
```

Podle specifikace ověřování Basic jsou přihlašovací informace posílány v hlavičce *Authorization* v odpovědi, ale jsou zakódovány algoritmem Base64 [5], takže vypadají jako zašifrované. Ve skutečnosti lze kódování Base64 dekódovat použitím libovolného dostupného dekodéru(např. http://atrey.karlin.mff.cuni.cz/~mkrs5246/?stranka=base64).

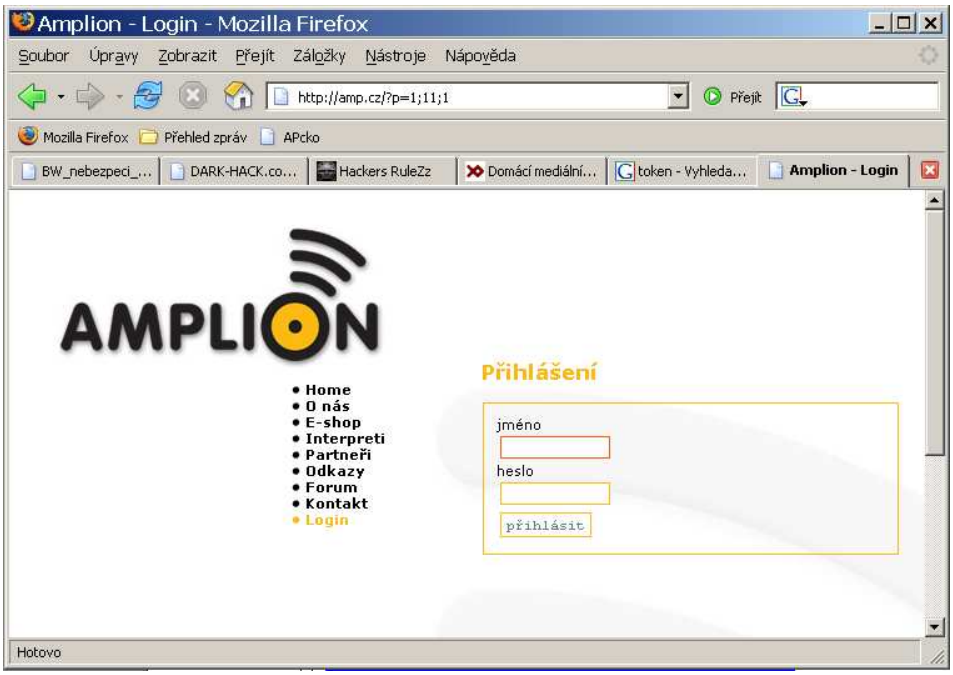
Jak je vidět, ověřování Basic je široce otevřené k útokům odposloucháváním, což je nejzávažnějším omezením tohoto protokolu.

4.6.1.2. Ověřování http: Digest

Ověřování Digest bylo navrženo pro poskytnutí vyšší úrovně bezpečnosti než té, kterou nabízí ověřování Basic. Ověřování Digest je založeno na podobném modelu jako Basic. Uživatel provede požadavek bez přihlašovacích údajů a webový server odpoví zprávou s hlavičkou *WWW-Authenticate* oznamující, že je nutné přihlášení pro přístup k chráněnému prostředku. Ale místo posílání uživatelského jména a hesla v kódování Base64 jako v případě ověřování Basic server vyzve klienta zasláním náhodné hodnoty zvané *nonce*. Prohlížeč pak použije jednosměrnou kryptografickou funkci pro vytvoření hodnoty *message digest* z uživatelského jména, hesla, dané hodnoty nonce, metody http a požadovaného URI. Funkce vytvářející digest, také známá jako hašovací algoritmus, je kryptografická funkce, kterou lze snadno vypočítat v jednom směru, ale je výpočetně nemožné jí obrátit, tedy najít reverzní funkci. Smyslem použití *nonce* v ověřování Digest je vytvořit větší množství klíčů, a tak někomu, kdo by chtěl provést databázový útok proti běžným heslům, ztížit situaci.

4.6.1.3. Formulářové ověřování

V protikladu k dosud diskutovaným mechanismům nespoléhá formulářové ověřování na funkce poskytované základními webovými protokoly jako HTTP a SSL(5). Jde o vysoce přizpůsobitelný přihlašovací mechanismus používající formulář obvykle sestavený v HTML z tagů <FORM> a <INPUT> zobrazující uživateli vstupní pole pro zadání jména a hesla. Poté, co jsou tyto informace načteny a vyhodnoceny nějakou serverovou logikou, pak v případě, že jsou platné, je klientskému prohlížeči odeslána nějaká forma oprávnění – token [9] pro použití při následných požadavcích. Díky své přizpůsobitelnosti a pružnosti je formulářové ověřování nejpopulárnější přihlašovací technikou používanou na Internetu.



Obr. 4.7 – ukázka typického Internetového přihlašovacího formuláře

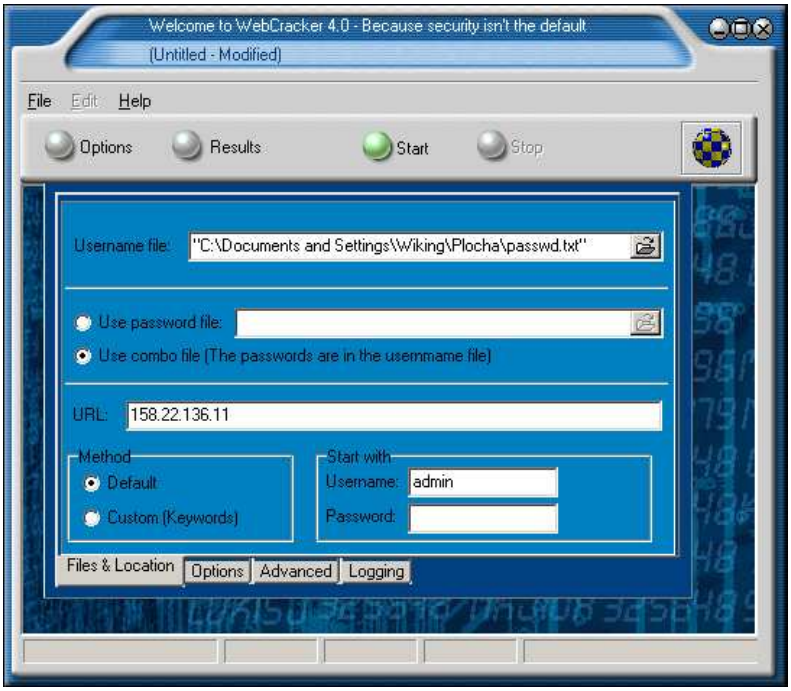
HTML podoba formuláře:

