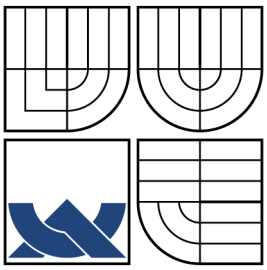
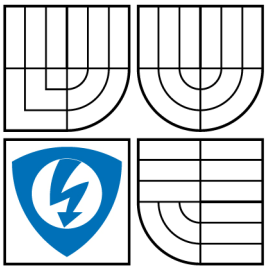


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

ÚTOKY NA AKTIVNÍ SÍŤOVÉ PRVKY

ATTACKS ON ACTIVE NETWORK ELEMENTS

BAKALÁŘSKÁ PRÁCE

BACHELOR´S THESIS

AUTOR PRÁCE — RICHARD ŠČEPKO
AUTHOR

VEDOUCÍ PRÁCE — ING. MICHAL POLÍVKA
SUPERVISOR

BRNO 2008

LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. Pan/paní

Jméno a příjmení: Richard Ščepko

Bytem: Trenčín, Súdna 3

Narozen/a (datum a místo): 11.9.1984, Trenčín

(dále jen „autor“)

a

2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií

se sídlem Údolní 244/53, 602 00, Brno

jejímž jménem jedná na základě písemného pověření děkanem fakulty:

Prof. Ing. Kamil Vrba, CSc., předseda oborové rady Teleinformatika
(dále jen „nabyvatel“)

Čl. 1
Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
   - □ disertační práce
   - □ diplomová práce
   - □ bakalářská práce
   - □ jiná práce, jejíž druh je specifikován jako .....

   (dále jen VŠKP nebo dílo)

Název VŠKP: Útoky na aktivní síťové prvky

Vedoucí/ školitel VŠKP: Ing. Michal Polívka

Ústav: Ústav Telekomunikací

Datum obhajoby VŠKP: .....

VŠKP odevzdal autor nabyvateli v*:

- □ tištěné formě – počet exemplářů 1
- □ elektronické formě – počet exemplářů 1

* hodící se zaškrtněte

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2
Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizovaní výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
   - □ ihned po uzavření této smlouvy
   - □ 1 rok po uzavření této smlouvy
   - □ 3 roky po uzavření této smlouvy
   - □ 5 let po uzavření této smlouvy
   - □ 10 let po uzavření této smlouvy

   (z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3
Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísni a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: …….

…….. ……
Nabyvatel Autor

ABSTRAKT

Tato bakalářska práce se zabývá problematikou bezpečnosti počítačových sítí. Zadáním práce jsou útoky na aktivní síťové prvky s cílem odchytu dat mezi dvěma uživateli. Ve vytvořené síti s topologií do hvězdy jsou připojené uživatelské stanice na aktivní prvek (switch). Dále jsou popsané jednotlivé způsoby útoků a význam protokolu ARP k MAC adresám. Je zde použitých několik programů, které dokáží získat a ovládnout kontrolu nad zařízením. Kvůli velkému množství aktivních prvků musí být upraven i zdrojový kód (GNU) těchto programů. Programy ARPtool a ARPoison vyžadují operační systém linux, konkrétně je zde použita distribuce Ubuntu. Program WinArpAttacker lze spustit i pod systémem Windows XP.V závěru jsou podrobně popsány a zhodnoceny výsledky odchytávání komunikace pomocí programu WireShark.

KLÚČOVÉ SLOVÁ
Počítačová sieť, ARP, ARPtool, ARPoison, WinArpAttacker, MAC, Switch, WireShark, Útok

ABSTRACT

The bachelor thesis deals with the topic of the security of computer networks. The tasks of the bachelor thesis are the attacks on active network elements with the aim of the catching of data between two users.
In the created structure with a stellate topology, the user stations have connect to the active element (switch). In the thesis, the individual ways of attacks and the significance of ARP proceedings to MAC addresses have describ. Several programmes have use in order to take control over the device. Due to a big amount of these active elements the source code of the programmes had to be alter. The work with the programmes ARPtool and ARPoison demanded the operational system Linux, in our case the distribution of Ubunt. The programme WinArpAttacker could be set off under the system Windows XP as well.
The achieved results and the description of the practical part are discuss in detail in the summary of the thesis. The result is the catching communication with the help of the programme WireShark.

KEYWORDS
Computer network, ARP, ARPtool, ARPoison, WinArpAttacker, MAC, Switch, WireShark, Attack

Ščepko, R. *Útoky na aktivní síťové prvky.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2008. 46 s. Vedoucí bakalářské práce Ing. Michal Polívka.

Prehlásenie

Prehlasujem, že svoju bakalársku prácu na tému **Útoky na aktívne sieťové prvky** som vypracoval samostatne pod vedením vedúceho práce a s použitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej bakalárskej práce ďalej prehlasujem, že v súvislosti s vytvorením tohto projektu som neporušil autorské práva tretích osôb, no hlavne som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a som si plne vedomý následkov porušení ustanovenia § 11 a nasledujúcich autorského zákona č. 121/2000 Sb., spoločne možných trestnoprávnych dôsledkov vyplývajúcich z ustanovení § 152 trestného zákona č. 140/1961 Sb.

V Brne dne ....

podpis autora

POĎAKOVANIE:

Ďakujem vedúcemu bakalárskej práce Ing. Michalovi Polívkovi za účinnú metodickú, pedagogickú i odbornú pomoc. Taktiež za ochotu spolupracovať pri realizácii mojich zapojení a za ďalšie cenné rády pri spracovaní mojej bakalárskej práce.

V Brne dňa ....
podpis autora

OBSAH

Zoznam obrázkov:

Zoznam tabuliek:

1. Úvod

V poslednom desaťročí sa zo siete prepájajúcej malú komunitu vedcov stal Internet globálnou sieťou, ktorú využívajú všetky typy ľudí. Skutočnosť, že Internet bol pôvodne plánovaný pre malý okruh ľudí s rovnakým záujmom má za následok, že sa nepočítalo s tým, že sa stane terčom útokov až v takej miere ako sa tomu deje dnes. Samozrejme, sieťové aktívne prvky sa tejto časti internetu rozhodne nevyhli a sú jeho dôležitou súčasťou.

V bakalárskej práci sa zaoberáme aktívnym sieťovým prvkom a to hlavne prepínačom (switch), jeho slabinami a tiež protokolmi ARP, TCP s ktorými pracuje tento prvok. Napadnutie sa musí posudzovať komplexne, nedá sa vytrhnúť z kontextu istý typ útoku, prípadne narušenia. Preto musí byť zohľadnená každá časť tejto problematiky. Tým sa zvýši miera zabezpečenia pred nedovoleným ovládnutím či útokom.

Úlohou tejto bakalárskej práce je hlavne zaoberať sa problematikou bezpečnosti a zabezpečenia. Zamerať sa na možnosti narušenia bezpečnosti aktívneho sieťového prvku. Taktiež použiť software, ktorý by umožňoval ovládnutie a donútenie prvku k úkonom, ktoré mu budú mnou zadané.

2. Rozdelenie sietí

Neprepínane siete

Siete, ktoré sú založené na ethernetovej technológii, je prenášaná všetka komunikácia na všetky porty aktívneho prvku, číže HUBu. HUB sám o sebe je pomerne neinteligentné zariadenie, ktoré len obnovuje prevádzku na všetky porty a rieši kolíziu paketový. ARP pakety sú doručované ku všetkým užívateľom, ktorí sú pripojený na toto zariadenie

Neprepínané siete

Siete, ktoré používajú ethernetový prepínač, ktorý sa snaží doručovať dáta len užívateľom portov, na ktorých sídlia komunikačné uzly. Veľkým plusom oproti HUBu je pokles sieťovej prevádzky tým, že vidí iba prevádzku, ktorá je adresovaná na tomto porte prepínača. Uvedené uzly nevidia prevádzku určenú na iných prepinačových portoch.
Tato metóda, ktorú prepínač používa ku stanoveniu, ako smerovať prevádzku, je úplné pasívna – udržuje cache ARP odpovedi, ktoré boli zaznamenane prechodom cez tuto sieť a smeruje prevádzku na špecifické hardwarové adresy k portom, z ktorých prichádzajú ARP odpovede. Keď je prepínač konfrontovaný s paktom, ktorý obsahuje hardwarovú adresu, ktorá sa ešte neobjavila v jeho cache, muší sa dočasné zachovať ako HUB, číže predošle tento paket na všetky porty, ktoré sú naň pripojene. Ten port, ktorý odpovie, je automaticky zaznamenaný so jeho ARP cache.

Problém nastáva, ak sa cache zaplní. Počet položiek je totiž obmedzený jej veľkosťou. Ak je teda zaplnená, paket je odoslaný na predchádzajúcu hardwarovú adresu, ma prepínač niekoľko možnosti ako zostať plne funkčným:

- prepínač sa prepne do režimu hubu pre tie adresy, ktoré nie sú v jeho cache. Je tým síce znížený výkon prepínača, ale dáta sú doručované naďalej.
- Prepínač sa môže rozhodnúť, že svoju cache pamäť jednoducho vyprázdni a bude ju obsadzovať znovu platnými dátami. Toto je vykonávané a zisťované až dovtedy, kým sa prepínač nedozvie umiestnenie všetkých portov v daných uzloch.

Nech je teda vybraná akákoľvek z týchto možnosti, prevádzka určená pre špecifické porty bude prenikať a bude viditeľná aj ostatným portom. Tu nastáva možnosť útoku pre prípadných útočníkov, ktorý môžu spustiť aplikáciu, ktorá ťaží z tohto prieniku. Zdielacia aplikácia (sniffer application) potichu odpočúva buď všetky, alebo len vybrane pakety na vedení a zaznamenáva ich pre neskorší prieskum.
Takýmto spôsobom môžu byt zaznamenávané hesla, citlivé a Ine dáta. Hacker by mohol zostať v systéme po pomerne dlhu dobu a získavať tak informácie o sieti bez toho aby bol detekovaný. To znamená, že na prepínanej sieti môže útočník získať prístup k sieťovej prevádzke v rozsahu väčšom, než by sa dalo v prvom momente očakávať a to tak, že vytvorí záplavu falošných ARP odpovediach, ktoré spôsobia, že sa zaplní cache pamäť v prepínači a ten je následné na to donútený správať sa ako hub.

2. 1. Malý historický prierez

Nie je to až tak dávno, čo sa používali v počítačových sieťach aktívne sieťové prvky typu HUB. Toto zariadenie ako také rozposielal prijaté pakety všetkými smermi, čím za prvé zahlcoval sieť a za druhé sa packet dostal aj tam kde nemal byť poslaný. Navyše ho musela odmietnuť klientska stanica, čo zaťažovalo samotné sieťové rozhranie. Zoberme si klasický príklad troch počítačov A, B, C, ktoré sú pripojené hubom. Zariadenie A odoslalo packet s požiadavkou na zariadenie B. Keďže hub nevedel, kde sa nachádza zariadenie B, poslal požiadavku aj zariadeniu C. Takto sa ľahko mohlo stať, že zariadenie C videlo celu komunikáciu medzi zariadeniami A a B.

Tento problém sa „vyriešil“ s príchodom switchov. Switch na rozdiel od hubu má v sebe tabuľku postavenú na základoch ARP. V tejto tabuľke sú uvedené MAC (Media Access Control) adresy spolu s portami, za ktorými sa nachádzajú. Ak teda na takomto switchy pošle zariadenie A požiadavku na zariadenie B, túto požiadavku zariadenie C vôbec neuvidí lebo sa k nemu nedostane. Na prvý pohľad jednoduché a geniálne. Ale ako každý protokol, aj tento má bohužiaľ svoje slabiny. Prepínané ethernet siete poskytujú vyššiu bezpečnosť než zdielené ethernet siete. Dáta sú až na výnimky prenášané len medzi komunikujúcimi stanicami na segmente, prístup k sieti aj tok dát je možné ďalej obmedziť konfigurovanými pravidlami. Napriek tomu sú prepínané siete zraniteľné útokmi [1].

2. 1. 1. Hlavné ciele útoku:

- odpočúvanie dát na LAN sieti
- odoslať falošné dáta do/zo siete
- pristúpiť bez oprávnenia do siete

2. 1. 2. Najčastejší terč útokov

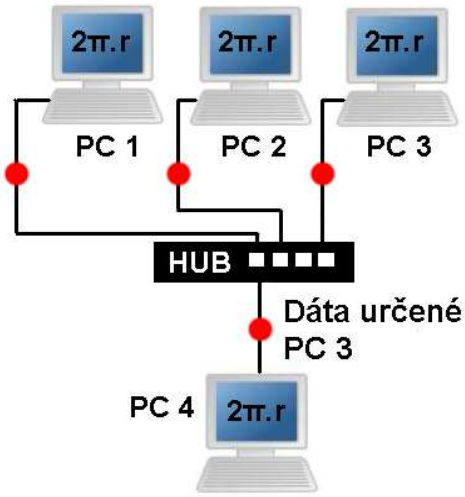
a) Zaplavovanie prepínačov v rámci s náhodne generovanými zdrojovými MAC adresami s cieľom prinútiť LAN prepínač prevádzať flooding, tj. odosielať dáta na všetky porty VLAN siete. Tento útok býva označovaný ako MAC flooding.
b) Útoky na zraniteľný ARP protokol, napríklad pomocou gratituous ARP s cieľom presmerovať dáta k útočníkovi. Ide o tzv. ARP spoofing.
c) Vloženie falošného DHCP servera do siete s cieľom predložiť užívateľom podvrhnutú bránu a menný server. Tomuto útoku môže predchádzať vyhľadávanie prevádzkového DHCP serveru.
d) Útoky na spanning tree protokol s cieľom destabilizovať sieť alebo presmerovať dáta smerom k útočníkovi.
e) Pokusy preniknúť cez virtuálnu sieť (VLAN) bez účasti smerovaču (VLAN hopping).
f) Útoky na služobné protokoly LAN prepínačov.
g) Pokusy o získanie neoprávneného prístupu do siete.
h) Pokusy o získanie neoprávneného prístupu k LAN prepínačom.

3. Aktívne sieťové prvky

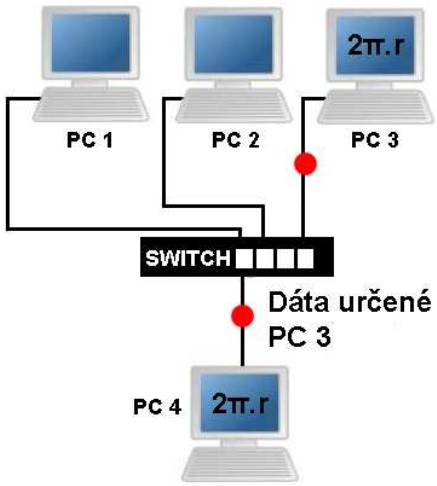
3. 1. Prepínač (Switch)

Prepínač (anglicky switch) je aktívny sieťový prvok, ktorý slúži ako „centrum počítačovej siete“. Pripája všetky zariadenia v počítačovej sieti a pracuje s dátami usporiadanými do dátových rámcov (frames). Prijaté rámce zosilní a pošle prostredníctvom kabeláže na port cieľového počítača. Tu je vidieť najväčší rozdiel medzi hubom a switchom pri spôsobe, akým posielajú rámce k cieľovému počítaču.

