VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

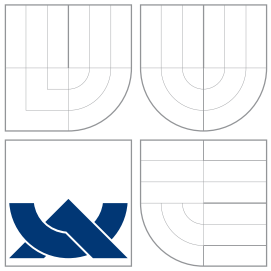
ANALÝZA A DEMONSTRACE VYBRANÝCH SÍŤOVÝCH ÚTOKŮ

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

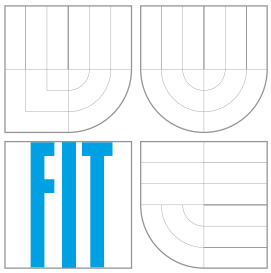
AUTOR PRÁCE
AUTHOR

JAKUB JIRÁSEK

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

ANALÝZA A DEMONSTRACE VYBRANÝCH SÍŤOVÝCH ÚTOKŮ

ANALYSIS AND DEMONSTRATION OF SELECTED NETWORK ATTACKS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — JAKUB JIRÁSEK
AUTHOR

VEDOUCÍ PRÁCE — Ing. PAVEL OČENÁŠEK
SUPERVISOR

BRNO 2009

Abstrakt

Cílem bakalářské práce je popsat a v reálném prostředí provést vybrané síťové útoky, které se vyučují na FIT VUT v Brně. K útokům jsou vytvořeny demonstrační úlohy tak, aby je bylo možné využít při výuce bezpečnosti počítačových sítí. První část práce se zabývá teorií k jednotlivým útokům a možnými bezpečnostními řešeními. Ve druhé části jsou pak popsány praktické implementace každého útoku.

Abstract

The aim of this Bachelor thesis is to describe and realize selected network attacks which are taught at the FIT, Brno University of Technology. The thesis includes demonstration exercises that can be used for teaching computer network security. First part of this work discusses theory for each attack and possible security solutions. In the second part the practical realizations (for each network attack) are described.

Klíčová slova

Lokální počítačová síť, síťové útoky, bezpečnost sítí, odposlouchávání, odchytávání paketů, man in the middle, spoofing, poisoning, flooding.

Keywords

Local Area Network, network attacks, network security, eavesdropping, packet sniffing, man in the middle, spoofing, poisoning, flooding.

Citace

Jakub Jirásek: Analýza a demonstrace vybraných síťových útoků, bakalářská práce, Brno, FIT VUT v Brně, 2009

Analýza a demonstrace vybraných síťových útoků

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Pavla Očenáška

.....
Jakub Jirásek
19. května 2009

Poděkování

Na tomto místě bych rád poděkoval vedoucímu této bakalářské práce, panu Ing. Pavlu Očenáškovi, za cenné rady, náměty a připomínky týkající se práce.

Obsah

Seznam obrázků

Kapitola 1

Úvod

Bezpečnost je v současné době velice často skloňovaným slovem. Ať už se jedná o práci, cestování nebo dětskou hračku, všude je požadována co největší bezpečnost. Nejinak je tomu i v oblasti informačních technologií a s tím souvisejících počítačových sítí. Základem pro vymýšlení a implementování bezpečnostních mechanismů však je, že víme, proti čemu zabezpečujeme, čemu se snažíme zabránit.

Zůstaňme tedy již čistě v oblasti počítačových sítí. Jistě ne nadarmo se říká: „Útočníci jsou vždy o krok napřed“. Většina bezpečnostních řešení je tedy až reakce na provedený útok. Stačí se podívat na technologie vymyšlené v počátcích počítačových sítí. V té době žádná bezpečnostní rizika nehrozila, jen výjimečně tedy tyto technologie obsahují nějaká bezpečnostní opatření. Velikou výhodou při zabezpečování počítačové sítě tedy může být znalost, na jakém principu síťové útoky fungují a jakým způsobem je útočník může provést.

Tím jsem se dostal k náplni této práce, kterou je popsat a v reálném prostředí provést vybrané síťové útoky. Výsledkem jsou pak demonstrační úlohy, určené pro praktické využití při výuce předmětu Bezpečnost počítačových sítí. V rámci tohoto předmětu byla studentům také většina těchto úloh autorem předvedena a vysvětlena.

Demonstrační úlohy jsou ve formě videí, vytvořených nasnímáním obrazovky během provádění útoku. Díky tomu je možné ukázat, jakým způsobem může útočník postupovat a jak pak takovýto útok vypadá v reálném prostředí. K úlohám, u kterých je to vhodné, jsou vytvořeny textové prezentace.

Vlastní text práce je rozdělen do dvou samostatných kapitol. Začátek první kapitoly je věnován klasifikaci síťových útoků, každá z dalších podkapitol se potom věnuje vždy jednomu konkrétnímu útoku. U nich je rozebrána teorie, kterou je potřeba pro pochopení útoku znát, poté následuje vysvětlení principu útoku a na závěr každé podkapitoly nastínění možných bezpečnostních řešení.

V kapitole následující je pak popsán postup, jakým byly jednotlivé útoky provedeny. Pokud byl k útoku použit specializovaný nástroj, je zde také stručně popsán.

Kapitola 2

Útoky v lokální počítačové síti

Náplní této kapitoly je popis všech vybraných útoků. Úvodní podkapitola se zabývá rozdělením síťových útoků a slouží jako jakýsi úvod k celé problematice. Každá z následujících podkapitol se potom zabývá jedním konkrétním útokem. Před vysvětlením principu každého z útoků jsou vždy uvedeny informace, které jsou pro jeho pochopení důležité.

2.1 Klasifikace útoků v síti

(„Podkapitola převzata z [9].“)

2.1.1 Pasivní útoky

Jedná se například o odposlouchávání komunikace nebo skenování sítě, kdy útočník pouze zjišťuje informace na dané síti, ale do komunikace nijak nevstupuje. Pasivní útoky často předcházejí nebo jsou součástí útoků aktivních.

Odposlouchávání komunikace (*eavesdropping*) U tohoto typu útoku útočník odposlouchává data přenášená po síti. Může se tak dostat k důvěrným informacím, jako jsou přihlašovací údaje, e-mail nebo telefonní hovor.

Analýza komunikace (*traffic analysis*) V tomto případě se také jedná o odposlech dat, nezajímá nás však jejich obsah (může být i šifrován). Cílem tohoto útoku je například získání informací o komunikujících stranách nebo o topologii sítě.

2.1.2 Aktivní útoky

Při tomto útoku již útočník aktivně vstupuje do komunikace, například modifikací přenášených dat nebo generováním falešného síťového provozu.

Podvržení identity (*masquerade*) Podvržení identity znamená, že se útočník vydává za jiného legitimního uživatele sítě.

Útok přehráváním (*replay attack*) Útoku předchází pasivní odposlech dat, která jsou pak opětovně zaslána za účelem neautorizované činnosti.

Modifikace dat (*modification*) Jedná se o změnu, zdržení nebo přehození pořadí původních dat.

Odepření služby (*Denial of Service*) Výsledkem je odepření služby běžnému uživateli, například zahlcením.

2.2 Skenování sítě

Základním stavebním kamenem pro realizaci útoků v lokální počítačové síti je zjištění, jak daná síť vypadá. Zajímá nás především její topologie, počet serverů/klientů, použitý operační systém atd. Můžeme na to jít dvěma odlišnými způsoby — fyzickým nebo softwarovým. Fyzickým je myšleno, že si celou síť projdeme a podíváme se, jak a co je kde připojeno. Tento způsob je sice jednoduchý, ale také dost neefektivní a to hlavně z těchto důvodů:

- často k síti vůbec fyzický přístup nemáme
- síť může být rozlehlá a potom tento způsob zabere nemálo času
- kdykoliv může dojít k její změně
- sledovat tímto způsobem, který klient je právě připojen a jaké služby využívá je prakticky nemožné

Všechny tyto problémy dokážeme vyřešit pomocí softwarového skenování, proto se zaměřím hlavně na něj.

2.2.1 Připojení uživatelé

Pro zjištění, zda-li je daný host právě připojený, postačí ping na jeho IP adresu. Program ping odešle speciální paket ICMP ECHO_REQUEST a pokud je cílový host připojený, dostaneme jako odpověď paket ICMP ECHO_REPLY. Takto se dají projít všechny adresy v lokální síti, čímž získáme přehled o připojených počítačích. V praxi se však tento způsob příliš nepoužívá, je hodně pomalý a ve velké síti může travat až několik hodin. Daleko lepší je použít například program `fping`, kterému můžeme s parametrem `-f` předat textový soubor se seznamem zjišťovaných IP adres. `Fping` navíc umí odeslat více dotazů najednou a celý proces tak značně urychlit.

2.2.2 Skenování portů

Další velice užitečnou informací je, jaké síťové služby na daném počítači běží. Jednotlivé programy se na aplikační vrtvě identifikují pomocí portů a pokud některá aplikace na daném portu naslouchá, musí být port otevřený. To, které porty má počítač otevřené, se zjišťuje tzv. skenováním portů. Programy pro skenování portů využívají paketů sloužících pro vytvoření TCP spojení (SYN - SYN/ACK - ACK), jak ukazuje obrázek 2.1.

Pro skenování se používá mnoho technik, většinou vycházejících z RFC standardu, jak by měl protokol TCP reagovat na různé akce. Zde jsou tři nejzákladnější („převzato z [4]“):

Navázání spojení Program se pokusí o úplné spojení s cílovým portem, viz obrázek 2.1

SYN sken V tomto případě již k uzavření spojení nedojde, protože klient navazující spojení ho nepotvrdí paketem ACK. Odešle SYN paket a počká na odpověď. Pokud se mu vrátí SYN/ACK, port je pravděpodobně otevřen, pokud je odpověď RST/ACK, port je nejspíše uzavřen. Výhodou tohoto skenování je, že protože nedojde k potvrzení spojení, nemusí být odhaleno (logováno).

FIN sken Místo paketu SYN je odeslán paket FIN, na který by měl cílový systém odpovědět RST paketem na všechny uzavřené porty.

2.2.3 Ukázka programu nmap

Pro skenování portů (a sítě obecně) je jednou z možností použití programu `nmap`. Jeho možnosti jsou opravdu velké, od zjištění připojených hostů, jejich otevřených portů, až po odhad použitého operačního systému (pozn.: z mých zkušeností funguje velice dobře, např. u Linuxu se pokusí rozeznat použité jádro a u MS Windows i případně instalovaný Service Pack). Existuje pro něj také grafické rozhraní `Zenmap`, které umí graficky zobrazit propojení počítačů.

Připojení uživatelé:

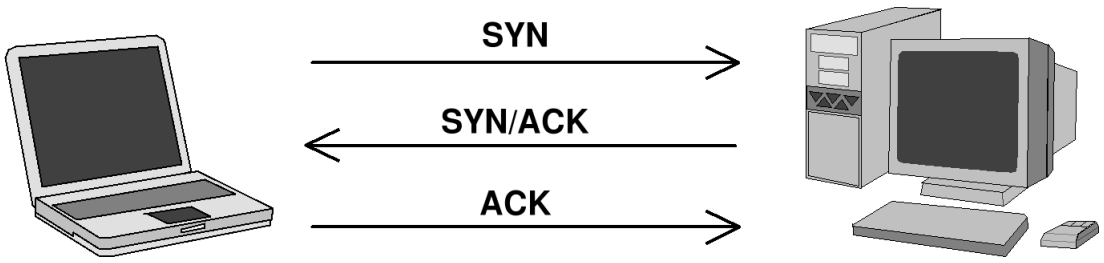
```
nmap -sP 192.168.1.0/24

Starting Nmap 4.76 ( http://nmap.org ) at 2008-11-01 13:26 CET
Host my.router (192.168.1.1) appears to be up.
MAC Address: 00:17:31:04:01:D2 (Asustek Computer)
Host server.xxxxxxx.net.upc.cz (192.168.1.2) appears to be up.
MAC Address: 00:01:02:9D:62:A0 (3com)
Host 192.168.1.5 appears to be up.
MAC Address: 00:13:02:7D:A1:2C (Intel Corporate)
Host 192.168.1.8 appears to be up.
Host 192.168.1.11 appears to be up.
MAC Address: 00:11:D8:01:D1:E7 (Asustek Computer)
Nmap done: 256 IP addresses (5 hosts up) scanned in 10.45 seconds
```

Sken portů:

```
nmap -sS 192.168.1.2

Starting Nmap 4.76 ( http://nmap.org ) at 2008-11-01 13:32 CET
Interesting ports on 192.168.1.2:
Not shown: 984 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
```



Obrázek 2.1: Vytvoření TCP spojení

```
22/tcp    open  ssh
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
110/tcp   open  pop3
111/tcp   open  rpcbind
113/tcp   open  auth
139/tcp   open  netbios-ssn
143/tcp   open  imap
445/tcp   open  microsoft-ds
548/tcp   open  afp
631/tcp   open  ipp
993/tcp   open  imaps
2049/tcp  open  nfs
10000/tcp open  snet-sensor-mgmt
MAC Address: 00:01:02:9D:62:A0 (3com)

Nmap done: 1 IP address (1 host up) scanned in 1.52 seconds
```

Odhad operačního systému (Suse Linux 2.6.22):

```
nmap -O 192.168.1.11

Starting Nmap 4.76 ( http://nmap.org ) at 2008-11-01 13:39 CET
Interesting ports on 192.168.1.11:
Not shown: 998 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
111/tcp open  rpcbind
MAC Address: 00:11:D8:01:D1:E7 (Asustek Computer)
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.22 - 2.6.23
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at
http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 5.63 seconds
```

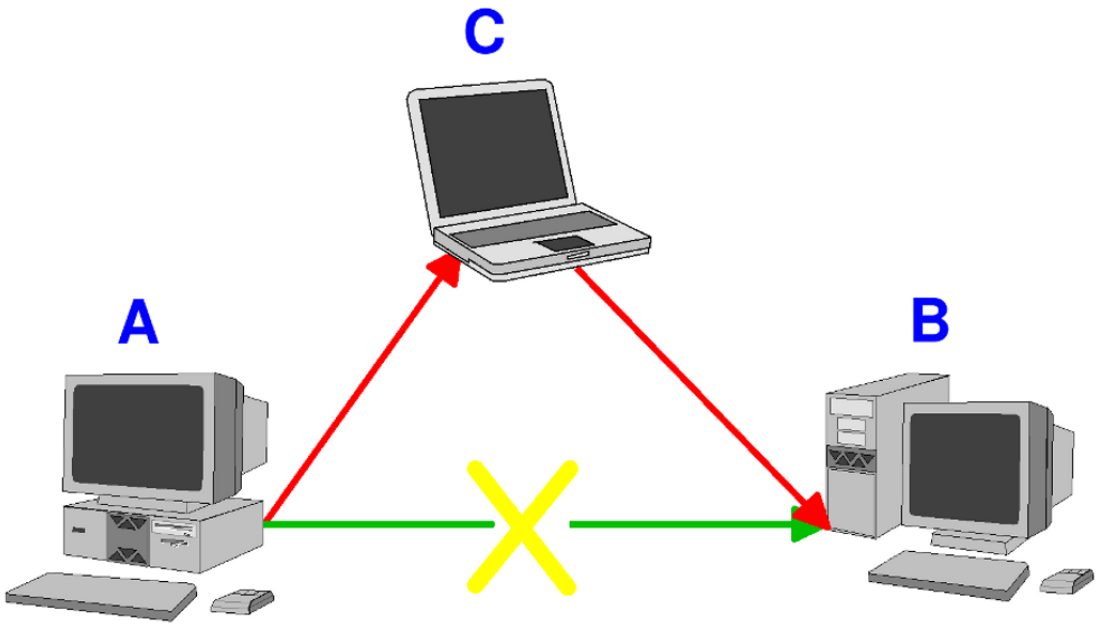
2.3 Postavení man-in-the-middle (MITM)

Při útocích v lokální počítačové síti se často využívá MITM postavení — útok ze středu. Útok samotný nijak nebezpečný není, takovéto postavení je však velice výhodné pro použití dalších technik, jako jsou například *odposlech na síti (eavesdropping)*, *útok opakováním (replay attack)*, *DoS útok* a další.

V této podkapitole se jen stručně zmíním, jak takové postavení vypadá a způsob, jak se dá dosáhnout nejjednodušeji. Složitějším útokům (*ARP Spoofing* 2.5, *Port Stealing* 2.6, *DHCP Spoofing* 2.7) jsou pak věnovány samostatné podkapitoly.