```
<form action="http://amp.cz/?p=1;11;1" method="post">
  jméno<br />
  <input    type="text"    name="uname"    size="12"    value="přihl.    jméno"
maxlength="25" /><br />
  heslo<br />
  <input type="password" name="pswrd" size="12" maxlength="32" value="" /><br
/>
  <input type="hidden" name="BlockID" value="1" />
  <input type="hidden" name="SendForm" value="Login_Login" />
  <input type="submit" value="přihlásit" />
</form>
```

Teď, když byly představeny základní typy přihlašovacích mechanismů, ukáži dva nástroje na útoky proti ověřování Basic a formulářovému ověřování.

4.6.2. WebCracker

WebCracker (http://www.antiserver.it/Password-Crackers/) je jednoduchý nástroj pro automatické hádání hesel při ověřování typu Basic, který využívá textové seznamy uživatelských jmen a hesel nebo kombinace obojího jako slovníky pro hádání. Vychází z odpovědi „HTTP 302 Object moved“, která indikuje, že pokus byl úspěšný, a najde všechny úspěšné kombinace jména a hesla ze zadaného souboru. Jinými slovy neskončí hádání po prvním úspěšném pokusu. Obrázek 4.8 ukazuje nastavování základních parametrů v grafickém uživatelském rozhraní.

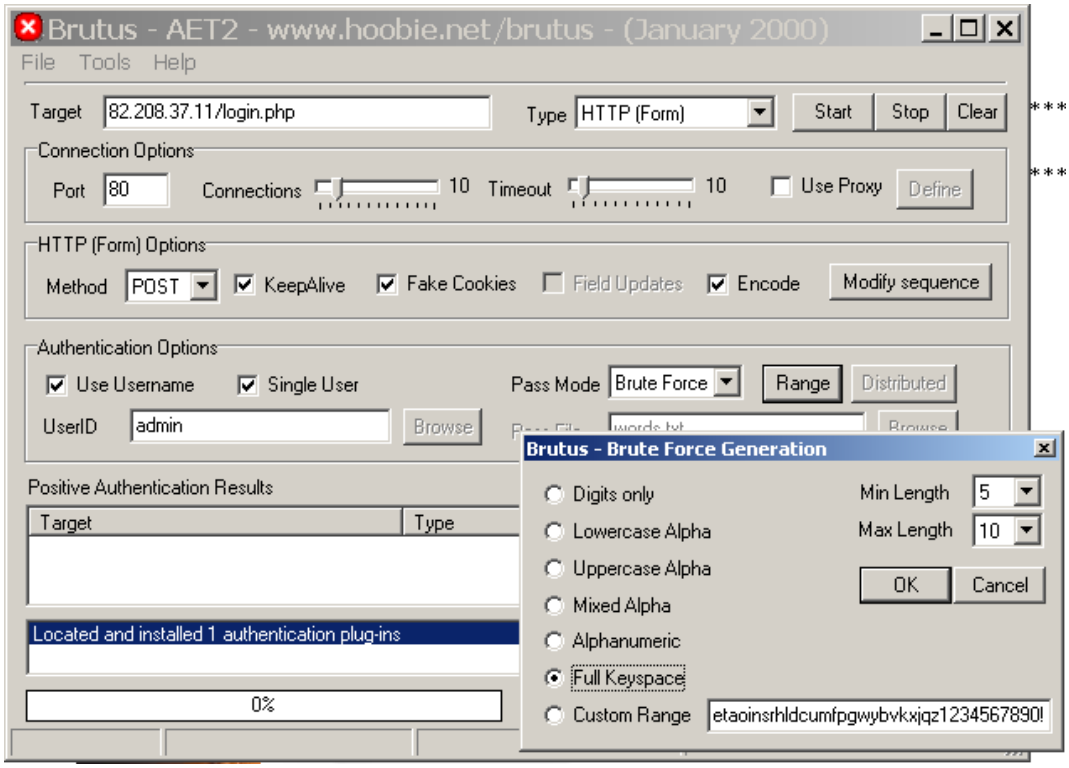


Obr. 4.8 – GUI programu webCracker

4.6.3. Brutus

Brutus (http://www.hoobie.net/brutus/brutus-download.html) je obecný nástroj pro hádání hesel, který obsahuje vestavěné funkce k útokům proti ověřování HTTP Basic a formulářovému ověřování, ale také podporuje další protokoly jako SMTP a POP3 [5]. Brutus může provádět jak slovníkové útoky, založené na připravených seznamech slov, tak útoky hrubou silou, kdy jsou hesla náhodně generována z dané množiny znaků. Například malá písmena nebo rozsah celé klávesnice, jak je vidět na obrázku 4.9.

Mimořádně vydařené jsou jeho schopnosti proti formulářovému ověřování, především funkce „Modify Seqence / Learn Form settings“. Ta umožňuje jednoduše zadat URL přihlašovacího formuláře a Brutus v něm automaticky vyhledá pole pro uživatelské jméno, heslo a všechna další pole včetně skrytých. Více ukáži v praktické části této práce. Další velmi užitečnou vlastností tohoto programu je možnost zadat, jaká odpověď je očekávána od přihlašovacího formuláře v případě úspěšného přihlášení. To je důležité kvůli velmi rozličnému charakteru formulářového ověřování, neboť je běžné, že servery používají často zcela unikátní stránky pro odezvu na úspěšné či neúspěšné přihlášení. Toto je jedna z hlavních překážek úspěšného hádání hesel u formulářového ověřování. S nástrojem Brutus lze přizpůsobit hádání hesel pro jakoukoliv odpověď, kterou cílový server používá. Jeho drobnou nevýhodou je, že občas oznámí pozitivní výsledek (tvrdí, že uhodnul heslo účtu), i když se mu to nepovedlo.



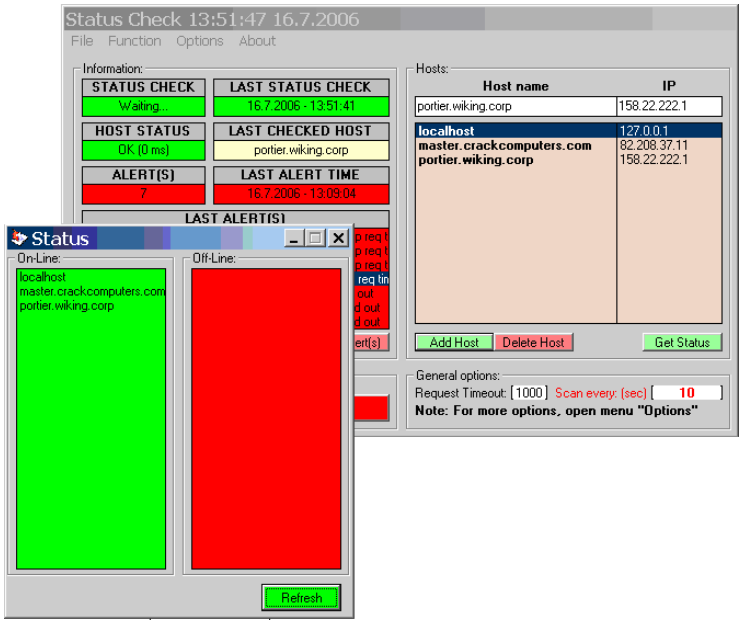
Obr. 4.9 – nastavování útoku hrubou silou v programu Brutus

5. Vlastní pokus o útok

K realizaci pokusu o útok jsem si vybral server www.funkynet.cz. Rozhodl jsem se tak proto, že vím, komu stránky, resp. server na kterém jsou provozovány, patří a tato znalost mi dovolovala vyloučit právní kolisi pokusu. Druhým důvodem bylo, že jsem předem věděl, že stránky a server jsou provozovány v domácích podmínkách a ne někde hostované.

Tyto stránky obsahují mnoho rad, typů, triků, informací, diskusní fóra, vše co souvisí s provozem komerční bezdrátové sítě s názvem Funkynet. Cílem je v praxi vyzkoušet dříve popsané techniky a úspěšně zrealizovat útok na tento server. Budu postupovat přesně tak, jak jsem popisoval jednotlivé po sobě jdoucí akce. Od zjišťování obecných informací přes trasování a skenování až po samotný pokus o útok.

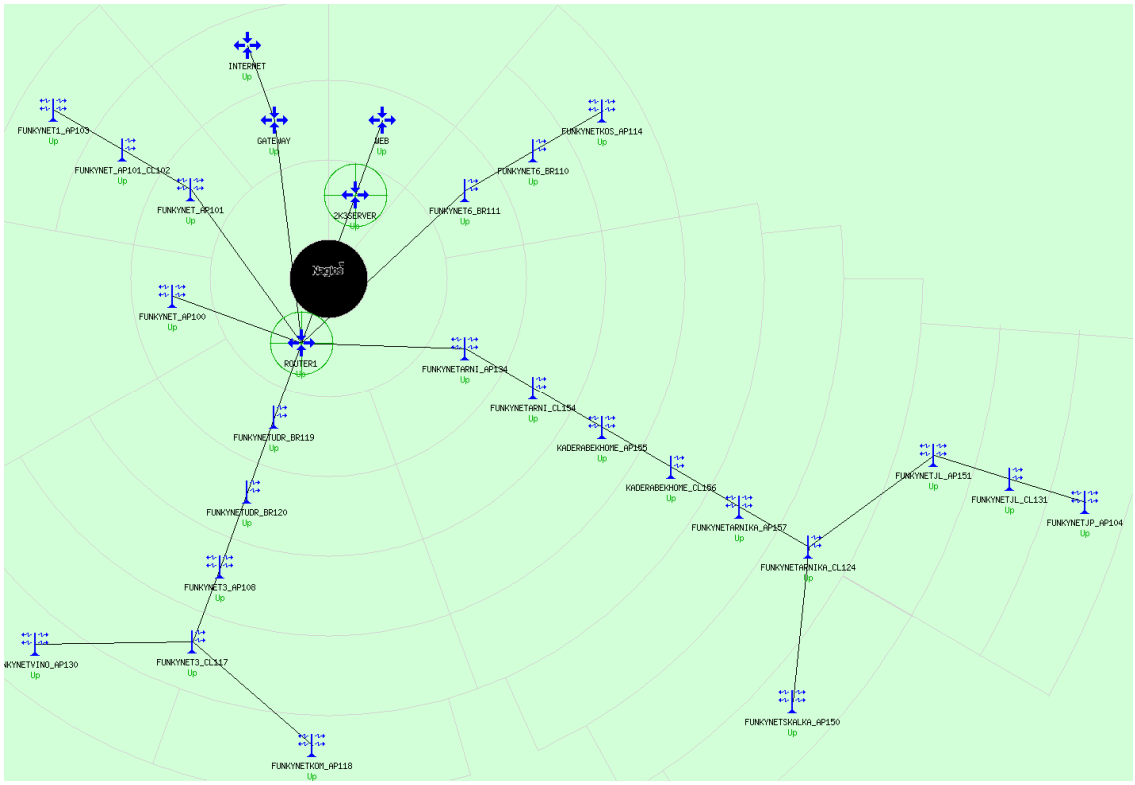
Co se týká technického vybavení, k realizaci jsem použil jeden stolní počítač IBM kompatibilní s procesorem AMD AthlonXP 3200+ , operační pamětí 1024MB DDR dual-channel a s operačním systémem Windows XP. Tento počítač byl připojen přes lokální sít a router do sítě UPC. Jako kontrolní počítač jsem používal Notebook společnosti Acer na architektuře Intel s operačním systémem též Windows XP. Konektivita byla u notebooku zajištěna pomocí ADSL připojení společnosti Český telecom. Dále jsem využil možnosti stáhnout si na stránkách www.funkynet.cz program z vlastní tvorby technického týmu kolem sítě Funkynet s názvem StatusCheck. Tento program sleduje vybrané stanice na Internetu a v pravidelných intervalech kontroluje jejich stav. Utilita se jeví jako velmi praktická pro kontrolu účinnosti některých útoků na dálku, kdy existuje okamžitá zpětná vazba z cílového systému a je jasné, zda má snažení nějaký efekt.



Obr. 5.1 – rozhraní programu StatusCheck

5.1. příprava půdy

Jako první jsem se rozhodl „oťukat“ si server a prohledat jej, zda na stránkách nenaleznu nějaké užitečné informace. K mému překvapení jsem nalezl detailní mapu páteřní sítě Funkynet. Seznam Access pointů, a dokonce jsem zde nalezl i operační systémy nainstalované na jednotlivých zařízení, což neshledávám jako příliš bezpečné.



Obr. 5.2 – mapa páteřní sítě

Z mapy lze vyčíst, že k serveru se stránkami vede cesta přes dvě brány, router s operačním systémem Linux a síťový server kde jsou nainstalovány Windows 2003 server. Ale k tomu více, až budu trasovat. Stránky jsou provozovány na počítači v mapě označeném jako WEB.

Další zajímavou a neméně důležitou informaci jsem objevil v článku „O Funkynetu“. V části o bezpečnosti a zaručování rovných podmínek všem klientům Funkynetu je na síti provozován monitorovací systém Nagios(http://www.nagios.org/). Když jsem studoval fóra, objevil jsem zmínku o zalogovaném pokusu o útok. Pokus byl logován programem ZoneAlarm. O pár příspěvků dále je uvedeno, že pokus byl veden z vnitřní IP adresy a že byl veden na port 137. Z podkapitoly inventarizace je známo, že port UDP port 137 je vyhrazen pro službu NetBIOS name service, která slouží jako distribuovaný systém mapování jmen pro sítě založené na MS Windows.

Dále jsem na stránkách objevil poštovní službu, kde z přihlašovací stránky je zřejmé, že server poskytuje tuto službu pomocí KerioMailServer6.

Jako další krok jsem se rozhodl prohledat zdrojový kód stránek. Kód jsem vzhledem k dostupné rychlosti připojení prohledával online, čili nebylo potřeba použít

žádný mnou dříve zmiňovaný nástroj ke stahování stránek na lokální počítač. Hledal jsem nedbale ponechané komentáře a osobní poznámky tvůrce stránek. Bohužel stránky jsou napsány profesionálem a žádné nadbytečné použitelné informace zde k nalezení nebyly. Jediné cenné, co jsem nalezl, byly automaticky vložené skripty programem Kerio, které pouze potvrdily fakt, že na serveru jako ochrana běží Firewall od této firmy.

Při prohledávání veřejně přístupných zdrojů, do prohlížeče www.seznam.cz jsem zadal klíčové slovo funkynet, jsem objevil na stránkách http://www.qcm.cz/nagios.php velmi zajímavou informaci, že jeden z týmu administrátorů kolem cílové sítě se podílel na české lokalizaci výše zmíněného monitorovacího systému Nagios. Dokonce o pár řádků níže jsem se dočetl, že k řádnému chodu toho monitorovacího systému je nezbytná přítomnost WEB serveru Apache.

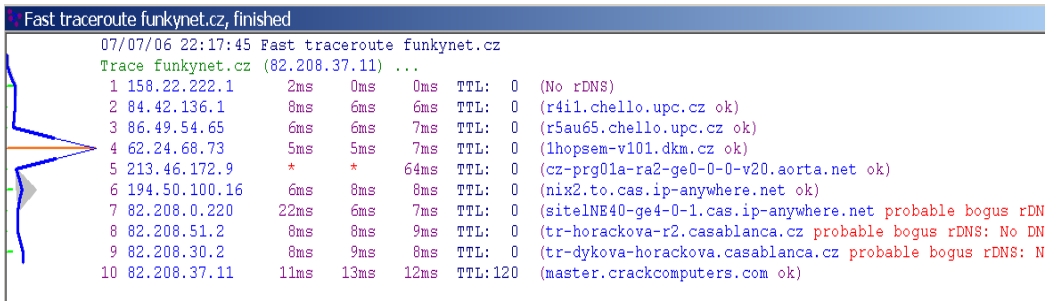
5.2. mapování a průzkum sítě

Prvním krokem v mapování sítě je identifikace domén a odpovídajících síťových adres náležících organizaci. Doménová jména representují organizaci na Internetu, a jsou tedy internetovým ekvivalentem obchodního jména společnosti.

V kapitole současné metody a techniky jsem popsal několik metod, jak se dotazovat do whois databází. Já jsem si vybral tu nejpohodlnější cestu, která se mi na platformě Windows nabízela, a to přímá návštěva serveru www.nic.cz. Po zadání klíčového výrazu funkynet.cz mi byly vráceny tyto informace: držitelem domény je společnost CRACK Computers s.r.o, smlouvu podepsal Marek Slivanský. Bohužel o doméně crackcomputers server www.nic.cz neví, takže po krátké práci s webovým prohlížečem, jsem zjistil, že doména crackcomputers náleží do .com a že stránky jsou totožné. Více informací o crackcomputers.com jsem získal pomocí programu SamSpade, zmíněného v kapitole o metodách a technikách. Na všech pozicích kontaktů je uveden Marek Slivanský. Pomocí integrované funkce v programu SamSpade, si také zjistíme IP adresy náležící našemu cíly.

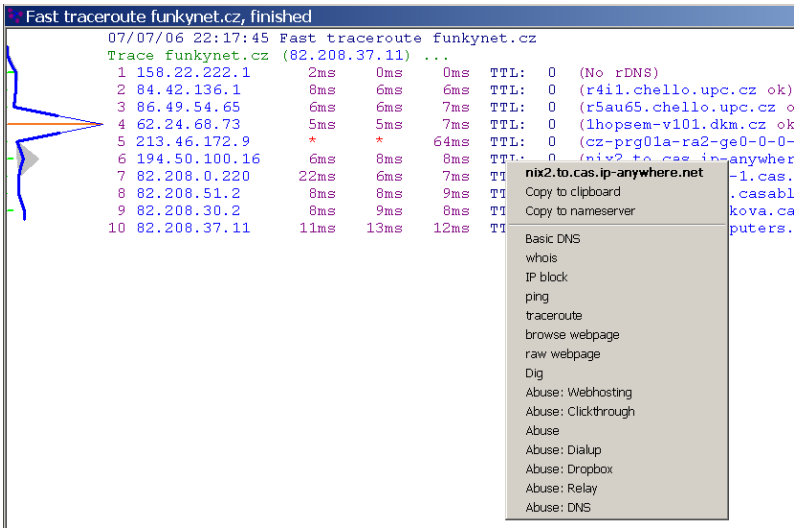
```
07/07/06 22:06:41 dns funkynet.cz
Canonical name: funkynet.cz
Addresses:
  82.208.37.11

