Vyššie odborné vzdelanie pri SPŠE Zochova 9, Bratislava

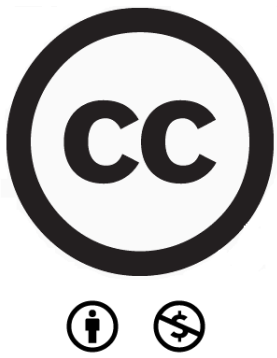
ABSOLVENTSKÁ PRÁCA

Zabezpečenie siete pomocou firewallu na operačnom systéme Linux

Vypracoval: Marián Jamrich

Konzultant: Ing. Jaroslav Abaffy

Vedúci práce: Ing. Zora Hledíková



2010

Poďakovanie

Ďakujem konzultantovi absolventskej práce Ing. Jaroslavovi Abaffymu za cenné rady a pripomienky, oponentovi mojej práce Mgr. Patrikovi Šankovi za vytvorenie podmienok na tvorbu mojej práce vo firme. V neposlednom rade sa chcem poďakovať Ing. Zore Hledíkovej za vedenie mojej práce a za cenné rady, ktoré mi pomohli pri písaní tejto práce.

Obsah

Zoznam obrázkov

1 Zoznam tabuliek

Úvod

Informatizácia vo firmách napreduje rýchlym tempom. Informačné technológie čoraz viac prenikajú do našich životov a stávajú sa našou súčasťou. Komunikačné prostriedky, medzi ktoré patrí určite Internet poskytujú veľký prínos pre komunikáciu po celom svete, ale aj veľké nebezpečenstvo, ktorému sa firma vystavuje.

Prvotnou motiváciou výberu tejto témy mojej absolventskej práce bol veľký záujem o operačný systém Linux, bezpečnostná politika počítačovej siete a písanie skriptov v interprete príkazov Bash. Cieľom práce bolo vytvoriť a realizovať firewall vo firemnom prostredí podľa mojich aktuálnych vedomostí. V tejto práci sa pokúsim čo najviac priblížiť spôsob akým som postupoval a aké som mal myšlienky pri tvorbe tejto práce. Prácu budem realizovať v Jednotnom majetkovom fonde na Odborárskom námestí č.3. Myslím si, že firewall by mal byť súčasťou každej aj malej firmy. Preto som sa rozhodol pre tento projekt.

2 Čo je firewall

2.1 Definícia

Firewall je zariadenie alebo skupina zariadení, ktoré chránia vnútornú sieť pred potenciálnymi útokmi a inými bezpečnostnými hrozbami z Internetu. Je určený na blokovanie proti neoprávnenému prístupu, ale zároveň umožňuje povolenie požadovaných služieb. V neposlednom rade chráni servery a hlavne vnútorné počítače a ich dáta pred útočníkmi.
Umožňuje teda definovať pravidlá pre obojsmernú komunikáciu z vnútornej siete a tak určovať bezpečnostnú politiku.

Iptables pracuje na úrovni kernelu t.j. samotné jadro operačného systému Linux rozhoduje o osude paketov smerov von zo siete LAN do Internetu a smerom do siete LAN z Internetu. V default konfigurácii pracuje na 1,2,3 a 4 vrstve RM OSI teda s portami, MAC a IP adresami. Keďže som chcel vyskúšať aj aplikačný firewall, musel som opatchovať kernel a pridať tak podporu pre 7. vrstvu teda filtrácia paketov na aplikačnej úrovni. Bolo treba opatchovať aj samotný IPTABLES, teda pridať podporu pre rozoznávanie paketov na úrovni protokolov.

Aby som si uľahčil robotu, tak som si napísal 2 skripty v bashi:

- Prvý skript slúži na pridávanie a odoberanie pravidiel vo firewalle na základe MAC a IP adresy.
- Druhý skript sú samotné pravidlá firewallu a kód na interaktívne spúšťanie skriptu.

Samotný server je postavený na platforme AMD, procesor Athlon 64 3200+ pracujúci na frekvencii 2000 MHz, operačná pamäť 1GB DDR1, HDD 200GB SATA.

2.2 Dôvody, prečo dáta treba dôkladne chrániť

Dáta sú výsledkom práce používateľa, často jedinečné, obsahujú množstvo investovanej práce, majú vysokú úžitkovú hodnotu, ich strata môže byť spojená s veľkými škodami. Zverejnením či sprístupnením nepovolaným osobám môžu tak vzniknúť morálne alebo aj materiálne škody (citlivé údaje súkromného charakteru, obchodného charakteru).

2.3 Pred čím firewall nechráni vnútornú sieť?

Firewall samozrejme nechráni vnútornú sieť tam, kde komunikácia cez firewall neprebieha. Zamestnanci vo firme môžu prenášať údaje na CD/DVD médiách, uploadovať na FTP serveri (pokiaľ je FTP povolené) a pod.

2.4 Pár spôsobov ako obísť firewall

Za jednu z najväčších nevýhod považujem, keď je možné do vnútornej sieti firmy bez väčších problémov pripojiť akékoľvek zariadenie napr. Wi-Fi router. Akokoľvek dobrý firewall sa tu obíde veľmi jednoducho, pretože zamestnanec môže priniesť do firmy už nakonfigurovaný Wi-Fi router a tak sa útočník môže veľmi ľahko dostať do vnútornej siete bez toho, aby išiel cez firewall. Niektoré veľké organizácie ako je napr. Národná banka Slovenska používa rušičky elektromagnetických vĺn, čiže sa nemožno pripojiť do siete z vonka. Menej účinné riešenie budem popisovať v mojej práci. Tým je napr. prístup cez firewall iba povoleným IP a MAC adresám alebo lepšie, kombináciou oboch adries. Ďalšia hrozba je nový IP protokol verzie 6, ktorý je prítomný už v systéme Windows XP, Vista a najnovšie aj Windows 7, ktorý je štandardne povolený a často sa na to nemyslí. Ide o to, že pokiaľ nemáme definované pravidlá vo firewalle pre IP ver.6, tak pred útokom cez takýto firewall nechráni prakticky nič. Takto môže útočník získať prístup priamo na pracovné stanice v sieti.