Přeloženo do češtiny, man-in-the-middle znamená *muž ve středu*. A přesně tak tato technika funguje. Útočník se dostane do středu veškeré komunikace v lokální síti a všechna odeslaná data přicházejí k němu. Ten je pak může analyzovat, zaznamenat nebo pozměnit. Takováto data pak může poslat původnímu adresátovi (a také většinou pošle), jinak by byl velice brzy objeven, protože by došlo k zastavení komunikace na síti. Zahlcení sítě může také nastat u velké sítě, kdy počítač útočníka nestíhá přeposílat pakety a nepřímo tak způsobí DoS útok (viz 2.9.4).

Jak může takové MITM postavení vypadat, je vidět na obrázku 2.2 — počítač A komunikuje s počítačem B, pakety však ve skutečnosti procházejí přes počítač C.



Obrázek 2.2: Man-in-the-middle útok

Zde jsou tři nejjednodušší, v praxi však výjimečně použitelné postupy pro získání MITM postavení:

Klienti propojení hubem Hub je pasivní síťový prvek, který jakékoliv přijaté pakety přeposílá na všechny své porty (všem připojeným klientům). Data potom přijme pouze ten počítač, kterému jsou určeny. To však lze velmi snadno obejít tzv. promiskuitním režimem síťové karty, která v tomto režimu přijímá veškerá data, i ta, která jí nejsou určena. V dnešní době však, vzhledem k poměru cena/výkon, huby ve většině případů nahradily přepínače (*switche*), případně směrovače (*routery*). Síťová karta v promiskuitním režimu je navíc v síti snadno rozpoznatelná, například pomocí programu `Sentinel`.

Nezabezpečená bezdrátová síť Do promiskuitního režimu lze uvést i některé bezdrátové síťové karty.

Speciální port přepínače Mnoho přepínačů má jeden speciální port, který slouží pro analýzu sítě. Pokud je uživatel na tento port připojen, přepínač na něj může přeposílat veškerou komunikaci.

2.4 Odposlouchávání komunikace

Cílem většiny síťových útoků je získat důvěryhodná data. Základem pro to je pasivní útok — odposlouchávání komunikace. I snaha o dosažení MITM postavení je většinou z důvodu, aby přes náš počítač procházela data, která nám nejsou určena, a my je mohli odposlechem získat.

Odposlech komunikace (resp. síťová analýza) jako takový útok není, běžně je využíván administrátory pro monitorování sítě, předcházení a případné řešení problému atd. Stejně tak programy pro odposlech dat jsou většinou určeny právě pro analýzu sítě.

(„Převzato z [1].“) Účelem síťové analýzy je dekódovat pakety a zobrazit síťový provoz v pro člověka čitelné podobě. Může se jednat o speciální zařízení s příslušným softwarem nebo o program běžící na běžném počítači. Zde je několik takových programů:

- `Wireshark` — V nekomerční oblasti v současnosti jeden z nejlepších programů. Obsahuje mnoho funkcí, grafické i textové rozhraní a podporu více než 400 protokolů.
- `Tcpdump` — Nejstarší a nejpoužívanější síťový analyzátor. Určen pro UNIX-like systémy, textové rozhraní.
- `WinDump` — Windows verze `tcpdump`.
- `Dsniff` — Sada programů, některé přimo určeny pro síťové útoky (např. odposlech hesel).
- `Ettercap` — Program určený převážně pro provádění útoků v přepínané síti.

2.5 ARP Spoofing

ARP Spoofing je v dnešní době asi nejčastější způsob, jakým se útočníci dostávají do MITM postavení. Protokol ARP byl navržen v době, kdy se na bezpečnost sítí nekladl velký důraz, neobsahuje tedy žádné bezpečnostní mechanismy. Při tomto útoku jde tedy o zneužití ARP protokolu ve prospěch útočníka.

2.5.1 Adresování v ethernetu

Každý počítač v síti má dvě adresy — MAC a IP. MAC (*Media Access Control*) adresa je uložena v paměti síťové karty a pro každou kartu je jedinečná. I když je adresa na kartě uložena v nepřepisovatelné paměti, lze ji změnit softwarově. Adresování pomocí MAC adres probíhá na druhé vrstvě ISO/OSI modelu (spojová vrstva), je tedy součástí až hlavičky odesílaného rámce. IP adresa je připojována k datům na vrstvě síťové (třetí vrstva, hlavička paketů). Tato adresa se přiřazuje pouze softwarově, není nijak vázána na konkrétní síťovou kartu. Jakmile počítač získá IP adresu, vytvoří se „neměnná“ dvojice MAC adresa - IP adresa, čehož právě využívá ARP.

2.5.2 Přepínač (*switch*)

K čemu potřebujeme dvojí adresování? Jako zprostředkovatel spojení v rámci lokální sítě většinou figuruje přepínač, který pracuje na druhé (spojové) vrstvě (existují i přepínače pracující na třetí a čtvrté vrstvě, zde je však pojmenování přepínač spíše zavádějící). Jeho směrování (respektive přepínání) pak funguje na základě CAM tabulky, ve které má přepínač uloženo, jakou MAC adresu má počítač připojený na který fyzický port. Naplnění této

tabulky pak funguje na jednoduchém principu — pokud přepínač přijme na určitý port rámec, podívá se do CAM tabulky a pokud k danému portu nemá přiřazenu MAC adresu, vytvoří v této tabulce nový záznam s daným portem a danou MAC adresou, kterou zjistí z hlavičky přijatého rámce (*MAC address source*). Stejně tak zjistí i cílovou MAC adresu rámce a podle záznamu v CAM tabulce odešle rámec na správný port. Pokud je hodnota cílové MAC adresy `ff:ff:ff:ff:ff:ff`, jedná se o broadcastovou doménu a přepínač odešle rámec na všechny porty (viz ARP).

2.5.3 ARP (*Address Resolution Protocol*)

ARP je poměrně jednoduchý protokol, jehož úkolem je pomocí IP adres zjišťovat MAC adresy — viz RFC 826 [2]. Dotaz s danou IP adresou rozešle všem počítačům v dané broadcast doméně (MAC adresa `ff:ff:ff:ff:ff:ff`). Jakmile počítač přijme tento paket, zjistí, zda je zjišťována jeho adresa. Pokud ano, odešle odpověď se svojí MAC adresou, jinak paket zahodí. Aby se počítač nemusel dotazovat při každé komunikaci, odpovědi si ukládá do lokální dočasné paměti (*cache*). Zde je ukázka, jak taková ARP dočasná paměť vypadá:

```
Address          HWtype  HWaddress           Flags Mask  Iface
192.168.1.100    ether   00:14:A5:E4:12:62   C           eth0
192.168.1.8      ether   00:18:F3:3E:D2:FF   C           eth0
192.168.1.5      ether   00:13:02:7D:A1:2C   C           eth0
```

2.5.4 Princip útoku

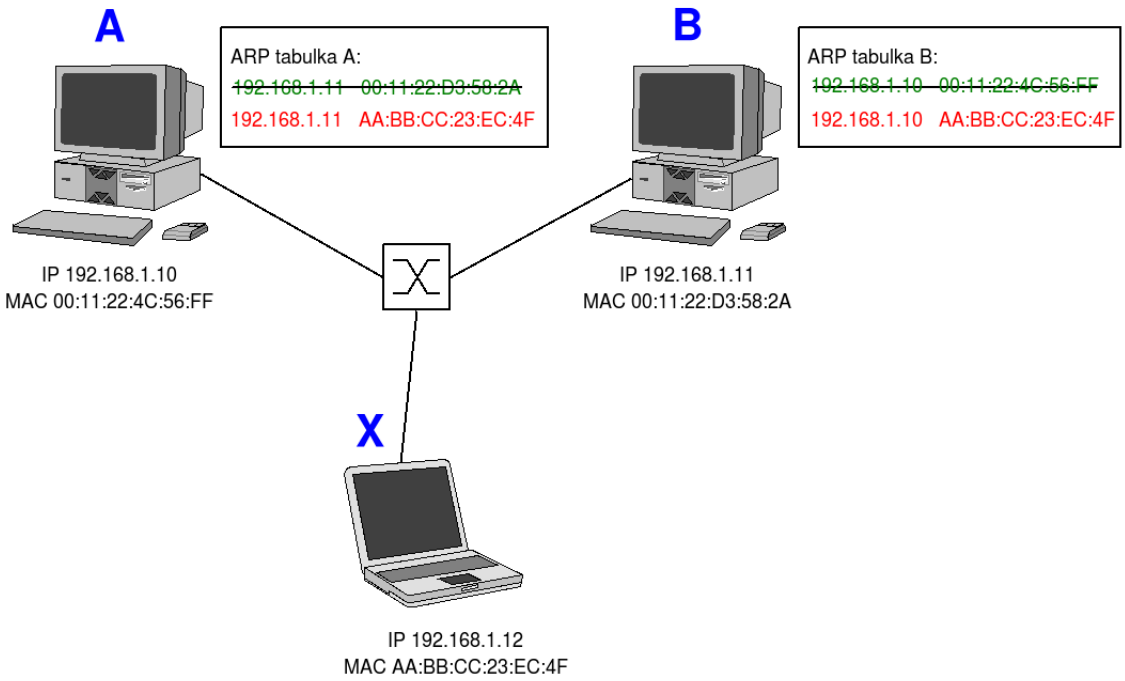
Kromě názvu útoku ARP Spoofing se v literatuře často setkáme s pojmenováním ARP Cache Poisoning. Z praktického hlediska jsou potom oba tyto názvy správné — *spoofing* značí útoky, při kterých se útočník vydává za někoho jiného, *cache poisoning* potom znamená změnu údajů v dočasné paměti (*cache*). Při tomto útoku děláme obojí — změnou údajů v dočasné paměti oběti se vydáváme za jiného uživatele sítě.

Princip útoku spočívá v tom, že útočník může odeslat oběti speciálně upravený ARP paket, ve kterém pro určitou IP adresu změní její MAC adresu — například na svoji. Pokud potom takto napadený počítač na danou IP adresu odešle paket, do jeho hlavičky uvede změněnou MAC adresu a přepínač proto tento paket odešle na počítač útočníka. Jak tento útok může vypadat v praxi, je zobrazeno na obrázku 2.3 („obrázek inspirován z [15]“). Mějme počítač A, který chce komunikovat s počítačem B. Před tím však dojde od útočníka (počítač X) k útoku ARP Spoofing, při kterém změní ARP tabulky obou počítačů a komunikace tak bude procházet přes jeho počítač.

2.5.5 Možnosti zabezpečení

Detekce

Jedním z důvodů, proč je tento útok tak oblíbený, je jeho složitá detekce. Pokud například útočník zaútočí pouze na jeden počítač v síti, je opravdu těžé ho odhalit. Pro zjištění tohoto útoku se tak používají monitorovací programy instalované přímo na koncové počítače. Pro Linux je to například program `ARP Watch` a pro Windows program `XARP`. Tyto programy potom monitorují a analyzují síťový provoz nad protokolem ARP a informují uživatele při podezření na útok.



Obrázek 2.3: ARP poisoning

Obrana

Záznamy do ARP tabulky počítače mohou být přidány ručně. Potom se jedná o tzv. statické záznamy, které jsou uloženy natrvalo a přijatými ARP pakety se nemění. Změna ARP tabulky tohoto počítače je pak pomocí ARP Spoofingu nemožná.

Druhou možností je zabezpečení na přepínači. Ten může na základě vytvořené tabulky s páry IP adresa — MAC adresa kontrolovat, zda ARP pakety mají tyto adresy nastaveny správně. Tento způsob zabezpečení se nazývá DAI (*Dynamic ARP Inspection*).

2.6 Port Stealing

U tohoto typu útoku útočník využívá principu, jakým přepínač vytváří záznamy ve své CAM tabulce a následně směruje jednotlivé pakety (viz podkapitola o přepínači 2.5.2). Tento útok by se kromě pojmenování „Port Stealing“ neboli „Kradení portů“ dal také nazvat „Switch CAM Poisoning“ neboli „Otrávení CAM tabulky přepínače“.

Jakým způsobem tedy „ukradnout port“? Musíme pozměnit záznam v CAM tabulce přepínače tak, aby měl k portu, ke kterému je připojen počítač útočníka, přiřazenu MAC adresu oběti. Toho lze dosáhnout tím způsobem, že z počítače útočníka pošleme jakýkoliv paket, který bude mít v hlavičce změněnou zdrojovou MAC adresu na adresu počítače oběti. Výhoda a současně i nevýhoda přepínače je potom to, jakým způsobem řeší konflikt v adresách — záznamy v CAM tabulce vždy upraví podle naposledy přijatého rámce. Výhoda pro útočníka tedy je, že mu stačí poslat jeden paket se změněnou zdrojovou adresou a přepínač si podle toho své záznamy aktualizuje. Nevýhoda potom je, že stačí, aby na přepínač přišel jeden rámec od počítače oběti a ten vrátí záznamy zpět do původního

stavu.

Pokud je tedy cílem útočníka dostat se tímto typem útoku do MITM postavení, je potřeba falešné pakety odesílat na přepínač opakovaně. A problém je tu ještě jeden, přeposlání dat původnímu adresátovi — počítači oběti. Jako útočník je totiž nemůžeme rovnou odeslat bez provedení změn, protože jakmile by dorazila na přepínač, ten by je podle záznamu v CAM tabulce chtěl poslat zpátky nám (většina přepínačů se však chová tak, že pokud je cílový port stejný jako zdrojový, rámec zahodí). Nejdříve tedy musíme vrátit do původního stavu záznamy na přepínači. Nejjednodušeji to provedeme tak, že přestaneme posílat naše falešné pakety a místo toho pošleme paket ARP Request. Jakmile pak na přepínač dorazí paket ARP Reply od počítače oběti, upraví si podle něj záznam v tabulce zpět na původní port.

Právě z těchto důvodů bude pro účinné provedení tohoto útoku v praxi nutné využít nástroje, který bude kradení a navracení portů provádět automatizovaně. Potom se však také jedná o nebezpečný a těžko odhalitelný útok.

2.6.1 Možnosti zabezpečení

Protože se jedná o útok na přepínač, je prakticky jedinou možností obrana přímo na přepínači. Zabezpečení je pak podobné jako v případě ARP Spoofingu (2.5.5, obrana na přepínači), kdy má přepínač v tabulce uloženy i kombinace MAC adres a portů. Pokud pak přijdou data s falešnou MAC adresou, jsou přepínačem zahozena.

2.7 DHCP Spoofing

V lokálních sítích bývá zvykem, že se síťové nastavení koncových zařízení provádí automaticky, pomocí DHCP (*Dynamic Host Configuration Protocol*). Hlavním úkolem DHCP serveru je přidělování IP adres, vždy z určitého rozsahu a na určitou dobu. Pokud dokáže útočník v počítačové síti nasadit vlastní DHCP server, může klientům přidělovat IP adresy podle vlastní potřeby. Větší nebezpečí je však v dalším síťovém nastavení, které DHCP server klientům poskytuje, jako je IP adresa výchozí brány a IP adresa DNS serveru.

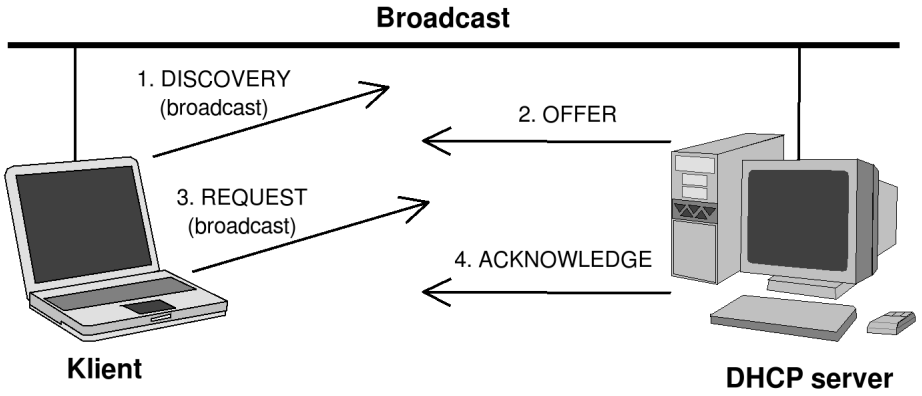
2.7.1 DHCP

Jakmile se počítač poprvé připojí do sítě, na něm běžící DHCP klient odešle na adresu broadcastu paket DHCP Discover. Pokud pak tento paket zachytí naslouchající DHCP server, odpoví paketem DHCP Offer, ve kterém nabídne počítači IP adresu a síťové nastavení. Pokud klient toto nastavení příjme, odešle paket DHCP Request. DHCP server si do souboru uloží záznam o přidělení IP adresy a vše potvrdí paketem DHCP ACK. DHCP server propůjčuje adresu vždy na určitou dobu, tzv. *lease-time*, nastavený v konfiguraci serveru. Pokud chce klient nastavení používat déle, musí si o něj po uplynutí této doby znovu požádat.