Každý dátový rámec je určený pre konkrétny počítač v sieti. Keďže hub nevie, na ktorý port má daný dátový rámec poslať, rozpošle ho na všetky porty – tento typ komunikácie sa nazýva aj „broadcasting“. Takto síce hub zabezpečí, že rámec sa dostane na príslušné PC, ale zbytočne zaťažuje komunikáciu v sieti tým, že rámce sa rozpošlú na všetky počítače a len „ten pravý“ ho spracuje, ostatné počítače ho ignorujú. Navyše, hub musí rozdeliť šírku komunikačného pásma (10 alebo 100 Mb/s) medzi všetky porty. To znamená, že čím viac počítačov je pripojených na hub, tým sa situácia zhoršuje – každý počítač posiela dátové rámce na hub a ten ich rozposiela znova na všetky porty (obr. 1.1)



Obr. 1.1: Tok dát cez HUB



Obr. 1.2: Tok dát cez SWITCH

Switch je sieťové zariadenie, ktoré pracuje na linkovej vrstve. Z toho vyplýva, že dáta adresuje podľa MAC adries v hlavičke linkového rámce. Keď dáta dostane, vyhodnotí, pre akú MAC adresu sú určené a dáta pošle iba na port, kde je cieľový počítač (zariadenie).Grafické zobrazenie tohto tvrdenia je vidieť na obrázku 1.2.
Aby switch vedel, na ktorý port dáta dorazia, má switch svoju vnútornú pamäť, do ktorej si zapisuje, na ktorom porte je ktorá MAC adresa. Táto pamäť je nazývaná CAM (Content Addressable Memory table) tabuľka. CAM tabuľka má obmedzenú kapacitu. Vnútorná štruktúra a reakcia tabuľky pri určitej udalosti sa líši od výrobcu k výrobcu.

Ak je switch aktívny, teda spustený, je jeho tabuľka zo začiatku prázdna (pokiaľ sme si tu nenastavili statické hodnoty). Pri príchode prvého paketu na switch sa paket pozrie na MAC adresu odosielateľa a do CAM tabuľky si uloží záznam, že táto MAC adresa leží na porte, odkiaľ paket prišiel. Switch tento paket ďalej rozošle na všetky porty okrem portu, odkiaľ paket prišiel. Pri príchode ďalšieho paketu si switch opäť poznačí adresu odosielateľa. Pozrie sa do CAM tabuľky a hľadá v nej cieľovú MAC adresu. Ak ju nájde, pozrie sa, na ktorom portu sa nachádza a tam paket pošle. Ak ju nenájde, je paket posielaný na všetky porty okrem portu, odkiaľ paket prišiel. Ak napríklad počítač zapojíme do iného portu, pri vyslaní prvého paketu bude CAM tabuľka aktualizovaná. Každý záznam v CAM tabuľke je po uplynutí nejakej doby zmazaný. Niektoré switche to môžu mať inak. Napríklad niektorý switch maže záznam až po uplynutí päťminútovej neaktivity danej MAC adresy.
Ku každému portu môže byť priradené viac MAC adries (napríklad je na danom porte pripojený switch). Ak switch dostane paket, ktorý je adresovaný pre počítač, tak je na rovnakom porte, paket už nikam neposiela (k portu je napríklad pripojený hub) [2], [3] .

3. 2. Smerovač (Router)

Smerovač (anglicky router) je sieťové zariadenie, ktoré sprostredkováva prenos dát medzi dvomi, alebo viacerými počítačovými sieťami v procese nazvanom smerovanie (anglicky routing). Smerovač prepája počítačové siete na úrovni vrstvy 3 modelu OSI.
Tento prvok je osadený dvomi alebo viacerými sieťovými rozhraniami, ktoré môžu, ale nemusia byť rovnakého typu. Smerovač analyzuje adresu každého datagramu, ktorý dostane na jednom zo svojich sieťových rozhraní od iného sieťového zariadenia a na

základe stavu sietí na iných sieťových rozhraniach rozhoduje, ktorému ďalšiemu sieťovému zariadeniu má datagram poslať, aby sa dostal bodu určenia.
Smerovače navzájom komunikujú a informujú sa o stave siete a smerovaní prostredníctvom zvláštnych komunikačných protokolov, napríklad ICMP.

Hlavnou úlohou smerovačov je vlastne zhodný s úlohou sieťovej vrstvy – teda postarať sa o doručenie paketov od ich pôvodného odosielateľa až ku konečnému príjemcovi. Smerovače teda musí prijímať rozhodnutia o tom, kadiaľ majú ďalej odoslať každý jednotlivý paket tak, aby sa dostal ku svojmu cieľu – zaisťovať to, čomu sa bežne hovorí smerovanie (routing). Rolu, ktorú ma smerovač plniť v bezpečnostnej štruktúre závisí od toho, kde je umiestnený v rámci siete, ktoré ma vzájomné spájať. Toto zariadenie sa hodí pre malé, poprípade pre siete s nižším rizikom napadnutia alebo pre sieťové segmenty [3].

3. 2. 1. Smerovač, ako zariadenie zabezpečovacieho obvodu

Úlohou smerovača je pre posielanie paketový medzi dvoma sieťovými segmentmi. Pri realizácii bezpečnostnej štruktúry sa na tuto skutočnosť často kráť zabúda. Na smerovač sa potom presúva veľa ďalších povinnosti, ktoré majú vplyv na jeho výkonnosť. Keďže väčšinou je smerovač braný do siete, bezpečnosť hrá v tejto roli podstatnú úlohu. Pravé z tohto dôvodu boli smerovane navrhnute s mnohými vstavanými bezpečnostnými prvkami ako sú napríklad paketové filtre, vlastnosti firewallu, preklad sieťových adries NAT, podpora VPN (Virtual Private Network).

Úloha smerovača, ako časti hĺbkovej ochrany sa môže pomerne dosť meniť v závislosti na tom aké prepojenia a súvislosti sa vyskytujú v celom bezpečnostnej schéme. Bolo by napríklad nerozumne, keby sa od smerovaču očakávali veci (hĺbková inšpekcia), ktoré by vykonával firewall implementovaný do siete. Väčšinou je lepšie nezaťažovať smerovač ďalšími úlohami, ale zameral sa na veci na ktoré je primárne predurčený.

3. 2. 2. Filtrácia paketov

Je jedna z veci, pre otere sa pravé použitie smerovača, keďže je schopný blokovať IP adresy určitého rozsahu. Tato silná stránka smerovača sa dá využiť vtedy, ak sa chceme toto zariadenie použiť v spolupráci s hĺbkovým (stateful) firewallom, pre filtrovanie prichádzajúcich a odchádzajúcich paketový.
Vstupne filtre sa implementujú na najvzdialenejšom bode siete, čo bude s najväčšou pravdepodobnosťou hraničný smerovač. Ak bude firewall vykonávať tuto činnosť, uľahčí a odľahčí tak činnosť na firewalle a môže tak byt využitý na činnosti na ktoré je primárne navrhnutý a prispôsobený ako napríklad hĺbkovú inšpekciu definovaných protokolov. Taktiež funkcie pre odchádzajúce pakety je vhodne premiestniť na smerovač, ktorý pracuje s ostatnými stateful firewallmy. Paketové filtre sú totiž prispôsobené pre blokovanie alebo povoľovanie prístupu z celého rozsahu siete. Vďaka členenie TCP/IP paketový a vďaka spôsobu, ktorým sa pakety porovnávajú so štandardným paketovým filtrom stačí pre zablokovanie prístupu určitému rozsahu sieťových adries jednoduché bitové porovnanie, ktoré sa dá len veľmi ťažko obísť [1], [6].

4. Spôsoby útoku

Asi najrozšírenejší typ útoku je ten, že útočník vyšle do siete nevyžiadaný (gratuitous) arp packet čím „presvedči“ stanicu, že on je tá pravá adresa kam ma ísť packet. Takto potom celá komunikácia medzi stanicou a serverom ide najprv cez útočníka. Proti tomuto typu útoku sa dosť ťažko bráni (niekedy nepomôžu ani statické arp záznamy), ale jedným z prejavov môže byť náhla zmena fingerprintu servera pri nadväzovaní SSR spojenia. Je známe, že ľudia zvyknutí na klikanie bez problémov odklepli zmenu fingerprintu (keď neviem, čo to je dám bez, veď sa ponáhľalo) a týmto spôsobom som sa dokázal nabúrať aj do inak celkom bezpečnej SSR čí scp komunikácie.
Toto sa dá riešiť uzamknutím Mac adresy na konkrétny port, vytvorením statickej arp tabuľky na switch s tým, že keď z nejakého portu začnú chodiť parkety, ktorých arp záznam nebude sedieť s mojim, tento port zablokujem.

4. 1. Predstierane (falošne) MAC adresy

Prevádzka na sieti môže byt prerušená, ak dva ethernetové adaptéry majú identické hardwarové adresy (alebo MAC – riadený prístup na médium). Tento problém by nemal nastať ak je výrobok od známeho výrobcu, pretože každý dodávateľ ma vyčlenený určitý počet adries, z ktorých vyrobenej karte priraďuje unikátnu adresu. Avšak každý ethernetový adaptér môže byt preprogramovaný na požadovanú adresu. Deje sa to síce zriedka, avšak hacker môžu preprogramovať ethernetový adaptér tak, aby sa tváril ako iný, čo vedie k strate komunikovať na oboch stranách. Ak si hacker vyberie rovnakú hardwarovú adresu ako ma miestny router, všetka vnútorná komunikácia s podsieťou bude prerušená.

4. 2. Predstierané (falošné) ARP pakety

Tak ako pri predstieraní falošnými MAC, môžu byt podobným spôsobom predstierane aj ARP odpovede používané využívané k prerušeniu, alebo presmerovaniu toku dať. Ak si predstavíme, že sypem A chce komunikovať so systémom B, takže vyšle ARP požiadavku s otázkou, čí „ktokoľvek ma IP adresu na uzle B, prosím, nech mi ju pošle.“ Hacker, ktorý sídli na zariadení C, ktorý uvidí tuto požiadavku by mohol predstierať packet s ARP požiadavkom s odpoveďou: „ja som uzol B a tu je moja IP adresa,“ ale vložil by miesto toho hardwarovú adresu uzla C. A ak nie je uzlu B zabránené v odpovedi, pravdepodobne odpovie. S ktorým systémom A vlastne ukončí komunikáciu, závisí na načasovaní odpovedi zbadaného uzla A.
Hacker môže kombinovať toto predstierane ARP s typom útoku na odopretie služby v systéme B, aby mu zabránil v odpovedi, kde možnými nastrojme ako toto vykonať môže byt záplava paketmi SYN alebo záplava správou zdroj nedostupný. Ak hacker spojí útoky uvedeným spôsobom, môže hacker zo systému C presvedčiť systém A, že systém C je skutočné systém B a nasadne podvracať systém A. Taktiež by mohol prepustiť prevádzku do systému B, takže ani A ani B by nezaznamenali niečo podozrivé. Existuje však niekoľko spôsobu, ako tomuto druhu útoku zamedziť. Jedným z nich je dsniffarpspoof (program na preposielanie dátovej cesty na útočníka) [2], [4], [12].

4. 3. Zneužívanie IP adries

IP (Internet Protokol) je nespoľahlivý protokol, používaný k doprave všetkých protokolov vyzej úrovne na internete. Nesie dôležité k doručeniu dát pri vysielaní datagramov. K ich adresácii využíva sieťovú adresu príjemcu uvedenú v záhlaví datagramu. Sieťová vrstva poskytuje službu bez spojenia. IP protokol teda nenadväzuje ani neudržuje spojenie, ani neudržuje spojenie o odoslaných dátach. Odoslaný datagram musí obsahovať informácie o adresátovi, o odosielateľovi a údaj o jeho poradí v správe.
Datagramy môžu prísť k príjemcovi v inom poradí. IP protokol nezaručuje jej doručenie, ale má záujem ju doručiť [15].
Tento protokol neposkytuje mechanizmus ako overovať, čí naozaj datagram pochádzajúci z daného uzla, pochádza naozaj z neho. V skutočnosti je to takmer neriešiteľné, pretože takmer každý uzol ma schopnosti jednať ako router, môže teda legitímne posielať prevádzku menom iného systému. Tento model však povoľuje uvedenú možnosť falšovania (predstierania) zdroja alebo cicala paketový.

4. 4. Predstierané (falšované) IP adresy

Častokrát je dôsledkom nesprávnej konfigurácie, okrem iných faktorov, aj problém predstierania falošnej IP adresy. Router, ktoré sú hraničné, nie sú častokrát nakonfigurované tak, aby odmietali prevádzku do alebo zo sieti ležiacich za nimi. Veľký poskytovatelia nemôžu účinné prevádzať filtráciu prevádzky, pretože sa neustále mení povaha adresných obsahov, ktoré obsluhujú. Tento problém je ešte ťažšie zvládnuteľný, ak pakety vstupujú do chrbticovej siete, pretože v tejto chvíli zdroj prevádzky už nie je známy a teda je považovaný za legitímny.
Súkromné adresy by nemali byt zbadane na hranici súkromného adresného priestoru a ktorá je rezervovaná pre interne použitie v nezávislých sieťach. Tieto adresy nie sú nikdy smerovane do siete internet.

4. 5. Falošný DHCP server

Väčšina klientov, či už linuxových alebo windowsových používa na svoje sieťové nastavenie údaje, ktoré získa automaticky z DHCP serveru. Bohužiaľ DHCP protokol nie je nijako zabezpečený, a preto nie je nič ľahšie ako túto skutočnosť zneužiť. Klient nevie, ktorý DHCP server používa a ak mu pridelíme správnu IP a sprevádzkujeme komunikáciu medzi nami a skutočným DHCP serverom, možno odchytávať celú komunikáciu. Stačí aby klienti mali ako predvolenú bránu práve náš falošný DHCP server (vďaka hlavne slepým Windows nastaveniam si ťažko niečo všimneme). Ak stanica posiela dáta smerom von, všetky dáta idú cez útočníka. Dáta idúce smerom zo skutočnej brány idú síce priamo klientovi, ale i túto komunikáciu je možné presmerovať. Skutočný DHCP server je možne dokonca aj vyfloodovať, podobne ako Mac tabuľku switchov predstieraním, že požiadavky prichádzajú z mnohých staníc. Kvalitnejšie a drahšie switch umožňujú ochranu priamo DHCP protokolu a to tým, že switch povolí DHCP komunikáciu len na určitý port(y). Teoreticky sa dá aj filtrovať každý DHCP packet a na základe tabuľky kde sú previazané IP s DHCP serverom takéto parkety filtrovať. Výborné možnosti v tomto poskytujú iptables v Linuxe [2].

4. 6. Zneužívanie UDP

Hlavička UDP je veľmi jednoduchá, keďže obsahuje len informáciu o zdrojovom a cieľovom porte, informáciu o dĺžke paktu a kontrolný súčet. Z toho je vidieť, že môže byt napodobnene všetko, preto je potrebne použiť autentifikáciu, ak sa ma týmto piketom dôverovať.
Kontrolný súčet pri UDP sa vypočítavá iba voliteľné. Ak ide o 16bitove pole rovne presne „0“ znamená to, že kontrolný súčet pre spomínaný prenos nebol vypočítaný a teda by nemal byt príjemcom kontrolovaný. Bez kontrolného súčtu však neexistuje spôsob ako detekovať porušenie paktu počas prenosu.
V súčasnej dobe neexistuje dôvoď na to aby tieto uzly nevyužívali kontrolne súčty, takže z toho vyplýva, že UDP pakety, ktoré majú toto pole vypnute, sú sporne a tým sa vyhýbajú detekcii. Ale je ťažké povedať, akú výhodu by z toho útočník mohol mat.