Ďalšou nebezpečnou metódou ako obísť firewall či už z vnútornej siete je použitie SSH tunelovania. Takto sa ľahko obíde firewall tzv. „bypassom“.

Dôležité je si uvedomiť to, že firewall poskytuje ochranu iba medzi dvoma sieťami od seba navzájom, nechráni jednotlivé uzly jednej podsiete od útokov z tej istej podsiete.

Pokiaľ získa útočník prístup na jeden uzol v sieti, získava tak prístup do celého uzla a teda aj napr. do celej siete. Riešením je rozdeliť sieť do jednotlivých subnetov (bezpečnostných domén), čím rozdelíme jednu sieť na viacero a znížime tým riziko zneužitia jednotlivých počítačov. Ak útočník získa prístup do jedného subnetu, tak pri dobrej konfigurácii firewallu sa nedostane do ďalších podsietí, čo je veľká výhoda.

3 Základné predstavenie protokolov a komunikácie

3.1 Čo je NAT

NAT (Network Address Translation alebo aj Network Masquerading) teda preklad verejnej IP adresy na lokálnu a naopak je spôsob úpravy sieťovej premávky cez smerovač (router) prepisom východiskovej a/alebo cieľovej IP adresy, často i zmenu čísla TCP/UDP portu pri prichádzajúcich IP paketoch tzv. presmerovanie portov (port forwarding).

3.2 Čo je routing

Routing (smerovanie) je proces, ktorý je potrebný v lokálnej sieti pre správne doručenie prenášaných paketov k cieľu. Toto smerovanie vykonáva smerovač, čo je špeciálne zariadenie pracujúce na 1,2,3 a 4 vrstve RM OSI.

3.3 Smerovanie vo WAN

Smerovanie vo WAN (Wide Area Network) zabezpečujú smerovače (Routery). Každý smerovač má v sebe svoju vlastnú smerovaciu tabuľku a tieto tabuľky si periodicky medzi sebou vymieňajú. Komunikujú pomocou smerovacích protokolov.

Routovanie môže byť **statické** a **dynamické**.

- Pri statickom routovaní sú smery a adresy sieti pevne definované. Hodí sa to sietí pre malé a jednoduché siete, ktorých topológia sa často nemení.

- Pri dynamickom routovaní si jednotlivé routre vymieňajú svoje routovacie tabuľky v pravidelných intervaloch. Každému routeru sa zadá len jeho najbližšie okolie a informácie o ďalších sieťach a ich vzdialenostiach získa od susedných routrov. Sieťovou vzdialenosťou sa rozumie počet sieti a podsieti, cez ktoré musí paket prejsť, aby dosiahol požadovaný cieľ.

Routovacie protokoly poznáme napr.:

IGRP je distance vektor algoritmus a protokol vyvynutý firmou Cisco. Interval obnovy smerovacích tabuliek je 90 sekúnd.

EIGRP je distance vektor algoritmus. Je novší ako IGRP. Ako metriku zohľadňuje šírku pásma (bandwidth), oneskorenie (delay), spoľahlivosť (reliability), zaťaženie (load).
RIP (ver.1) je distance vektor algoritmus teda používa jednoduchý algoritmus založený na počte skokov k cieľu. Interval obnovy smerovacích tabuliek je 30 sekúnd. Nevie pracovať s VLSM!
RIP (ver.2) vychádza z RIP verzie 1. Ako metriku taktiež používa počet skokov. Je to vylepšená verzia 1 a podporuje už VLSM.
BGP umožňuje smerovanie medzi viacerými autonómnymi systémami alebo doménami a poskytuje informácie o smerovaní a dostupnosti ostatným BGP systémom.
OSPF je link-state vektor teda algoritmus na základe stavu spojov v sieti. Používa Dijikstrov algoritmus.

4 Typy firewallov

4.1 Paketový firewall

Pracuje na **3. vrstve** OSI modelu. Je najjednoduchší typ firewallu. Sieťovú prevádzku analyzuje na základe MAC adresy, zdrojovej a cieľovej IP adresy. Nie je tu možnosť pridávať stavové pravidlá. Poskytuje menšie ochranu.

4.2 Stavový firewall

Pracuje na **4. vrstve** OSI modelu. Vie si spájať jednotlivé relácie (Session). Je rozšírenejší typ firewallu. Pracuje ako paketový filter, ale je tu podpora stavových pravidiel. Je tak možné definovať odchádzajúce pravidlo pre pakety a vstup do lokálnej siete je zamietnutý, ale vďaka stavovému pravidlu prepustí už nadviazané spojenia z lokálnej siete.

4.3 Aplikačný proxy firewall

Pracuje až na aplikačnej **7. vrstve**, ale iba v prípade podporovaných protokolov. Napr. HTTP, FTP atď. Vie blokovať aj jednotlivé príkazy, ako je PUT, GET u FTP a podobne. Typických príkladom je program Squid, ktorý je dobre použiteľný v Linuxe.