07/07/06 22:07:46 dns crackcomputers.com
Canonical name: crackcomputers.com
Addresses:
  82.208.37.11
```

Dalším krokem, když je známa adresa cíle, je trasování. Tento proces pomůže určit do jisté míry topologii sítě a pomůže také najít potenciální přístupové cesty. K trasování použiji můj oblíbený program SamSpade. Výhodou tohoto programu je fakt, že jednotlivé výstupy lze rychle zpracovávat na vstupu dalších funkci. Stačí kliknout pravým tlačítkem na crackcomputers.com ve výstupu whois databáze a vybrat položku trace.



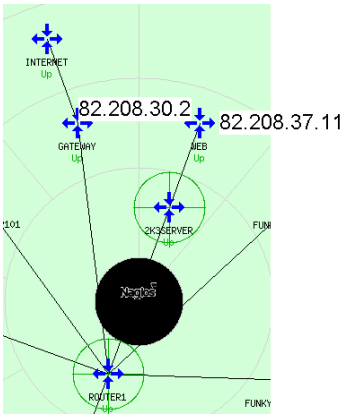
Obr. 5.3 – výstup programu SamSpade

Podobným způsobem lze pomocí pravého tlačítka zkoumat jednotlivé směšovače na cestě ke crackcomputers.com.



Obr. 5.4 – volby pro položky výstupu trasování

Je vidět cestu paketů přes několik směšovačů až k cíli, aniž by byly blokovány, lze tedy říci, že cílový počítač je zapnutý a naslouchá na síti. První zařízení na cestě paketů je router v mém pokoji. Zařízení 2 až 5 náleží mému providerovi, společnosti UPC. Zařízení 6 je brána mezi internetem a Wifi sítí společnosti Casablanca INT s.r.o. Trasovaní nám navíc pomohlo rozšířit mapu páteřní sítě o IP adresy klíčových zařízení.



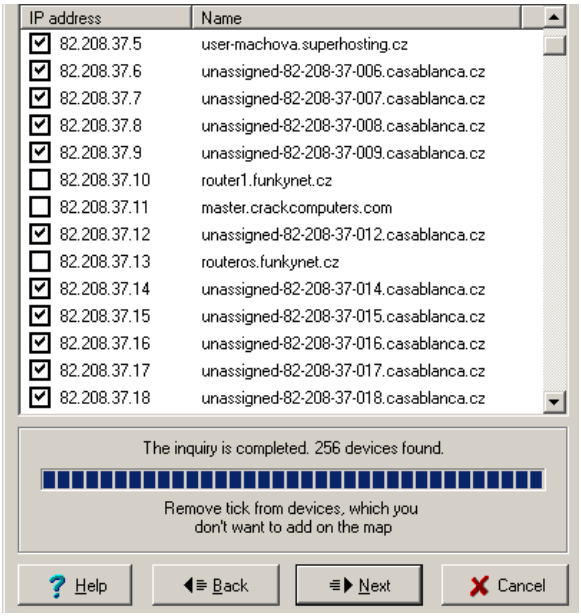
Obr- 5.5 – detail mapy sítě

Z výše uvedených střípků informací se pomalu vykreslil obraz cíle. Jedná se malou firmu, která na trhu působí zhruba 6 let „zabývající se návrhem, realizací a správou počítačových sítí, stavbou zakázkových PC sestav od domácích po profesionální (servery, video). Poradenství HW a SW samozřejmostí.“(http://www.b2m.cz/firma/B2M-AX28a2f1/crack-computers-s-r-o). Projekt Funkynet jako bezdrátové sítě je nápad z poloviny roku 2004, kdy byl původně plánován jen jako privátní síť. Z malé sítě se brzy stala síť komerční se širokým spektrem nabízených služeb.

Získal jsem podrobnou mapu páteřní sítě s adresami i se seznamem operačních systémů nainstalovaných na některých zařízeních a podařilo se nám lokalizovat počítač, na kterém jsou provozovány stránky www.funkynet.cz.

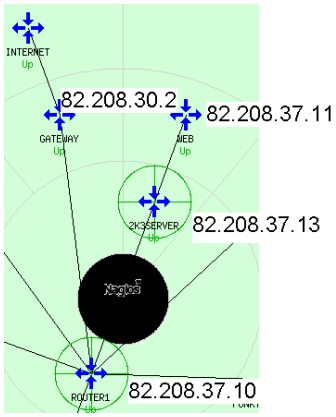
Nyní se pomocí různých utilit a postupů, jako je ping a scanování portů pokusím zjistit, na kterých IP adresách se nacházejí ostatní funkční systémy dostupné z internetu.

Jeden ze základních kroků mapování sítě je automatický hromadný ping na interval IP adres, umožňující identifikovat v rámci tohoto intervalu živé systémy. K tomu jsem použil program FriendlyPinger.



Obr. 5.6 – výstup programu FriendlyPinger

Díky hromadnému pingu se podařilo identifikovat zbylá dvě zařízení na mapě sítě. A tím je kompletní představa o struktuře sítě, ve které se nachází cílový počítač s běžícími stránkami.



Obr. 5.7 – doplněná mapa cesty k webovému serveru

Nyní, když jsou známy všechny potřebné informace o cílovém serveru, na kterém jsou provozovány stránky, je čas přistoupit ke skenování portů cílového počítače. Ke skenování jsem použil program nmap. Tento program pomůže kromě identifikace otevřených portů také určit typ operačního systému, který je provozován na cílovém počítači.