4. 7. Zneužívanie ICMP

Predstierane ICMP pakety môžu byt využite k vytváraniu situácie odopretím služby tým (Domain o Service), že sa indikácia chyb nesprávne propaguje cez sieť. Pretože ICMP je relačné nezávislý protokol, nedá sa autentifikovať, či prijatý ICMP paket skutočné pochádza z údajného uzla. ICMP chyby sa bežné považujú za dočasné, sú aktuálne len v období útočníkovho kontinuálneho predstierania chybných sieťových sprav. Ako náhle však tieto pakety prestanú dochádzať k obeti, tato porucha skončí.

4. 7. 1. Nedostupný cieľový uzol

Existuje veľká časť protokolov ICMP paketový so správou o tom, že ciel je nedosiahnuteľný. Z toho je zrejme, že ciel, čí u ide o sieť, uzol alebo špecifický port je z rozličných príčin nedosiahnuteľný. Ak uzol obdri takýto paket, ktorý hovorí, že požadovaná sieť je nedostupná, môže prestať prenášať dáta čo spôsobí narušenie komunikácie.

4. 7. 2. Uzol neodpovedá

Paket so správou, že uzol prestal odpovedať je zamietavá odpoveď, ktorá signalizuje vysielaciemu systému, že prijímateľ nie je schopný spracovávať jeho dátový tok. Ako náhle je prijatý paket takéhoto druhu, od vysielacieho uzla sa očakáva, že prevádzka k príjemcovi bude znížená. Ak nejaký útočník vie, že uzly A a B spolu bez problémov komunikujú, mohol by tuto komunikáciu narušiť vyslaním predstieraním ICMP paketový so správou, že uzol prestal komunikovať. Dokonca by bolo ďaleko účinnejšie, keby hacker tieto informácie o ne komunikácii zaslal na obidva uzly, keďže každý uzol by sa domnieval, že druha strana si Žela ukončiť komunikáciu [5].

5. Detekcia

5. 1. Systém detekcie narušenia (IDS)

Systém detekcie narušenia môže byt definovaný ako súbor nástrojov, Metod a zdrojov, ktoré nám umožňujú identifikovať, sprístupniť a hlásiť neautorizovane a neschválené sieťové aktivity. Deteguje také aktivity prevádzky, ktoré môžu, alebo nemusia byt narušené. Detekcia narušenia je jednou z časti celkového ochranného systému. Nie je to teda samostatne ochranne opatrenie.
Dalo by sa to uviesť v príklade, kde by firewall reprezentoval uzamknutú branú, systém detekcie by sme mohli brat ako zabezpečovací hlásič – alarm a strážne psy na objekte by mohla byt považovaná ako prevencia proti narušeniu.

5. 2. Typy IDS systémov:

- uzlovo orientovane systémy detekcie (Host - based intrusion - detection systém, HIDS)
- sieťovo orientovane systémy detekcie (Network - based instrusion - detection systém, NIDS)
- kombinácia týchto dvoch

HIDS vyžaduje určitý software, ktorý umiestnený na tomto systéme a môže tak skenovať aktivity, ktoré sa deju na všetkých uzloch. Zápise ľubovoľnú udalosť do bezpečnostnej databáze a overí, čí sa náhodou tato udalosť nezhoduje so záznamami chybných udalosti, ktoré sú uvedené v dosiahnutej znalostnej databáze.

NIDS sa obvykle zaraďuje so siete sériovo analyzuje sériové pakety, z ktorých potom zisťuje pripadne napadnutie. Prijíma všetky pakety v zvláštnom segmente siete, pomocou Metod. Z týchto Metod to môže byt napríklad vetvenie alebo zrkadlenie portov. Starostlivo rekonštruuje bitový tok a analyzuje z neho pripadne zavädne správanie. Väčšina týchto systémov je schopná zaznamenávať súčinnosť, generovať alebo hlásiť výstrahu pri zistení sporných prípadoch. Tieto schopnosti NDIS sú ponúkané aj na väčšine výkonných routeroch.

Hybridný IDS je kombináciou HIDS, ktoré monitorujú prevádzku na uzlovom systéme a NIDS, ktorý monitoruje sitovú prevádzku. Základný režim pre IDS je, že HIDS alebo NIDS zbiera pasívne dáta, predspracovávajú ich a klasifikujú. Podľa toho vedia posúdiť a zistiť, čí informácie spadajú pomimo ich rámec normálnej činnosti, je porovnávaný s databázou znalosti [2].

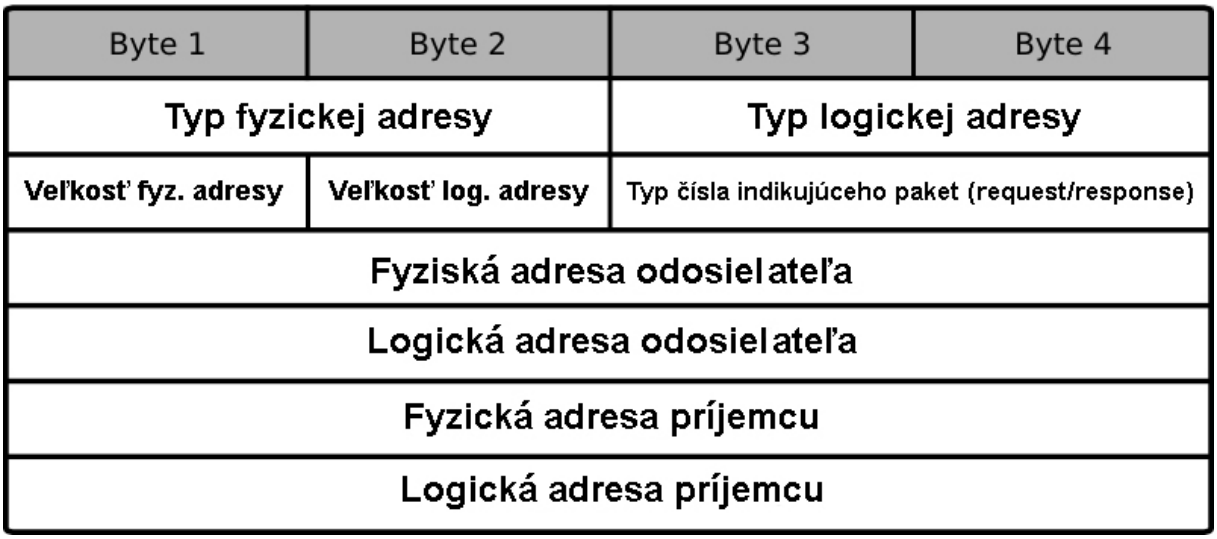
NIDS	HIDS
Široké použitie (dozerá na všetky sieťové činnosti)	Úzke použitie (dozerá iba na špecifické činnosti)
Jednoduchšie nastavenie	Zložitejšie nastavenie
Vhodnejší na detekciu externého napadnutia	Vhodnejší na detekciu interného napadnutia
Lačnejšia implementácia	Drahšia implementácia
Detekcia je založená na tom, čo môže byt zaznamenane v celej sieti	Detekcia je založená na tom, čo môže zaznamenať každý jednotlivý hostiteľ
Preskúmava hlavičku paktu	Ignoruje hlavičku paktu
Odozva v takmer v reálnom čase	Odozva je zvyčajné až vtedy, čo bol zaznamenaný podozrivý vstup v logovacom súbore
Nezávislý na operačnom systéme	Špecifický operačný systém
Sieťový útok detekujú ako dôsledok analýzy užitočnej záťaže	Deteguje lokálny útok skôr než je napadnutá vlastná sieť
Deteguje neúspešné útoky na sieť	Verifikuje úspešné, alebo zlyhane útoky

1.1 Hostiteľský alebo sériovo orientovane systémy detekcie narušenia

6. Protokoly

6. 1. Protokol ARP

Tento protokol ARP (Address Resolution Protocol**)** sa v počítačových sieťach s IP protokolom používa k získaní ethernetovej (MAC) adresy susedného stroje z jeho IP adresy. Používa sa v situácii, kedy je treba odoslať IP datagram na adresu ležiacu v rovnakej podsieti ako odosielateľ. Dáta sa teda majú poslať priamo adresátovi, u neho však odosielateľ pozná iba IP adresu. Pre odosielanie prostredníctvom napr. Ethernetu ale potrebuje poznať cieľovú ethernetovú adresu. V skutočnosti ARP nie je obmedzený len na prevod IP adries na MAC adresy, ale dokáže mapovať ľubovoľný typ logickej adresy na hocijaký typ fyzickej adresy (obr. 1.4)



Obr. 1.3: Hlavička ARP paketu

Preto vysielajúci odošle ARP otázku (ARP request) obsahujúcu hľadanú IP adresu a údaje o sebe (vlastná IP adresu a MAC adresu). Túto otázku si posiela linkovým broadcastom – na MAC adresu identifikujúci všetkých účastníkov danej lokálnej siete (v prípade Ethernetu na ff:ff:ff:ff:ff:ff). ARP dotaz neprekročí hranice dané podsiete, ale všetka k nej pripojené zariadenia dotaz obdržia a ako optimalizační krok si zapíšu údaje o jeho odosielateľovi (IP adresu a odpovedajúcu MAC adresu) do svojej ARP cache. Vlastník hľadanej IP adresy potom odošle tazateli ARP odpoveď (ARP reply) obsahujúcu vlastnú IP adresu a MAC adresu. Tu si dotazovateľ zapíše do ARP cache a môže odoslať datagram.

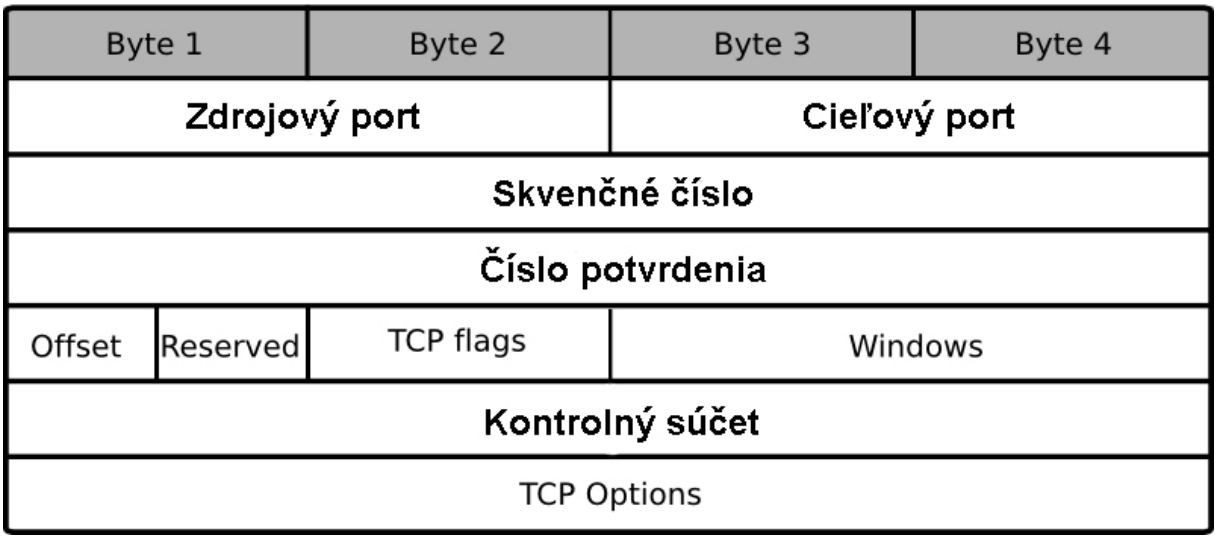
Informácie o MAC adresách odpovedajúcich jednotlivým IP adresám sa ukladajú do ARP cache, kde sú uložené do vypršania svojej platnosti. Nie je teda potreba hľadať MAC adresu pred odoslaním každého datagramu – jednou získaná informácia sa využíva opakovane. V rade operačných systémov (Linux, Windows XP) sa dá obsah ARP cache zobraziť a ovplyvňovať príkazom arp. Alternatívou pre počítač bez ARP protokolu je používať tabuľku priradením MAC adries IP adresám definovanou iným spôsobom, napríklad pevne konfigurovanou. Tento prístup sa používa predovšetkým v prostredie sa zvýšenými nároky na bezpečnosť, pretože v ARP sa dá podvádzať – miesto skutočného vlastníka hľadané IP adresy môže odpovedať niekto iný a stiahnuť tak k sebe jeho dáta.

ARP sa používa iba pre IPv4. Novšia verzia IP protokolu (IPv6) používa podobný mechanizmus nazvaný Neighbor Discovery Protocol (NDP, „objavovanie susedov“). Aj keď sa ARP v praxi používa takmer výhradne pre preklad IP adries na MAC adresy, nebol pôvodne vytvorený iba pre IP siete. ARP sa môže použiť pre preklad MAC adries veľkého množstva rôznych protokolov na sieťové vrstve. ARP bol taktiež prispôsobený tak, aby vyhodnocoval iné typy adries fyzickej vrstvy: napríklad ATMARP sa používa k vyhodnoteniu ATM NSAP adries v protokole Classical IP over ATM [11], [15].

6. 2. Protokol TCP

6. 2. 1. Stručný popis TCP protokolu

Skutočnosť, že Internet bol pôvodne plánovaný pre malý okruh ľudí s rovnakým záujmom má za následok, že sa nepočítalo s tým, že sa stane terčom útokov až v takej miere ako sa tomu deje dnes. Samozrejme TCP (Transmission Control Protocol) protokol ako časť Internetu sa týmto útokom taktiež nevyhol.
Bol vydaný v septembri 1981. Je to spojovo orientovaný protokol, ktorý spoľahlivo prenesie údaje medzi počítačmi v sieti. Spoľahlivo v tomto prípade znamená, že TCP garantuje, že všetky doručené údaje sú správne a v prípade porušenia alebo straty paketu sa prenos opakuje.
Každý TCP paket obsahuje hlavičku (obr. 1.5), ktorá je v RFC definovaná nasledovne:



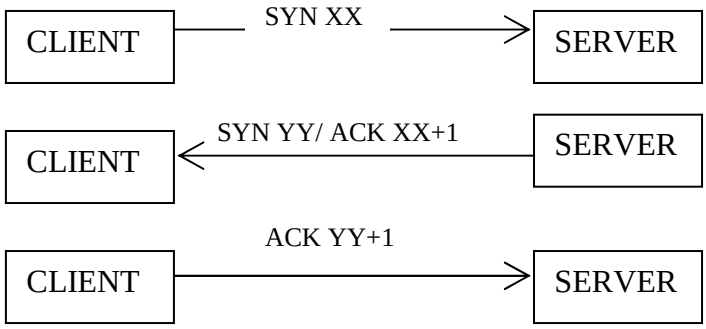
Obr. 1.4: Hlavička TCP paketu

TCP po prijatí údajov z vyšších vrstiev tieto údaje rozdelí na menšie časti – segmenty a použije IP protokol, ktorý z týchto segmentov spraví datagramy. Nakoniec sa datagramy vkladajú do sieťových paketov, ktoré môžu byť smerované v sieti.
Keď paket príde do cieľa, tak IP zásobník vo prijímajúcom uzle postupne z paketu vyberie datagram a z toho sa ďalej vyberie segment. Segment sa pošle do TCP zásobníka, kde sa potvrdí jeho platnosť. Napokon TCP zásobník môže preusporiadať všetky segmenty a poslať ich aplikácií. TCP poskytuje obojsmernú komunikáciu, takže tento proces prebieha v obidvoch smeroch [6].