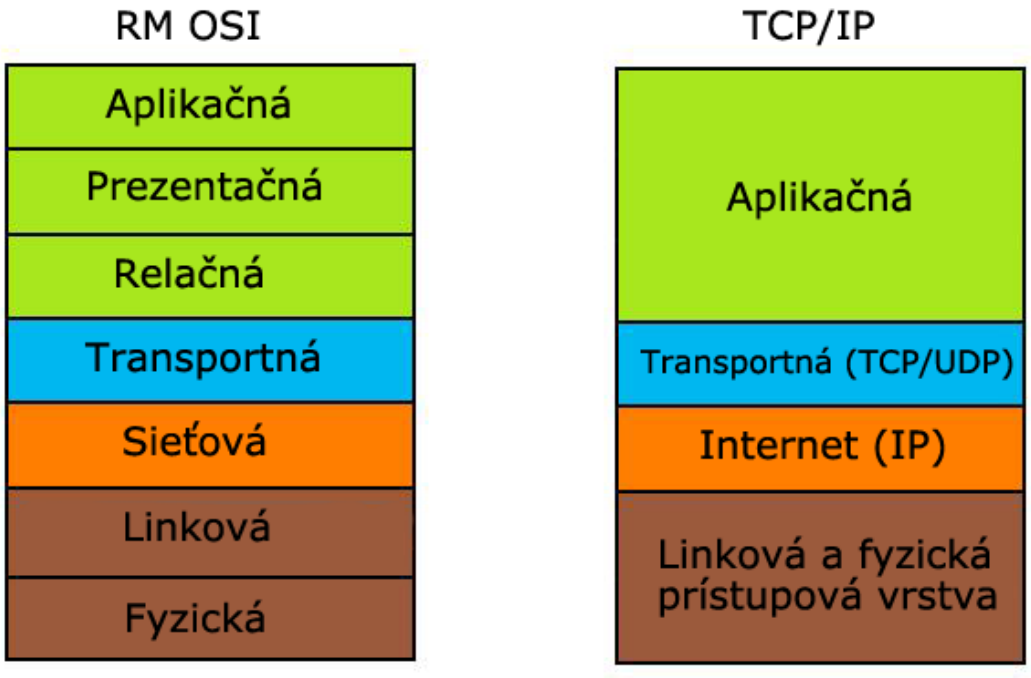
4.4 Aplikačný firewall

Pracuje na **7. Vrstve** OSI modelu. Rozhoduje na základe MAC, IP, portov a protokolov. Vie obmedziť protokoly typu P2P, rôzne sieťové protokoly, ktoré potrebujú hry, IM,

5 TCP/IP a Referenčný model OSI (RM OSI)

5.1 Transmission Control Protocol (TCP)

Je spojovo orientovaný, spoľahlivý komunikačný protokol transportnej vrstvy. Protokol vychádza z referenčného modelu OSI. Ako vidieť na obrázku, TCP/IP protokol má 4 vrstvy. RM OSI pozostáva so 7. Vrstiev.