```
Interesting ports on master.crackcomputers.com (82.208.37.11):
Not shown: 1653 closed ports
PORT     STATE    SERVICE
20/tcp   open     ftp-data
25/tcp   open     smtp
53/tcp   open     domain
80/tcp   open     http
110/tcp  open     pop3
119/tcp  open     nntp
135/tcp  filtered msrpc
136/tcp  filtered profile
137/tcp  filtered netbios-ns
138/tcp  filtered netbios-dgm
139/tcp  filtered netbios-ssn
143/tcp  open     imap
179/tcp  filtered bgp
389/tcp  open     ldap
443/tcp  open     https
445/tcp  filtered microsoft-ds
465/tcp  open     smtps
563/tcp  open     snews
636/tcp  open     ldapssl
888/tcp  open     accessbuilder
993/tcp  open     imaps
995/tcp  open     pop3s
1025/tcp open     NFS-or-IIS
1337/tcp open     waste
3306/tcp open     mysql
3389/tcp open     ms-term-serv
8080/tcp open     http-proxy
Device type: general purpose
Running: Microsoft Windows 2003/.NET
OS details: Microsoft Windows 2003 Server SP1

Nmap finished: 1 IP address (1 host up) scanned in 5.719 seconds
```

Z výstupu je vidět, že na cílovém počítači běží operační systém MS Windows 2003 Server se service packem 1. Také je vidět, že server naslouchá na velkém množství otevřených portů, ale část z nich je filtrována nějakým filtrem nebo firewallem.

Pro dotvoření představy o tom, jaké služby doopravdy na serveru běží a jaké z nich jsou v praxi napadnutelné, provedl jsem inventarizaci pomocí programu NetScanTools. NetScanTools posbíraly bannery z TCP portů 20(ftp), 25(smtp), 110(pop3), 119(nntp), 143(imap), 1337 a 3306. Na portech 20 a 1337 se hlásí jako automatická odezva varování firewallu, že veškerá komunikace přes ony konkrétní porty je monitorována a logována.

Aby byl obraz opravdu kompletní, použil jsem ještě program ScanLine. Jeho výstup je o to cennější, že pod operačním systémem Windows nabízí jednoduchou cestu ke skenování UDP portů.

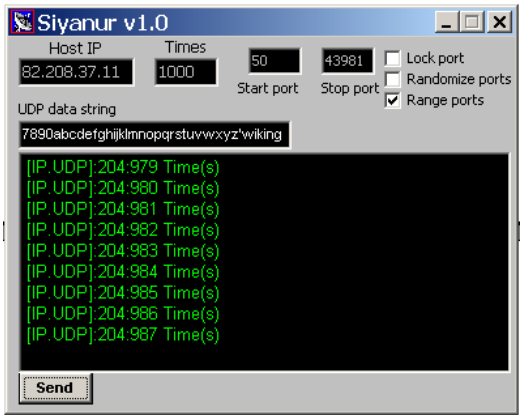
```
C:\jira\hackovani - bakule\sid>sl -bh 82.208.37.11
ScanLine (TM) 1.01
Copyright (c) Foundstone, Inc. 2002
http://www.foundstone.com

Scan of 1 IPs started at Sun Jul 16 12:44:38 2006
------------------------------------------------------------------------------
82.208.37.11
Responded in 47 ms.
8 hops away
Responds with ICMP unreachable: Yes
TCP ports: 25 53 80 110 119 143 389 443 465 563 636 1025 1028 1723 3306 3389 8080
UDP ports: UDP ports: 67 111 123 135 137 138 161 191 192 256 260 407 445 500 514 520
1009 1030 1033 1034 1035 1037 1041 1058 1060 1091 1352 1434 1645 1646 1812 1813 1900
1978 2002 2140 2161 2631 2967 3179 3327 3456 4045 4156 4296 4469 4802 5631 5632 11487
31337 32768 32769 32770 32772 32773 32774 32778 32779 32780 32781 32782 32783 32784
32785 32786 32787 32788 32789 32790 43981

TCP 25:
[220 master.funkynet.cz ESMTP ready]

TCP 80:
[HTTP/1.1 200 OK Date: Sun, 16 Jul 2006 10:44:44 GMT Server: Apache/1.3.33 (Win32)
PHP/5.0.5 X-Powered-By: PHP/5.0.5 Connection: close Content-Type:   text/html]

TCP 110:
[+OK POP3 server ready <2404.1153046685@master.funkynet.cz>]

TCP 119:
[200 NNTP server ready]

TCP 143:
[* OK IMAP4rev1 server ready]

TCP 389:
[0 a]

TCP 3306:
[w i #HY000Host 'r4i29.chello.upc.cz' is blocked because of many connection errors;
unblock with 'mysqladmin flush-hosts']

TCP 8080:
[HTTP/1.1 200 OK Content-Length: 1433 Content-Type: text/html Content-Location:
http://82.208.37.11:8080/iisstart.htm Last-Modified: Fri, 21 Feb 2003 16:48:30]
------------------------------------------------------------------------------
```

Na praktické ukázce je vidět, že díky přepínači –b lze získat i bannery, a pro úplnost, přepínač –h skryl zavřené porty, aby výstup byl o něco přehlednější.

5.3. Pokus o útok typu odepření služby

Při podrobném zkoumání všech doposud sesbíraných informací, zvláště pak při zkoumání výstupů ze skenerů portů, mě zaujala pasáž, kde program ScanLine vypisuje seznam UDP portů, které jsou otevřené a naslouchají na cílovém počítači. Tento fakt mě přiměl vážně uvažovat o variantě vyzkoušet jako první útok typu UDP flood. Navíc jsem se obával, že zvýšená aktivita na síti způsobená mými pokusy o útok nutně vyvolá zpětnou vazbu ze strany administrátora a ten pak uzavře všechny dosud otevřené přístupové cesty.

Na internetu jsem na stránkách http://www.antiserver.it/Denial-Of-Service/index.html vyhledal co nejnovější utility využívající slabin komunikace na portech UDP. Stáhnul jsem si program Siyanur v1.0 a nejnovější verzi programu pjam. Program HGod, který jsem k útoku zkoušel jako první, jsem měl již stažený na lokálním počítači. Bohužel implementace UDP flood útoku ve zmiňovaném programu, který v sobě spojuje hned několik metod DoS útoků, není zřejmě ještě dokonale dořešena a ve stávajících podmínkách program vracel při pokusech o UDP flood chybu. Druhým programem, který jsem k útoku na UDP porty použil, byl program Siyanur v1.0. Program je vyveden v praktickém grafickém rozhraní a navíc dovoluje definovat řetězec dat, který se má při zaplavování portů posílat. Program dovoluje jednoduše definovat rozsah portů, které mají být napadeny. Přes všechny tyto nesporné výhody cílový systém nevykazoval žádné viditelné reakce. Podobným způsobem jsem se snažil zaútočit i na dva systémy, které leží ve Funkynetu mezi webovým serverem a bránou do Internetu. Bohužel bez odezvy.

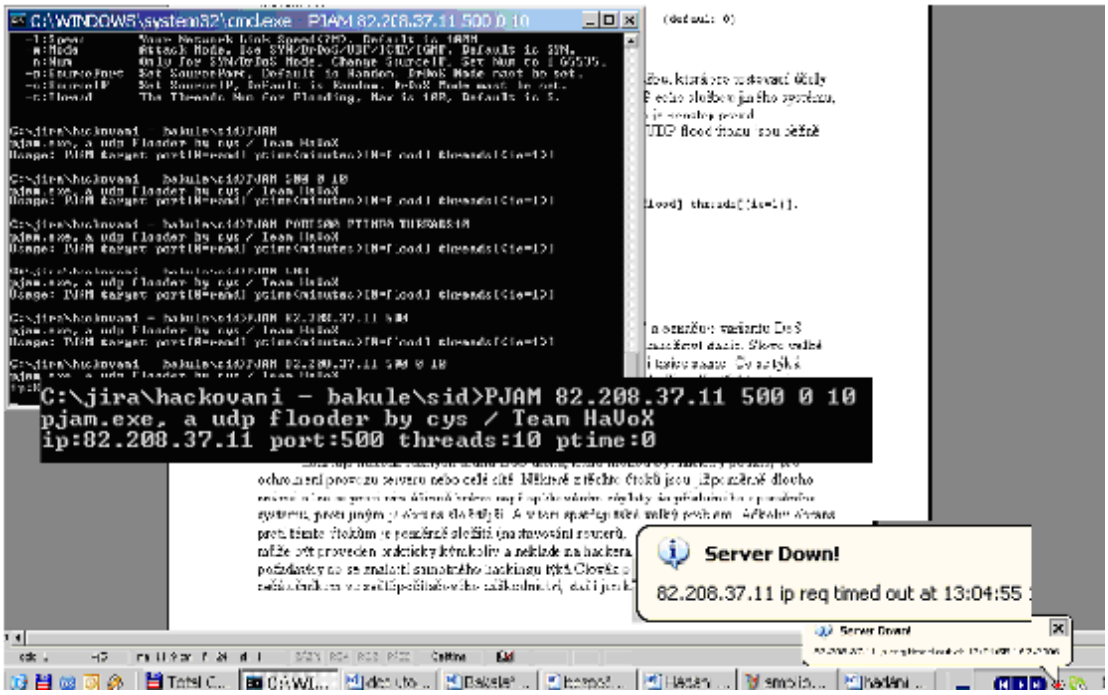


Obr. 5.8 – rozhraní programu Siyanur v1.0

Posledním programem, který jsem použil při pokusu o realizaci úspěšného útoku je program pjam. Program se spouští společně s parametry z příkazové řádky.

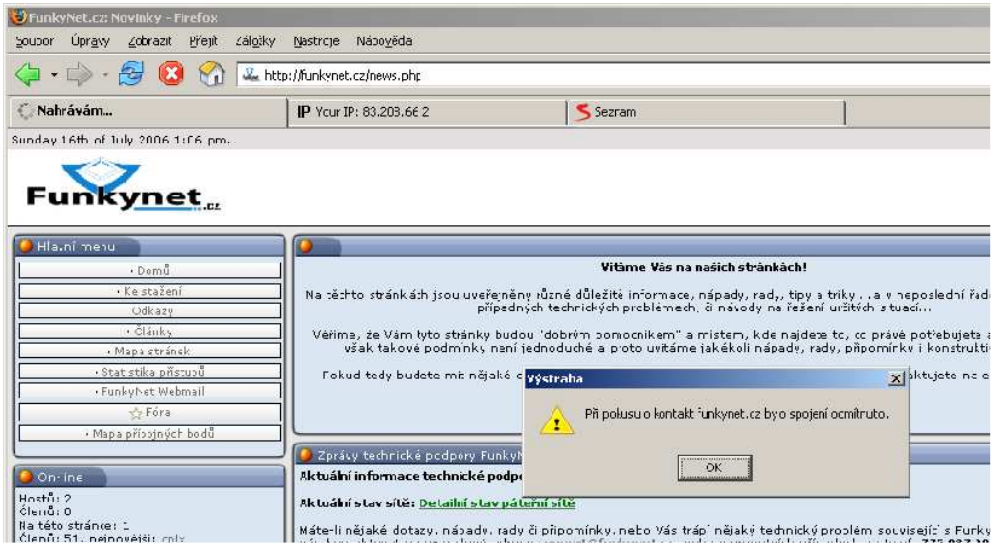
```
C:\...\sid>pjam
pjam.exe, a udp flooder by cys / Team HaVoX
Usage: pjam target port[0=rand] ptime(minutes)[0=flood] threads[(ie=1)]
```

Spustil jsem program s příslušnými parametry, kdy UDP port jsem vybíral podle výstupu programu ScanLine. Během několika chvil, kdy běžela procedura zaplavování cíle, zahlásil program StatusCheck ztrátu odezvy ze serveru funkynet.cz.



Obr. 5.9 – ztráta odezvy z www.funkynet.cz

Jako kontrolu stavu, který indikoval můj počítač, jsem spustil prohlížeč na druhém kontrolním počítači s odlišným bodem připojení. Prohlížeč na notebooku indikoval nedostupnost stránek funkynet.cz, jak je vidět na obrázku 5.10.



Obr 5.10 – kontrolní pokus o připojení na stránky

Přerušil jsem útok běžící v okně s příkazovou řádkou a provedl několik dalších kontrolních testů. Provedl jsem pokus o získání odezvy ECHO REPLY, ale cílový server neodpovídal.