6. 2. 2. Nadviazanie spojenia

TCP spojenie je zvyčajne vytvorené trojcestným „handshakingom“. Ak chce uzol A poslať paket uzlu B, tak nastaví len riadiaci bit SYN a vloží náhodne generované číslo do poľa „sequence number“ hlavičky paketu. Toto náhodne generované číslo sa nazýva ISN (initial sequence number).
Uzol B prijme paket s nastaveným bitom SYN, a teda vie, že uzol A sa snaží nadviazať spojenie a momentálne sa snaží synchronizovať sekvenčné číslo. Uzol B preto odpovie odoslaním paketu, v ktorom budú nastavený bity SYN a ACK. Uzol B generuje jeho vlastné sekvenčné číslo a vloží ho do poľa „sequence number“ hlavičky paketu. Taktiež

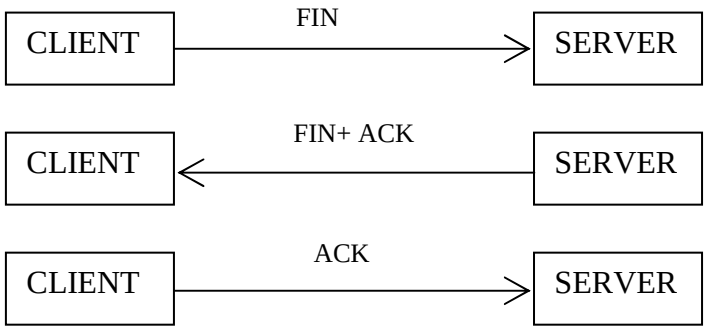
vloží do poľa „acknowledgment number“ číslo rovné ISN+1, tak signalizuje, že ďalšie sekvenčné číslo očakáva z uzla A.
Uzol A prijme tento paket s nastavenými bitmi SYN a ACK. Overí, či číslo v poli „acknowledgment“ je správne, potom vytvorí tretí paket, tento raz len s nastaveným bitom ACK. V poli „acknowledgment“ tohto paketu je sekvenčné číslo vyššie o 1 oproti predchádzajúcemu paketu, potvrdzuje uzlu B, že prijal SYN a ACK (obr. 1.6). Taktiež indikuje, ktoré sekvenčné číslo ďalej očakáva. V tomto bode je trojcestný handshaking ukončený a TCP spojenie medzi uzlami A a B je vytvorené, údaje môžu byť prenášané oboma smermi [3], [7].



Obr. 1.5: Nadviazanie spojenia v TCP

6. 2. 3. Ukončenie spojenia

Zatiaľ čo spojenie nadväzuje väčšinou klient, tak ukončiť spojenie môže ľubovoľná strana. Strana, ktorá prvá odošle TCP segment s príznakom FIN, vykonáva tzv. aktívne ukončenie spojenia (obr. 1.7), zatiaľ čo druhá strana musí vykonať pasívne ukončenie. Teoreticky je možné aj súčasné ukončenie spojenia.
Ak vykoná jedna strana aktívne ukončenie spojenia, tak už nemôže odosielať údaje (nemôže odoslať segment s príznakom PSH). Druhá strana však môže v odosielaní údajov pokračovať až dovtedy, pokiaľ sama neukončí spojenie. Medziobdobie od aktívneho ukončenia spojenia do ukončenia spojenia nazývame polo uzavretým spojením [3], [7]..



Obr. 1.6: Ukončenie spojenia v TCP

Príznaky TCP	Problém
Žiadny	Paket so žiadnym príznakom nie je ani inicializácia relácie (SYN), ani prostriedok prúdu (ACK), ani ukončenie relácie (FIN/RST). Nie je časťou žiadnej platnej transakcie TCP.
SYN/FYN	Tato kombinácii príkazov znamená, ako inicializácia relácie (SYN), tak ukončenie relácie (FIN), CO je nezmyselná podmienka
SYN/RST	Uvedená kombinácia príznakov znamená ako inicializáciu relácie (SYN), tak ukončenie relácie (RST), CO je nezmyselná podmienka
SYN/FIN/ACK	Tato kombinácia príznakov značene ako inicializácia relácie (SYN), tak prostriedok prúdu (ACK), ale aj ukončenie relácie (FIN), CO je tiež nezmyselná podmienka
SYN/RST/ACK	Zmienená kombinácia príznakov znamená ako inicializáciu relácie (SYN), tak prostriedok prúdu (ACK), ale aj ukončenie relácie (RST), CO je taktiež nezmyselná podmienka.
Všetky príznaky	Kombinácia príznakov často nazývaná „vianočný stromček“, spojuje problém príznakov inicializácie, stredného prúdu a ukončenie príznaku Peha URG (ktoré sú samé o sebe platne)

1.2 Príznaky komunikácie protokolu TCP

6. 2. 4. Princíp útoku

Ide o útok typu DoS (Denial of Services), jeho cieľom teda nie je vstúpiť do systému, získať alebo poškodiť údaje. DoS útoky spôsobujú obvykle dočasnú nefunkčnosť určitej služby napr. HTTP alebo SMTP. Podstatou SYN flooding útoku je využitie three - way handshake pri vytváraní spojenia protokolu TCP.

6. 2. 5. Postup útoku

Pri zostavovaní TCP spojenia, klient posiela na server paket s nastaveným príznakom SYN. SYN flooding nerobí nič iného, len začne odosielať veľké množstvo takýchto paketov, akoby chcel normálne komunikovať, ale nevykoná už tretiu záverečnú fázu, zostavenia spojenia. Takto dochádza na stroji, ktorý je cieľom útoku, k postupnému zaplneniu vyrovnávacích pamätí pre polootvorené spojenia.
Toto je však len základný postup útoku, ktorý má jeden nedostatok a to, že je veľmi jednoduché vypátrať odkiaľ je útok vedený a správca serveru by mohol podniknúť účinné protiopatrenia. SYN flooding bol teda doplnený o IP spoofing. Za normálnych okolností samozrejme nemožno dosť dobre falšovať IP adresu, pretože chceme, aby sa údaje vrátili späť na náš počítač. V prípade SYN flooding, keďže nestojíme o spojenie, nie je vôbec problém vydávať sa za niekoho iného. Okrem maskovacieho účelu prináša toto „vylepšenie“ ešte ďalší prvok. Keď totiž počítač, ktorého IP adresu si útočník „požičal“ na zamaskovanie svojej totožnosti, dostane paket SYN+ACK od napadnutého servera, tak odpovie RST paketom, ktorý okamžite polootvorené spojenie ukončí. Tak sa značne znižuje sila útoku, a tak si útočníci vyberajú IP adresy strojov, ktoré sú v danom čase nedostupné. Cieľ útoku je tak nútený držať polootvorené spojenie až do vypršania timeoutu, ktorý sa obvykle pohybuje okolo 75 sekúnd.

6. 2. 6. Cieľ útoku

Cieľom je dosiahnuť, aby server nebol schopný prijímať ďalšie pokusy o nadviazanie spojenia a teda sa stáva nedostupným. Horšou alternatívou môže byť úplné vyčerpanie voľnej pamäte, ak nie je obmedzený maximálny počet spojení – to s veľkou pravdepodobnosťou spôsobí „pád“ serveru s možným poškodením údajov.

6. 2. 7. Obrana

Často sa hovorí, že najlepšou obranou je prevencia. Táto metóda je potenciálne najspoľahlivejšie, ale prakticky nedosiahnuteľná. Ako už bolo spomenuté vyššie, pri SYN flooding sa používa falšovanie IP adries. Keby všetci poskytovatelia mali na svojich smerovačoch filtrovacie pravidlo, ktoré by zabraňovalo prechodu smerom von paketom so zdrojovou IP adresou, ktorá sa v tejto sieti nemôže vyskytovať, tak by bolo po probléme. Bolo by totiž výrazne jednoduchšie vypátrať aspoň približne miesto, kde sa nachádza útočník a v spolupráci s jeho poskytovateľom Internetu by sa ukončila jeho nekalá činnosť.

Riešením, ktoré v podstate nič nerieši je zväčšenie priestoru pre polootvorené spojenia. Útočník len zvýši intenzitu útokov a celá obrana je „prelomená“. Ďalšou metódou obrany by mohlo byť zníženie timeout pre polootvorené spojenie, ale aj tu platí, že zvýšením intenzity útokov, by útočník dosiahol rovnaký výsledok. Navyše pri príliš krátkom timeoute, by mohol server zrušiť aj korektné spojenia.

Podstatne účinnejšou ochranou je použitie firewallov. Firewall sa predradí serveru tak, že každé spojenie prejde cez neho. Firewall potom pri prijatí SYN paketu otvorí úplne spojenie so serverom, čím dosiahne to, že server nedrží polootvorené spojenie. Pokiaľ žiadateľ o spojenie správne dokončí nadviazanie spojenia, tak firewall funguje ako relay – teda presmeruje pakety medzi žiadateľom a serverom. Je zrejmé, že takýto firewall musí byť špeciálne zariadenie, pretože v opačnom prípade, by šlo len o presunutie problému na iný stroj.

Ďalšou účinnou zbraňou je inteligentné filtrovanie paketov. Server by v tomto prípade udržoval spojový zoznam všetkých SYN paketov za určitý časový úsek. Pokiaľ by počet SYN paketov prichádzajúcich na jeden soket presiahol určitú mieru, porovnala by sa charakteristika týchto paketov so staršími záznamami a pokiaľ by to vyzeralo podozrivo, tak by sa spojenie okamžite stornovalo. Samozrejme ani toto riešenie nie je ideálne, pretože od detekčného algoritmu vyžaduje dostatočnú „inteligenciu“ a rýchlosť.

Inou možnosťou je využitie tzv. SYN – cookies. Keď server odpovedá na prvý SYN paket, pridáva do odpovede aj iniciálne sekvenčné číslo. Server vypočíta hash číslo zo zdrojovej IP adresy, portu a ďalších údajov žiadateľa o spojenie, odošle SYN+ACK odpoveď a polootvorené spojenie úplne zahodí. Keď príjme ACK paket obsahujúci sekvenčné číslo, ktoré odpovedá tomu, čo mohol sám vygenerovať, otvorí spojenie akoby existovalo polootvorené spojenie. Aj táto metóda má svoje pre aj proti. Pozitívom je relatívna jednoduchosť a účinnosť. Nevýhodou je, že existuje (zrejme len teoretická) možnosť uhádnutia hash a tiež aj to, že výpočet hash funkcie trvá určitý čas, čo zvyšuje zaťaženie stroja. Preto sa táto metóda začína používať až v okamihu, keď sú vyrovnávacie pamäte pre polootvorené spojenie plné. Zabráni sa tak zrúteniu, prípadne nedostupnosti serveru , ale u veľmi navštevovaných serverov sa nevyhneme veľkému preťaženiu.

6. 2. 8. Skryté SYN

Mnoho implementácii TCP je náchylných k útoku vyčerpania zdrojov, ktorý je známy ako Záplava SYN, v ňom je vytváraný veľký počet relácii zaslaním veľkého množstva žiadosti o synchronizáciu, CO v konečnom dôsledku spôsobí rozvrat vo využívanej pamäti. Pri tomto druhu zátoku býva zaslaný veľký počet paketový SYN na otvorený port do fronty obsluhy ešte nevyriadenej žiadosti o spojenie. Port, ktorého sa to dotýka odošle spať paket SYN/ACK, vytvorí ďalší vstup vo fronte neobslúžených žiadosti o spojenie a Čaka na dokončenie procesu trojstranne ho odsúhlasenia. V tejto fázy je spojenie často nazývané termínom „polootvoreným“. Riadenie týmto systémom uberá systému konečne množstvo pamäte, takže veľké množstvo prijatých SYN paketový vedie k zlyhaniu dokončenia trojstranne ho odsúhlasenia a narastá množstvo spotrebovanej pamäti.
Ak predstierajú pakety SYN, že sú zo zdroja, ktorý v skutočnosti neexistuje, nie je prijatá zodpovedný paket SYN/ACK a fronty nevyriadených požiadavkou o spojenie a rozrastá.

6. 2. 9. Spätný rozptyl

Tento termín sa vzťahuje k odpovedi na SYN/ACK, ktorými odpovedá uzol, ktorý obdŕžal predstierane SYN pakety. Ak je tato zdrojová adresa pôvodného paktu predstieraná, potom bude paket SYN/ACK zaslaný pravé na tuto adresu, CO môže spotrebovať celu šírku pásma predstieraného uzlu alebo siete. Spätný rozptyl môže byť detekovaný ako záplava SYN/ACK paletou, bez toho, aby bol vyslaný prvotný SYN paket. Ak predstieraný uzol existuje, pravdepodobne odošle RST paket k predstieranému vysielaciemu uzlu, čím sa dosiahne to, že sa zvýši šírka pásma. Tento RST paket je neočakávaný boží dar pre náš predstieraný uzol, pretože môže toto spojenie uzatvoriť, uvoľniť nevybavene spojenie z fronty a tým sa vyhnúť niektorým nepríjemným efektom vyplývajúci z

6. 3. Nedostatky protokolu a jeho zneužitie

6. 3. 1. Slabiny protokolu TCP

Protokol TCP bol navrhnutý pre prostredie ARPANETu, v časoch keď ešte nikto netušil, že Internet môže byť tak nehostinné miesto. Z toho vyplývajú aj slabiny, ktoré sú navyše často znásobované nesprávnou implementáciou TCP.
Jednotlivé segmenty TCP protokolu nie sú medzi jednotlivými stanicami nijako autentizované, takže pokiaľ je možné podvrhovať pakety, tak je zároveň možné podvrhovať údaje v rámci TCP spojenia. Táto slabina sa využíva predovšetkým v útokoch typu TCP hijacking.

6. 3. 2. Možnosť odpočúvania komunikácie

Ďalší problém protokolu TCP vyplýva z absencie, akéhokoľvek kódovania prenášaných údajov. Preto je možné kdekoľvek na trase medzi komunikujúcimi počítačmi A a B odpočúvať prenášané údaje, bez toho, aby sa to ktorákoľvek strana dozvedela.
Predstavme si napr. ethernetovú sieť. Ethernet používa pre pripojené počítače jeden prenosový kanál, o ktorý sa tieto počítače delia. Hlavička ethernetového rámca obsahuje okrem iného informácie o príjemcovi. Pri bežnej komunikácii preberajú počítače z prenosového kanála len údaje pre seba. Sieťové rozhranie je však možné nakonfigurovať tak, aby prijímalo všetky údaje prenášané v danom ethernetovom segmente – tento stav sa nazýva promiskuitný mód.
V praxi je teda možné uvedením sieťového rozhrania do promiskuitného módu monitorovať všetky dátové toky, ktoré však u neanonymných služieb obsahujú aj príslušné autorizačné údaje (login, heslo).
Proti použitiu takto odchyteného hesla sa možno brániť použitím iných autentifikačných metód, napr. jedno rázové heslá, avšak ani táto metóda nie 100% bezpečná, pretože existujú spôsoby ako ju obísť.