Přesný popis tohoto protokolu je popsán v RFC 2131 [7].

2.7.2 Princip útoku

Není žádné pravidlo, které by určovalo, na jaké adrese nebo kolik může v jedné lokální síti naslouchat DHCP serverů. Pokud je tedy jeden z počítačů v síti v moci útočníka, může na něm tuto službu spustit. To ještě ale nemusí znamenat, že nově připojený počítač dostane nastavení od falešného DHCP serveru, protože:



Obrázek 2.4: DHCP

- Pokud se počítač připojuje do stejné sítě, DHCP klient si většinou pamatuje, na jaké adrese naslouchá DHCP server a posílá paket DHCP Request přímo na jeho adresu
- Jestliže je požadavek vyslán na adresu broadcastu, příjme počítač první nabídku, která mu přijde. Právo přidělit danému počítači nastavení tedy „vyhrává“ DHCP server, který byl v dané chvíli nejrychlejší

Samotné spuštění DHCP serveru by tedy útočníkovi nemuselo stačit. Řešení spočívá ve vyřazení stávajícího DHCP serveru. Každý takový server může přiřadit nově připojenému počítači IP adresu pouze v případě, že má ještě nějaké volné.

Mějme například síť: 192.168.1.0/24. Adresy 192.168.1.0 a 192.168.1.255 jsou vyhrazené, samotný DHCP server pak má adresu 192.168.1.1. Rozsah, ze kterého může udělovat IP adresy, je tedy nastaven na 192.168.1.2–192.168.1.254, což je 253 adres. V dané lokální síti tedy může být najednou připojeno maximálně 253 počítačů. Pokud DHCP server přidělí všechny IP adesy ze svého rozsahu, přestane odpovídat na dotazy DHCP Discover (případně DHCP Request), protože už nemá co nabídnout.

A přesně tohoto může útočník využít. Zahltí DHCP server požadavky o přidělení IP adresy, až vyčerpá všechny adresy z jeho rozsahu a ten tím pádem přestane odpovídat. Potom může spustit svůj vlastní DHCP server a síťové nastavení bude klientům přicházet od něj.

2.7.3 Možnosti zabezpečení

(„Podkapitola převzata z [13].“)

Detekce

Pro detekci falešného DHCP server je potřeba monitorovat síťový provoz, v tomto případě hlavně pakety DHCP Discover a DHCP Offer. Pokud máme možnost sledovat komunikaci po celé síti, můžeme využít pasivní detekce, při které pouze naslouchámе a kontrolujeme DHCP požadavky a odpovědi. Můžeme tak podle počtu odpovědí na paket DHCP Discover zjistit, že se v síti nachází více jak jeden DHCP server. Stejně tak budeme kontrolovat, zda pakety DHCP Offer mají zdrojovou IP a MAC adresu legitimního serveru.

Druhou možností je detekce aktivní. Spočívá v tom, že vyšleme všesměrový požadavek DHCP Discover a zkontrolujeme, zda nám přišla pouze jedna odpověď a je od našeho legitimního DHCP serveru.

Obrana

Jednou z možností obrany proti DHCP Spoofingu je nepoužívat v síti DHCP server, ale u koncových zařízení nastavit síťové informace ručně. Nastavením vysokého parametru `lease-time` (touto hodnotou server určuje, na jakou dobu počítači danou adresu propůjčuje) je možné omezit DHCP požadavky od klientů a tím tak zmenšit riziko, že případně dostanou odpověď od falešného DHCP serveru.

Proti tomuto útoku se lze bránit také přímo na přepínači. Jedná se o tzv. DHCP Snooping, při kterém jsou porty na přepínači rozděleny na důvěryhodné (*trusted*) a nedůvěryhodné (*untrusted*). Služby jako DHCP nebo DNS potom mohou běžet pouze na počítači připojeném k důvěryhodnému portu. Pokud pak přijde na přepínač odpověď od DHCP serveru na nedůvěryhodný port, ten ji zahodí.

2.8 DNS Spoofing a Pharming

DNS systém poskytuje služby, které uživatelům počítačové sítě, hlavně potom sítě Internet, výrazným způsobem ulehčují práci. Ani tento systém se však nevyhl útokům. V současnosti se navíc jedná o jeden z největších bezpečnostních problémů Internetu a právě zabezpečení DNS je dnes velmi diskutované téma různých konferencí (u nás například LinuxAlt 2008, Adam Tkáč: Bezpečnost DNS DNSSEC).

Útoků na DNS systém existuje celá řada. Nejnebezpečnějším je potom útok, jehož výsledkem je tzv. Pharming. Co je Pharming, co je DNS a k jakým útokům je náchylný je náplní této podkapitoly.

2.8.1 DNS

Celé téma DNS je poměrně dost rozsáhlé a určitě by vydalo na více než pár odstavců. V této podkapitole tedy uvádím pouze informace, které se týkají útoků. Jak tento protokol funguje je popsáno v RFC 1035 [6].

Pro jednoznačné adresování počítačů v síti slouží tzv. IP adresy. Jedná se o 32-bitová čísla, pomocí nichž se počítače na síťové vrstvě identifikují. Pro člověka je však takovýto způsob adresování nepřirozený a těžko by si zapamatoval větší množství adres. Z toho důvodu vznikl DNS (*Domain Name System*), který slouží k překladu IP adres na doménová jména a naopak (DNS poskytuje i další služby, jako například informace pro doručování pošty či IP telefonii). Jedná se o službu na aplikační vrstvě (číslo portu je 53) typu klient-server, pro přenos požadavků pak používá protokol nazvaný DNS protokol. Klient pošle na server požadavek (například pro překlad doménového jména na IP adresu) a DNS server, pokud zná na tento požadavek odpověď, odešle ji klientovi. Pro přenos takovýchto dotazů se používá především transportní protokol UDP, klient tedy nemusí se serverem navazovat a ukončovat spojení, jednoduše odešle požadavek a čeká na odpověď. Protokol TCP je potom použit v případě, kdy je odpověď příliš dlouhá (více než 512 bytů).

V rámci sítě Internet potom vznikla celá hierarchická struktura DNS serverů, které se o překlad adres starají. Adresy v síti Internet jsou rozděleny na domény a o každou takovouto

doménu se stará tzv. autoritativní DNS server, který obsahuje informace pro překlad adres dané domény, případně adresy na autoritativní servery subdomény. Na vrcholu tohoto stromu jsou potom tzv. kořenové jmenné servery, které mají informace o autoritativních serverech pro domény nejvyšší úrovně.

2.8.2 Útoky na DNS

Cílem útoků na DNS je podvržení odpovědi serveru, vedoucí většinou k tzv. Pharmingu (viz podkapitola Pharming 2.8.3). Typů těchto útoku je několik, já se zde (stejně jako v celé této publikaci) zaměřím na útoky v rámci lokální počítačové sítě.

DNS Spoofing typu MITM

Tento útok využívá MITM postavení (viz podkapitola 2.3). Pokud veškerá komunikace prochází přes náš počítač (z pohledu útočníka), stačí nám DNS dotaz odchytnout a buď přeposlat na legitimní DNS server, nebo odeslat námi vytvořenou odpověď.

DNS Spoofing s využitím DHCP Spoofingu (nasazení falešného DNS serveru)

O tomto typu útoku jsem se zmínil již v předchozí podkapitole DHCP Spoofing (2.7). Důležitá je potom ta informace, že kromě IP adres DHCP server klientům posílá také adresu DNS serveru. Pokud jsme tedy v síti schopni provést útok DHCP Spoofing, můžeme nově připojeným klientům vnutit náš DNS server a mít tak pod kontrolou všechny DNS dotazy a odpovědi.

Útoky na DNS v rámci sítě Internet

Oba výše zmíněné útoky jsou v praxi proveditelné pouze v rámci lokální sítě, MITM postavení i DHCP Spoofing jsou pro ně specifické. V rámci sítě Internet jsou pak tyto útoky většinou typu *cache poisoning* — cílem útočníka je změna informací přímo na legitimním DNS serveru.

2.8.3 Pharming

(„Tato podkapitola byla inspirována článkem [17].“) Pharming je novodobý typ útoku, jehož cílem je získání soukromých údajů od uživatelů. Dosahuje toho podvržením webové stránky, na které pak důvěřivý uživatel tyto údaje sám zadá.

Stejného principu využívá i starší typ útoku — Phishing. Rozdíl mezi těmito útoky je v tom, jakým způsobem uživateli podvrženou stránku „vnutí“. Phishing k tomu využívá tzv. sociální inženýrství. Jedná se o techniku, při které je uživatel-oběť slovně přesvědčen, aby sám na stránku s podvrženou adresou vstoupil a zadal tam svoje údaje. V praxi se jedná například o podvodný (na první pohled však věrohodný) e-mail, ve kterém je oběť požádána, aby se přihlásila ke svému internetovému bankovnímu účtu. Součástí je pak odkaz, údajně vedoucí na tento přihlašovací formulář. Takto podvržená stránka pak může být po vzhledové stránce nerozeznatelná od originální, po zadání přihlašovacích údajů se však dostanou do rukou útočníka.

Útok nazvaný Pharming je pak ještě sofistikovanější a proto také nebezpečnější. K podvržení webové stránky totiž dojde i při zadání její adresy do prohlížeče. Není tedy potřeba žádných podvodných e-mailů.

Pharming je vlastně dotažení útoku DNS Spoofing. Uživatel zadá doménovou adresu webové stránky, na DNS dotaz o překlad tohoto jména však prohlížeč oběti obdrží útočníkem podvrženou IP adresu. Výsledkem pak je, že se oběť ocitne na falešné webové stránce, což však nemusí vůbec poznat. Právě díky této transparentnosti z pohledu oběti je tento útok tak nebezpečný.

2.8.4 Možnosti zabezpečení

Oběma útokům typu DNS Spoofing uvedených výše vždy předchází jiný útok. Základem zabezpečení je tedy zabránit útočníkovi dostat se do MITM postavení nebo nasadit falešný DHCP server. Je také možné využít tzv. důvěryhodného portu přepínače, tak jak bylo vysvětleno v podkapitole o možnostech zabezpečení proti DHCP Spoofingu (2.7.3).

Pokud se nám nepodaří zabránit DNS Spoofingu, obrana proti Pharmingu je pak poměrně složitá. Cílem tohoto útoku je téměř vždy získat důvěrné informace, proto pokud vstupujeme na stránku, kde budeme takovéto informace odesílat, je dobré zkontrolovat adresu stránky, přečíst si případný certifikát apod.

2.9 Denial of Service útoky

Dne 9. února 2000 byly na několik hodin vyřazeny z provozu servery Yahoo!, CNN.com, eBay a eTrade. Jednalo se o DDoS útok (*Distributed Denial of Service*), z velké části o tzv. falšované „šmoulí“ útoky (*smurf attacks*, viz dále).

> Útoky typu DoS (*Denial of Service* — odepření služby) stojí komerční firmy miliony dolarů ročně a jsou vážným problémem libovolného systému nebo sítě. Zmíněné ztráty jsou způsobeny nedostupností systémů, ztrátou tržeb a nutností analyzovat a napravit vzniklý problém. Důsledkem DoS útoku je v podstatě narušení nebo kompletní znepřístupnění služby pro její běžné uživatele. Jedná se tedy téměř vždy o zlý úmysl a útok často nevyžaduje skoro žádné vědomosti, protože vhodné nástroje jsou běžně dostupné. [4]

(„Většina následujícího textu této podkapitoly byla převzata z [4] a [14].“)

2.9.1 Útočníci

Abych navázal na předchozí odstavec této podkapitoly, zmíním nejdříve tři odlišné skupiny útočníků, provádějících tyto útoky:

Script kiddies Za většinou (D)DoS útoků stojí tzv. script kiddies. Tento výraz pochází z Ameriky a je poměrně výstižný. Jsou takto označováni lidé (většinou mladí — proto kiddies), kteří dané problematice příliš nerozumí, ale v rukou mají hotový skript (program), který prakticky celý útok provede za ně. V tomto případě jde při provádění útoku většinou o zábavu, případně o zviditelnění — většími útoky se jistě budou zabývat média. Právě díky těmto typům útočníků jsou však tyto programy (a s tím související útoky) velmi rozšířeny a ve většině případů také zabezpečeny.

„Profesionální hackeři“ Na druhé straně stojí útočníci, kteří tyto útoky provádějí za účelem zisku. Žádná společnost poskytující nějaký druh webových služeb nestojí o vyřazení serveru z provozu. Pokud se navíc jedná o velké mezinárodní společnosti

(jako například ty, které se staly cílem útoku v únoru roku 2000), mohou jejich ztráty dosáhnout obrovských sum, nemluvě o jejich dobrém jméně. A právě toho využívají tito útočníci (většinou se jedná o organizované zločinecké spolky) a vydírají společnosti. Pokud jim zaplatí, nic se nestane, pokud ovšem ne, začnou útočit...

Nechtěné Neboli *Unintentional DoS*. Přesně jak říká název, jedná se o útok nechtěný. Mějme například server CNN.com, který denně navštíví statisíce uživatelů. Tento zpravodajský server je na to zvyklý a tudíž i připravený. Co když ale na svou stránku umístí odkaz (například odkud daný článek čerpal), který vede na server, který na takou zátěž není stavěný? Takto odkazovaný server je pak kvůli obrovskému nárustu návštěvnosti často nedostupný nebo se úplně zhroutí. („Převzato z [14]“)

2.9.2 Útoky zahlcením

Při útocích zahlcením (*DoS Flood*) se útočník snaží oběti zablokovat přístup do sítě. Dosáhnout toho může například zahlcením přenosového pásma nebo vyčerpáním síťových prostředků počítače oběti.

ICMP Flood

Co jsou to ICMP pakety a k čemu se využívají je již napsáno v podkapitole 2.2.1. Spolu se SYN Flood se také jedná o jeden z nejčastějších DoS útoků, hlavně pro jeho jednoduchost a účinnost.

Běžným použitím programu `ping cílová_adresa` však oběť příliš nezatížíme, program totiž mezi jednotlivými odeslanými ECHO_REQUEST pakety čeká jednu vteřinu. Parametrem `-i` (interval) však tuto hodnotu můžeme změnit, v našem případě především zmenšit. Jako běžný uživatel až na hodnotu 0.2 sekundy, jako super-uživatel pak až na hodnotu 0, při které program odesílá pakety jak nejrychleji dokáže. S patřičnými právy můžeme použít přepínač `-f` (*flood ping*), který k takovému DoS útoku přímo vybízí — program odesílá požadavky s nulovým intervalem, přičemž za každý odeslaný paket vypíše na standartní výstup tečku a při přijetí odpovědi jednu tečku smaže. Můžeme tak pěkně v reálném čase sledovat, jak stíhá/nestíhá cílová stanice odpovídat. V příkladu je navíc přepínačem `-s` zvětšena velikost paketu, aby byl útok ještě účinnější:

```
# ping -f 192.168.1.6 -s 65000
PING 192.168.1.6 (192.168.1.6) 65000(65028) bytes of data.
.....................................................................
.....................................................................
.....................................................................
.............................
--- 192.168.1.6 ping statistics ---
257 packets transmitted, 4 received, 98% packet loss, time 3079ms
rtt min/avg/max/mdev = 70.668/110.936/149.705/28.990 ms, pipe 13,
ipg/ewma 12.030/88.047 ms
```

Jak je vidět, za něco málo přes tři sekundy jsme odeslali 257 paketů a odpovědi přišly pouze čtyři — počítač ani nestíhá odpovídat. A jakmile se trochu uvolní jeho zdroje, okamžitě je zabere pro odpověď. Po dobu, po kterou necháme tento program běžet, bude cílový počítač prakticky neschopný komunikovat po síti.

Smurf attack

Výše provedený útok má však několik nevýhod. Za prvé, jsme jeho přímým účastníkem, naše šířka pásma je totiž zabírána námi odesílanými pakety a navíc i odpověďmi. Pokud je navíc cíl připojený k síti rychlejší linkou než útočník, ten pak nemá moc šanci jeho pásmo zahltit.