```
C:\...\sid>ping 82.208.37.11 -t

Příkaz PING na 82.208.37.11 s délkou 32 bajtů:

Vypršel časový limit žádosti.
Vypršel časový limit žádosti.
Vypršel časový limit žádosti.
Vypršel časový limit žádosti.
Vypršel časový limit žádosti.
Vypršel časový limit žádosti.

Statistika ping pro 82.208.37.11:
Pakety: Odeslané = 6, Přijaté = 0, Ztracené = 6 (ztráta 100%),
```

Stejný test jsem provedl i z kontrolního počítače. Výstup potvrzoval prvním strojem indikovaný stav.

Tato situace, kdy sytém ani stránky nefungovaly, trvala bez změny asi 30minut. Po této době začal systém vracet kladnou odezvu na dotazy programu ping, ale stránky byly stále nedostupné z obou počítačů.

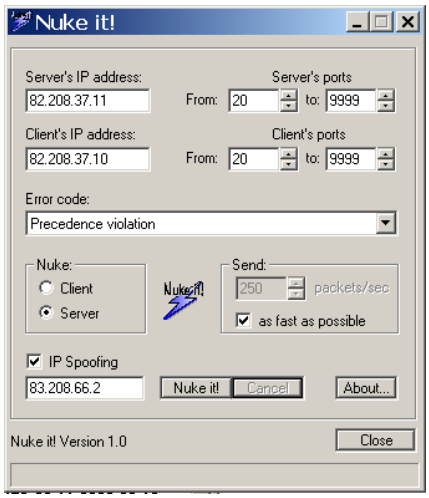
Jako další postupný krok k analýze cílového systému jsem se rozhodl použít program ScanLine k opětovné inventarizaci cílového systému. Při porovnávání se stavem, který program ScanLine zaznamenal před útokem, jsem nalezl několik odlišností, ze kterých asi nejpodstatnější byla v počtu otevřených portů.

```
C:\jira\hackovani - bakule\sid>sl -bh 82.208.37.11
ScanLine (TM) 1.01
Copyright (c) Foundstone, Inc. 2002
.
.
.
UDP ports: 53 123 137 138 445 500 1027 1034
.
.
.
Scan finished at Mon Jul 17 01:27:03 2006

1 IP and 267 ports scanned in 0 hours 0 mins 9.92 secs
```

Na cílovém systému došlo k radikálnímu omezení otevřených UDP portů. Následné testování pomocí opětovného pokusu o UDP flood pomocí programu pjam nepřináší žádné odezvy.

Po dalších asi 30 minutách kontrolou prohlížeče na testovacím počítači shledávám stránky opět plně funkčními. Na počítači, ze kterého byl veden útok, se stránky nezobrazují. Několik krátkých ověření z různých prohlížečů mne vede k závěru, že pravděpodobně došlo k blokaci IP adresy útočícího počítače. Pomocí programu Anonymizer (viz. Kapitola Zabezpečení) získávám v prostředí Microsoft Explorer novou identitu a testuji stránky www.funkynet.cz. Stránky se bez jakýchkoliv náznaků problémů načetly a nebyl problém jimi surfovat. Stav, při němž nemohu stránky bez změny identity navštívit, přetrvává stále.

Následně jsem testoval na cílovém systému relativně velké množství utilit umožňujících realizaci různých variant DoS útoků jako je SYN flood či Smurf. Příkladem takových pokusů může být i program Nuke-it! ze stejného skladiště programů jako všechny ostatní, v této části zmiňované programy. Jak se používá, lze vidět na obrázku 5.11.



Obr. 5.11 – program Nuke it!

Další program, který jsem testoval proti serveru funkynet.cz, byl dříve zmiňovaný HGod.

```
C:\jira\hackovani - bakule\sid>hgod 82.208.37.11 25,53,80,110,119,143,
389,443,465,563,636,1025,1028,1723,3306,3389,8080 -d:0 -s:83.208.66.2
============================== HUC DoS Tool V0.5==============================
================ By Lion, Welcome to http://www.cnhonker.com=================

      Start                    SYN                    flood                  -
>82.208.37.11:25,53,80,110,119,143,389,443,465,563,636,1025,102
      8,1723,3306,3389,8080. Thread:10.
      Ctrl+C to Quit