6. 3. 3. Zneužívanie TCP

Je relatívne ťažké napodobiť a predstierať TCP spojenie pokiaľ hacker neovládne kontrolou nad routrom na trase medzi dojmy systémami. Ak by došlo na už ustanovenom TCP spojení, môže byt útočník udentifikovaný.
Existujú však iste zneužitia na sieťovej úrovni, ktoré môžu byt spustene napodobenými a predstieranými paketmi. Jedným z takýchto možnosti je spôsob, kedy môžu byť použite znovu zaslane TCP dáta alebo TCP príznaky, uzly môžu byt zasypane záplavou paletou SYN alebo spätným rozptylom. Pake ry SYN a RST môžu, aj keď by nemali, obsahovať dáta a v niektorých prípadoch sa dá predpokladať sekvenčné číslo.
Ak TCP pošle paket opäť, mal by byt identicky ako tom pôvodný. Môže byt počas prenosu fragmentovaný, ale po zrekonštruovaní dátovej časti, mala by byt totožná s pôvodným paktom. Ako náhle zbadá IDS znovu poslaný paket (s platným kontrolným súčtom), ktorý ma odlišný obsah od pôvodného paktu, potom bude predpokladať, že ide buď o chybu TCP/IP alebo ide o útok.
Starostlivá prehliadka hlavičky TCP odhalí, že paket obsahuje príznaky, ktoré sú v normálnej prevádzke vzájomné nezlučiteľné. Napríklad príznak SYN by nemal byť nikdy videný v doprovode príznaku FIN a RST, pretože účinok výslednej kombinácie nekorešponduje s akoukoľvek legitímnou činnosťou TCP zásobníka.

6. 3. 4. TCP reset

Z pohľadu bežného používateľa Internetu sa nezdá byť tento útok príliš zaujímavý, pretože pokiaľ mu niekto preruší sťahovanie obrázku z WWW stránok, tak jednoucho natiahne stránku znovu. Pokiaľ sa preruší prenášanie emailu, tak sa servery pokúsia o prenos znovu. Úplne inak sa však k tomuto problému stavajú ISP.

Hlavné smerovače, ktoré zaisťujú prepojenie s miestnymi ISP alebo zabezpečujú konektivitu do sveta, si totiž vymieňajú smerovacie informácie o topológií zvyšku Internetu pomocou protokolu BGP. Tento protokol používa pre svoj transport práve TCP

a jeho relácie nemajú trvanie pár minút, ale skôr niekoľko dní alebo dokonca mesiacov. Rozpad takéhoto spojenia má približne rovnaký dôsledok ako fyzické rozpojenie média, ktorý smerovače spája. Pokiaľ teda úspešne zaútočíme na všetky BGP spojenia daného ISP, tak si môžeme byť istý, že na moment neprejde z jeho siete do zvyšku Internetu ani paket. Pokiaľ sa bude tento útok niekoľkokrát opakovať, tak je pravdepodobné, že ostatné smerovače ohodnotia túto sieť ako nestabilnú a na pár minút ju vyradia zo svojich smerovacích tabuliek. Tento problém už pocíti aj bežný používateľ.

Hoci o samotnej možnosti útoku sa vie už veľmi dávno, pôvodne sa predpokladalo, že v prípade použitia hrubej sily bude musieť útočník vyskúšať každú kombináciu sekvenčného čísla (2^{32}). Napríklad Sean Convery a Matthew Franz vo svojej práci o bezpečnosti protokolu BGP vypočítali, že taký útok by trval 142 rokov.

6. 3. 5. Princíp útoku

Hlavnou myšlienkou pri útoku TCP reset je falošne ukončiť vytvorené spojenie. Predstavme si, že máme vytvorené spojenie medzi uzlami A a B. Teraz tretí uzol C, vytvorí falošný paket, ktorý ma rovnaký zdrojový port a IP adresu uzla A, cieľový port a IP adresu uzla B a navyše aj aktuálne sekvenčné číslo aktívneho TCP spojenia medzi uzlami A a B. Útočník nastaví RST bit vo sfalšovanom pakete, takže keď ho uzol B prijme, tak automaticky ukončí spojenie. Toto spôsobí „denial of services“, dovtedy pokiaľ spojenie nie je obnovené.

6. 3. 6. Cieľ útoku

Aplikácie a protokoly, ktoré potrebujú dlhšie trvajúce spojenia sú najzraniteľnejšie. Príkladom môže byť Border Gateway Protokol (BGP). BGP je zraniteľné práve preto, lebo sa spolieha na dlhodobo udržiavané spojenie medzi uzlami. Ak sa spojenie preruší, tak obnovenie spojenia zaberie určitý čas a dovtedy môže vzdialený uzol spôsobiť „route flapping“. „Route flap“ je akákoľvek zmena v smerovaní, ktorá spôsobí, že smerovacia tabuľka sa zmení. Smerovač, ktorý pravidelne stráca spojenie a spôsobuje „route flapping“ na ostatných BGP smerovačoch bude mať problém s aktualizáciou svojej smerovacej tabuľky. Keďže BGP je použitý na mnohých hraničných smerovačoch, tak to môže mať negatívny vplyv na mnohých koncových používateľov.

6. 3. 7. Postup útoku

Zrušiť prebiehajúce TCP spojenie nie je pre útočníka bez možnosti odpočúvania vôbec jednoduchá záležitosť. Potrebuje poznať 5 údajov:

- IP adresu prvého smerovača
- IP adresu druhého smerovača
- Zdrojový port
- Cieľový port
- Sekvenčné číslo TCP spojenia

Prvé dva údaje sú útočníkovi známe, pretože vie, medzi kým chce komunikácie prerušiť. Cieľový port tiež nie je žiadnym tajomstvom, pretože BGP nadväzuje spojenie na porte 179. Zdrojovým portom by malo byť náhodné číslo v rozsahu 1 - 65 535. V praxi sa ale všetky porty použiť nedajú a niektoré čísla nie sú pre tento účel použiteľné. Aj tak by ale ostávalo dosť možností, ktoré by musel útočník vyskúšať.

Bohužiaľ väčšina OS nevolí tento port náhodne, ale sekvenčne od predikovateľného začiatku. Watson uvádza, že napríklad firmware IOS 12.0(8) najpoužívanejších CISCO smerovačov nadväzuje spojenie z portu 11778 a každé ďalšie potom z portu čísla zvýšeného o 512. Pretože smerovače zvyčajne nadväzujú veľmi málo TCP spojení, tak je šanca na odhad tohto čísla veľmi vysoká, pravdepodobne by stačilo vyskúšať 50 možností.
Posledným potrebným údajom pre úspešné vykonanie TCP reset je sekvenčné číslo TCP spojenia. Ako už bolo spomenuté vyššie, tak to nie je neprekonateľný problém.

6. 3. 8. Obrana

Nanešťastie neexistuje 100% ochrana, ale je niekoľko spôsobov, ako útočníkovi sťažiť prácu.

1. Znížiť veľkosť okna pri BGP spojení a upraviť firmware tak, aby zdrojový port vyberal celkom náhodne. To útočníkovi značne predĺži čas potrebný na prerušenie spojenia.
2. Ďalší spôsob spočíva v mechanizme popísanom v RFC - 2385. BGP je prenášané TCP protokolom, kde ku každému paketu pridáva smerovač signatúru, aby bolo možné overiť, že sa jedná o autentický paket. Nevýhodou tohto riešenia je, že zaťažuje už aj tak dosť vyťažené centrálne procesory smerovačov a je možné, že paradoxne otvorí cestičku k inému útoku.
3. Posledná možnosť je filtrovať BGP pakety tak, aby smerovač zabránil útočníkovi vôbec posielať tieto pakety. V miestach kde je na jednom segmente viac BGP smerovačov je ale potrebné, aby filtrovanie nasadili všetky pripojené smerovače.

7. Skenovanie portov

Skenovanie portov je činnosť, pri ktorej sa zisťuje, či určitý TCP/ UDP port je otvorený alebo uzavretý. Pod označením otvorený port sa myslí taký port, pre ktorý existuje program, ktorý odpovedá na zaslané dotazy. Keď pre daný port nie je spustený žiadny program, tak je port uzavretý. Je tu ešte možnosť, že pre náš port existuje program, ktorý odpovedá na dotazy, ale rozhoduje sa komu odpovie a komu nie. Keď sa teda pokúsime zistiť, či je náš port otvorený alebo uzavretý, tak sa môže stať, že z jedného počítača sa dozvieme, že je otvorený a z iného, že nie je. Takýto port sa označuje za filtrovaný.

Skenovanie portov nie je nelegálna činnosť. Každý program, ktorý používa TCP/UDP protokoly pre sieťovú komunikáciu si najprv oskenuje port, na ktorom chce komunikovať, aby zistil či vôbec existuje možnosť na komunikáciu. Jedným z programov, ktorý možno použiť na skenovanie portov je program *nmap*.

7. 1. Porty

TCP protokol umožňuje udržiavať viac spojení v jednom čase. Toto je možné vďaka portom. Port je unikátne celé číslo v rozmedzí od 1 do 65 535. Pre väčšinu TCP spojení platí, že cieľový port je známy ako ustanovujúca hodnota. Napr. web prehliadač sa zvyčajne pripája na port 80, poštový klient posiela email na port 25. Dôvodom prečo sa klienti pripájajú na špecifický port je, že vedia, že serveri na týchto portoch počúvajú.
TCP spojenie môže byť ale nadviazané z ľubovoľného portu, hoci istá skupina portov je rezervovaná a nedostupná pre použitie ako zdrojový port (predovšetkým porty 1-1024). V jednom čase môže pre daný port existovať len jedno odchádzajúce spojenie. Po vytvorení spojenia je kombinácia zdrojového portu, IP zdroja, cieľového portu a IP zdroja unikátnym identifikátorom, ktorý možno použiť na rozlíšenie medzi všetkými aktívnymi TCP spojeniami. Hoci toto je len malá časť z problematiky o TCP protokole, je to postačujúce na to, aby sme sa mohli venovať útokom na tento protokol [8], [10].

7. 2. Typy skenovania portov:

7. 2. 1. SYN scan

Táto technika sa niekedy nazýva aj polootvorené skenovanie, pretože sa nenaviaže plné spojenie. Pri zaslaní SYN paketu vráti počítač paket s príznakom SYN a ACK. Nmap ale už naspäť potvrdenie nepošle (pošle RST – žiadosť o ukončenie spojenia) – stačí mu to k tomu, aby vedel, že na danom porte služba počúva. K tomuto typu skenovania sú potrebné administrátorské práva, inak jadro systému nie je schopné vytvárať také pakety.

7. 2. 2. TCP connect() scan

Je to základná forma skenovania, využívaná skoro všetkými jednoduchými skenermi. Na každý port sa pokúša pripojiť cez funkciu connect(). Ak sa to podarí, tak na danom porte počúva, v opačnom prípade je port nedostupný.
Tento typ skenovania môže používať ľubovoľný používateľ, je to defaultný typ skenu. Jeho veľkou nevýhodou je, že je veľmi ľahko vypátrateľný. V log súboroch sa nachádzajú údaje o veľa službách, ktoré prijali funkciu accept() spojenie a ihneď boli ukončené.

7. 2. 3. FIN scan

Pri tomto skenovaní je poslaný na daný port TCP paket s príznakom FIN, ktorý hovorí, že náš počítač chce ukončiť spojenie. Keďže je prenos TCP duplexný, čaká sa ešte na odpoveď zo skenovaného počítača. Ten pošle odpoveď, pokiaľ je daná port otvorený, inak nedostane nič.

7. 2. 4. Null scan

Paket null scanu, nemá nastavené žiadne príznaky. Pretože pri normálnom spojení je nastavený minimálne príznak ACK, dopytovaný počítač pošle späť oznámenie o chybe (príznakom RST). Z tohto môžeme opäť usúdiť, ktorý port je otvorený, a ktorý je zavretý.

Poznámka: Pri skenovaní FIN, Xmas a Null skenovaný počítač posiela späť pakety s príznakom RST. Microsoft si zaviedol svoj „štandard“, takže tento typ skenovania nemožno použiť na systémoch Windows95/NT. Naopak je uľahčené rozpoznávanie operačného systému a to tak, že ak nám tieto skenov oznámia, že všetky porty sú uzavreté a SYN scan ukáže niektoré otvorené, tak sa jedná o windowsový systém.

7. 2. 5. Idle proxy scan

Idle proxy scan je jednou z možností ako skryť svoju identitu pri skenovaní. Tomuto druhu skenovania sa niekedy hovorí zombie scan, pretože je potrebné, aby zariadenia, cez ktoré sa scan vykonáva neboli zaťažené. Zariadenie musí mať navyše predvídateľnú sekvenciu generovanú IPID. V lokálnych sieťach možno použiť napr. niektoré tlačiarne od HP, počítače s Windows 95.Postup scan je nasledujúci:

Na zombie pošleme SYN ACK paket a z RST paketu, ktorý dostaneme späť získame IPID. Pošleme paket na preverovaný port zariadenia, ktoré nás zaujíma. Paket má zdrojovú adresu nášho zombie. Pokiaľ zariadenie odpovie na pokus o otvorenie spojenia SYN ACK paketom (port je otvorený), náš zombie odošle RST, pretože on žiadne také spojenie nenadväzoval. Pritom sa posunie v sekvencii generovania IPID.
Ak skenované zariadenie odpovie RST (port je zatvorený), bude ho náš zombie ignorovať. Znovu skúsime zombie pomocou SYN ACK a kontrolujeme IPID v RST pakete, ak sa posúvala sekvencia raz, tak je port zatvorený, ak sa posúvala viackrát, tak je otvorený.
Pokiaľ budeme kontrolovať logy, tak na zombie aj na cieľovom stroji to bude vyzerať, akoby sa skenovali navzájom.

7. 2. 6. ACK scan

Jedná sa o pokročilejšiu metódu, ktorá sa obvykle používa k detekcii firewallov. Môže sa dozvedieť, či je náš firewall stavový alebo obyčajný paketový filter. Stavový firewall posudzuje len prvý paket – pokiaľ ten prejde, tak prejde všetko, inak neprejde nič. Paketový firewall blokuje prichádzajúce ACK pakety.
Pri tomto scane nmap odošle na zvolený port SYN paket s náhodným *Sequence* a *Acknowledge* číslom. Ak príde ako odpoveď RST, tak je port nefiltrovaný. Ak nepríde nič, alebo dôjde ICMP unreachable, je port pravdepodobne filtrovaný. Týmto typom scanu, nemožno nikdy dostať otvorený port, slúži hlavne k testovaniu.

7. 2. 7. Window scan

Tiež sa jedná o mierne pokročilú techniku a podobá sa technike ACK scan. Rozdiel je len v tom, že dokáže detegovať aj otvorené porty. Platí to však len pre isté operačné systémy, ktoré určitým špecifickým spôsobom reagujú na určité nastavenie veľkosti okna [2], [7].