Zde pak mohou přijít na řadu tzv. *smurf attacks* neboli „šmoulí“ útoky. Ty využívají toho, že můžeme změnit zdrojovou IP adresu ICMP paketu. Tu nastavíme na adresu oběti a takto změněné dotazy odešleme na několik počítačů. Všechny odpovědi pak budou chodit právě na adresu oběti. Útok můžeme navíc umocnit tím, že neposíláme dotazy na jednotlivé počítače, ale na adresu celé sítě (broadcast). Spolu se změněnou adresou zdroje je pak oběť vystavena veškerým odpovědím od všech počítačů v dané síti.

SYN Flood

Tento typ DoS útoku zneužívá ustavení spojení u TCP komunikace, viz obrázek 2.1. Jakmile oběti přijde paket s příznakem SYN, předpokládá, že s ním chce klient ustavit spojení. Alokuje si tedy zdroje pro toto spojení, odešle zpět paket s příznaky SYN + ACK a čeká na paket s příznakem ACK, který by spojení dokončil. Ten již však útočník nepošle. Po určité době oběť opakuje dotaz SYN + ACK a pokud stále nepřichází odpověď, celou inicializaci zruší a uvolní alokované zdroje.

Na takovémto chování není nic špatného a z hlediska běžného provozu je naprosto v pořádku. Pro útočníka jsou to však otevřené dveře pro aplikování útoku s názvem SYN Flood. Nejdůležitější jsou zde pak tyto dvě výše zmíněné informace: „alokuje si tedy zdroje“ a „po určité době“. Útočník tedy jednoduše zahltí oběť pakety pro ustavení spojení (příznak SYN), kdy pro každý takovýto přijatý paket si oběť alokuje své zdroje (záznam v tabulce). Tyto zdroje samozřejmě nejsou nekonečné. A i díky malé velikosti těchto TCP řídících paketů (42 bytů) jich je útočník za určitý čas schopen poslat mnohokrát více, než jich oběť zruší pro neaktivnost. Po vyčerpání veškerých prostředků, které má oběť vyčleněny pro inicializaci, tak dojde k odepření služby — server není schopen přijmout žádné další (legitimní) spojení. Stejně jako v předchozím typu útoku může útočník změnit zdrojovou adresu paketu a není tak „obtěžován“ pakety SYN + ACK - odpovědi od oběti.

TCP, UDP Flood

Jedná se o další útoky využívající zahlcení oběti určitým typem paketů — jak paketů bezstavového transportního protokolu UDP, tak stavového protokolu TCP. Ve druhém případě můžeme paketům nastavovat různé příznaky, jako například SYN (viz SYN Flood), RST, ACK, FIN, případně jejich kombinace. Reakce na tyto příznaky jsou popsány v protokolu, ne vždy jsou však dodržovány a různé operační systémy na ně mohou reagovat jinak.

2.9.3 Útoky využívající chyb v programech

K odepření služby může dojít také po chybě určitého programu, operačního systému či logice hardwaru. Při těchto typech útoků se útočník snaží o umělé vyvolání takovéto chyby. Nástrojům, který chyby vyvolávají, se říká *exploity*. Většina takovýchto chyb je obvykle rychle odhalena a také opravena, což je i případ dvou následujících útoků.

Teardrop

Maximální velikost IPv4 paketu je 65 535 bytů (je to dáno 16-ti bitovým prostorem v hlavičce paketu pro uložení jeho velikosti). Pro fyzický přenos paketů po síti se však používá tzv. MTU (*Maximum Transmission Unit* — maximální přenosová jednotka), jejíž maximální velikost pro Ethernet je 1500 bytů. Pokud tedy chceme poslat paket větší než je maximální velikost MTU, musíme ho rozdělit a poslat po částech. Tomu se říká fragmentace a chyby v její implementaci využíval útok *teardrop*, neboli slza.

Útočník poslal na adresu oběti speciálně fragmentovaný paket, kdy se druhá část fragmentu překrývá s první (druhá část začíná v první). Problém potom nastal na straně cílového operačního systému. Ten předpokládal, že druhá část fragmentovaného paketu bude začínat až za koncem první. Takhle mu najednou vyšla při napojování druhé části paketu adresa jeho začátku záporná (resp. obrovská kladná) a došlo k chybě.

Ping of death

Ping of death (ping smrti) využíval podobně jako *teardrop* fragmentování paketů. Jak je zmíněno výše, maximální velikost paketu je 65 535 bytů. Běžnou komunikací nemohla nastat situace, že by vznikl paket větší velikosti. Tato situace tedy nebyla v operačním systému nijak ošetřena. Při provádění útoku *ping of death* však útočník poslal paket ICMP ECHO_REQUEST větší velikosti, než byla adresovatelná. Jak je to možné? Útočník samozřejmě nemohl vytvořit paket větší. Mohl však posílat jednotlivé fragmenty, které na straně příjemce po sestavení větší paket vytvořily, což vedlo ke zhroucení systému.

2.9.4 MITM

Při postavení MITM (viz kapitola 2.3) jdou všechny pakety přes počítač útočníka. Je tedy čistě na něm, jestli tyto data přepošle dál (původnímu adresátovi) nebo ne. Pokud tak neučiní, může některým (případně všem) uživatelům v dané síti úplně zablokovat síťovou komunikaci nebo odepřít určité služby (nebude například přeposílat pakety na port FTP apod.)

2.9.5 Možnosti zabezpečení

Proti útokům typu zahlcení je obrana poměrně složitá. Přicházející komunikaci je totiž potřeba odfiltrovat ještě před tím, než k našemu počítači dorazí — nejlépe na hraničním směrovači. Používáním správně nastaveného firewallu můžeme dopad útoku také zmírnit. Například při útoku ICMP Flood nebudeme na takové množství ICMP paketů odpovídat, čímž dojde pouze k polovičnímu zahlcení linky.

Obrana proti útokům využívajících chyb je potom v pravidelné aktualizaci programů a operačního systému. Například v Linuxu byl útok *teardrop* opraven již v jádrech 2.0.x a 2.2.x, v dnešní době je tedy prakticky nepoužitelný.

2.10 WEP Crack

V posledních asi deseti letech se v oblasti počítačových sítí velmi rozšířila nová technologie, nazývána Wi-Fi. Jedná se o bezdrátové propojení počítačů, dříve určené pro připojení klientů do lokální sítě, později začalo být využíváno i k propojení na delší vzdálenosti,

například k síti Internet. Wi-Fi neboli WLAN (*Wireless LAN*) vychází ze specifikace IEEE 802.11.

Největším bezpečnostním problémem bezdratových sítí je fakt, že veškerá data jsou přenášena pomocí elektromagnetických vln vzduchem. K odposlechu takto přenášených dat potom stačí být v dosahu těchto vln — oproti drátovým sítím tedy odpadá nutnost být k síti fyzicky připojen. Základem zabezpečení těchto sítí je proto šifrování přenášených dat. Nejjednodušší metodou je potom použití protokolu WEP, která, přestože je poměrně snadno napadnutelná, je i v dnešní době (zvláště u malých či domácích sítí) nejpoužívanější.

WEP je šifrovací protokol sloužící k zabezpečení bezdrátových lokálních sítí, uvedený v roce 1999. Sama zkratka WEP potom značí: „*Wired Equivalent Privacy*“, což volně přeloženo do češtiny znamená: „Zabezpečení na úrovni drátových sítí“. Na šifrování dat používá šifrovací algoritmus RC4 s jedním tajným klíčem o délce 40 bitů nebo 104 bitů, jehož spojením s 24 bitovým inicializačním vektorem vzniká 64 bitový nebo 128 bitový klíč (někteří výrobci podporují i klíč 256 bitový), nazývaný WEP klíč.

Největší slabost tohoto protokolu je potom právě v použití šifrovacího algoritmu RC4. Tyto slabosti byly popsány autory *Scott R. Fluhrer, Itsik Mantin* a *Adi Shami* v dokumentu *Weaknesses in the Key Scheduling Algorithm of RC4* v roce 2001 ([8]), ve kterém také popisují, jak se díky tomu dá při dostatečném množství inicializačních vektorů (4 až 6 milionů) WEP klíč prolomit. V roce 2004 pak hacker s přezdívkou *KoReK* tento útok vylepšil, kdy na získání klíče stačí několikrát méně vektorů (0,5 až 2 miliony). Protože v takovémto „vylepšování“ pokračovalo mnoho odborníků i v následujících letech, prolomení WEP klíče je dnes s využitím techniky *Packet Injection* otázkou maximálně pár minut — metoda PTW, viz dokument *Breaking 104 bit WEP in less than 60 seconds* od autorů *Erik Tews, Ralf-Philipp Weinmann a Andrei Pyshkin* [3].

2.10.1 Možnosti zabezpečení

Protože je slabost WEP klíče dobře známá, pro zabezpečení bezdrátových sítí existují i další bezpečnostní řešení. Jednou z možností je přidat filtrování MAC adres. Na přístupovém bodě je administrátorem vytvořen seznam MAC adres počítačů a připojit se může pouze uživatel, jehož adresa je na seznamu.

Protože však změna MAC adresy není složitá, bezpečnějším řešením je použití metod WPA nebo WPA2.

WPA Hlavní změna proti WEP spočívá v pravidelné výměně klíče mezi přístupovým bodem a klienty. Autentizace potom může spočívat buď pomocí hesla (WPA-PSK), nebo uživatelského jména a hesla (WPA-EAP).

WPA2 Poskytuje nejlepší zabezpečení, hlavně co se podnikových sítí týká. Pro šifrování dat používá algoritmus AES a pro autentizaci protokoly EAP-TLS nebo PEAP. Ověřování uživatelů potom probíhá na serveru (tzv. RADIUS server, viz. 802.1x, [5]).

2.11 SSH downgrade attack

Jako poslední ukázku útoků v síti jsem vybral tento poměrně nenápadný, zajímavý a při nezabezpečení také velmi účinný útok na protokol SSH, nazvaný *SSH downgrade attack*.

2.11.1 SSH

SSH (*Secure Shell*) je síťový protokol, sloužící k zabezpečené komunikaci dvou koncových zařízení (klient a server), popsán v RFC 4251 [10]. Přenášená data protokol SSH šifruje pomocí asymetrické kryptografie, využívá tedy principu veřejného a privátního klíče (více o tomto tématu například v knize: *Stallings, William, Network security essentials: applications and standards, 2007* [9]). Protokol vyvinul v roce 1995 Tatu Ylönen (Helsinki University of Technology, Finland) jako náhradu za protokoly *rlogin*, *TELNET* a *rsh*, které posílají přihlašovací údaje v nezabezpečené formě. V roce 1996 pak byla představena vylepšená verze tohoto protokolu a protože jsou tyto verze nekompatibilní, byla původní verze nazvána SSH-1 a novější verze SSH-2.

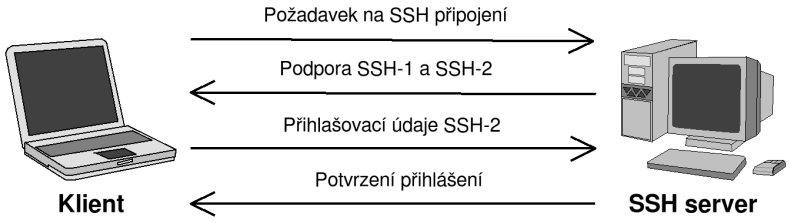
2.11.2 Navázání spojení

Celá komunikace začíná u klienta, který pošle serveru požadavek na ustavení spojení přes SSH. Pokud je na serveru (na daném portu) SSH služba aktivní, pošle jako odpověď klientovi jeden z následujících řetězců:

- SSH-2.xx — Server podporuje pouze SSH-2
- SSH-1.99 — Server podporuje SSH-1 i SSH-2
- SSH-1.51 — Server podporuje pouze SSH-1

U každé odpovědi je za pomlčkou napsáno, co který z řetězců pro klienta znamená. První a třetí odpověď je zřejmá. Pokud server podporuje pouze jednu verzi protokolu, není možné použít jinou. Jestliže klient daný protokol nepodposruje, není možné se k server připojit. Když server odpoví řetězcem „SSH-1.99“, znamená to, že podporuje obě verze SSH a je na klientovi, kterou z nich si vybere.

Klient tedy odpoví serveru, že přijímá připojení na vybraném protokolu a zašle mu přihlašovací údaje. Na základě přihlašovacích údajů pak server klienta autorizuje a ustaví SSH spojení.



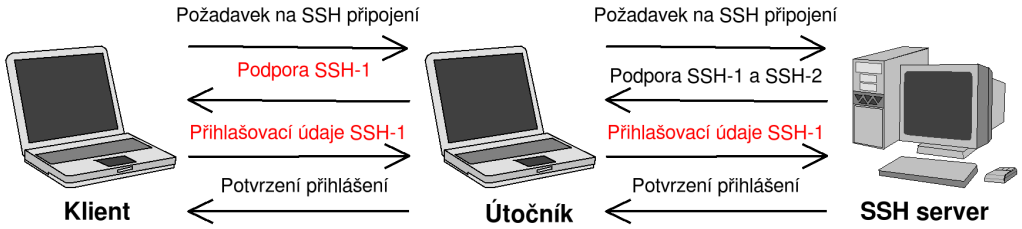
Obrázek 2.5: SSH připojení

2.11.3 Princip útoku

Z pohledu útočníka je pak nejdůležitější tento fakt — starší verze SSH-1 používá na rozdíl od SSH verze 2 slabou a prolomitelnou autentizační metodu. Cílem tohoto útoku je tedy přesvědčit obě komunikující strany, aby pro komunikaci použily protokol SSH-1 — proto také název útoku: „*SSH downgrade attack*“. K tomu se musíme v první řadě dostat do komunikace mezi klientem a serverem. Jak toho dosáhnout je popsáno například v podkapitole ARP Spoofing 2.5 nebo Port Stealing 2.6.

Pokud server podporuje pouze SSH verze 1, máme jako útočník vyhráno a přihlašovací údaje můžeme odchytit rovnou. Naopak, pokud podporuje pouze SSH-2, máme smůlu, v současnosti jsou pro nás tato data v reálném čase nerozluštitelná.

Nejzajímavější je pak situace, pokud server podporuje obě verze protokolu — odpověď je tedy řetězec „SSH-1.99“. Většina SSH klientů je implicitně nastavena tak, že při možnosti výběru zvolí protokol vyšší — verzi 2. Jako útočník tedy do této komunikace aktivně vstoupíme a změníme odpověď serveru z řetězce „SSH-1.99“ na řetězec „SSH-1.51“. Když pak ke klientovi dorazí takto změněná odpověď, bude se domnívat, že server podporuje pouze protokol SSH-1. Jak taková komunikace vypadá je zobrazeno na obrázku 2.6.



Obrázek 2.6: SSH downgrade attack

2.11.4 Možnosti zabezpečení

Pro tento útok je základním předpokladem, že se útočník nachází mezi oběma komunikujícími stranami. V první řadě je tedy potřeba zabránit, aby se útočník do takovéhoto postavení dostal.

Tím patrně nejúčinnějším zabezpečením přímo proti tomuto útoku je pak podpora pouze SSH-2 protokolu na straně serveru. Odepíráme tak sice možnost připojit se klientům, kteří podporují pouze SSH-1, nejenom z důvodu tohoto útoku by se však tato verze již používat neměla.

Kapitola 3

Demonstrace vybraných útoků

Cílem této kapitoly je popsat praktické provedení vybraných útoků. Jak bylo zmíněno v úvodu, znalost útočníkova postupu může být pro bezpečnostního technika velikou výhodou — útoky jsou proto popisovány přímo z pohledu útočníka. V první podkapitole je potom popsáno konktrétní prostředí, ve kterém byly útoky prováděny.

UPOZORNĚNÍ! Veškeré níže uvedené skutečnosti a postupy slouží pouze k informačním účelům. Jakékoliv jejich použití (s výjimkou auditu vaší sítě) může být nezákonné a kvalifikováno jako trestný čin.