      = / =
```

kromě metody SYNflood, která je demonstrována uvedeným výstupem na příkazové řádce, jsem testoval i ostatní dostupné metody, ale bez žádoucích výsledků.

Bohužel se mi již nepodařilo ani z jednoho z počítačů, které jsem měl při pokusu o realizaci útoku na webové stránky, žádných pozitivních výsledků v oblasti útoků typu DoS dosáhnout. Velmi rychlá odezva ze strany administrátora mi za daných podmínek znemožnila zopakovat a ověřit DoS útok, či realizovat jiný, podobný, který by mohl presentovat probírané techniky útoků.

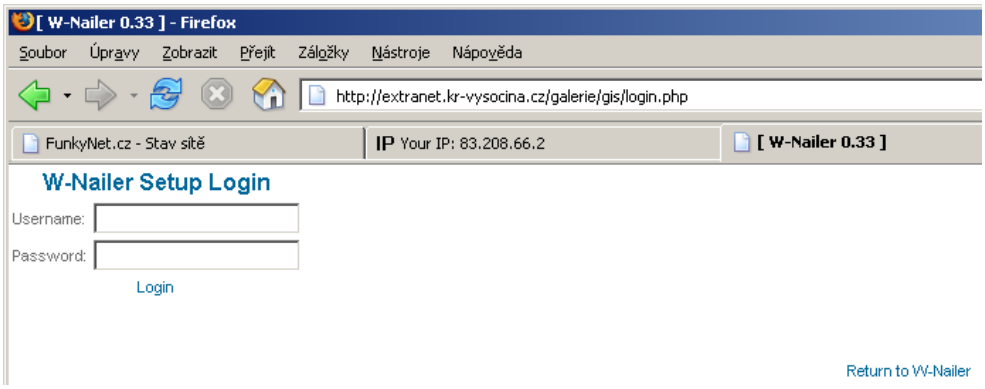
5.4. Hádaní hesel

Vzhledem k velmi silné politice přihlašování na stránkách funkynet.cz jsem se rozhodl prezentovat techniky hádání hesel na dvou jiných webových stránkách. První z nich jsou hudební stránky Amplion (http://www.amp.cz), které mimo jiné obsahují i elektronický obchod s hudebními nosiči. Druhým pokusným cílem se staly extranetové stránky krajského úřadu kraje Vysočina(http://extranet.kr-vysocina.cz/). Nejdříve na stránkách kraje Vysočina předvedu útok na formulářové ověřování pomocí programu Brutus. Následovat bude ještě jedna demonstrace lámání administrátorského přístupu, a to na stránkách prvně zmiňovaných, hudebního serveru Amplion. K tomuto kroku jsem se uchýlil po zjištění, že útok brutální silou na účet „admin“ na stránkách funkynet.cz představuje pro naše účely nepřiměřeně dlouhý časově náročný úkon. Proces hádání silných hesel je běh na dlouhou trať a ve většině případů, kdy heslo obsahuje znaky celé klávesnice a má délku větší než sedm znaků, je tento pokus neúspěšný. Proto se cílem vhodným k presentování technik hádání hesel staly stránky s minimální nebo žádnou bezpečnostní politou, což si myslím, že zvláště u stránek státní správy není moc chvályhodný postoj k problematice bezpečnosti.

Základním krokem při pokusu o útok na přihlašovací formulář je jeho lokalizace na stránkách. Asi nejjednodušším způsobem je využití služeb nějakého vyhledávacího portálu jako je třeba Google(http://www.google.com). Klíčovými výrazy by pak měly být „název_ serveru“ a výraz „login“, někdy lze zkusit i klíčové slovo „admin“. Klasickým příkladem je cesta k formuláři webových stránek Amplion, kdy přihlašovací formulář je ve výsledcích hledání na prvním místě.

5.4.1. Aplikace programu Brutus na formulář ověřování webových stránek kraje Vysočina

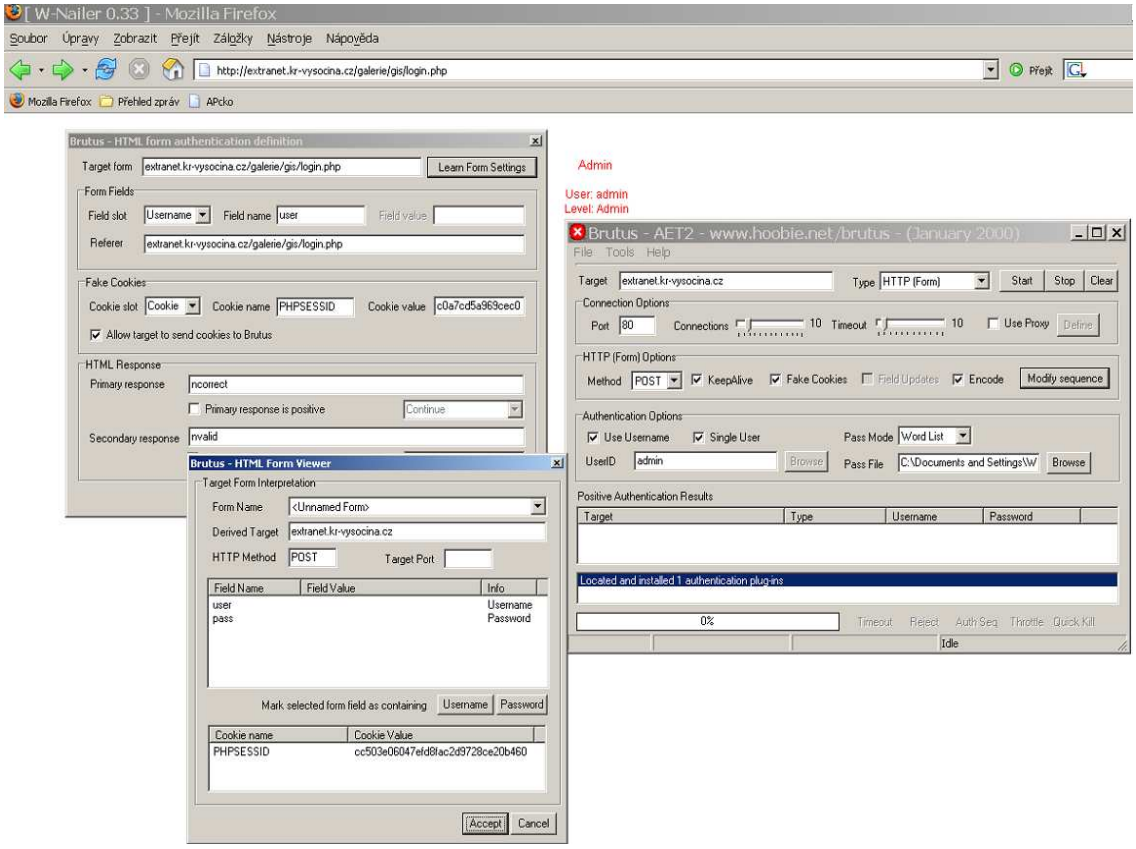
Pomocí vyhledávače Google (klíčová slova „vysočina login“) jsem našel přesnou adresu přihlašovacího formuláře.



Obr. 5.12 – přihlašovací formulář

Tato adresa bude výchozí při nastavování procedury programu Brutus. Spustil jsem utilitu Brutus. Klíčové je správné nakonfigurování programu. Cílem útoku je

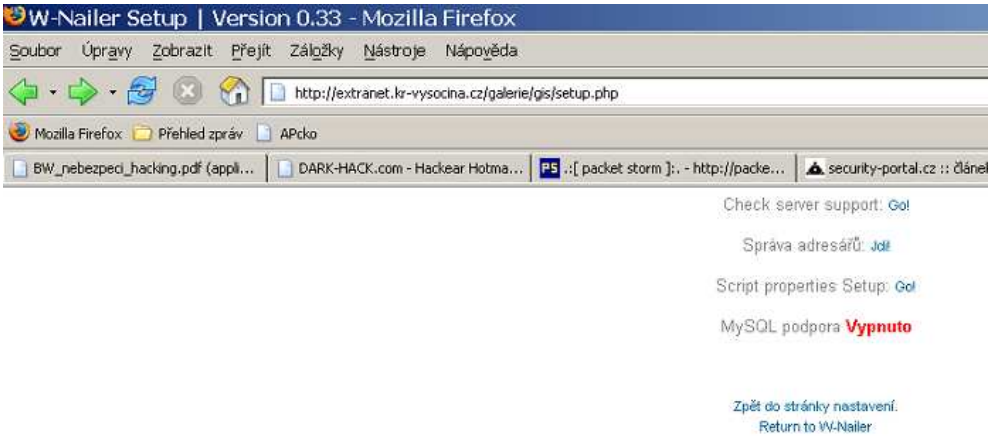
extranet.kr-vysocina.cz a typ útoku bude nastaven na „HTTP(form)“. Program mimo jiné nabízí i možnosti útoku na HTTP(Basic) nebo POP3 a SMTP. Při výběru možnosti formulářového útoku se cílový port automaticky nastaví na port číslo 80, čili s tímto nastavením není potřeba nic dělat. Dalším krokem teď bude rozpoznávání formuláře. Následuje volba možnosti „Modify Sequence“ v hlavním okně a do volného pole přijde adresa cílového formuláře, v tomto případě „http://extranet.kr-vysocina.cz/galerie/gis/login.php“. Nyní se lze pokusit o automatické rozpoznávání vstupních polí nebo je možné zadat je ručně. Pokud je voleno ruční zadávání, je potřeba otevřít paralelně zdrojový kód stránky a najit názvy příslušných polí. Automatické rozpoznávání, volba „Learn from Settings“ otevře nové okno, kde program sám rozpozná všechna vstupní pole včetně skrytých a identifikuje metodu přihlašování. Na uživateli teď zůstává jen potvrdit správné přiřazení polí „username“ a „password“ a volbu potvrdit tlačítkem „Accept“. Vše lze vidět přehledně na obrázku 3.13 níže.



Obr. 5.13 – nastavení programu Brutus na formulář ověřování

Posledním krokem, který je třeba udělat, je vybrat skupinu uživatelských jmen a hesel, ze kterých má program Brutus hádat. Já jsem volil pouze jedno uživatelské jméno, a to konkrétně „admin“. Slovník, respektive textový seznam hesel jsem si vytvářel sám, a zahrnuje pouze základní varianty a jejich variace vhodné k uživatelskému jménu „admin“. Příkladem může být: admin, admin123, administrace, [prázdné_heslo], atd.

Nyní by mělo být vše připraveno a stačí stisknout tlačítko start. Vzhledem k malému počtu hesel ve slovníku, je proces hádání takřka bleskový. V případě útoku hrubou silou mohou pokusy bezvýsledně probíhat celé dny, což musí bezpodmínečně nutně vzbudit pozornost na straně administrátorů cílových stránek. Pokud jde vše podle plánu, je za moment hotov výstup z programu Brutus, v tomto případě magická kombinace, kterou jsem se nakonec rozhodl neuvést v rámci bezpečnosti stránek krajského úřadu. Nahoře na obrázku 5.13 v pozadí je vidět stránka extranetu kraje vysočina se statusem a úrovní práv. Na obrázku 5.14, který lze vidět níže, je zachyceno menu voleb přístupných pouze administrátorovi stránek.



Obr. 5.14 – administrační menu

Na stránkách kraje Vysočina jsem chtěl pouze demonstrovat postup, jak lze napadnout stránky a využít neopatrnosti administrátorů k útoku. Vzhledem k faktu, že stránky náleží státní správě, neshledal jsem vhodným demonstrovat na nich možné dopady takových útoků. K tomu nám poslouží druhé stránky, a to stránky serveru Amplion. O faktu prolomení hesla na stránkách jsem nikoho z adminů neinformoval.

5.4.2. Útok na stránky serveru Amplion

Jak jsem již napsal v úvodu této kapitoly, nejvhodnějším způsobem jak najít formulář na cílových stránkách, je využít služeb některého z veřejných portálů.

V případě stránek Amplion zašla bezostyšnost administrátora tak daleko, že na uhodnutí kombinace uživatelského jména a hesla nebyly potřeba žádné automatizované programy. Kombinace „admin“/“admin“ mě pouze utvrdila v pocitu, že největší nebezpečí pro webové stránky představují k bezpečnosti lhostejní správci. Na obrázku 4. můžete vidět status přihlášení na serveru, který stránky vrátily jako odezvu na dotázání se na stav přihlášení.

Obr. 5.15 - Potvrzení administrátorských práv na stránkách amp.cz

V případě hudebních stránek Amplion jsem již žádné etické zábrany neměl, takže jsem se na závěr rozhodl presentovat praktické dopady takových úspěšných útoků na stránky. Jak je vidět na obrázku 5.16, během několika málo minut jsem změnil výchozí domovskou stránku serveru Amplion a na titulní stranu jsem umístil vlastní text. Změnu jsem omezil „pouze“ na titulní stranu, protože proti serveru Amplion nechovám žádné antipatie. Ale samozřejmě se tu nabízí možnost celé stránky smazat, či změnit ceny hudebních nosičů v nabídce elektronického obchodu. Potom by škody mohly být opravdu znatelné.

Hned, jak jsem pořídil potřebnou dokumentaci o provedení útoku, kontaktoval jsem elektronickou poštou administrátora stránek a informoval jsem ho o tom, že jsem stránky úspěšně napadl. Útok jsem realizoval v pátek večer, k nápravě ze strany administrátora došlo v pondělí, někdy v ranních až dopoledních hodinách. Administrátor se mě dosud nesnažil nijak kontaktovat.

Obr. 5.16 – výsledný efekt úspěšného útoku na webové stránky

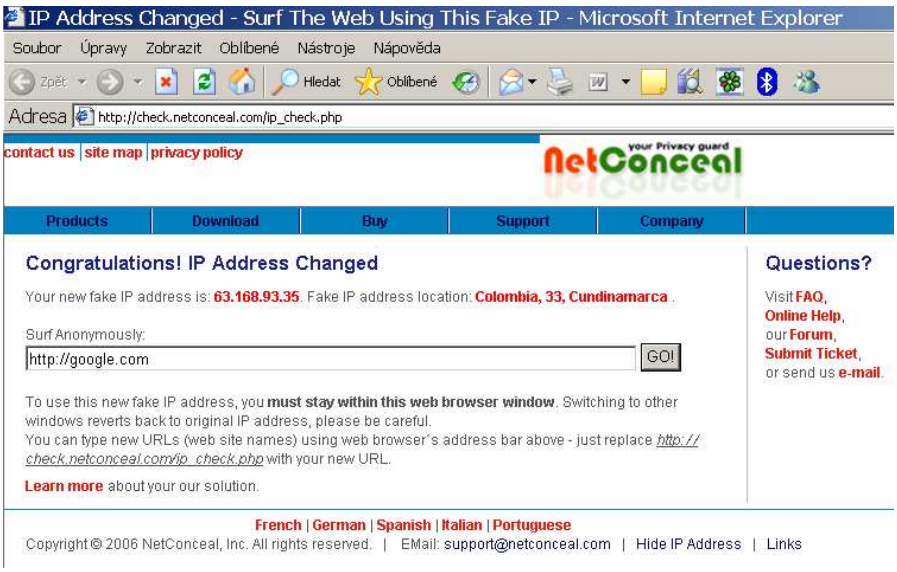
6. Zabezpečení

V této kapitole se budu věnoval bezpečnostní stránce problematiky. Ukážu pohled na problém ze dvou stran. Na jedné straně stojí bezpečnost útočníka, tedy metody jak se schovat a zůstat v bezpečí a na druhé straně stojí problém zabezpečení cílového systému.

6.1. Bezpečnost útočníka

Útoky na webové stránky jsou, jak známo, nelegální činnost. Proto by při páchání takových nekalostí mělo být prioritou každého hackera snaha zůstat v tajnosti. Jeho předčasné odhalení v ranné fázi příprav, ať už se jedná o trasování či fázi inventarizace služeb, nutně vede k bezpečnostním opatřením na straně cílového systému v podobě zásahů technických administrátorů systému. To by mohlo vést k předčasnému ukončení snažení útočníka, nebo alespoň k výraznému ztížení jeho podmínek. Cílové systémy mají v záloze mnoho utilit k detekování podezřelých aktivit na síti, ale o tom budu mluvit později v samostatné kapitole. Proto je nezbytně nutné, všechny aktivity maskovat. Nabízí se několik různých metod, které nejlépe fungují společně.