7. 3. Ochrana proti skenovaniu portov

V súčasnosti existuje na trhu množstvo programov, ktoré dokážu detegovať pokus o skenovania portov na počítači. Jedným z takýchto nástrojov je aj program Portsentry. Tento program deteguje pokusy o skenovanie vybraných portov nášho systému a dokáže v závislosti na konfigurácií okamžite reagovať. Sú dostupné tri typy akcii:

- Zabránenie ďalšiemu pokusu o spojenie zo stroja, z ktorého sken pochádza (zablokovanie spojenia pomocou firewallu alebo úpravou smerovacej tabuľky)
- Zápis detegovaných pokusov do logu
- Vykonanie ľubovoľného vopred špecifikovaného príkazu.

7. 3. 1. Portsentry ponúka tri režimy detekcie scanov

V režime *Classic mode* sa portsentry po štarte naviaže na zvolené TCP a UDP porty a čaká na prichádzajúce spojenie. V tomto režime deteguje len „connect“ skeny. V režime *Enhanced Stealth mode* sa naviaže na špecifikované porty pomocou bind(), ale pre analýzu spojenia používa tzv. raw socket a deteguje klasické connect(), SYN, FIN, Xmas, NULL scany a ďalšie.
V režime *Advanced Stealth Mode* portsentry najskôr zistí, ktoré porty vo zvolenom rozsahu sú otvorené a zapamätá si ich. Ďalej „stráži“ všetky ostatné porty zvoleného rozsahu. Pokiaľ je neskôr otvorený ďalší port, portsentry to zaregistruje a pokiaľ je port otvorený, tak ho ignoruje. Tak sa vyhne falošným poplachom v prípadoch, keď sa napríklad pri službe FTP otvára dátové spojenie na ďalšom porte.

7. 3. 2. Pasívna alebo aktívna ochrana

Portsentry ponúka niekoľko spôsobov, ako na detegovaný scan reagovať. Zapisovanie udalostí je riešené cez syslog(). Záleží teda na konfigurácii syslogu či pre logy portsentry vyhradíme zvláštny súbor alebo necháme udalosti zapisovať štandardným spôsobom.
Keďže portsentry ponúka možnosť pri detegovaní skenu vykonať určitý príkaz, môžeme to využiť napr. k poslaniu e-mailu, SMS správy a pod. Dôležitým aspektom je možnosť zablokovať prístup ku vzdialenému stroju, z ktorého sken prichádza a to buď úpravou smerovacej tabuľky alebo pomocou firewallu. Je však potrebné si uvedomiť, že automatické blokovanie IP adries potenciálne útočníkovi umožňuje navonok zneprístupniť služby poskytované takto chráneným systémom (DOS útok). Ako náhle útočník zistí, že je jeho prístup blokovaný, môže server zahltiť pakety s podvrhnutými zdrojovými IP adresami, a tak znemožniť prístup na server z použitých adries. Ak teda na poskytujeme kritickú službu, je rozumnejšie porty len monitorovať a na jednotlivé prípady reagovať individuálne [8], [16].

8. Útoky

Útok zvyčajne nasleduje až po skenningu, keď útočník nadobudne určité znalosti o sieti, spozná topológiu a získa prehľad o službách v sieti. Útoky môžu mať mnoho cieľov. V tejto práci bude cieľom odopretie služby (DoS), odpočúvanie sieťovej komunikácie (MAC flooding) a vydávanie sa za niekoho iného (DNS cache poison, ARP cache poison).

8. 1. Denial of Service

Dalo by sa to preložiť ako zamedzenie alebo odoprenie služby. Typickým cieľom útoku sú webové servery, ktoré po uskutočnení tohto útoku nie sú schopné spracovávať požiadavky od klientov, t. j. nezobrazujú web stránky. Ďalším príkladom sú počítače v sieti, ktoré s počítačmi v iných sieťach (Internet) komunikujú cez jediný prístupový bod – bránu. Ak ju bude útočník systematicky zahlcovať nezmyselnými požiadavkami, nebude stíhať plniť požiadavky od legitímnych užívateľov. Tým môže dôjsť k úplnému znefunkčneniu tejto služby alebo aspoň k jej celkovému spomaleniu. V konečnom dôsledku to užívateľ pocíti „pomalým“ spojením. Špeciálnym (a viac používaným) prípadom DoS útoku je DDoS – Distributed Denial of Service. Ako názov napovedá, ide o rozloženú formu tohto útoku. Pod pojmom rozložená si čitateľ môže predstaviť viacero útočníkov (počítačov), ktoré spolupracujú na zahltení služby. Takto vedený útok má obrovskú silu, čo dokazuje fakt, že takto boli zneprístupnené webové stránky, napr. amazon.com alebo microsoft.com.

8. 1. 1. SYN attack

Funguje na podobnom princípe ako SYN scan popísaný na strane 24. Útočník vyrobí TCP paket so SYN príznakom, často s falošnou zdrojovou adresou. Server odpovie SYNACK paketom, ktorý však útočník ignoruje. V prípade falošnej zdrojovej adresy je paket zahodený príjemcom. Spojenie sa nachádza opäť v polootvorenom stave. Útočník pošle dostatočne veľa paketov na to, aby vyčerpal počet pripojení, ktoré môže
server naraz obsluhovať. Takto zahltený server nemôže prijímať požiadavky od obyčajných klientov.
Príklad obsahuje jeden TCP SYN paket, ktorý je posielaný na cieľovú adresu v nekonečnej slučke. To je dosiahnuté pomocou parametru loop funkcie send().

```
tcpSYN=IP(dst="google.com")/TCP(dport=80, flags="S")
send(tcpSYN, loop=1)
```

8. 1. 2. Smurf attack

Smurf attack neútočí na stanicu obete priamo, miesto toho používa broadcasting. Zostrojí
ICMP echo request paket so zdrojovou adresou obete. Cieľová adresa je broadcastová adresa nejakej siete. Ako náhle sa paket dostane do tejto siete, všetky stanice odpovedia ICMP echo reply paketom na zdrojovú adresu, na adresu obete. Pokiaľ bude útočník

neustále posielať takéto pakety, dôjde na strane obete k úplnému zahlteniu komunikačnej linky. Kód útoku je opäť veľmi jednoduchý.

```
smurf =IP(src="target.com", dst="192.168.1.255")/ICMP()
send(smurf, loop=1)
```

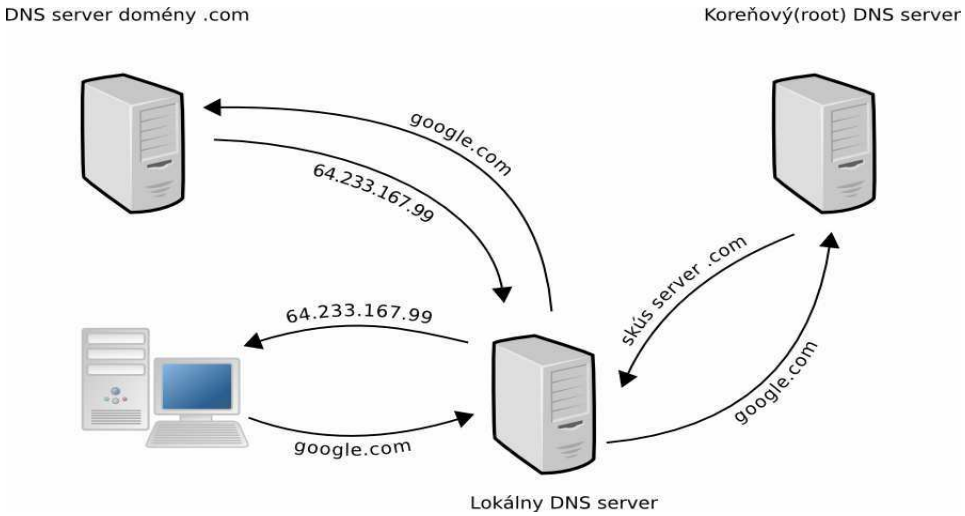
8. 1. 3. Broadcast storm

Funguje podobne ako Smurf attack, s tým rozdielom, že nezahlcuje IP adresu ale rovno celú sieť. Rovnako ako v prípade Smurf útoku, útočník vysiela ICMP echo request pakety so zdrojovou aj cieľovou adresou broadcastovej adresy siete. Ak paket dorazí do siete, hneď sa rozpošle všetkým staniciam. Každá stanica odpovie ICMP echo reply paketom na broadcastovú adresu, teda každému. Týmto spôsobom dôjde k silnému obmedzeniu priepustnosti siete. Kód je veľmi podobný ako v predchádzajúcom príklade.

```
storm=IP(src="192.168.1.255", dst="192.168.1.255")/ICMP()
send(storm, loop=1)
```

8. 2. DNS cache poisoning

DNS (Domain Name Service), je služba, ktorá prekladá názvy domén na IP adresy, napríklad google.com preloží na 64.233.167.99. DNS servery sú organizované hierarchicky, teda ak DNS server nevie preložiť názov, opýta sa svojho nadradeného servera (obr. 1.8).
Užívateľov počítač nepozná IP adresu google.com, tak sa spýta DNS servera. Vo väčšine prípadov lokálneho DNS servera na sieti alebo DNS servera poskytovateľa internetového pripojenia. Ak ani ten nepozná IP adresu, pýta sa koreňového (root) servera. Ten odporučí spýtať sa DNS servera domény najvyššej úrovne, v tomto prípade .com. Tento server pozná odpoveď, takže pošle IP adresu lokálnemu DNS, ktorý ju prepošle užívateľovmu počítaču. Ak počítač dostane od nadradeného DNS servera odpoveď, uchová si ju na určitú dobu, aby sa nemusel zbytočne pýtať.



Obr. 1.7: Preklad domény na IP adresu

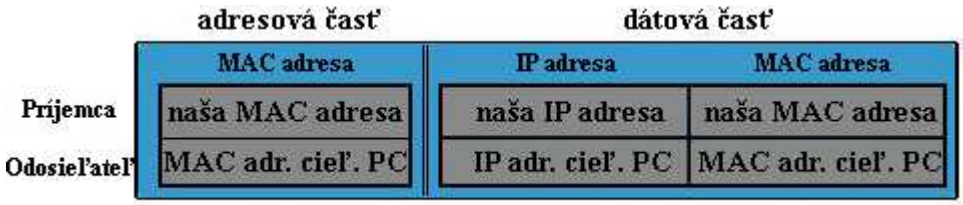
Obeť pošle svojmu DNS serveru požiadavku na preloženie názvu ebanka.sk. Útočník požiadavku spozoruje a vyrobí DNS paket, ktorý obsahuje informáciu o tom, že názvu ebanka.sk odpovedá IP adresa 123.123.123.123. Ako zdrojovú adresu uvedie adresu DNS servera a pošle paket obeti. Pokiaľ útočníkov paket príde k obeti skôr než od servera, útok je úspešný.
Po úspešnom útoku názov ebanka.sk odpovedá IP adrese 123.123.123.123. Útočník môže na tejto adrese vyrobiť výzorovo dokonalú kópiu webovej stránky ebanka.sk. Užívateľ vôbec nepozná, že je na falošnej stránke a zadá svoje prihlasovacie údaje. V tomto okamihu má útočník všetky informácie, aby sa mohol prihlásiť do skutočnej ebanky pod užívateľovou identitou.

8. 3. ARP Cache poisoning

Táto technika útoku využíva slabiny, ktoré sú obsiahnuté v ARP protokole. Cieľom tejto techniky je dostať dáta niekoho iného k nášmu počítači, aby boli pre nás dostupné.

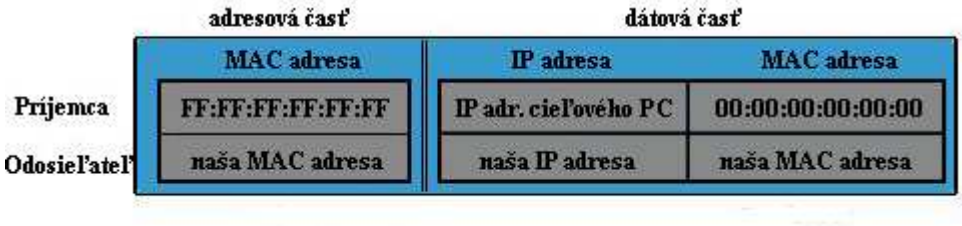
Táto technika sa dá využiť v sieti, kde je spojovacím zariadením switch. Switch pracuje na linkovej vrstve a adresuje pomocou MAC adries. Každý switch má vnútornú pamäť, do ktorej si ukladá MAC adresy zariadení v sieti a priraďuje si k nim porty, ku ktorým sú tieto zariadenia pripojené. Táto skutočnosť je podrobne obsiahnutá v kapitole, ktorá sa zaoberá samotnými aktívnymi prvkami v sieti. Tá sa nachádza na strane číslo 8.

Protokol ARP primárne slúži k prekladu IP adries na MAC adresy, ale môže slúžiť aj pre iné preklady). Medzi IP adresou a MAC adresou nie je žiadna matematická spojitosť, preto tento preklad nemôže prebehnúť výpočtom. Preklad nastane vtedy, ak chceme komunikovať s počítačom, s ktorým switch predtým nekomunikoval. Najčastejšie je známa iba IP adresa cieľového počítača, ale pre adresáciu je potrebná aj MAC adresa. Počítač preto pošle paket, ktorý sa nazýva ARP Request, ktorý pre adresovanie používa iba MAC adresy, keďže ide o protokol, ktorý pracuje na linkovej vrstve. V dátovej časti tohto paketu sa nachádzajú štyri položky pre adresy. ARP Request paket slúži ako žiadosť, aby sa ozval počítač, ktorý má svoju IP adresu rovnú IP adrese príjemca v dátové oblasti („IP adr. cieľového PC“) paketu. Na obrázku 1.9 je zachytené, ako bude vyplnených všetkých šesť adries v pakete žiadajúceho o ARP preklad.



Obr. 1.8: Príklad vyplnenia adresy pri pakete žiadajúceho o ARP preklad

MAC adresa v tvare FF-FF-FF-FF-FF-FF je špeciálna a ide o broadcastovú adresu. Keď má nejaký paket tuto adresu nastavenú, je poslaný switchem na všetky porty a zachytia ho všetky počítače. Každý počítač, ktorý tento paket dostane, ho z analyzuje a porovná IP adresu príjemcu (IP adresa cieľového PC) v dátovej časti zo svojou. Ak ale nenastala zhoda paket zahodí a ďalej sa mu už nevenuje. Ak ale zhoda nastane, počítač odpovie späť nášmu a nastaví v pakete všetky adresy (obr. 2.0).



Obr. 1.9: Príklad vyplnenia adresy pri ARP Reply

Počítač, ktorý prijal paket si z odpovede vytiahne MAC adresu odosielateľa z dátovej časti (MAC adr. cieľového PC – z dátovej časti, tá z adresnej časti sa môže počas cesty paketu zmeniť). Teraz však počítač vie, akú MAC adresu má daný počítač. Aby nemusel tento preklad robiť vždy, urobí si záznam do pamäti, že k určitej IP adrese patrí určitá MAC adresa. Tento záznam je uložený počas určitej doby v pamäti, potom je vymazaný a preklad prebehne znovu. Tejto pamäti hovoríme ARP Cache. Protokol ARP bol navrhovaný v dobách, kedy sa o bezpečnosť veľmi nestaralo, preto nemá žiadne ochranné mechanizmy. Naše situácia je takáto: máme sieť troch počítačov prepojených switchem, máme počítače Útočník, Obeť a Brána. Brána je počítač, ku ktorému je pripojený Internet, a všetky počítače, pokiaľ chcú do Internetu, musí komunikovať práve cez neho.