3.1 Popis prostředí

Všechny útoky byly prováděny v rámci jedné lokální sítě s (maximálně) šesti klienty a jedním serverem. K propojení koncových zařízení slouží bezdrátový směrovač, pro většinu útoků je však využit jako přepínač (pomocí drátového připojení). V tabulce 3.1 jsou síťové informace účastníků útoků, na obrázku 3.1 je potom znázorněna topologie sítě a použité operační systémy — černou přerušovanou čárou je znázorněno připojení bezdrátové, plnou drátové. Zeleně je pak naznačeno připojení počítače oběti a červeně útočníka. Počítače, které nejsou uvedeny v tabulce, nebyly pro účely demonstrace útoků využity.

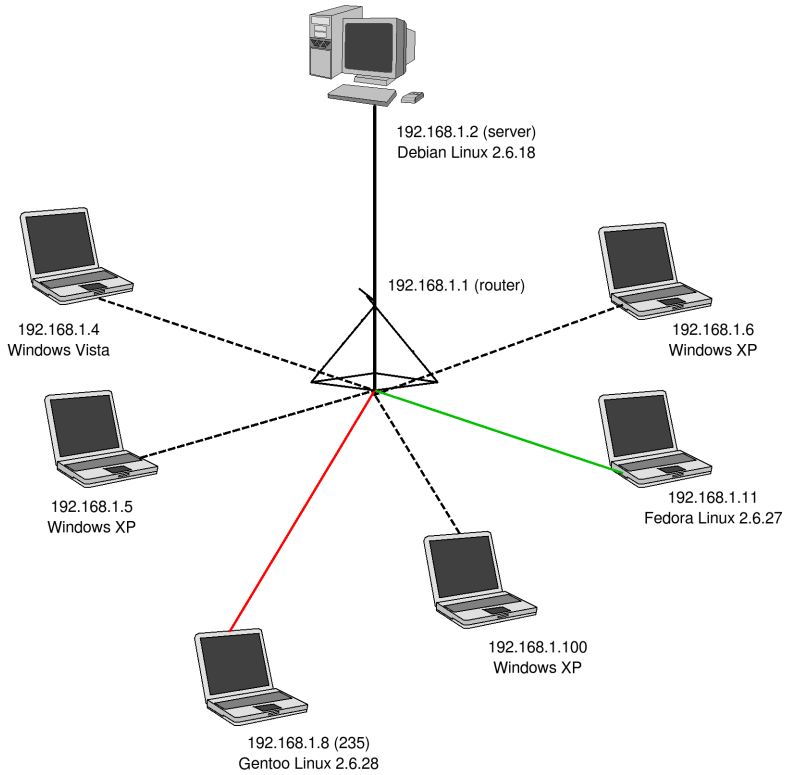
Počítač označený jako „server“ je funkční síťový server s běžnými službami, jako jsou DHCP, DNS, FTP, web, tisk atd.

Tabulka 3.1: Síťová nastavení účastníků útoků

Role	**IP adresa**	**MAC adresa**	**Doménové jméno**
Útočník	`192.168.1.8`	`00:18:f3:3e:d2:ff`	—
Oběť	`192.168.1.11`	`00:11:d8:17:9a:66`	`fedora`
Server	`192.168.1.2`	`00:01:02:9d:62:a0`	`server`
Směrovač	`192.168.1.1`	`00:17:31:04:01:d2`	`router`

3.2 ARP Spoofing

Cílem tohoto útoku je dostat se do MITM postavení a díky tomu tak získat údaje, ke kterým bychom se dostat neměli. Takovýmito údaji jsou většinou přihlašovací jméno a heslo, stejně tak ale můžeme zjistit, které webové stránky oběť navštívila (například pomocí DNS dotazů)



Obrázek 3.1: Znázornění topologie sítě

nebo si přečíst odesílaný/přijímaný e-mail atd. Způsoby, jak provést tento útok, popíši dva. Nejdříve s využitím textového nástroje `arpspoof` a potom pomocí specializovaného programu `Ettercap`.

Před započetím útoku si necháme vypsat ARP tabulku počítače oběti:

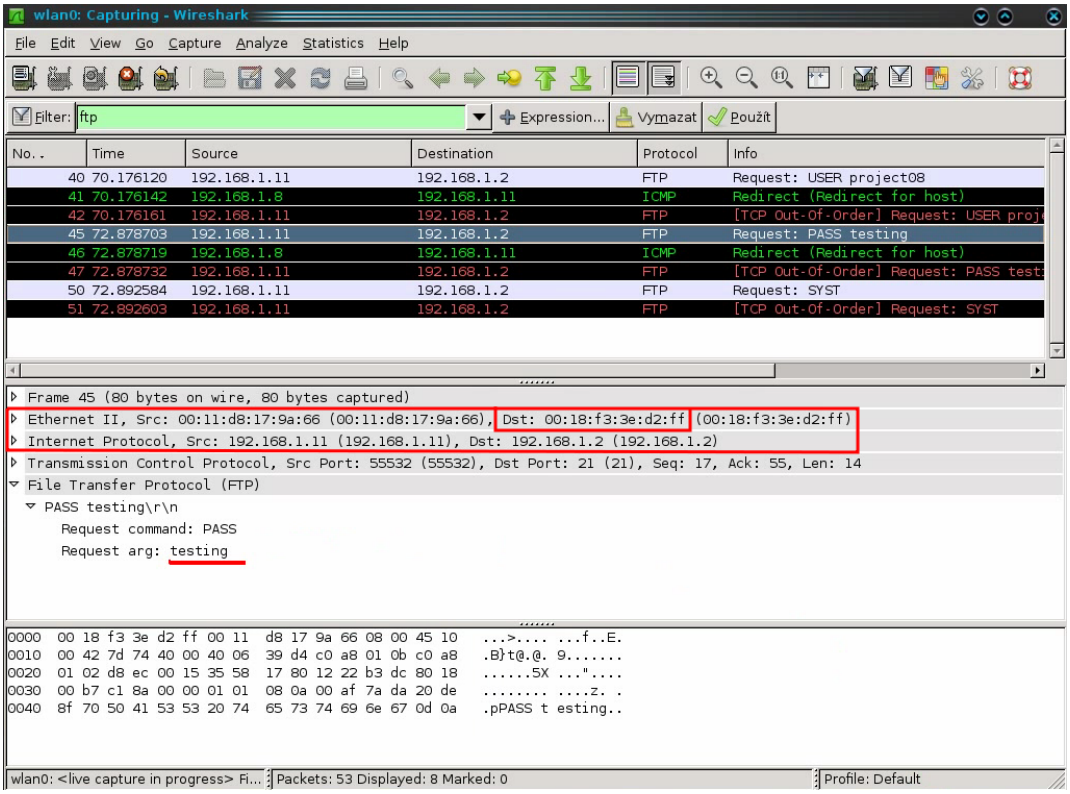
```
[root@fedora project08]# arp -n
Address          HWtype  HWaddress           Flags Mask  Iface
192.168.1.2      ether   00:01:02:9d:62:a0   C           eth0
192.168.1.8      ether   00:18:f3:3e:d2:ff   C           eth0
192.168.1.1      ether   00:17:31:04:01:02   C           eth0
```

Vidíme, že obsahuje tři záznamy, přičemž můžeme předpokládat, že jsou všechny správné. Pokud tedy zkusíme z počítače oběti provést ping na jiný počítač, vše proběhne v pořádku a my, jako útočník, žádný paket nezachytíme.

Nyní se tedy pokusíme pomocí ARP Spoofingu přesměrovat tuto komunikaci přes náš počítač. Ke změně záznamů v ARP tabulce počítače oběti použiji nástroj `arpspoof` z balíku `dsniff`:

```
arpspoof -i wlan0 -t 192.168.1.11 192.168.1.2
```

Program začne přes rozhraní `wlan0` odesílat na IP adresu 192.168.1.11 (adresa počítače oběti) falešné ARP Reply pakety, ve kterých je změněna zdrojová IP adresa na adresu 192.168.1.2. K počítači oběti tak dorazí ARP odpověď, ze které se dozví, že počítač s IP adresou 192.168.1.2 má MAC adresu `00:18:f3:3e:d2:ff`, což je však adresa naše (útočníka). Výsledkem tedy je, že si oběť podle této odpovědi aktualizuje svoji ARP tabulku



Obrázek 3.2: Wireshark - odchycení FTP hesla

a pokud nyní bude chtít poslat data na adresu 192.168.1.2, do hlavičky paketu přiřadí MAC adresu útočníka. Že to takto bude opravdu fungovat, se můžeme opět přesvědčit pomocí programu `ping` — nyní již přijdou ICMP ECHO_REQUEST pakety z počítače oběti na počítač útočníka, i když jsou původně určeny pro IP adresu 192.168.1.2. Na počítači útočíka pak můžeme pomocí programu `Wireshark` v hlavičce těchto odchycených paketů vidět, že IP adresy jsou sice správné, ale cílová MAC adresa je nastavena na adresu naši (viz obrázek 3.2).

V tuto chvíli se však u nás veškerá komunikace z počítače oběti zastaví. Abychom ji přeposlali původnímu adresátovi, musíme v jádře operačního systému tohle přeposílání povolit:

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

Pokud tedy na náš počítač dorazí paket, který má jinou IP adresu než naši, tak ho přepošleme, tektokrát však již správně. Veškerá odchozí komunikace od oběti na adresu 192.168.1.2 tedy půjde přes náš počítač. Díky tomu potom můžeme odchytit DNS dotazy nebo přihlašovací údaje na FTP server, jak je vidět na obrázku 3.2.

O úplné MITM postavení se však nejedná, protože odpověď je poslána přímo počítači oběti — pro dosažení úplného MITM postavení bychom museli současně posílat falešné ARP odpovědi i cílovému počítači. Pokud navíc chceme takto otrávit více počítačů současně, je lepší použít nějakého automatizovaného nástroje.

Provedení útoku pomocí programu Ettercap

Popis nástroje `Ettercap` je uveden v příloze B, zde již tedy přímo popíši, jak provést útok ARP Spoofing v konkrétním prostředí. Jako cíl útoku (Target 1) vybereme pouze počítač s IP adresou 192.168.1.11. Target 2 necháme prázdný. Cílem tedy bude, že jakákoliv odchozí komunikace z počítače oběti půjde přes nás. Pokud bychom nechali prázdný i Target 1, dosáhli bychom tzv. úplného MITM postavení, došlo by tedy k otrávení ARP tabulek všech počítačů v síti a přes náš počítač by tak procházela veškerá data.

V menu *Mitm* vybereme položku *Arp poisoning...* a spustíme útok (*Start - Start sniffing*). `Ettercap` tak od této chvíle bude každých 10 sekund odesílat na adresu oběti falešné ARP odpovědi. Výsledek je pak vidět při zobrazení ARP tabulky počítače oběti, kdy pro všechny IP adresy je nastavena útočníkova MAC adresa:

```
[root@fedora project08]# arp -n
Address          HWtype  HWaddress           Flags Mask  Iface
192.168.1.2      ether   00:18:f3:3e:d2:ff   C           eth0
192.168.1.8      ether   00:18:f3:3e:d2:ff   C           eth0
192.168.1.1      ether   00:18:f3:3e:d2:ff   C           eth0
```

Pokud se nyní opět oběť připojí na FTP server, `Ettercap` automaticky tyto přihlašovací údaje odchytí a vypíše:

```
FTP : 192.168.1.2:21 -> USER: project08 PASS: testing
```

Útok potom můžeme ukončit v menu *Start - Stop sniffing* a *Mitm - Stop mitm attack(s)*. `Ettercap` tím kromě ukončení odposlechu také vrátí ARP tabulky zasažených obětí do původního stavu.

3.3 Port Stealing

Jak je uvedeno v podkapitole zabývající se teorií tohoto útoku (2.6), pro dosažení úplného MITM postavení je dobré použít automatizovaného nástroje. Takovým nástrojem je například program `Ettercap`. Provedení tohoto útoku je pak velmi podobné, jako v ukázce předchozí, jenom je potřeba vybrat jako typ MITM útoku Port Stealing. Z tohoto důvodu jsem se rozhodl raději názorně předvést základní princip, jakým se dá „ukradnutí portu“ dosáhnout „ručně“. Program `Ettercap` pak celý tento proces automatizuje a přidává možnost odchycená data přeposlat původnímu adresátovi (počítači oběti).

Nejdříve si jednoduchým způsobem ukážeme, že se v MITM postavení nenacházíme. Pomocí protokolu SSH se tedy připojím na počítač oběti a při zapnutém odchytávání paketů (například pomocí programu `Wireshark`) provedu ping z počítače oběti na adresu serveru. Ping proběhne úspěšně a na počítači útočníka žádný ICMP paket nezachytíme.

Nyní se pokusíme na přepínači „ukradnout port“. K tomu budeme potřebovat odeslat speciálně upravený ARP Reply paket. A přesně k tomuto účelu nám poslouží program `arpoison`. Ještě před tím pomocí programu `arping` zjistím MAC adresu cílového počítače (jedná se o počítač, se kterým chce oběť komunikovat) a potom použiji program `arpoison` následujícím způsobem:

```
arpoison -i eth0 -d 192.168.1.8 -s 192.168.1.2 -t 00:18:f3:3e:d2:ff
         -r 00:01:02:9d:62:a0 -w 1
```

Jednotlivé parametry určují:

```
-i název síťového rozhraní
-d cílová IP adresa
-s zdrojová IP adresa
-t cílová MAC adresa
-r zdrojová MAC adresa
-w čas v sekundách mezi odeslanými pakety
```

Nejdůležitější je potom parametr uvádějící zdrojovou MAC adresu, která je nastavena na MAC adresu cílového počítače. Ve chvíli, kdy pak tento ARP Reply paket dorazí na přepínač, ten si podle jeho zdrojové adresy aktualizuje CAM tabulku. Výsledkem je, že přepínač má k fyzickému portu, ke kterému jsme připojeni my, asociovánu námi vynucenou MAC adresu. Pakety určené pro cílový počítač tedy dorazí k našemu počítači.

3.4 DHCP Spoofing

Před nasazením našeho falešného DHCP serveru je dobré „vyřadit ze hry“ server původní (pro danou síť server legitimní). Nejdříve tedy provedeme útok DHCP Flooding, kterým vyčerpáme všechny jeho IP adresy z adresového prostoru. K tomu použiji nástroj `dhcpx` z balíku `IRPAS`:

```
dhcpx -vv -i eth0 -A -D 192.168.1.2
```

Uvedená IP adresa je adresa, na které běží napadený server. Na tuto adresu bude program posílat falešné DHCP požadavky (z náhodně vygenerovaných MAC adres) a my ho necháme běžet do té doby, dokud nevyčerpáme všechny adresy z adresového prostoru DHCP serveru.

Po úspěšném vyřazení regulérního DHCP serveru nám nic nebrání spustit náš vlastní server. Ten se tak stane jediným fungujícím DHCP serverem v síti a na všechny požadavky přijaté od této chvíle bude odpovídat DHCP server, který je pod naší kontrolou. Můžeme samozřejmě využít některý z běžně využívaných DHCP serverů (například `dhcpd`), já jsem pro názornost využil serveru, který nabízí program `Ettercap`. Jeho nastavení provedeme v menu *Mitm - Dhcp spoofing...*. V zobrazeném dialogu vyplníme rozsah přidělovaných IP adres, síťovou masku a IP adresu DNS serveru. Protože se jedná o útok typu MITM, jako implicitní bránu `Ettercap` automaticky nastaví naši IP adresu. Útok potom spustíme v menu *Start - Start sniffing*. Při každém přidělení síťových informací nás o tom Ettercap detailně informuje. (Pozn. k obrázku 3.3: IP adresa útočníka je v této ukázce 192.168.1.248.)

Výsledkem tohoto útoku tedy je, že oběť má nastavenu implicitní bránu na IP adresu útočníka. Veškerý odchozí provoz jde tedy přes útočníkův počítač. Opět se tedy jedná o neúplné MITM postavení, protože zachytíme pouze data odchozí. O tom se můžeme přesvědčit jednoduchým způsobem, kdy provedeme ping z počítače oběti na nějaký vzdálený server. Na počítači útočníka zachytíme pakety ICMP ECHO_REQUEST, odpověď již však ne, ta půjde rovnou na počítač oběti. Pokud tedy chceme dosáhnout úplného MITM postavení, musíme využít jednoho z předchozích dvou typů útoků. Jak si ale ukážeme hned v následující podkapitole (3.5), může tento útok vést i k dalšímu, tentokrát již hodně nebezpečnému útoku.