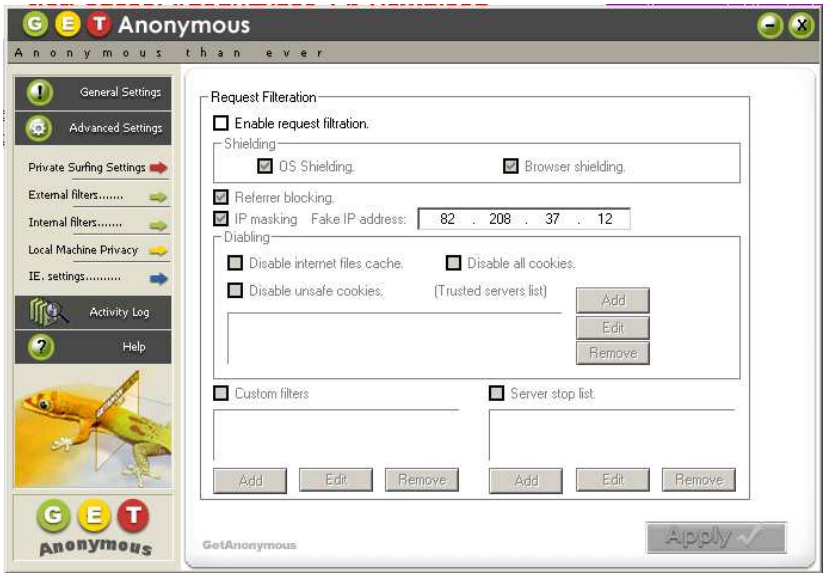
První z opatření, která se zdají v souvislosti s maskováním vhodná, je skrývaní skutečné IP adresy útočníkova stroje. Nabízí se mnoho utilit k tomu vhodných. První, o které budu mluvit je NetConceal Anonymizer(http://www.brothersoft.com/internet/_online_privacy/netconceal_anonymizer_download_39832.html). Program si stáhne aktuální stavy proxy serverů z listu a vybere náhodnou IP adresu a lokaci, kterou přidělí.



Obr. 6.1 – nově přidělená IP adresa programem Anonymizer

Druhým podobným a rozhodně ne méně vydařeným programem je GET Anonymus(http://www.brothersoft.com/internet/online_privacy/getanonymous_download_18021.html). Ba spíše naopak. Program je ještě šikovnější. IP adresu lze zvolit

libovolně a tuto IP adresu dokáže předstírat i při používání některých programů z příkazové řádky.



Obr. 6.2 – nastavování programu GETAnonymous

Ještě se zmíním o jedné možnosti, jak řešit problém s případnou blokací IP adresy útočníkova stroje. Většina domácích připojení, kromě wifi [5], využívá vymožeností tzv. DHCP služby. Jedná se o dynamické přidělování IP adres serverem k tomu určeným, který patří poskytovateli připojení. Když dojde k odpojení například kabelového modemu, jak od internetu, tak od zdroje el. Energie na dobu potřebnou k vymazání paměti s nastavením (cca 10s by mělo stačit), po opětovném napojení modem vyšle na DHCP server žádost o nové nastavení, s čímž obdrží i novou IP adresu. U adsl připojení to bývá ještě jednoduší a nová ip adresa se často dostává s každým novým připojením se k internetu.

Podezřelé aktivity na síti, jako je skenování portů či jiné sondovací akce, však můžou vzbudit nežádoucí pozornost na straně administrátorů cílového systému. Bez ohledu na to, zda je podezřelá aktivita vyvíjena z Prahy nebo Singapuru, nebo z té či oné IP adresy. Proto je potřeba se snažit tyto aktivity zamaskovat. První z možností, která se nabízí, je využití nadstandardních funkcí skenovacího programu nmap. Pokud je cílová síť dobře administrovaná, nabízí program nmap možnost takzvaných klamných skenů. Během opravdového skenu běží zároveň ještě několik dalších (klamných) skenů, které se na cílovém počítači jeví, jako by opravdu probíhaly z jiných počítačů. Pro administrátory je potom velmi těžké určit, které aktivity jsou pravé a které ne.

Další velmi šikovnou utilitou, o které napíšu, je program tcp-junkie [7]. Jeho prvotním účelem bylo testovat firewally, ale velmi rychle se zabydlel u nejednoho hackera. Tento program je vlastně náhodným generátorem TCP paketů napsaným v jazyce Perl. Generuje velké množství náhodných TCP paketů z náhodných zdrojových adres, zdrojových portů a s náhodnými příznaky. Čili vlastně nasimuluje obrovský provoz na síti a portech cílového systému, takže je pak pro administrátory velmi složité mezi tím obrovským množstvím spojení odhalit to pravé nebezpečí.

6.2. Bezpečnost cílového systému

Jediná možná obrana proti prosakování informací do veřejných zdrojů je zavedení striktní bezpečnostní politiky. Asi jedinou možností je zavést interní klasifikaci informací na ty, u kterých je jedno, zda se dostanou ven, a na ty, u kterých to žádoucí není. Další preventivní opatření by mohlo spočívat ve výchově technického personálu k tomu, aby přispíval do veřejných konferencí z anonymních adres.

V kapitole Současné metody a techniky – průzkum sítě jsem se jen dotknul problematiky technik používaných k průzkumu sítě. Přesto je třeba pomýšlet na aktivní obranu i proti do té doby presentovaným, relativně mírným technikám. Mnoho komerčních detektorů průniků do sítě je schopných takové pokusy detekovat. Mluvím zde o tzv. NIDS – Network Intrusion Detection Systems (více na http://www.pavlis.net/articles/288.aspx), které by neměly na žádné síti chybět. Jsou zdatným pomocníkem pro včasné odhalení možného budoucího útoku. Také jeden z nejlepších volně šířitelných detektorů, Snort (http://www.snort.org/dl), autora Marty Roesche, umí tyto metody detekovat. A pokud není žádoucí pasivní role při sledování nežádoucích aktivit na síti a je preferována aktivní obrana, lze vyzkoušet program RotoRouter (http://www.windowsecurity.com/pages/article_p.asp?id=969) autora jménem Humle ze skupiny Rhino9. Tato utilita zaznamenává pokusy o trasování programem traceroute a generuje matoucí odpovědi.

Další presentovanou technikou sběru informací byl hromadný ping a skenování portů. Hromadné pingy je třeba detekovat, protože mohou napovědět o snahách o průnik do sítě. V některých případech je dokonce nutné je blokovat.

Včasnou detekcí lze nejenom zachytit počátek útoku, ale je možné se také pokusit identifikovat útočníka. K detekci lze použít některý z IDS programů, jako je snort, nebo třeba Nuzzler (http://www.majorgeeks.com/downloadget.php?id=2759&file=13&evp=e94ca1928b082db9455e6b0e622fad9d). Na straně počítačů, které jsou pingy ohroženy, existuje mnoho různých utilit. Jenže valná většina z nich jsou pro operační systém Unix. Ve světě Windows je situace poněkud horší. Kromě zmiňovaného Nezzlera, je k dispozici ještě shareware Genius (http://www.indiesoft.com/), který sice umí spolehlivě rozeznat TCP ping, ale ICMP ECHO detekovat neumí.

Co se prevence týká, tak tok ICMP dat lze samozřejmě filtrovat. Je však třeba postupovat velmi opatrně, protože ICMP protokol složí k diagnostice síťového provozu. Nelze tedy bezmyšlenkovitě blokovat veškerá ICMP data, protože by to mohlo vést k výpadkům monitorovacích systémů (pokud jsou nasazeny). Poskytovatel připojení například často monitoruje hraniční směšovače nebo i jiná zařízení, dle přání zákazníků. Taky je třeba si uvědomit, že protokol ICMP používá mnohem více typů zpráv, než je ECHO a ECHO REPLY. Je tedy třeba pečlivě zvážit, které zprávy jsou potřeba, a tedy je nelze blokovat, a které jsou naopak zbytečné. Dobrým nápadem se pak jeví povolit ICMP komunikaci pouze mezi systémy, které jí vyžadují. To se dá zajistit vhodnou konfigurací firewallu pomocí ACL (access control list. Metoda k omezení používání zdrojů pouze pro autorizované subjekty).

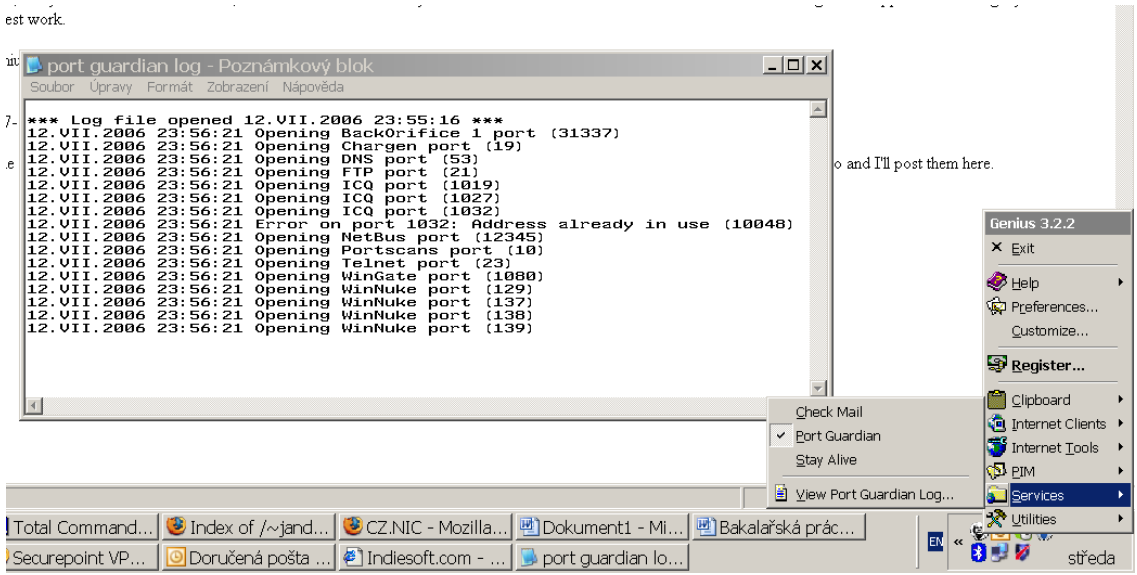
Další zajímavou myšlenkou je odstranit ICMP ECHO a ICMP0 REPLY komunikaci z jádra systému a realizovat ji na aplikační úrovni. Pak je velmi snadné kontrolovat konfigurací na úrovni serveru, kdo bude odpovědi ve formě ICMP REPLY paketů dostávat a kdo ne. Tuto myšlenku rozpracoval Tom Ptacek a v Linuxu ji realizoval

Mike Schiffman ve formě démona pingd(http://www.janderson.ca/~janderson/pingd/), o čemž více píše George Kurtz v knize Hacking bez tajemství [3].

Detekování skenu portů umožní udělat si představu o tom, kdy může proběhnout útok na naše systémy a kdo ten útok podnikne. Z perspektivy unixových uživatelů můžeme detekovat skeny pomocí programu Scanlogd (http://www.openwall.com/scanlogd/). Program PortSentry (http://yolinux.com/TUTORIALS/LinuxTutorialInternetSecurity.html#PORTSENTRY) umí skeny nejenom detekovat, ale umí na ně i aktivně reagovat. Jednou takovou reakcí může být automatické nastavení filtru, který bude blokovat pakety přicházející z útočníkova systému.

Detekce podobných aktivit je velmi důležitá. Zachycení skenu totiž většinou indikuje snahu o průzkum sítě. Po průzkumu může následovat plnohodnotný útok. Je třeba si ale uvědomit, že útočník může zfalšovat IP adresu, ze které provádí sken. Pokud bude zablokována, může se stát, že bude zablokován úplně „nevinný“ systém, který nemá s útočníkem nic společného.