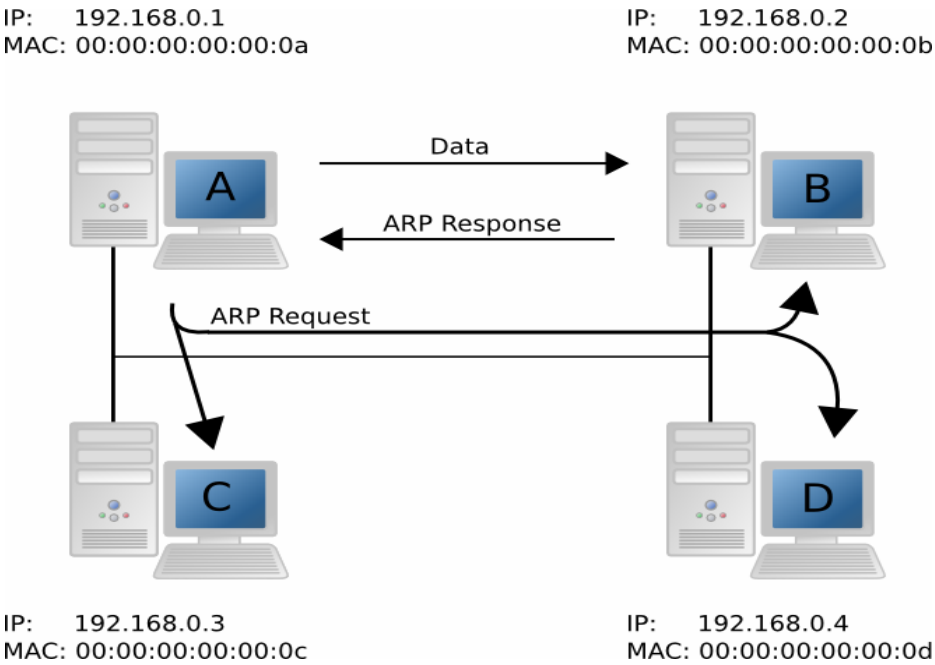
Útok spočíva v tom, že Obeti pošleme paket, v ktorom dostáva informáciu, že Brána má MAC adresu rovnakú ako Útočník. Ďalej pošleme paket Bráne, že obeť má MAC adresu rovnakú ako Útočník. Tím docielime, že počítače pre vzájomnú komunikáciu budú dosadzovať MAC adresu Útočníka a switch pošle dáta nám. My si dáta prezrieme a pošleme ich ďalej, avšak teraz už vyplníme správnou MAC adresu. Útok sa nám podaril, pretože protokol ARP si vôbec nestráži, či o tieto dáta žiadal alebo nie. Ako náhle už má záznam vytvorený, akýmkoľvek paketom ARP Reply mu môžeme záznam zmeniť. Jediná podmienka ktorá musí byť splnená, je, aby už bol záznam (priradený MAC adresy k IP adrese), ktorý chceme zmeniť, v cieľovom počítači vytvorený. Toto však taktiež nie je problém, pretože môžeme poslať napríklad ping s falošným odosielateľom. Počítač dostane tieto dáta a sám si uloží odosielateľovu IP a MAC adresu. K úplnom dokončeniu chýba len to, aby neuplynula doba, počas ktorej sú dáta uchovávané v ARP Cache (táto doba záleží na operačnom systéme). Toho docielime tým, že tieto falošné ARP Reply pakety budeme posielať napríklad každých desať sekúnd.

Pre hromadné otrávenie ARP Cache sa používajú taktiež takzvané ARP Gratuitous (zbytočné) Reply pakety. Ide o normálny ARP Reply, ktorému nepredchádzal ARP Request a MAC adresa príjemca je nastavená na FF:FF:FF:FF:FF:FF . Preto ho prijme všetky počítače na sieti, a pokiaľ majú pri IP adrese obeti už záznam, zmení ho na našu MAC adresu.

Existujú taktiež ARP Gratuitous Request (je to dotaz na svoju IP adresu). Tento paket sa posiela napríklad pri zmene našej IP adresy alebo spustenie operačného systému. Slúži k zisťovaniu, či nie je táto IP adresa už obsiahnutá [11], [12], [14].

Vzorová sieť (obr. 2.1.) obsahuje štyri stanice, ktoré sú pripojené k lokálnej sieti. Každá stanica má vlastnú IP aj MAC adresu. Stanica A chce poslať dáta počítaču B, avšak nepozná jej fyzickú adresu. Pošle ARP Request všetkým zariadeniam v sieti. ARP Request znamená niečo v zmysle „Kto má IP adresu 192.168.0.2, nech pošle svoju MAC adresu na 192.168.0.1“. Žiadosť zachytí každá stanica, no odpovie len tá stanica, ktorej IP adresa sa zhoduje s adresou v žiadosti. Odpoveď („Ja, 192.168.0.2, mám MAC adresu 00:00:00:00:00:0b“) pošle stanici A. Stanica A si do svojej ARP tabuľky uloží záznam

o tom, že IP adrese 192.168.0.2 prislúcha MAC adresa 00:00:00:00:00:0b. Následne môže stanica A posielať dáta stanici B. Platnosť údajov v tabuľke je časovo obmedzená. Ak stanica A dlho nekomunikovala so stanicou B, tak by „zabudla“ jej MAC adresu a musela by ju horeuvedeným spôsobom znovu získať.
Veľkou nevýhodou ARP protokolu je, že nepoužíva žiadnu autentizáciu, čo znamená, že na ARP Request môže odpovedať hocikto a nikto to nezistí. Ďalší bezpečnostný nedostatok spočíva v tom, že stanica príjme ARP Reply bez toho, aby predtým posiela-



Obr. 2.0 Princíp ARP

la ARP Request. Tieto dve vlastnosti zneužíva útok zvaný ARP cache poisoning [14]. Útočník, stanica D, pošle stanici A ARP Reply paket s IP adresou stanice B ale svojou MAC adresou. Stanica A si aktualizuje ARP tabuľku, čo spôsobí, že všetky pakety určené pre stanicu B sa dostanú k útočníkovi.
Útočníkovi stačí napísať tento kód aby dosiahol úspešný útok.

```
>>> ether=Ether(dst="00:00:00:00:00:0a")
>>> arp=ARP(op="is-at", hwsrc="00:00:00:00:00:0d",psrc="192.168.0.2",
pdst="192.168.0.1")
>>> ARPPoison=ether/arp
>>> sendp(ARPPoison, loop=1, inter=5)
```

8. 4. MAC flooding

V lokálnych sieťach sa používaju dva druhy zariadení na pripojenie počítačov k sieti – hub a switch. Základný rozdiel medzi týmito dvoma zariadeniami spočíva v tom, že hub posiela pakety na všetky porty, na rozdiel od switchu, ktorý vie, na ktorý port má paket poslať, aby prišiel príjemcovi. MAC flooding je spôsob ako zo switchu spraviť hub. Switch si do tabuľky ukladá dvojice hodnôt MAC:PORT, čím si pamätá, kam má čo poslať. Kapacita pamäte switchu však nie je neobmedzená, čo využíva tento typ útoku. Útočník zahlcuje switch paketmi s vymyslenou MAC adresou, čím narastá jeho

tabuľka. Pokiaľ tabuľka narastie do určitej veľkosti (blížiacej sa kapacite pamäte), tak sa switch dostane do tzv. failopen režimu. V tomto štádiu funguje ako hub, teda posiela pakety na všetky porty, čo môže využiť útočník na odpočúvanie komunikácie ostatných počítačov v sieti. Útočník sa týmto spôsobom môže dostať k citlivým dátam, napríklad k heslám v otvorenom tvare.
Nech aa:bb:aa:bb:aa:bb je fyzická adresa switchu a funkcia random_MAC() generuje reťazce obsahujúce náhodné MAC adresy. Potom kód útoku by vyzeral nasledovne:

```
>>> while True:
>>> eth=Ether(dst="aa:bb:aa:bb:aa:bb", src=random_MAC ())
>>> sendp(eth)
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
```

8. 5. DHCP Spoofing

Útok využíva faktu, že na jednej sieti môže bežať viac DHCP serverov. Ďalší fakt, ktorý tomuto útoku napomáha je, že regulérne servery nie sú „príliš“ rýchle. Vytvorením nového DHCP servera sa dá docieliť, ak obeť spustí počítač, podstrčiť údaje od útočníka. V odstrčených údajoch môžu byť dáta s falošnou gateway alebo DNS server.

Ak sa podstrčím falošný DNS server, môže vytvoriť útok zachytávajúci oba smery toku dát. Toho docielime tým, že na všetky dotazy bude odpovedať našou IP adresou a zo svojho počítača bude vytvorený fiktívny proxy server. Princíp spočíva v tom, že ak je vytvorená sieť s niekoľkými počítačmi, kde jeden beži ako DHCP server a ďalší slúži ako gateway. Keď sa počítač pripojí do siete po prvýkrát, pošle na sieť paket DHCP Discover (ide o broadcast). týmto paketom žiada, aby sa mu ozvali DHCP servery. DHCP server mu odpovedá DHCP Offer, v ktorom mu ponúka parametre. Takto odpovedajú všetky DHCP servery. Platí však pravidlo najrýchlejšieho (záleží na implementácii DHCP klienta). Klient odpovedá najrýchlejšiemu serveru paketom DHCP Request, kde oznamuje požiadavku o získanie týchto parametrov. Server mu následne pošle DHCP Ack, v ktorom oznamuje svoj súhlas s požiadavkou. Takto získal klient IP adresu. V tomto prípade by stačilo mať iba rýchlejší DHCP server a útok by sa podaril. Pokiaľ už ale počítač bol niekedy v sieti pripojený, je postup iný. Počítač pošle iba DHCP Request serveru, od ktorého naposledy získal IP adresu. V pakete žiada o svoju poslednú IP adresu. Cieľový DHCP server mu žiadosť potvrdí paketom DHCP Ack (môže ju aj zamietnuť a poslať mu inú.

Parameter „lease time“ určuje server a hovorí klientovi, ako dlho má danú IP adresu priradenú. Klient si musí vždy pred uplynutím tejto doby predĺžiť platnosť priradením. Pokiaľ tak neučiní, server si označí danou IP adresu ako voľnú a môže ju dostať niekto iný.

V predošlej popisovaný problém sa dá obísť tým, že vyčerpáme všetky IP adresy, ktoré DHCP server priraďuje. Ako náhle nemá DHCP server voľné IP adresy pre priradenie, prestane odpovedať na pakety DHCP Discover. Musí počkať nejaký čas, než uplynie doba priradenia IP adries (lease time) obsadených počítačom. Teraz, keď sa počítač spustí a bude žiadať o svoju starú IP adresu, nedostane žiadnu odpoveď alebo dostane zamietajúcu odpoveď. V tom prípade väčšina klientov pošle do siete paket

DHCP Discover a správajú sa, ako keby do danej siete neboli nikdy pripojení. V tejto chvíli nastáva opäť miesto pre falošný DHCP server, ktorý môže ponúkať napríklad zabrané IP adresy, keďže sa už nemusí obávať, že by nastal nejaký konflikt. Keď už bude mať získaných klientov, ktorých potreboval, môže útok na regulérny DHCP server ukončiť [11].

8.5 Port stealing

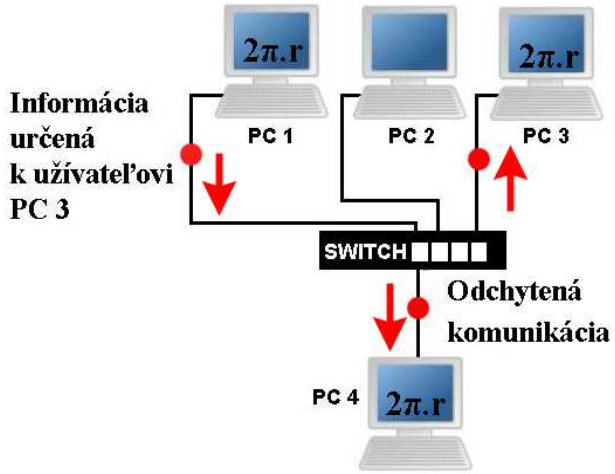
Útok tohto typu je založený na doslovnom kradnutí portov. Toho je dosiahnuté vďaka tomu, že switch si aktualizuje CAM tabuľku pri príjme paketu.

K tomuto útoku je potrebné zistiť akú má naša obeť MAC adresu. Po tomto zistení začneme posielať pakety, ktoré budú mať cieľovú MAC adresu rovnakú ako je naša MAC adresa. Zdrojová MAC adresa bude nastavená na MAC adresu obete. Switch dostane tento paket a bude si myslieť podľa toho, ako sú nastavené MAC adresy v pakete, že obeť bola pripojená na port, odkiaľ prišiel paket a preto si upraví CAM tabuľku. Keďže cieľová MAC adresa je na rovnakom porte, nebude paket posielaný ďalej. Cieľovú MAC adresu môžeme taktiež nastaviť na broadcast. Takéto nastavenie je potrebné, ak je obeť pripojená na inom switchy. keď na switch príde nejaký paket pre obeť, bude switchom poslaný k nášmu počítaču. My si ho prezrieme a aby sme ho mohli doručiť obeti, potrebujeme CAM tabuľku opraviť.

Opravenie CAM tabuľky vykonáme tak, že prestaneme posielať pakety pre ukradnutie portu a pošleme paket ARP Request. Obeť nám odpovie ARP Reply. Keď sa tento paket dostane na switch, tak sa switch opäť pozrie na zdrojovú MAC adresu a opraví si CAM tabuľku. Teraz je CAM tabuľka opravená. Zostáva už len čas, než dorazí paket ARP Reply k nám. Ako náhle dorazí, máme istotu, že tabuľka je opravená a zachytený paket pošleme obeti. Celý proces sa môže opakovať.
Tento útok má nevýhodu v tom, že keď obeť odošle akýkoľvek paket, CAM tabuľka sa obnoví. Preto musia byť poslané pakety na kradnutie portu rýchlejšie. Taktiež sa stáva, že nejaké pakety vďaka tomu nezachytíme. Je potrebné ďalej čakať, než k nám dorazí paket ARP Reply a poslať dáta ešte pred jeho prijatím. Tým sa ale riskuje, že dáta sa odošlú príliš skoro a ona sa vráti (CAM tabuľka sa nestihne zotaviť) [5], [11].

9. Praktická časť

Mojou úlohou bolo zrealizovať útok za účelom získania informácie, či dát, ktoré mali byť určené inému účastníkovi a nie mne ako užívateľovi. K tomu bolo potrebné vytvoriť alebo získať program, ktorý by bol schopný takýto druh útoku zrealizovať. Keďže mojim cieľom mal byť aktívny prvok, bol zvolený switch. Bol použitý osem portový od firmy Micronet. Bolo potrebné zistiť a overiť pravdivosť získavania dát pomocou tzv. pretekania CAM tabuľky v ktorej má prvok uložené informácie o užívateľoch pripojených na jeho porty.