3.5 DNS Spoofing a Pharming

V této podkapitole popíši, jak dosáhnout nasazení falešného DNS serveru s využitím DHCP Spoofingu. Provedení útoku typu DHCP Spoofing je vysvětleno v podkapitole 2.7, následu-

Obrázek 3.3: DHCP Spoofing - Ettercap

jící popis tedy bude od kroku, kdy se počítač oběti připojil do sítě a od našeho falešného DHCP serveru dostal námi zvolené síťové nastavení. Nejdůležitější položkou je v tomto případě IP adresa DNS serveru (obrázek 3.4).

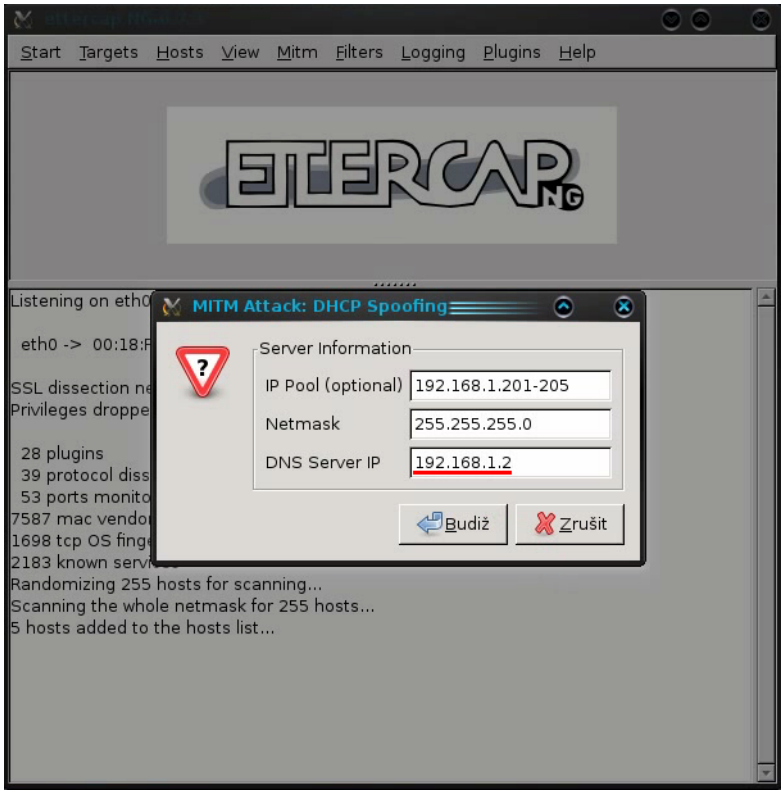
Druhým cílem našeho útoku (po nasazení falešného DNS serveru) bude podvrhnout oběti falešnou webovou stránku, neboli Pharming.

Provedení Pharmingu není složitý proces. Musíme pouze upravit (v našem případě přidat) záznam o námi podvrhované stránce v konfiguraci DNS serveru, aby při překladu této adresy byla oběti doručena podvržená IP adresa.

Jako podvrhovanou stránku jsem zvolil informační systém naší fakulty, jehož adresa je „wis.fit.vutbr.cz“. Pro tohle doménové jméno je autoritativní DNS server umístěn mimo naší síť a v normálním případě by náš lokální DNS server požadavek o překlad tohoto jména přeposlal dál. My však pro tuto adresu vytvoříme na našem DNS serveru nový záznam (tzv. zónu). Pokud pak na server přijde požadavek o překlad této adresy, ten ji najde ve svých záznamech a vrátí zadanou IP adresu. Zóna pro naši konkrétní adresu by pak mohla vypadat následovně (použitý DNS server je `Bind 9`):

```
zone 'wis.fit.vutbr.cz' {
      type master;
      file '/etc/bind/db.fit';
      allow-update { none; };
};
```

Pokud nyní na server dorazí požadavek o překlad adresy „wis.fit.vutbr.cz“, bude ji hledat v souboru „db.fit“. Musíme tedy ještě vytvořit tento soubor a v něm informaci pro překlad.



Obrázek 3.4: DNS Spoofing - Nastavení IP adresy falešného DNS serveru

Ta bude vypadat následovně:

```
wis.fit.vutbr.cz.   IN   A   192.168.1.2
```

Narozdíl od legitimní IP adresy stránky tak server odpoví IP adresou 192.168.1.2. Poslední věcí, která nám zbývá, je vytvořit na této adrese vzhledově podobnou (stejnou) stránku, jako je stránka původní. Pokud nic netušící oběť potom zadá do prohlížeče adresu „wis.fit.vutbr.cz“, najede jí na první pohled stejný informační systém, jaký očekává. Ve skutečnosti se však jedná o stránku falešnou.

3.6 DoS útok - ACK Flood

DoS útoků typu zahlcení oběti síťovým provozem existuje velká řada. Já jsem zvolil tento poměrně jednoduchý, ale pěkně názorný útok - zahlcení oběti potvrzovacími ACK pakety. Využívám k tomu *exploit* představený v lednu roku 2000, viz [16]. Jedná se o program v jazyce C, který na adresu oběti odesílá ACK pakety z náhodných IP adres. Jeho síla pak spočívá především v počtu takovýchto paketů, který je (měl by být) omezen. („Převzato z [16].“) Výsledkem útoku je, že server zahlcený těmito pakety není schopen jakékoliv komunikace po síti.

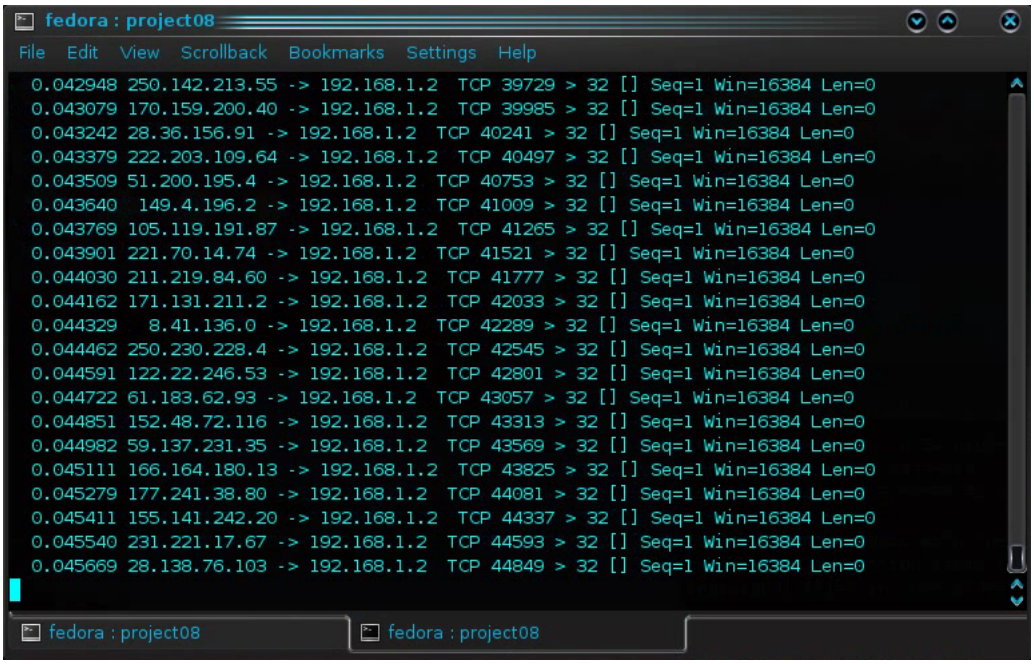
Teď tedy k provedení tohoto útoku. Aby bylo vidět, že skutečně dojde k vyřazení serveru, samotný útok spustím z jiného počítače. Problémem útoků zahlcením je totiž to, že stejně tak jako linka oběti je zahlcována i linka naše (útočníka). Snadno se tedy může stát, že místo počítače oběti zahltíme sami sebe.

Pomocí SSH protokolu se tedy připojím na vzdálený počítač a z něj spustím *exploit* `stream.c`:

```
[root@fedora dos]# ./stream 192.168.1.2 32 0
stream.c v1.0 - TCP Packet Storm

Resolving IPs...Sending...
```

Naším primárním cílem je na serveru 192.168.1.2 vyřadit službu SSH, která běží na portu 32 (druhý parametr). Třetím parametrem potom můžeme změnit velikost odesílaných paketů, nula značí velikost implicitní. V programu `TShark` (textová verze `Wiresharku`) můžeme vidět, jak vypadají odesílané paktety — obrázek 3.5.



Obrázek 3.5: DoS útok - ACK Flood, výpis programu TShark

Nyní se zkusíme z našeho počítače, který teď představuje běžného uživatele, připojit k serveru pomocí SSH:

```
$ ssh -p 32 project08@192.168.1.2
ssh: connect to host 192.168.1.2 port 32: No route to host
```

Jak je vidět, připojení se nezdařilo. Výsledkem útoku ale není pouze odepření služby SSH, nýbrž vyřazení celého serveru (z pohledu síťových služeb). Na serveru běží mimo jiné služby DNS, FTP či webový server, které jsou po dobu útoku také nedostupné.

Zotavení serveru z tohoto typu útoku je pak velmi rychlé. Brzy po ukončení zaplavování pakety jsou všechny služby opět k dispozici.

3.7 WEP Crack

Tato ukázka by se také dala nazvat: „Jak pomocí programu Aircrack-ng prolomit WEP klíč během 5 minut“. I přesto, že k uskutečnění tohoto útoku potřebujeme provést několik

různých operací, na všechny nám bude stačit jediný komplexní nástroj, kterým je právě `Aircrack-ng`.

3.7.1 Aircrack-ng suite

(„Podkapitola převzata z [11].“)

`Aircrack-ng suite` je sada konzolových (textových) programů, určených na prolamování 802.11 WEP a WPA-PSK klíčů. Jedná se o open-source projekt šířený pod licencí GPL-2, fungující jak pod unix-like systémy, tak pod Windows. Nyní tedy stručně popíši programy z tohoto balíku, které budeme k provedení útoku potřebovat.

Airmon-ng

Účelem tohoto jednoduchého programu je ovládat bezdrátové rozhraní Wi-Fi karty, konkrétně přepínat ho z „Managed“ módu do „Monitor“ módu a zpět. Implicitně je rozhraní v režimu „Managed“, ve kterém přijímá pouze pakety pro něj určené. Po přepnutí do režimu „Monitor“ rozhraní přijímá veškeré pakety, které zachytí. Jedná se o stejný princip, jako je promiskuitní režim u rozhraní drátových. Po spuštění nad určitým rozhraním program vytvoří rozhraní nové (nad stejnou fyzickou bezdrátovou kartou), ovšem v režimu „Monitor“. Rozhraní je odstraněno po jeho zastavení (volba „`stop`“).

Použití:

```
airmon-ng <start|stop> <interface> [channel]
```

kde:

```
<start|stop> Spuštění nebo zastavení daného rozhraní
<interface>  Určení rozhraní
[channel]    Nastavení kanálu, na kterém bude rozhraní naslouchat
```

Aireplay-ng

Hlavní funkcí tohoto nástroje je tzv. *packet injection*. Jedná se o techniku, při které cíleně generujeme síťový provoz. Může jít například o falešnou autentizaci, pakety pro odhlášení nebo v našem případě o falešné odpovědi na ARP dotazy (viz dále).
Použití:

```
aireplay-ng <options> <interface>
```

V `options` potom jako jeden z parametrů uvedeme, jaký typ *packet injection* chceme použít. Pro nás jsou důležité tyto:

```
-9 Injection test
-1 Falešná autentizace
-3 ARP request replay útok
```

Airodump-ng

`Airodump-ng` slouží k samotnému odchytávání paketů, které bezdrátová síťová karta zachytí. Do souboru potom ukládá WEP inicializační vektory pro budoucí použití programem `aircrack-ng`. Na standartní výstup za běhu vypisuje všechny důležité informace, jako MAC adresy komunikujících stran, počet odchycených paketů za vteřinu, použité šifrování atd.

Aircrack-ng

Posledním článkem pro provedení tohoto útoku je nástroj na samotné prolomení WEP klíče. Nástroj `aircrack-ng` k tomu využívá inicializační vektory uložené v souboru programem `airodump-ng`. Program umí obě dvě metody — jak metodu PTW, tak i FMS/KoReK. Kterou má program použít, je určeno parametrem při jeho spouštění. Já použiji metodu PTW, protože je rychlejší.

3.7.2 Předpoklady k provedení útoku

Ještě před samotným popisem útoku zde zmíním čtyři podmínky, nutné (kromě té poslední) k provedení tohoto útoku:

- Dostatečná blízkost k přístupovému bodu
- Minimálně jeden připojený aktivní klient
- Wi-Fi rozhraní s podporou „Monitor“ módu
- Fungující *packet injection*

Abychom se vůbec mohli k danému přístupovému bodu (*Access Point*) připojit, musíme být v jeho dostatečné blizkosti. Je však důležiré si uvědomit, že i když danou síť vidíme a dokážeme přijmout pakety po této síti putující, nemusí to ještě znamenat, že budeme schopni se k dané síti připojit. Přístupový bod má většinou silnější signál než běžná bezdrátová karta, můžeme se tedy ocitnout v takové vzdálenosti, kdy pakety sice přijímáme, ale nejsme schopni je odeslat a asociovat se tak s přístupovým bodem.

Druhá podmínka souvisí s principem tohoto útoku. Pokud na síti nebude žádný aktivní klient, nemůžeme odchytit žádné pakety a tudíž ani nezískáme žádné inicializační vektory, potřebné pro prolomení klíče.

Další podmínkou je bezdrátová síťová karta s podporou „Monitor“ módu. Ne všechny karty totiž tento mód podporují.

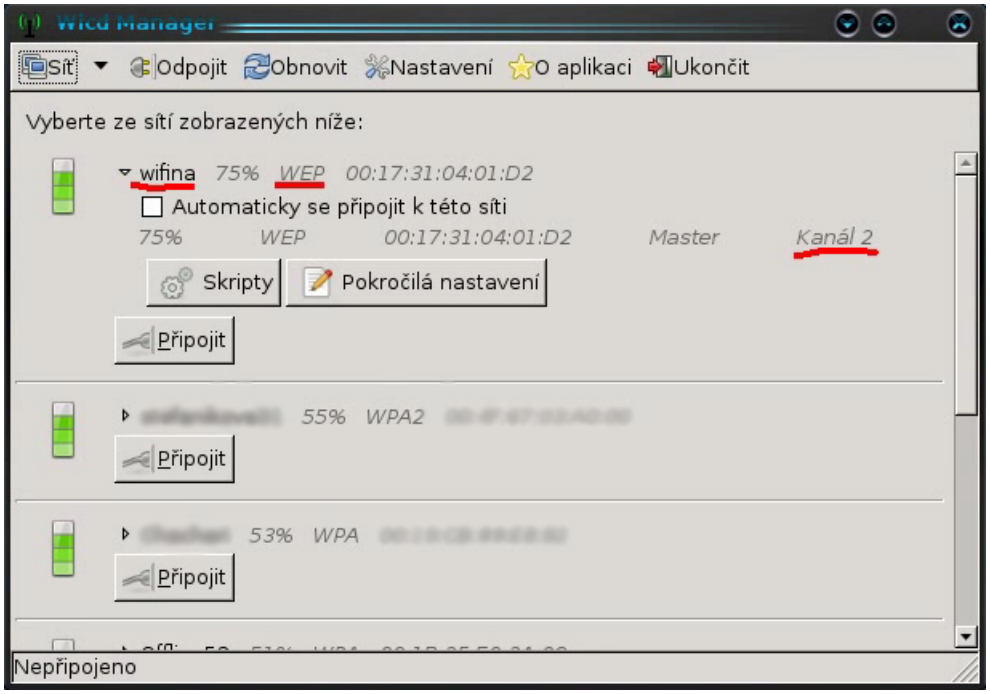
Poslední podmínka sice není nezbytně nutná, bez *packet injection* však bude útok trvat mnohonásobně déle.

3.7.3 Praktická ukázka

Všechny výše uvedené předpoklady máme splněné, můžeme se tedy konečně vrhnout na samotné provedení útoku. V první řadě si musíme vybrat bezdrátovou síť, ke které se budeme chtít připojit. Pro vypsání bezdrátových sítí v dosahu, stejně tak jako jejich zabezpečení a kanál, jsem použil program `Wicd Manager`. Vybrali jsme si tedy síť zabezpečenou pomocí WEP a s dobrým signálem, pojmenovanou (SSID) *wifina*.