Pro operační systém Windows je k dispozici několik málo utilit detekujících jednoduché skeny portů. Jeden z nich, o kterém jsem již mluvil v souvislosti s hromadnými pingy, je program Genius. Tento software poskytuje mnohem více funkcí než pouhou detekci skenů. Ale co se týče detekce skenů, tak Genius analyzuje v definovaném časovém intervalu požadavky o napojení a v případě, že detekuje sken, zobrazí okno s IP adresou a DNS jménem počítače, který sken prováděl.



Obr. 6.3 – záznamy provozu na portech programu Genius

6.2.1. Obrana proti útokům typu DoS

Ochrana proti DoS útokům patří mezi netriviální problémy i v dnešní době. Slabých míst, která může případný útočník využít, je totiž, jak jsem již naznačil v kapitole Současné metody a techniky – útoky typu DoS, mnoho. A vlastně čím jsou

systémy složitější, tím více možností jak chybovat při programování vyvstává před jejich tvůrci. Toho vydatně využívají útočníci a stále vytvářejí nové a nové utility k páchání útoků typu DoS. Obecně platí, že velká síla je v řádném a promyšleném nakonfigurování operačního systému a firewallu.

6.2.1.1. Chyby v implementaci protokolů

Proti tomuto druhu útoku je velmi těžké se bránit, protože příčiny leží v chybném kódu. Jediná opatření, která jsou k dispozici, jsou omezena pouze na pravidelné aktualizování systému a včasné aplikování bezpečnostních záplat od výrobce softwaru.

6.2.1.2. Útoky na síťovou vrstvu

V oblasti ochranných prvků v rámci OS je typickou volbou použití vestavěného packetového filtru, umožňujícího mimo jiné omezit maximální počet spojení z konkrétní IP adresy nebo podsítě. Další možností bývá například povolení tzv. „SYN-cookies“. Tato technika upravuje reakci OS na žádost o vytvoření spojení — příjem SYN packetu. Namísto vyhrazení místa v tabulce spojení jako odpověď odeslán SYN/ACK packet, který má iniciální sekvenční číslo *(Initial Sequence Num ber,* dále jen „ISN“) nastavené na výsledek jednocestné (hash) funkce aplikované na zdrojovou a cílovou adresu, zdrojový a cílový port a časově omezenou tajnou informaci. Při přijetí ACK packetu je potom ověřena shoda vráceného ISN pomocí stejného výpočtu. V případě shody je spojení přijato, v opačném případě odmítnuto. Tato metoda tedy efektivně znemožňuje provedení SYN-flood útoku, nebot‘ nelze vyčerpat tabulku spojení pomocí pouhého zahlcení cílového systému proudem SYN packetů.

6.2.1.3. Útoky hrubou silou.

Aby bylo možno úspěšně zabránit UDP záplavě, je třeba buď zakázat všechny UDP služby na každém uzlu sítě nebo aktivovat firewall filtrující všechny příchozí požadavky UDP služeb. Poněvadž UDP služby jsou primárně navrženy pro interní diagnostikování, melo by být dostačující zavedení pravidla o zákazu přístupu UDP služeb z Internetu. Nedoporučuje se úplný zákaz veškerého UDP provozu, mohlo by tím totiž dojít k odmítnutí některé legitimní aplikace, která používá UDP jako svůj transportní mechanismus.

6.2.2. Bezpečnostní politika hesel

Asi jediným, zato nejúčinnějším opatřením proti hádání hesel a hrubé síle je kombinace silné politiky hesel a silné politiky blokování účtů. Aplikace by měla zablokovat účet již po malém počtu neúspěšných pokusů o přihlášení, aby snížila možnost tohoto útoku. Je dobré dávat pozor na útok typu odepření služeb proti aplikaci s přehnaně paranoidní politikou blokování účtů. Zlomyslný útočník by mohl zablokovat všechny účty na systému. Dobrým kompromisem se pak jeví možnost blokovat uživatelské účty jen dočasně na krátkou dobu. Řekněme deset minut. Toto opatření efektivně snižuje rychlost hádání hesel. Díky silné politice nepůjde žádné heslo uhodnout. Efektivně velký prostor klíčů pro hesla, větší než osm alfanumerických znaků, spolu se silnou politikou blokování účtů omezuje možnost útoku na hesla hrubou silou.

7. Závěr

V této práci jsem si dal za cíl seznámit s problematikou útoků na webové stránky. Snažil jsem se nastínit metody a postupy které se používají při vyhledávání důležitých informací a pak jsem se věnoval popisování základních typů útoků na webové stránky. Prezentované programy a postupy při jejich aplikaci jsem se snažil v praktické části ukázat na konkrétních cílech.

K ukázce realizace základních technik mapování sítě a inventarizace perfektně posloužila síť FunkyNet.cz. Bohužel zvýšená podezřelá aktivita na síti z mé strany, která vyeskalovala pokusem o vyvolání chyby typu „odmítnutí služby“, vedla k radikálním bezpečnostním opatřením zavedeným následně po pokusu o útok na síti FunkyNet.cz. Vzhledem k omezení, že pokusy byly realizovány na platformě s operačním systémem Windows XP, se již nepodařilo vyvolat odezvu na pokusy o útok typu „odepření služby“. Pro realizaci pokročilých technik útoků na webové servery je prakticky nezbytné pokusy realizovat z unixové platformy. Její možnosti a především technické zázemí na Internetu daleko přesahují schopnosti a podporu systému Windows v této oblasti.

Po útocích typu DoS jsem úspěšně presentoval techniky hádání/lámání hesel na dvou webových serverech. V obou případech byla příčina mého úspěchu v nedbalosti administrátorů těchto stránek. Bohužel v každodenním životě se s tímto jevem neopatrnosti lze setkat na každém kroku Internetem a já shledávám skutečností, že někteří uživatelé a administrátoři se asi nikdy nepoučí, dokud se jich tento problém nedotkne osobně.

V poslední části jsem se snažil nastínit problematiku řešení některých základních bezpečnostních opatření jak na straně útočníka, tak na straně cílového systému. Nesnažil jsem se ukázat konkrétní postupy při nastavování firewallů a různých filtrů. Mým cílem bylo spíše čtenáře seznámit s problematikou obecně a pouze ukázat jakým směrem by se měla konkrétní opatření ubírat. Uvedl jsem několik obecných zásad bezpečnostní politiky informací a probral zásady a pozitivní důsledky tvorby silných hesel a silné politiky přihlašování, které byly dobře patrné z neúspěšných pokusů o prolomení hesla k uživatelskému účtu „admin“ n a stránkách FunkyNet.cz.

Samozřejmě vím, že má práce na téma útoky na webové stránky nemůže být vyčerpávající co se informační obsáhlosti týče. Daná problematika je tak rozsáhlá, že vystačí naplňovat informacemi desítky výtisků knih, které v dnešní době chrlí komerční nakladatelství. Navíc k rychlému nárůstu objemu informací o této problematice velkým podílem přispívá stále přetrvávající boom v oblasti výpočetní techniky. Proto jsem se snažil presentovat základní techniky a vytvořit ucelený pohled na jeden segment počítačové kriminality. Záměrně jsem se vyhnul zacházení do detailů v souvislostech s komplikovanými pokročilými technikami ovládnutí vzdáleného počítače, kterými se hodlám zabývat v budoucnosti.

8. Seznam použitých zdrojů a literatury

[1] Kolektiv autorů. The Phenomenon Of Computer Crime: Making-A-Difference, červen 2006 online publokace: http://www.mobrien.com/computer_crime1.htm

[2] STERLING, Bruce. Zátah na Hackery - Řád a chaos v elektronickém pohraničí. 1992. online publikace: http://stuff.mit.edu/hacker/hacker.html

[3] McCLURE, Stuart a SCAMBRAY, Joel a KURTZ, George.Hacking bez tajemství: 3. aktualizované vydání, vydání první. Brno:Computer Press 2003, ISBN80-7226-948-8

[4] SCAMBRAY, Joel a SHEMA, Mike.Hacking bez tajemství:webove aplikace.vydání první.Brno:Computer Press 2003. ISBN 80-7226-769-8

[5] Internetová encyklopedie Wikipedia, http://en.wikipedia.org/wiki/Main_Page

[6] Informační server ROOT.CZ, článek „Noční můra jménem SYN flooding“, Michal Krause, září 1999, http://www.root.cz/clanky/nocni-mura-jmenem-syn-flooding

[7] Stránky zabývající se problematikou hackingu Antiserver, sekce download, http://www.antiserver.it/index.html

[8] RNDr. Libor Dostálek, Velký průvodce protokoly TCP/IP a systémem DNS, Computer Press 2002, e-book - http://www.cpress.cz/knihy/tcp-ip-bezp/

[9] Jiří Peterka, Co (ne)najdete ve slovníku: token, časopis COMPUTERWORLD, číslo 48, ročník 1993, archiv článků: http://www.manualy.sk/archiv/a348c120.htm

9. Přílohy

9.1 Seznam obrázků a tabulek