Obr. 2.1: Realizovaný útok pomocou programu Arptool

V ďalšej časti som sa pokúsil otestovať útok získavania dát pomocou portov.
Pod systémom Windows som skúšal program macshift.exe. Je písaný v programovacom jazyku C++. Tento program mal náhodne generovať z externého súboru MAC adresy. Ibaže mojím nepochopením aplikácie program menil MAC adresy len na úrovni sieťovej karty. Čo v mojom prípade nebolo účelom.

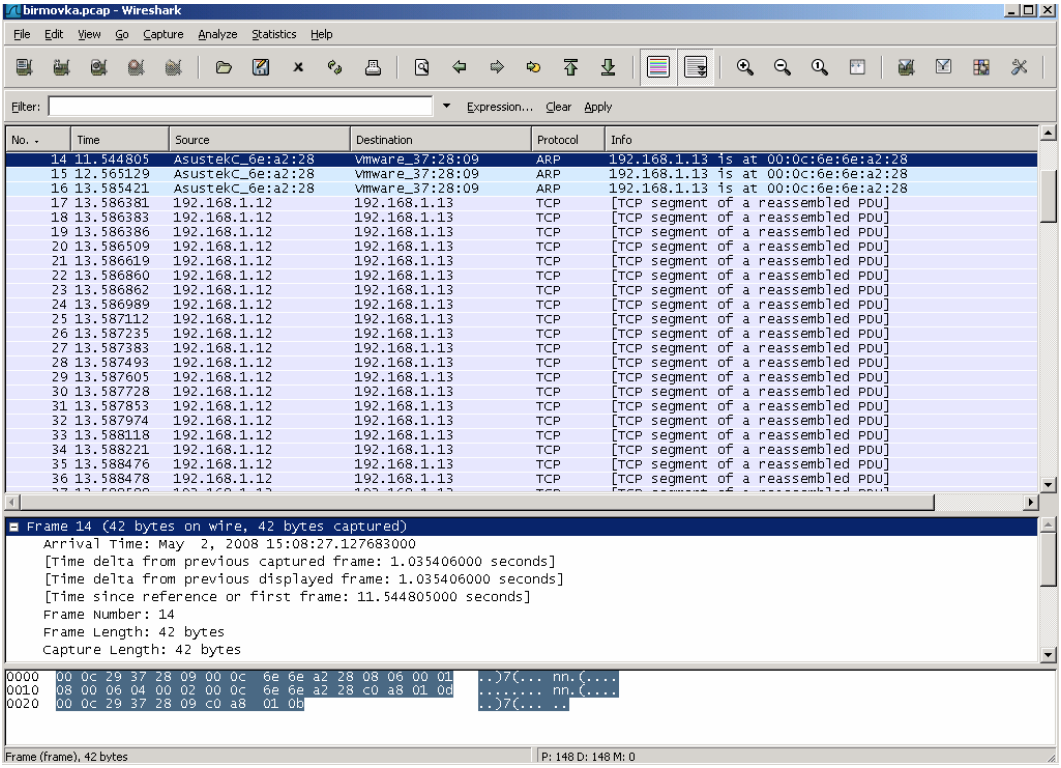
K realizácii útoku zahltením switchu MAC adresami bol použitý program WinArpAttacker.exe, ktorý je voľne stiahnutý. Tento program je užívateľsky, ale aj graficky príjemný, takže v ňom bolo možné nastaviť rôzne typy útokov ako ARP Flood, ARP Spoof ako aj útok na bránu (gate). No aj napriek niekoľkým pokusom a zmenám v nastavení sa nepodarilo dospieť k prijateľným výsledkom. K pravidelnému generovaniu náhodných MAC adries sme dospeli až programom Arptool verzie 0.0.1, ktorý sa vyznačoval ľahším ovládaním a masívnou generovanou silou MAC adries. Ten je spustiteľný pod operačným systémom na Unixovej báze, čo vyžadovalo inštaláciu linuxovej distribúcie Ubuntu verzie 7.10. Ide o program, ktorého kód je voľne šíriteľný pod licenciou GNU GPL.

Keďže bolo potrebné zohľadniť individuálnosť switchov používaných v LAN sieťach, museli byť v programe zmenené niektoré parametre. Priamo v zdrojovom kóde bolo potrebné zmeniť či upraviť rýchlosť, akým bude generovanie prebiehať. Taktiež šlo o časové hladisko a zmena milisekúnd. Keďže samotný kód, ktorým bol program vytvorený bol písaný v jazyku C. Nastavenie však muselo zohľadňovať schopnosť daného zariadenia spracovávať falošné prijímané MAC adresy. Samotná realizácia útoku prebiehala bez väčších problémov. Najskôr bolo potrebné získať IP adresu útočiaceho počítača príkazom `ifconfig -all`. Samotný príkaz bol zadaný a spúšťaný z aktuálneho adresára v tvare `./arpoison IP_SIETOVEJ_KARTY_UTOCNIKA`.

Spustenie a ovládanie programu prebiehalo pomocou terminálového okna. Tento útok neprebiehal úplne stopercentne, keďže vcelku uspokojivé výsledky bolo možné dosiahnuť pri zapnutí aktívneho prvku – switch. Po opakovanom zahlcovaní switch reagoval znížením prenosovej rýchlosti. Toto obmedzenie mohlo byť spôsobné nie len softwarovým, ale aj hardwarovým vybavením daného zariadenia. Takto testované boli dva switche. Jeden jednoduchší (Micronet SP 608k EtherFast 10/100M) s ôsmymi portami a druhý s dvadsaťštyri portový od firmy HP (HP procurve 2626 J4900A). Ten prebiehajúci útok vďaka svojmu vybaveniu znášal podstatne lepšie. Bol schopný spracovávať omnoho väčšie množstvo falošne generovaných MAC adries. Pri jednoduchšom prvku sa priepustnosť pohybovala okolo 250 až 300 MAC tabuliek, častokrát aj s väčším kolísavým rozdielom. Ak ale bol použitý podstatne zložitejší switch od spoločnosti HP, stúpol tento počet na 1500 až 2200 náhodne vygenerovaných a spracovaných MAC adries. Samotná priepustnosť mnou vytvorenej siete sa podstatne zvýšila. Ak spomeniem samotný odchyt komunikácie, v mnou vytvorenej siete, pri komunikácii dvoch počítačov, útok vykazoval určitú nespoľahlivosť. Pri tomto útoku bolo možné odchytiť dáta pri úplnom vypnutí aktívneho prvku a jeho opätovnom zapojení do zdroja elektrického napätia. Alebo pri z resetovaní tohto zariadenia. Z tohto poznatku sa dá vysloviť záver, že switch (Micronet) sa po ustálení útoku čiastočne zotavil a bol schopný ďalej spracovávať prijímané dáta v podobe falošných MAC adries. Tento stav sa však odrazil na výraznom poklese prenosu samotných dát.

Posledným útokom na switch bol program umožňujúci kradnutie portov. Princíp tohto útoku je popísaný na strane 34 pod názvom port stealing. Aj k tomuto programu bolo potrebné použiť operačný systém Linux (v mojom prípade opäť šlo o Ubuntu). K realizácii bolo potrebné vytvoriť identickú sieť ako v predchádzajúcich testoch a napojiť na aktívny prvok. Po nastavení a overení komunikácie medzi počítačmi bolo možné pristúpiť k spusteniu programu. Bol použitý program Arpoison verzie 0.6.

Ide o program kradnutia portov. Toto je dosiahnuté vďaka tomu, že každý switch a teda aj môj, si aktualizuje CAM tabuľku pri príjme paketu. V mojom prípade bolo potrebné získať MAC adresu obete. Teda užívateľa, na ktorého mal byť vykonaný útok. Po získaní tohto poznatku, bolo možné pristúpiť k samotnému útoku. Začal som pomocou programu posielať pakety, ktoré mali mať cieľovú MAC adresu rovnakú mojej MAC adrese. Teda zdrojová MAC adresa bola nastavená na MAC adresu obete. Spustenie prebiehalo z terminálového okna. Bolo potrebné zadať potrebné parametre ako: arpoison -i SIETOVA_KARTA -d IP_NASA -s IP_OBETE -t MAC_NASA -r MAC_OBETE -w 1.
Aby bolo možné sledovať odchytenú komunikáciu, bol zároveň spustený program Wireshark, pri ktorom bolo možné podrobne sledovať mnou realizovaný útok s výsledkom odchytenej komunikácie.



Obr. 2.2 Odchytená komunikácia pomocou programu Wireshark

Pri samotnom spustení programu bolo zaujímavé sledovať čas, počas ktorého nastal akýsi prvotný reakčný impulz, pri ktorom začal aktívny prvok reagovať na mnou vytvorené podnety. Pomocou zazdieľaného súboru bola simulovaná komunikácia medzi dvoma užívateľmi. V mojom prípade išlo o súbor fotografii, ktoré boli prenášané. Na odchytenej komunikácii je vidieť odchytené segmenty posielaným súborom od jedného užívateľa k druhému. Ak bol tento proces spustený ešte pred tým, ako došlo k samotnej komunikácii a prenosu dát, switch na posielané dáta zareagoval takmer okamžite. Čas, pri ktorom bolo vidieť odchytenú komunikáciu v programe WireShark, bol podstatne menší. Ak ale prenos dát už prebiehal a útok bol následne spustený, reakčný čas bol podstatne väčší.

Pokiaľ sú útoky realizované spôsobom, že paketom je nastavená cieľová MAC adresa identická s mojou, teda útočiacov MAC adresou, je tento spôsob ťažko vysledovateľný. Je však možné pozorovať nezvyčajné správanie sa stavovej LED diódy na aktívnom prvku. V jednej z diskusii bolo spomínaná možnosť zistenia útoku Port stealing, ktorý by mal byť vysledovateľný pomocou častých ARP dotazov a podľa MAC adresy pri prichádzajúcich dátach.
Spomínaný program z tejto, ale aj predchádzajúcich častí je súčasťou kompaktného disku priloženého k bakalárskej práci.

10. Záver

V dnešnej dobe existuje určite mnoho ďalších spôsobov, ako získať údaje, ktoré sú určené pre úzke skupiny, či jednotlivcov. Úlohou bakalárskej práce nie je zmapovať alebo zdokumentovať všetky typy takýchto útokov. Zameral som sa na zariadenie, ktoré je časté pri lokálnych sieťach. Ide o prvok siete – switch.

Bolo potrebné fyzicky vytvoriť a prepojiť sieť, upraviť program s voľne šíriteľnou licenciou (GNU) pre moje potreby, no hlavne z dôvodu rôznych typov týchto zariadení a taktiež teoretické znalosti získané počas prípravy bakalárskej práce overiť v praktickej rovine. Podarilo sa mi otestovať niekoľko softwarových aplikácii, ktoré sa venujú či už samotným útokom, alebo spôsobu kradnutia portov. Po otestovaní týchto aplikácii bolo možné porovnať získané výsledky a vyvodiť určitý záver, ktorý potvrdzuje mnohé poznatky z teoretickej časti tejto práce. Žiadna aplikácia či program nie sú univerzálne pre daný typ zariadenia, keďže samotné zariadenia pochádzajú od rôznych výrobcov, s inou konfiguráciou, firmwarom, ale aj hardwarovým vybavením. Tento poznatok bol viditeľný pri použití iného typu switchu, než toho s ktorým som dovtedy pracoval. V tomto prípade bolo vidieť ako sa dané zariadenia vysporiadali pri masívnom generovaní MAC adries. Mne sa podarilo overiť pravdivosť faktu pretekania CAM tabuľky, či kradnutia portov pri aktualizácii už spomínanej tabuľky. A to aj pri úprave kódu z jedného z programov, vyžadujúci si zmenu určitých parametrov. Na odchytenej komunikácii, ktorá bola určená obeti, bolo možné podrobne analyzovať dosiahnuté výsledky zo získaných dát.

Zoznam skratiek:

ARP (Address Resolution Protocol)

UDP (User Datagram Protocol)

TCP (Transmission Control Protocol)

MAC (Media Access Control)

RST (ReSeT)

BGP (Border Gateway Protokol)

ISP (Internet service provider)

IP (Internet Protocol)

DoS (Denial of Services)

HTTP (Hypertext Transfer Protocol)

SMTP (Simple Mail Transfer Protocol)

NDP (Neighbor Discovery Protocol)

IDS (Instrusion - Detection systém)

HIDS (Host - based Intrusion - Detection system)

NIDS (Network - based Instrusion - Detection system)

ICMP (Internet Control Message Protocol)

DHCP (Dynamic Host Configuration Protocol)

SSR (Scalable Source Routing)

VPN (Virtual Private Network)

VLAN (Virtual Local Area Network)

CAM (Content Addressable Memory table)

IPv4 (Internet Protocol version 4)

WWW (World Wide Web)

IPID (doplnim)

SMS (Short Message Service)

DDoS (Distributed Denial of Service)

GNU (doplnim)

LAN (Local Area Network)

NAT (Network Adress)

OSI (Open System Interconnection Reference Model)

RFC (Request For Connection)

Použitá literatúra

[1] SCAMBRAY, J., MCCLURE, S., KURTZ, G. Hacking bez záhad. Praha: Computer Press, 2005. 592 s. ISBN 978-80-247-1502-5.

[2] ENDORF, Carl, SCHULTZ, Eugene, MELLANDER, Jim. Hacking - Detekce a prevence počítačového útoku. Praha: Grada, 2005. 356 s. ISBN 80-247-1035-8.

[3] DOSTÁLEK, Libor, KABELOVÁ, Alena. Velký průvodce protokoly TCP/IP a systémem DNS. 3. vyd. Praha: Computer Press, 2002. 542 s. ISBN 80-7226-675-6.

[4] NORTHCUTT Stephen, ZELTSER Lenny, WINTERS Scott, W. RITCHEY Ronald. Bezpečnost počitačových sítí . Brno CP Books, 2005. ISBN 80-251-0697-7

[5] The Ease of Spoofing, http://www.linux.cz/noviny/2000-11/clanek10.html

[6] Filip O., Nebezpečné TCP?, http://www.lupa.cz/clanek.php3?show=3409

[7] Krause M., Noční můra jménem SYN flooding, http://www.root.cz/clanky/nocni-mura- jmenem-syn-flooding

[8] Hédl M., NMAP Populárny port scanner, http://home.tiscali.cz:8080/dingo/ref_nmap.html

[9] NMAP a jeho možnosti, http://www.actinet.cz/bezpecnost_informacnich_technologii/l19/cl7/st1/j1/Nmap_a_jeho_moznosti.html

[10] Häring D., Ochrana před scanováním portů: Portsentry, http://www.linux.cz/noviny/2000-11/clanek10.html

[11] Odposloucháváme data na přepínaném Ethernetu, http://www.lupa.cz/clanky/odposlouchavame-data-na-prepinanem-ethernetu/

[12] ARP – Tool. Obmedzenie klienta na LANke, http://blackhole.sk/arp-tool-obmedzenie-klienta-na-lanke

[13] http://www.governmentsecurity.org/archive/t2605.html

[14] Z. Trabelsi and H. Rahmani. Detection of Sniffers in an Ethernet Network. In ISC, pages 170–182, 2004. Available from World Wide Web: http://springerlink.metapress.com/openurl.asp?genre=article&issn=0302-9743&volume=3225&spage=170.

[15] RAJMIC, P. Základy počítačové sazby a grafiky. pracovní text předmětu BZSG Skripta, FEIKTVUT v Brně, 2007, 128 stran