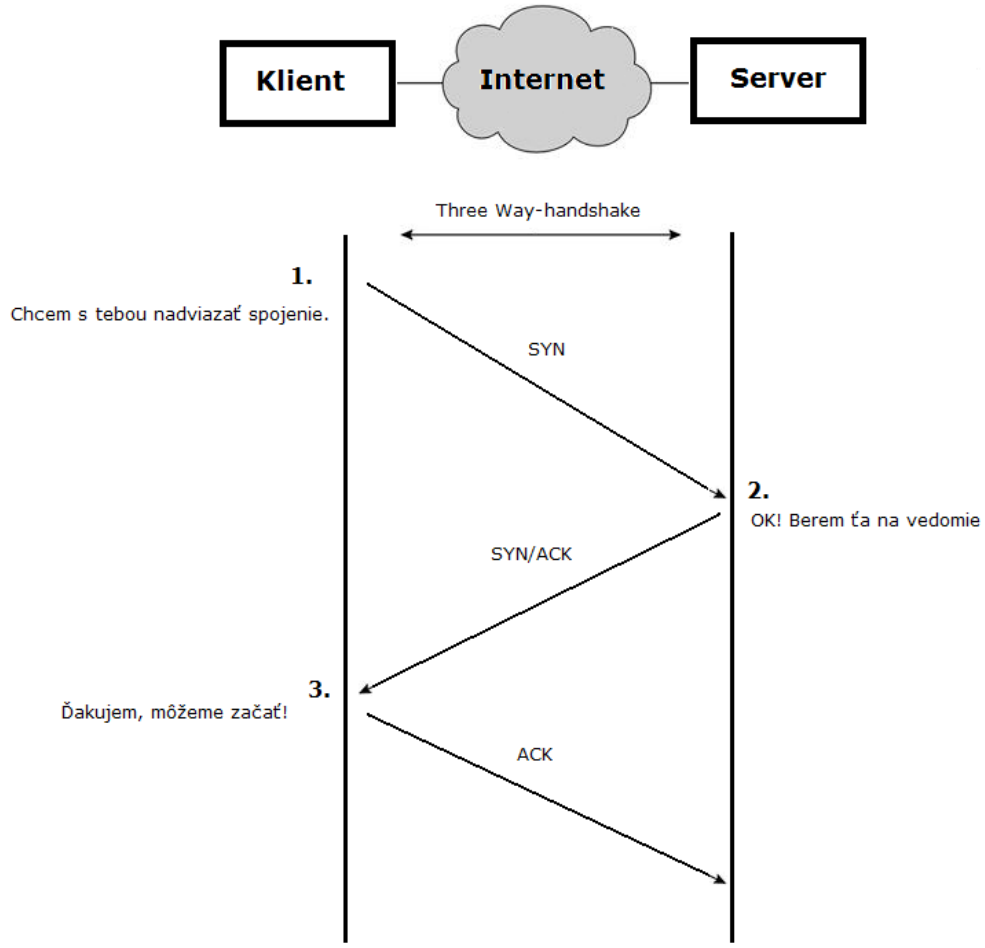


Obrázok 1 - Porovnanie TCP/IP a RM OSI

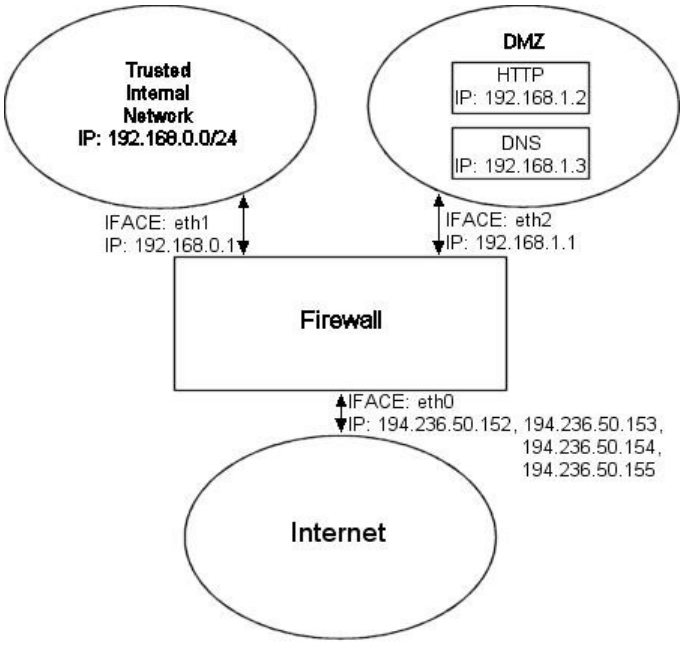
5.2 Three Way-Handshake (trojsmerné potrasenie rukou)

Skôr ako začnem písať o druhoch útokov na serveri, treba si uvedomiť ako prebieha bežná regulárna TCP komunikácia tzv. 3 Way-Handshake. Ako názov napovedá, ide o komunikáciu pozostávajúcu z troch krokov:

- zdrojový počítač vyšle TCP paket s nastaveným príznakom "SYN" (žiadosť o komunikáciu) teda synchronizačný paket
- cieľový počítač odpovedá paketom s nastavenými príznakmi "SYN" a "ACK" (pripravený na komunikáciu)
- zdrojový počítač odpovedá paketom s nastaveným príznakom "ACK" (začíname komunikovať)



Obrázok 2. – Klient/server komunikácia



Obrázok 3. – Firewall, DMZ a lokálna sieť

6 Niektoré spôsoby útokov

6.1 Význam útokov

Predovšekým slúžia tieto útoky na harvérové limity daného servera. Inými slovami, cieľom je poslať na server toľko paketov, na ktoré server nemôže odpovedať v dôsledku veľkého množstva paketov alebo kvôli pozmenenej zdrojovej IP adrese paketu.

Útokom typu DOS/DdoS sa nevyhli ani veľké internetové stránky Yahoo, Buy.com, eBay, Amazon, Datek, E*Trade alebo CNN.

6.2 DOS (Denial Of Service)

Tento útok, spočíva v zahltení cieľového servera falošnými požiadavkami na službu, kde útočník útočí na hardvérové limity servera. Výsledkom takého útoku je neschopnosť cieľového stroja obslúžiť legitímne požiadavky legitímnych klientov, pretože je zahltený a zamestnaný obsluhovaním falošných požiadaviek útočníka t.j. falošnými SYN paketami, kde server odpovedá na „spoofed“ IP adresy.
Server normálne odpovedá na legitímne požiadavky používateľa SYN/ACK a čaká na spätnú reakciu. V prípade úspešného prijatia používateľom zašle spätne odpoveď ACK/FIN a tým sa vytvorí stabilné spojenie medzi serverom a klientskym počítačom. Problém je ale v tom, že

server začne nadväzovať spojenie na IP adresy, ktoré sú falošné(spoofed) a nie je teda možné vytvoriť korektné spojenie. A tak server stále vyčkáva na odpoveď, aby mohol dokončiť toto „napoly otvorené“ spojenie, čo môže trvať aj niekoľko minút (záleží na nastavení servera a firewallu). Týmto útočník dosiahne znefunkčnenie daného servera.
Riešenie:
Jedným z mnohých riešení je analyzovať pripojenia na server z IP adresy a vyhodnocovať či ide o legitímne požiadavky z jednej IP adresy pri NAT alebo ide o útočníkovú IP adresu.

6.3 DDoS (Distributed Denial Of Service)

Útok DDos je úspešný len v tom prípade, keď počet falošných požiadaviek, ktoré dokáže útočiaci stroj poslať za jednu sekundu na cieľový stroj je vyšší, než počet požiadaviek, ktoré dokáže cieľový stroj plnohodnotne obslúžiť za sekundu. Úspešnosť útoku sa teda môže zvýšiť, ak na cieľ posiela falošné požiadavky niekoľko útočiacich strojov s nezávislým HW a nezávislým prenosovým pásmom. To je útok DDoS. V praxi sa počet útočiacich strojov pohybuje v rádovo stoviek až tisícou tzv. botnet. Tieto útočiace stroje sú spravidla pripojene do jedného botnetu a útok je teda distribuovaný a decentralizovaný, ale centrálne ovládaný a koordinovaný.