Ze všeho nejdříve musíme zajistit odchytávání paketů. Pomocí programu `airmon-ng` tedy přepneme bezdrátové rozhraní do „Monitor“ módu, otestujeme *packet injection* a spustíme `airodump-ng` pro odchyt inicializačních vektorů. Náš vybraný přístupový bod určíme pomocí jeho MAC adresy (parametr `bssid`) a také uvedeme kanál, na kterém daná síť běží.

Aby od nás přístupový bod vůbec přijímal pakety, musí s ním být asociována naše MAC adresa. Pro tuto asociaci použijeme program `aireplay`, pomocí něhož se budeme pokoušet každých 10 sekund o falešnou autentizaci.



Obrázek 3.6: Wicd Manager - výběr sítě

Abychom celý útok výrazným způsobem zrychlili, budeme navíc generovat falešné ARP dotazy. Při každém takovém dotazu totiž přístupový bod tento paket rozešle všem klientům a vygeneruje nový inicializační vektor. Díky tomu jsme schopni zachytit v kratším čase daleko víc inicializačních vektorů. Pro rozesílání těchto dotazů opět použijeme program `aireplay-ng`, tentokrát s volbou `-3`.

V této chvíli nám k získání WEP klíče chybí poslední věc, a to spustit `aircrack-ng` nad souborem, který jsme určili při spouštění programu `airodump-ng`. Parametr `-z` potom programu říká, že chceme použít metodu PTW.

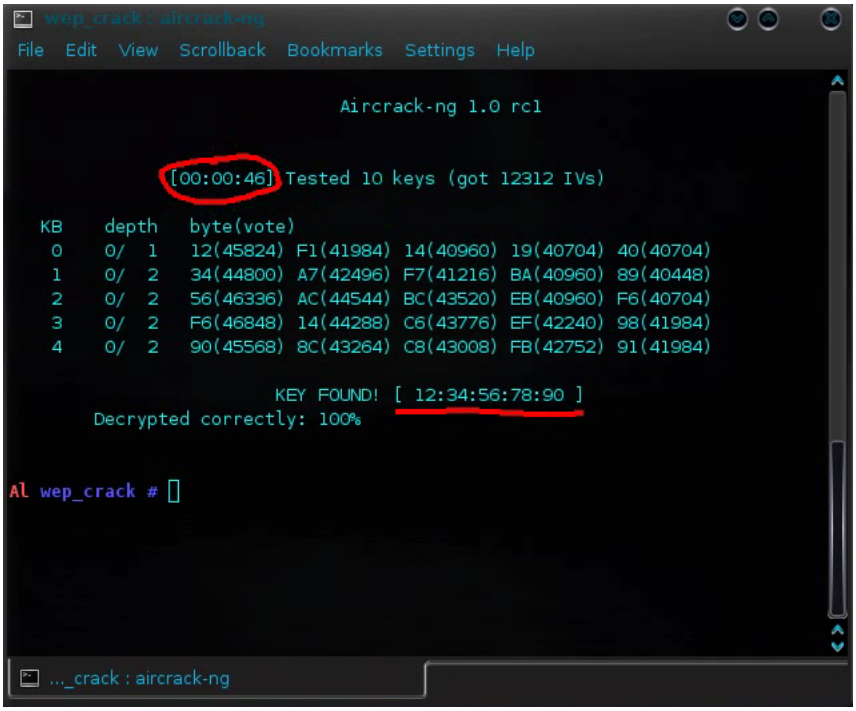
Jaký je výsledek? Samotné prolamování klíče trvalo 46 vteřin. Celý útok pak trvá kolem pěti minut, po kterých jsme schopni se k takto zabezpečené bezdrátové síti připojit. Cílem této ukázky tedy bylo ukázat, že tato metoda zabezpečení, ač v současnosti hojně používaná, je určitě nedostatečná.

3.8 SSH downgrade attack

Náplní této podkapitoly je ukázat, jak lze získat přihlašovací údaje pro připojení k počítači přes protokol SSH. Nastavení komunikujících stran je potom běžným způsobem — server podporuje protokol verze SSH-1 i SSH-2, klient pak také, s tím že preferovanou verzí je SSH-2.

Základním požadavkem pro provedení tohoto útoku je, že se (opět z pohledu útočníka) nacházíme v MITM postavení vzhledem k oběma komunikujícím stranám, neboli pakety mezi klientem (obětí) a SSH serverem procházejí přes náš počítač. V ukázce jsem zvolil metodu ARP Spoofing (s programem `Ettercap`), která je podrobně popsána v kapitole 2.5.

V této chvíli sice dokážeme odposlechnout veškerou komunikaci mezi oběma stranami, pokud se ale klient pokusí připojit na server pomocí SSH protokolu, vzhledem k jejich na-



Obrázek 3.7: WEP Crack - Aircrack-ng

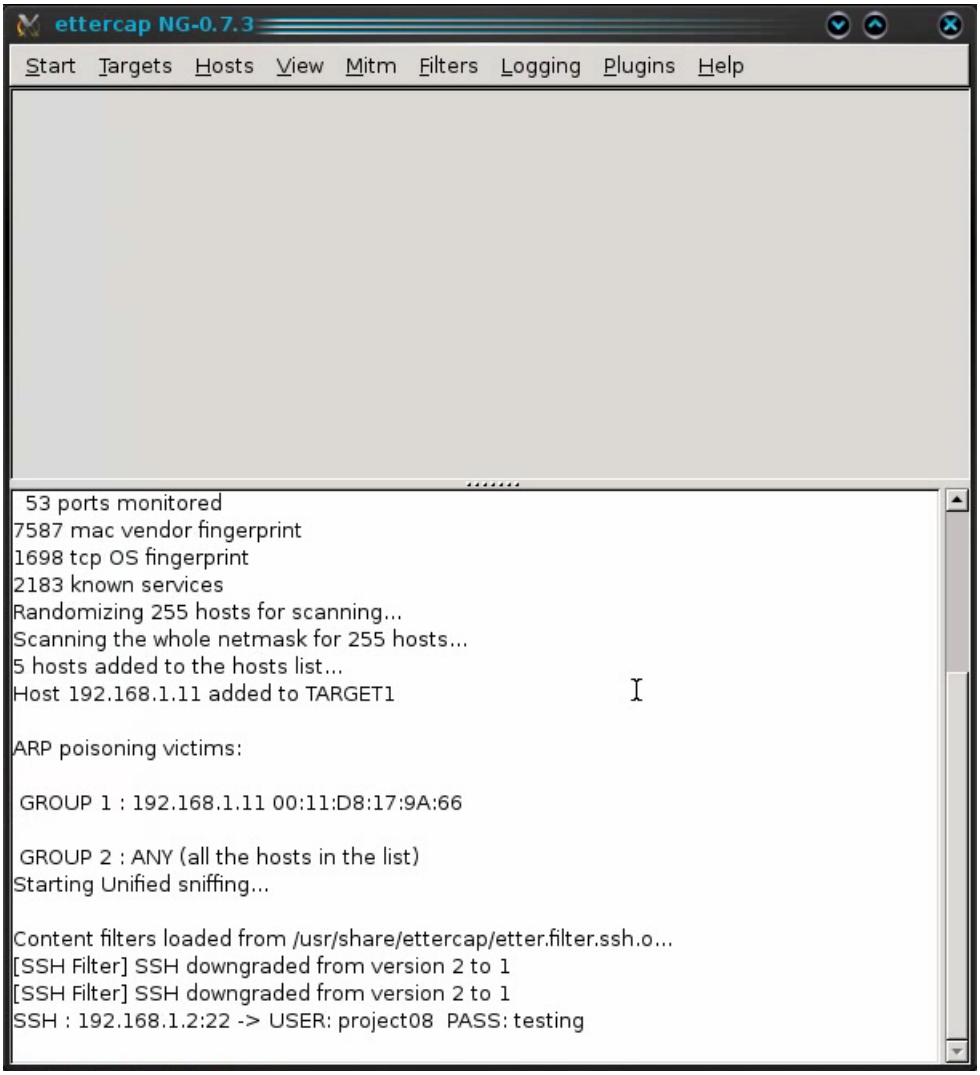
stavení se použije SSH-2 a my odchytíme pouze zašifrované pakety, které nejsme schopni dešifrovat. Při zahájení komunikace přes protokol SSH tedy musíme do této komunikace aktivně vstoupit a pozměnit odpověď serveru. K tomu nám opět poslouží `Ettercap`, u kterého můžeme nastavit tzv. filtr. Jedná se vlastně o předpis (podobný jazyku C), který `Ettercapu` říká, na základě jakých pravidel a jakým způsobem má konkrétní paket pozměnit. Zde uvádím ukázku, jak takový filtr vypadá právě pro *SSH downgrade attack*:

```
if (ip.proto == TCP) {
  if (tcp.src == 22) {
    if ( replace('SSH-1.99,,, ''SSH-1.51') ) {
      msg('[SSH Filter] SSH downgraded from version 2 to 1\n');
    } else {
      if ( search(DATA.data, 'SSH-2.00') ) {
        msg('[SSH Filter] Server supports only SSH version 2\n');
      } else {
        if ( search(DATA.data, 'SSH-1.51') ) {
          msg('[SSH Filter] Server already supports only version 1\n');
        }
      }
    }
  }
}
```

Nejdříve v něm testujeme, jestli paket splňuje podmínky — musí se jednat o spojení TCP na protokolu 22 (standartní port SSH). Pokud je tomu tak, pokusíme se změnit řetězec „SSH-1.99“ na „SSH-1.51“. Jestliže se nám to povede, vypíšeme zprávu o úspěšném *down-*

grade útoku. V opačném případě vypíšeme, jaká verze SSH protokolu bude pro komunikaci použita.

Filtr máme napsaný, pomocí nástroje `etterfilter` ho tedy zkompilujeme do binární podoby, které program `Ettercap` rozumí. Jakmile pak tento filtr načteme, automaticky se spustí filtrování všech paketů a jejich případná změna. Když se klient pokusí k serveru připojit nyní, jako odpověď mu přijde, že server podporuje pouze protokol verze SSH-1. Po přihlášení klienta jsme potom jako útočník získali na tento server jeho přihlašovací údaje.



Obrázek 3.8: SSH downgrade attack - Ettercap

Kapitola 4

Závěr

Problematika bezpečnosti počítačových sítí mě vždy zajímala a sám jsem chtěl zjistit, jaké možnosti mají útočníci při provádění útoků. Tato práce mě tedy opravdu bavila, přišla mi zajímavá a naučil jsem se mnoho nového. Také jsem rád, že jsem se mohl o své výsledky podělit se svými spolužáky.

Nejprve bylo potřeba problematiku útoků a bezpečnostních mechanismů v počítačových sítích nastudovat. Výsledkem je teoretická část práce, která představuje ucelený přehled o nejčastějších útocích prováděných v rámci lokální sítě.

Hlavním cílem této práce pak bylo vytvořit demonstrační úlohy pro předvedení síťových útoků. Jako svůj hlavní přínos tak považuji demonstrační videa, přímo nasnímaná při provádění útoků. Protože jsem sám měl příložitost tyto úlohy studentům předvést, myslím si, že se mi tento cíl splnit podařilo. Kromě teoretické výuky tak mají studenti možnost vidět i praktické provedení útoků v reálném prostředí.

Velice zajímavé a podnětné by jistě bylo pokračování v této práci z pohledu druhého, tedy praktické ukázky vybraných bezpečnostních opatření.

Literatura

[1] Angela Orebaugh, E. D. T.: *Ethereal Packet Sniffing*. Syngress Publishing, 2004, iSBN 1-932266-82-8.

[2] David C. Plummer: RFC 826 - An Ethernet Address Resolution Protocol. `http://www.ietf.org/rfc/rfc826.txt`, 1982.

[3] Erik Tews, Ralf-Philipp Weinmann, Andrei Pyshkin: *Breaking 104 bit WEP in less than 60 seconds*. 2008.

[4] Joel Scambray, G. K., Stuart McClure: *Hacking bez tajemství, 2. aktualizované vydání*. Computer Press, 2002, iSBN 80-7226-644-6.

[5] P. Congdon: RFC 3580. IEEE 802.1X Remote Authentication Dial In User Service (RADIUS). `http://www.ietf.org/rfc/rfc3580.txt`, 2003.

[6] P. Mockapetris: RFC 1035. Domain names - implementation and specification. `http://www.ietf.org/rfc/rfc1035.txt`, 1987.

[7] R. Droms: RFC 2131. Dynamic Host Configuration Protocoll. `http://www.ietf.org/rfc/rfc2131.txt`, 1997.

[8] Scott R. Fluhrer, Itsik Mantin, Adi Shami: *Weaknesses in the Key Scheduling Algorithm of RC4*. 2001.

[9] Stallings, W.: *Network Security Essentials: Applications and Standards, Third Edition*. Prentice Hall, 2007, iSBN 0-13-238033-1.

[10] T. Ylonen, C. Lonvick: RFC 4251. The Secure Shell (SSH) Protocol Architecture. `http://www.ietf.org/rfc/rfc4251.txt`, 2006.

[11] WWW stránky: Aircrack-ng suite. `http://www.aircrack-ng.org/doku.php`.

[12] WWW stránky: Ettercap. `http://ettercap.sourceforge.net/index.php`.

[13] WWW stránky: Bráníme se odposlechu: specifické útoky. `http://www.lupa.cz/clanky/branime-se-odposlechu-specificke-utoky/`, 15.8.2006.

[14] WWW stránky: Seriál Útoky typu DoS. `http://www.lupa.cz/serialy/utoky-typu-dos/`, 2006.

[15] WWW stránky: Nejznámější útoky v síti Ethernet. `http://connect.zive.cz/node/714`, 20.7.2007.

[16] WWW stránky: New DoS attack tool released (stream.c, raped.c, ACK). http://www.securiteam.com/unixfocus/5YP0I000DG.html, 21.1.2000.

[17] WWW stránky: Rhybaření střídá pharming. http://www.lupa.cz/clanky/rhybareni-strida-pharming/, 31.3.2005.

Dodatek A

Seznam demonstračních videí

Všechna demonstrační videa jsou umístěna na přiloženém CD-ROM.

1. ARP Spoofing
2. ARP Spoofing - Ettercap
3. Port Stealing
4. DHCP Flooding & Spoofing
5. DNS Spoofing & Pharming
6. DoS - ACK Flood
7. WEP Crack
8. SSH downgrade attack

Dodatek B

Popis programu Ettercap

Ettercap je nástroj pro man-in-the-middle útoky v lokální počítačové síti. Mezi jeho vlastnosti patří odposlouchávání komunikace, filtrování paketů a mnoho dalších. Podporuje velké množství protokolů (včetně šifrovacích) a obsahuje funkce pro analýzu sítě a připojených uživatelů. („Převzato z [12]“.)

B.1 Důležité položky hlavního menu

Options - Promisc mode Přepnutí síťové karty do/z promiskuitního módu.

Options - Set netmask Nastavení masky sítě.

Sniff - Set pcap filter... Filtr přímo pro knihovnu pcap, která zajišťuje odchytávání paketů.

Sniff - Unified sniffing... Určení síťového rozhraní a přepnutí do vedlejšího menu.

B.2 Vedlejší menu

Start Spuštění a zastavení vybraného útoku.

Targets Určení cíle (cílů) útoků — oběti.

Hosts Seznam koncových zařízení v síti.

View Informace o síti, statistiky...

Mitm Výběr typu útoku.

Filters Načtení filtru.

Logging Nastavení a spuštění logování.

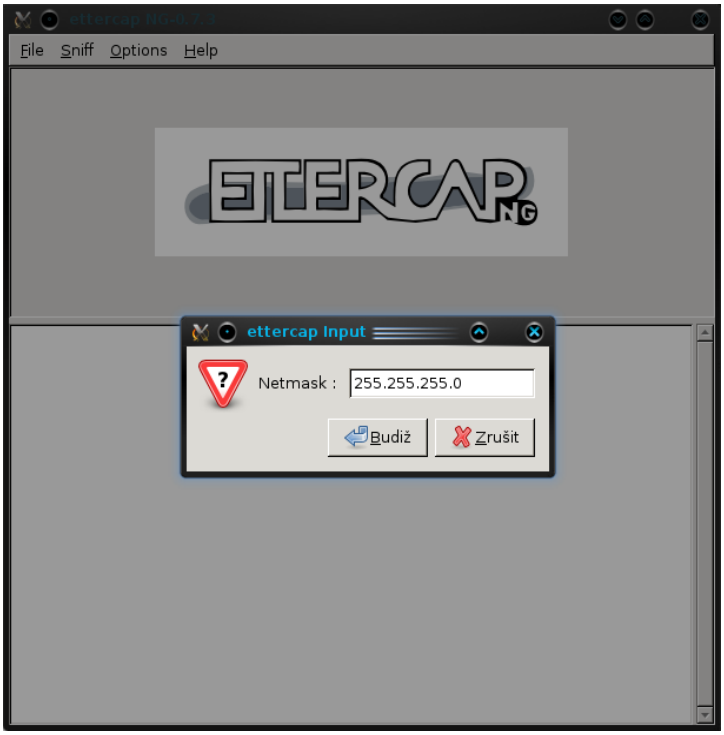
Plugins Zásuvné moduly.