6.4 SYN Flooding (SYN záplava)

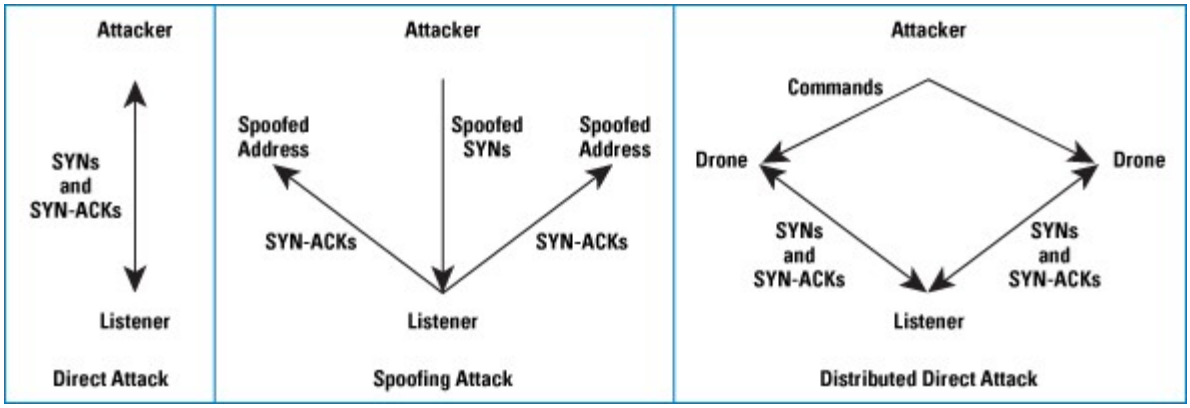
Veľmi nepríjemný druh útoku z DoS útokov je tzv. "SYN flood" teda zaplavanie cieľového servera značným množstvom synchronizačných paketov. SYN je príznak v hlavičke TCP paketu, ktorý znamená, že sa bude vytvárať spojenie. Tým sa zahltí server požiadavkami na ktoré nevie dostatočne rýchlo odpovedať.

6.5 UDP Flooding

Útok je založený na zasielaní UDP paketov s falošnou adresou odosielateľa. Keďže UDP je protokol bez nadväzovania spojenia, útočník zasiela UDP pakety na náhodné porty cieľového počítača. Ak počítač prijme UDP paket adresovaný na port, ktorý nevyužíva nijaká aplikácia, zašle odosielateľovi ICMP paket s hlásením o chybe. Ak útočník použije dostatočne veľa UDP paketov, na sieti vznikne veľká (dvojnásobná) prevádzka a počítač zamrzne.

6.6 ICMP Flood

Útočník začne posielať veľké množstvo paketov (napr. ICMP ECHO) na cieľový server. Pakety zvyknú mať falošnú IP adresu odosielateľa a často sú zasielané z niekoľkých serverov súčasne. Cieľový server začne zasielať odpovede a dôjde k zahlteniu linky.



Obrázok 4. – Normálna kominikácia, spoof attack, DD attack

7 Pravidlá firewallu IPTABLES

7.1 Pakety vieme kontrolovať v rôznych fázach

INPUT

Prijaté pakety, ktoré prešli smerovaním a sú určené nejakej lokálnej službe. Dá sa určiť rozhranie, z ktorého prišli. Pokiaľ pravidlá dovoľujú, budú predané žiadanej aplikácii.

OUTPUT

Lokálne vytvorené odchádzajúce pakety, ktoré ešte neprešli smerovaním. Dá sa zistiť, ktorým rozhraním budú zrejme odoslané. Po smerovaní pokračujú do POSTROUTING.

FORWARD

Pakety, ktoré prešli smerovaním a sú určené inému počítaču. Mali by sa poslať ďalej. Je známe vstupné a výstupné rozhranie. Pokiaľ je preposielanie schválené, pokračujú do POSTROUTING. Preposielanie sa musí povoliť priamo v jadre.

PREROUTING

Ide o nové, prichádzajúce pakety do firewallu, ktoré ešte neprešli smerovaním/routovaním. U nich je možné zmeniť cieľovú IP adresu a port. Ďalej pokračujú buď do reťaze INPUT (nie sú smerované, ostávajú v samotnom firewalle) alebo do FORWARD (sú smerované a prechádzajú z jednej sieťovej karty do druhej).

POSTROUTING

Pakety, ktoré prešli smerovaním a opustia počítač. Miesto pre zmenu zdrojovej IP adresy.

7.2 Stavové pravidlá

INVALID: paket nesúvisí s nijakým existujúcim spojením

ESTABLISHED: paket súvisí so spojením, ktoré už bolo nadviazané

NEW: paket iniciuje nové spojenie

RELATED: paket iniciuje nové spojenie, ale súvisí s už existujúcim spojením. Príkladom je napr. FTP spojenie, kedy sa okrem riadiaceho kanála vytvára nové spojenie na inom porte. Aktivujeme ho stavovým pravidlom „-m state –state **STAV**“.

7.3 Definícia osudu paketu

ACCEPT – pakety sú štandardne prijaté a ďalej spracované

REJECT – paket bude zničený a jeho odosielateľovi sa odošle informácia, že jeho paket nebol prijatý cieľovou aplikáciou/službou

DROP – paket bude úplne zničený a jeho odosielateľovi nebude zaslaná žiadna správa

DNAT - prepíše cieľovú adresu paketu na adresu definovanú voľbou **--to-destination**. Táto akcie je platná iba v tabuľke NAT v reťazci PREROUTING

SNAT - prepíše zdrojovú adresu paketu na adresu definovanú voľbou **--to-source**. Platné v tabuľke NAT v reťazci POSTROUTING

MASQUERADE - platné iba v tabuľke NAT v reťazci POSTROUTING

8 Stavové a bezstavové firewally

Paketové filtre sa dajú rozdeliť na dve skupiny. Prvú skupinu tvoria **stavové**, no a druhú **bezstavové** filtre. Líšia sa implementáciu a spôsobom vyhodnocovania paketov / spojení.

8.1 Bezstavové firewally

Takéto firewally kontrolujú každý paket nezávisle na ďalších. Sú menej náročné na pamäť.

8.2 Stavové firewally

Stavové filtre si udržiavajú informácie o spojeniach. Tieto informácie používajú na rozhodovanie o osude paketov. Majú väčšie nároky na pamäť.

9 Stratégie konfigurovania firewallu

9.1 Paranoidná

Všetko je zakázané , povolené sú iba vybrané funkcie (túto metódu považujem za lepšiu) a budem ju aj preferovať pri tvorbe pravidiel firewallu.

9.2 Benevolentná

Všetko je povolené, zakázané sú iba vybrané funkcie.

V podstate sú možné dva prístupy riadeniu prevádzky cez firewall:

1. Čo nie je povolené je zakázané
2. Čo nie je zakázané je povolené

9.3 Moja stratégia zostavovania firewallu

9.3.1 Prvý prípad

V prvom prípade sa musia definovať služby, ktoré sú nevyhnutné pre fungovanie danej firmy. Všetky ostatné služby, ktoré nie sú povolené sa dáta neprenášajú cez firewall a sú blokované. Metóda tejto implementácie je náročnejšia na zdokumentovanie prípadných hrozieb a môže sa až blížiť k obmedzovaniu užívateľov, ale je ľahšie monitorovateľná a bezpečnejšia. Postupne sa teda vyvíjajú definície pravidiel vo firewalle.

9.3.2 Druhý prípad

V druhom prípade sa identifikujú služby, ktoré sú potencionálne nebezpečné a zakážu sa. Tento prístup je flexibilnejší a vedie k menej obmedzujúci riešeniam, avšak tu je riziko zneužitia užívateľmi, pretože ak zablokujeme napr. službu SSH, tak si používateľ môže ľahko zmeniť port a tak získať prístup k svojmu serveru.

Ja budem používať „paranoidný“ spôsob riešenia firewallu. Z Internetu do siete a zo siete do Internetu zakážem všetko čo nie je povolené a postupne budem služby povoľovať podľa požiadaviek pracovníkov.

10 Kľúčové aspekty navrhnutia môjho firewallu

10.1 Filtrácia paketov na sieťovej vrstve

Ak pracuje firewall na sieťovej vrstve (IPTABLES), tak filtrujeme pakety. V tomto prípade sa vyhodnocujú hlavičky paketov pre prichádzajúce a odchádzajúce pakety.

Nevýhoda je, že takýto firewall nevie filtrovať protokoly (pozn. touto podporou sa zaoberám v mojej práci).

Výhoda takýchto firewallov je, že sú rýchlejšie ako aplikačné firewally.

10.2 Filtrácia cez proxy servery na aplikačnej úrovni

Nevýhoda takýchto firewallov je, že sú pomalšie ako paketové firewally, ale vedia rozlišovať jednotlivé protokoly a teda ich povolovať/zakazovať. Ich výhoda je aj tá, že slúžia ako buffer požiadaviek pre protokoly http, https a ftp.

10.3 Vypnutie nebezpečných služieb

Za nebezpečné služby som považoval také, ktoré prenášajú dáta a heslá v nešifrovanej forme čo je veľmi nebezpečné. Takéto služby som na serveri samozrejme vypol a používal som SSH. Medzi takéto nebezpečné služby patrí:

- telnet (slúži na vzdialené pripájanie do systému)
- rlogin (slúži na vzdialené zadávanie príkazu)
- NFS (Network File System)
- TFTP (triviálny FTP protokol)
- SNMP (Simple Network Management Protocol – slúži na zber dát zo systému)
- Finger (slúži na informácie o užívateľoch a systéme)
- X server (je grafické okno) - beží pod právami roota

10.4 Nahradenie nešifrovaných protokolov

Medzi veľmi používané protokoly patri napr. IMAP, POP, SMTP, FTP, HTTP. Tieto protokoly sa však dajú „zabaliť“ do TLS alebo SSL protokolu, ktorý obalí komunikáciu do šifrovaného tunela a takto ochráni dáta a heslá pred odchytením útočníkom.

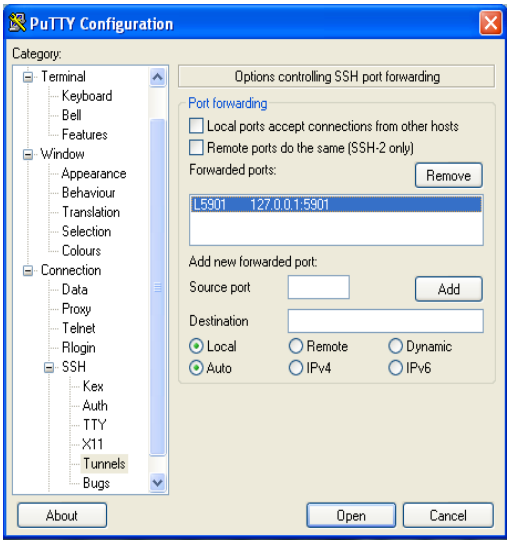
10.5 VNC cez SSH tunel

VNC je protokol, ktorý služi na vzdielané grafické pripájanie na pracovnú plochu iného počítača. Keď som sa pripájal cez VNC na môj server, uvedomil som si, že celý prenos a teda aj zadané heslo nie je šifrované, čo by mohol byť veľký problém a hocikto by mi to mohol odchytiť a dokonca aj celú komunikáciu.
Riešením je spraviť šifrovaný tunel cez SSH, ktorý je šifrovaný. Funguje to následovne:

- v programe putty som nastavil tunel s lokálnym portom 5901 a cieľovou IP adresou 127.0.0.1 a portom 5901 na ktorom počúva VNC server na mojom Linux serveri.
- spustil som samotný VNC server príkazom „vncserver :1 -localhost“ pod užívateľom admin. (:1 je displej 1 a -localhost – kde som chcel, aby počúval VNC démon)
- prihlásil som sa následne cez putty na SSH server pod užívateľom admin
- potom som sa pripojil z lokálneho PC na IP adresu 127.0.0.1 port 5901
- teraz už je zahájená šifrovaná komunikácia cez SSH tunel

10.6 Implementácia pravidiel do firewallu

Samotný skript a pravidlá firewallu sú popísané v technickej dokumentácii v prílohe B na strane číslo 33.