Help Nápověda.

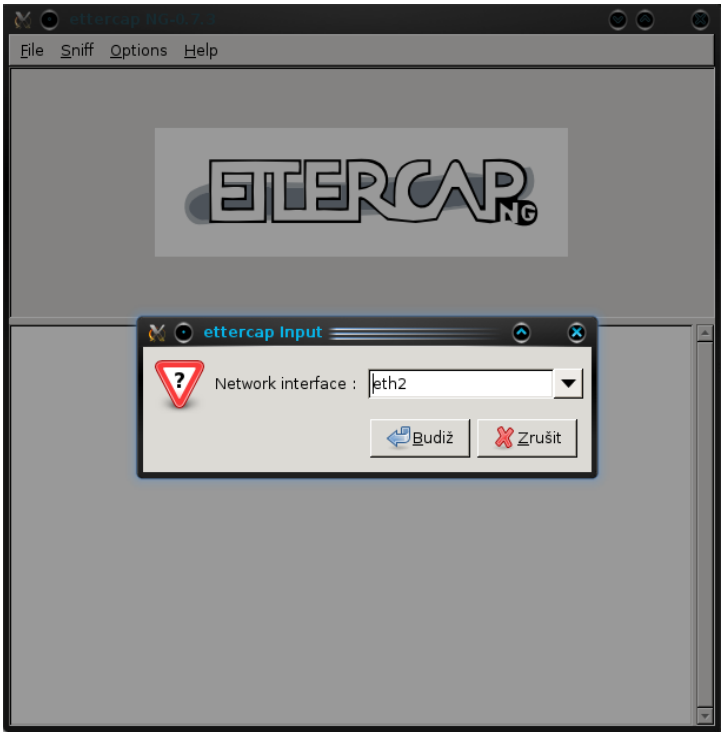
B.3 Ukázka provedení útoku ARP Spoofing

Spustíme `Ettercap` v grafickém módu:

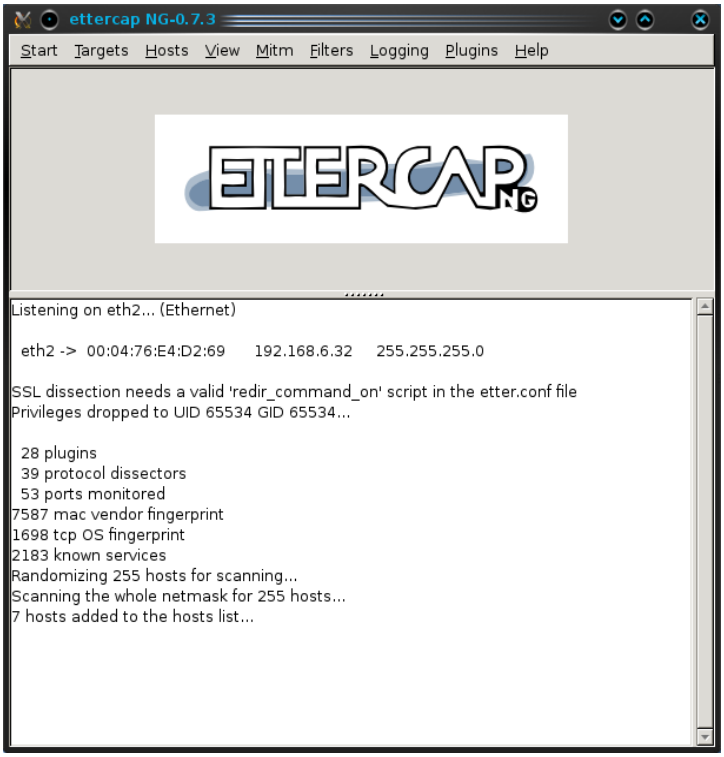
```
#ettercap -G
```



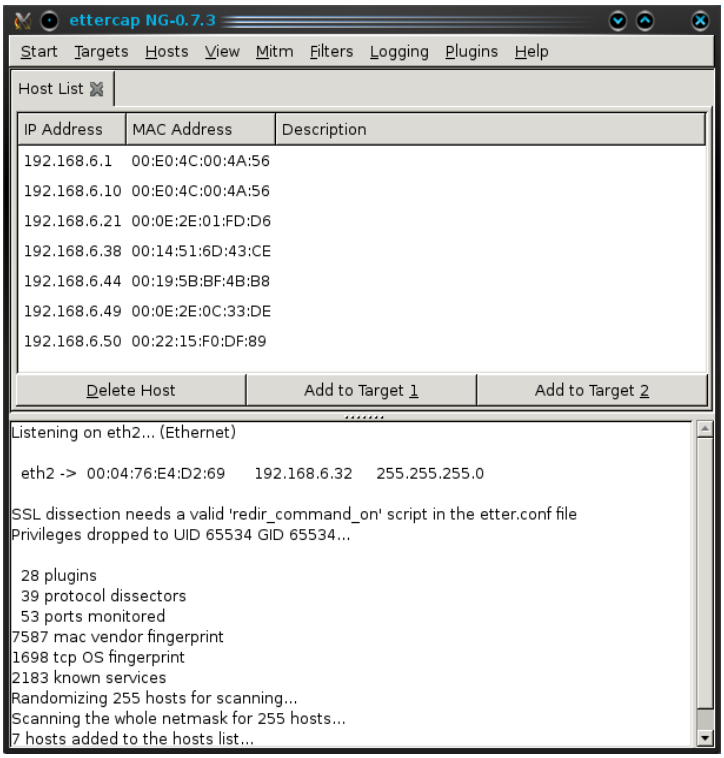
Obrázek B.1: Nastavíme masku sítě (*Options - Set netmask*)



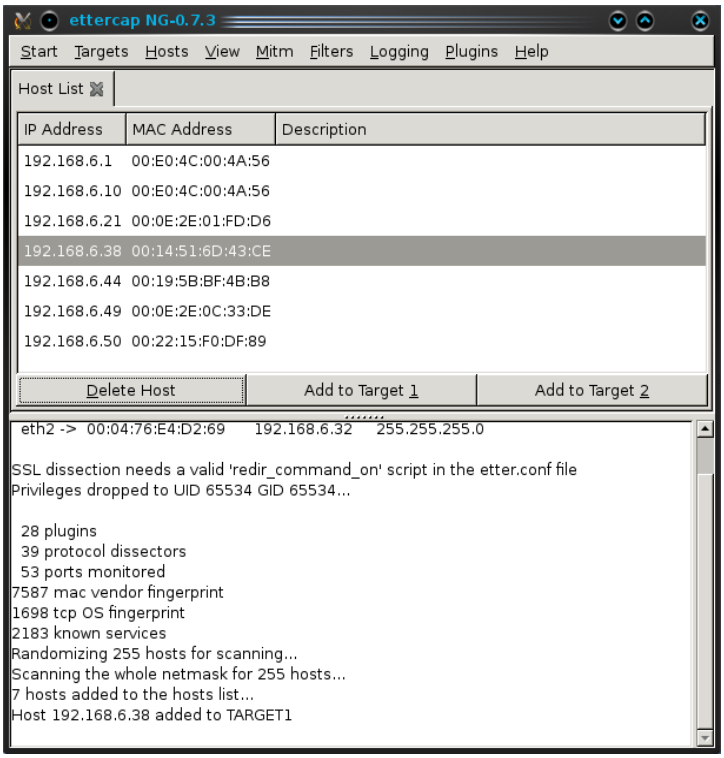
Obrázek B.2: Vybereme síťové rozhraní (*Sniff - Unified sniffing...*)



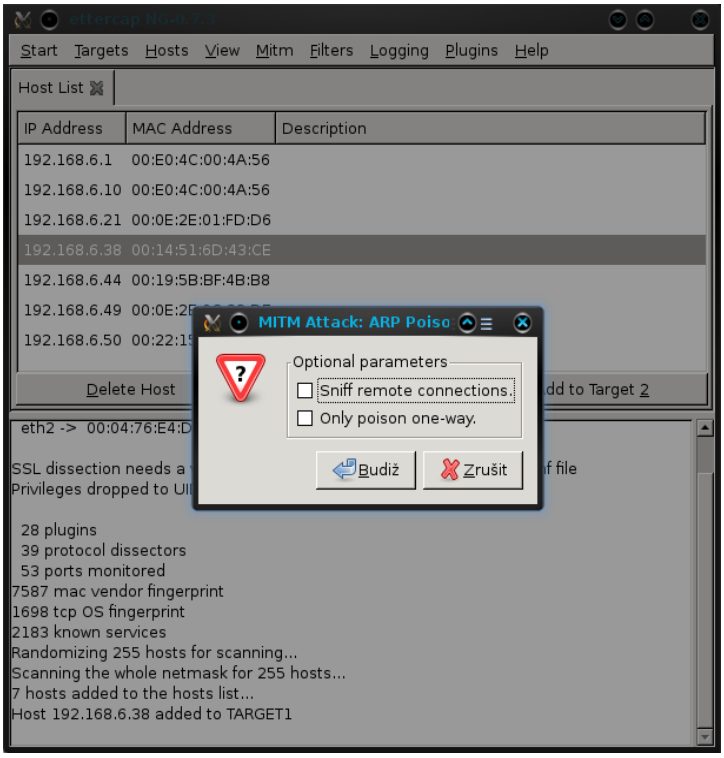
Obrázek B.3: Naplníme seznam hostů v síti (*Hosts - Scan for hosts*)



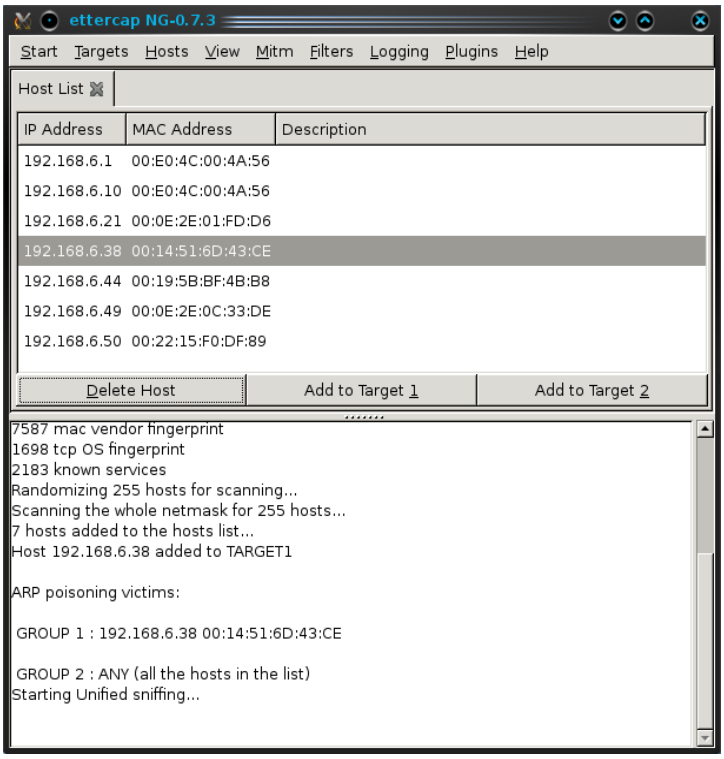
Obrázek B.4: Zobrazíme IP a MAC adresy hostů v síti (*Hosts - Hosts list*)



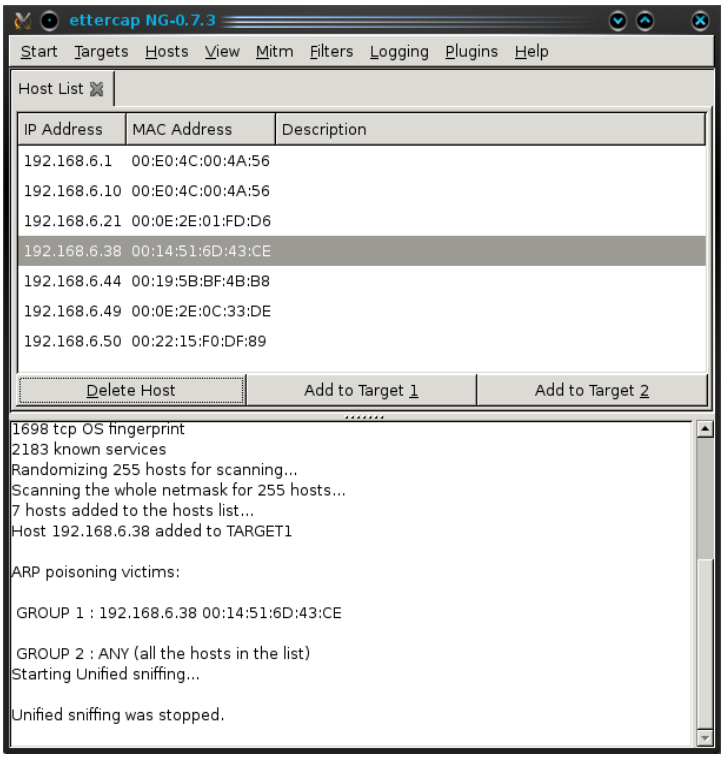
Obrázek B.5: Vybereme cíl útoku (*Add to Target 1*)



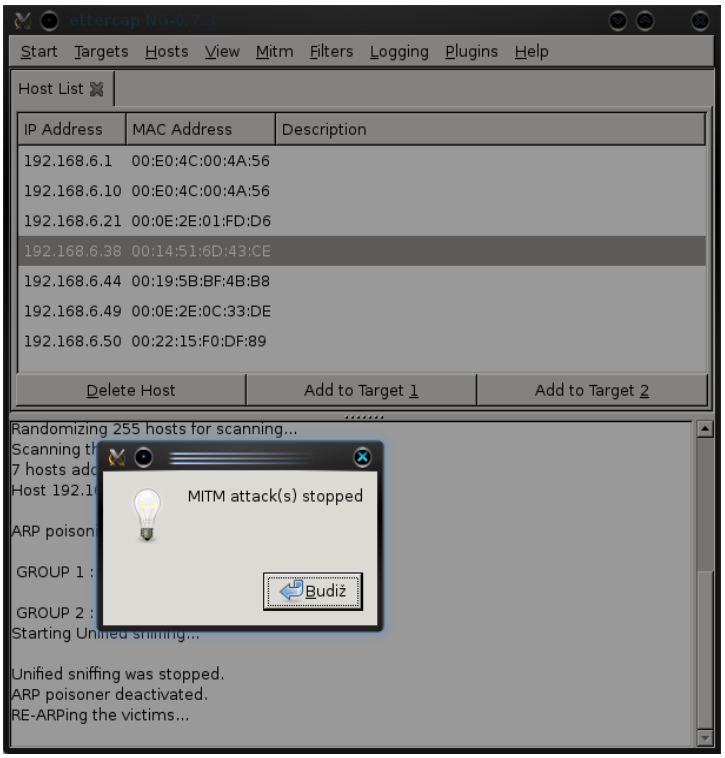
Obrázek B.6: Vybereme typ útoku ARP Spoofing (*Mitm - Arp poisoning...*) — pokud v nabídce zaškrtneme první volbu („*Sniff remote connections.*“), půjde přes náš počítač i komunikace směřující mimo naši síť, druhá možnost („*Only poison one-way.*“) potom přes náš počítač přesměruje pouze komunikaci z TARGET1 k TARGET2



Obrázek B.7: Spustíme odposlech (*Start - Start sniffing*) — `Ettercap` začne automaticky odchytávat přihlašovací jména a hesla



Obrázek B.8: Zastavíme odposlech (*Start - Stop sniffing*)



Obrázek B.9: Ukončíme ARP Spoofing (*Mitm - Stop mitm attack(s)*)