Obrázok 5. – Putty – nastavenie tunela

Obrázok 6. – Šifrovaný SSH tunel

Samotný SSH server počúva na neštandardnom porte 99, ktorý som zvolil, kvôli skriptovým útokom, ktoré sa dejú na Internete. V Iptables som nastavil pravidlo, ktoré hovorí o tom, že ak sa niekto pokúsi pripojiť na SSH server z inej IP adresy ako je tá moja, tak ho má ihneď zalogovať + nastavil pravidlo, ktoré hovorilo, že pokiaľ sa niekto pripojí z danej IP adresy v priebehu 120 sekúnd, tak dočasne uzavrie SSH port. Vytvoril som automatický skript, ktorý tieto logy vyhodnocuje a posiela mi ich na e-mail každý deň. Skúsil som teda nastaviť server, aby počúval na štandardnom porte 22 a v priebehu 8 hodín (od 0:00 do 8:00 rána) sa mi objavilo v emaily 7 nových IP adries, ktoré sa pokúšali dostať na môj server:

123.142.80.122 (**South Korea**), 210.17.183.83 (**Hong Kong**), 221.179.220.183 (**China**), 59.40.182.186 (**China**), 61.146.115.71 (**China**), 67.202.67.138 (**United States**), 89.173.132.27

Obrázok 7. – User friendly pridávanie pravidiel do firewallu

Samotný skript je popísaný v technickej dokumentácii v prílohe A na strane číslo 30.

Číslo porty	Protokol	Otvorené
25	smtp	stále
53	dns	stále
80	http	od 7 do 18 hod
99	ssh	od 7 do 23 hod
110	pop	stále
443	https	od 7 do 18 hod
3128	proxy	od 7 do 18 hod
5900	vnc	od 7 do 23 hod
5901	vnc	od 7 do 23 hod

Tabuľka 1 – Časová tabuľka povolených portov

11 Implementácia podpory pre 7. Vrstvu

11.1 Fedora 11-> Kernel 2.6.31, Iptables-1.4.5 + l7-filter

Niekedy je potrebné zakázať užívateľom sťahovať dáta s P2P sietí, pretože je to raj vírusov a dochádza k zahlteniu siete desiatkami otvorených spojení. Iptables ponúka riešenie, ktoré som aplikoval na serveri. Išlo o tieto kroky:

- **Stiahnutie potrebných súborov**

cd /usr/src/kernels/
wget http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.31.tar.bz2
wget http://iptables.org/projects/iptables/files/iptables-1.4.5.tar.bz2
wget http://ufpr.dl.sourceforge.net/sourceforge/xtables-addons/xtables-addons-1.18.tar.bz2
wget http://ufpr.dl.sourceforge.net/sourceforge/l7-filter/netfilter-layer7-v2.22.tar.gz
wget http://ufpr.dl.sourceforge.net/sourceforge/l7-filter/l7-protocols-2009-05-28.tar.gz

- **Rozbalenie archívov**

for archive in *.bz2; do tar -jxf $archive;done
for archive in *.gz; do tar -zxf $archive;done

- **Pridanie podpory iptables layer7 do kernelu a zapnutie**

cd /usr/src/kernels/kernel-2.6.31

 # Opatchovanie kernelu. Prepínač -p 1 znamená, že kernel je o úroveň nižšie
patch -p1 < ../netfilter-layer7-v2.22/kernel-2.6.25-2.6.28-layer7-2.22.patch

 # Vojdeme do konfigu kernelu a zapneme podporu pre iptables layer7
make menuconfig

Networking options ---->
[*] Network packet filtering framework (Netfilter) ---->
Core Netfilter Configuration ---->
Netfilter connection tracking support
[*] "layer7" match support

- **Skompilovanie a nainstalovanie modulov a samotného kernelu**

make && make modules && make modules_install && make install

- **Patchovanie a inštalácia Iptables 1.4.5**

 # Vstúpime do adresára
cd /usr/src/kernels/iptables/extensions

 # Skopírujeme knižnicedo adresára
cp ../../netfilter-layer7-v2.22/iptables-1.4.3forward-for-kernel-2.6.20forward/libxt_layer7.* .

 # Nahradíme „exit_error(“ na „xtables_error(“ v súbore „libxt_layer7.c“
sed -i 's/exit_error(/xtables_error(/' libxt_layer7.c

 # Ideme o úroveň nižšie
cd ..

 # Následuje opatchovanie iptables
patch -p1 -NE < ../iptables-1.4.2-tarpit.diff
cp /usr/src/linux/include/asm-generic/bitsperlong.h /usr/src/linux/include/asm/

 # Samotné kompilovanie Iptables a inštalácia do /usr
./configure --with-ksource=/usr/src/linux \
--prefix=/usr --with-xtlibdir=/lib/xtables --libdir=/lib --enable-libipq --enable-devel
make && make install

 # Vojdeme do adresára s layer7 protokolmi
cd /usr/src/kernels/l7-protocols-2009-05-28/

 # Nainštalujeme ich
make install

- **Xtables-addons compile**

cd /usr/src/xtables-addons
./configure --with-xtables=/lib --prefix=/usr \
--mandir=/usr/share/man --infodir=/usr/share/info --libexecdir=/lib \
--with-ksource=/usr/src/linux --includedir=/usr/include/

 # Preložíme a nainštalujeme
make && make install

- **Kompilácia jadra (Kernel 2.6.31)**

 # Skopírovanie konfigurácie aktuálneho kernelu do adresára s novým jadrom
cp /boot/config-`uname -ra` /usr/src/linux/2.6.31/.config

 # Samotné skompilovanie nového jadra
make && make modules && make modules_install && make install

11.2 Otestovanie funkčnosti podpory layer7

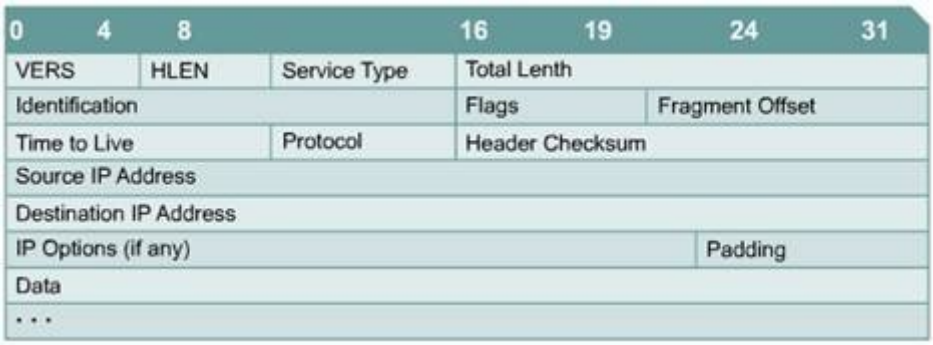
iptables -m layer7 –h
iptables -m ipp2p -h

Aby som otestoval, či firewall naozaj blokuje P2P prevádzku, dal som sťahovať film. Výsledok bol úspešný, P2P prevádzka bola buď úplne zastavená, alebo v horšom prípade sa sťahovanie začalo, ale rýchlosťou okolo 2-10 kB/s čo je zanedbateľné.

11.3 Ako pracuje rozoznávanie protokolu firewallom

Všetky IP pakety pozostávajú z dát z horných vrstiev plus IP hlavička.

IP hlavička pozostáva:



Obrázok 8. – Anatómia IP paketu

Informácie z hlavičky poskytujú hornej vrstve protokolu definovanie dát v pakete.

Iptables používa práve časť paketu „protokol“ kde je definovaný typ paketu. Na základe protokol definition skenuje každý paket na prítomnosť nechceného protokolu, ktorý sme definovali s použitím regulárnych výrazov (Regular Expression). Napr. bittrorrent ma túto definíciu:

^(\x13bittorrent protocol|azver\x01$|get /scrape\?info_hash=get /announce\?info_hash=|get /client/bitcomet/|GET /data\?fid=)|d1:ad2:id20:|\x08'7P\)[RP]

RESUMÉ

Informatizácia vo firmách napreduje rýchlym tempom. Informačné technológie čoraz viac prenikajú do našich životov a stávajú sa našou súčasťou. Ak by som bol dlhšie na praxi a mal by som možnosť ďalej pracovať na bezpečnostnej politike siete, implementoval by som ešte technológiu IDS (Intrusion Detection System), ale z časového hľadiska sa mi to bohužial nepodarilo.

Computerisation in companies is progressing rapidly. Information technologies increasingly permeate our lives and become our part. If I was to practice more and I should continue to work on network security policy, I would have implemented technology, IDS (Intrusion Detection System), but over time I find it unfortunately failed.

ZÁVER

Na záver môžem skonštatovať, že sa mi podarilo spraviť všetko čo som mal v pláne, dokonca aj niečo naviac. Filtrácia protokolov aj samotná kompilácia fungovala správne a pravidlá sa mi podarilo nastaviť tak, ako som chcel a ako to bolo treba pre účely firmy.

Ďalším cieľom práce bolo pomerne jednoducho zabezpečiť sieť z Internetu a nastaviť pravidlá pre rôzne služby a vzdialený prístup na server. Tvorba bezpečnostných pravidiel je dlhodobá záležitosť a nedá sa nastaviť ihneď, pretože až čas ukáže či bolo súčasné pravidlá dobre nastavené.

Ak by som bol dlhšie na praxi a mal by som možnosť ďalej pracovať na bezpečnostnej politike siete, implementoval by som ešte technológiu IDS (Intrusion Detection System), ale z časového hľadiska sa mi to bohužial nepodarilo.

Zoznam použitej literatúry

[1] Mark. G. Sobbel 2007 - Mistrovství v linuxu příkazový řádek, shell, programování
[2] Dostupné na internete http://www.iana.org/assignments/icmp-parameters
[3] Dostupné na internete http://www.cyberciti.biz/
[4] Dostupné na internete http://www.slacksite.com
[5] Dostupné na internete http://l7-filter.sourceforge.net/protocols
[6] Dostupné na internete http://linax.wordpress.com/2009/09/16/slackware-13-kernel-2-6-31-iptables-1-4-5-l7-filter-tarpit-ipp2p
[7] Dostupné na internete http://www.utoronto.ca/web/HTMLdocs/Book/Book-3ed/appb/mimetype.html
[8] Bob Toxen 2003 - Bezpečnost v Linuxu
[9] man iptables
[10] man ssh

12 Prílohy

12.1 Príloha A - Praktický príklad bash skriptu

Tento skript som navrhol tak, aby išlo interaktívne pridávať a mazať IP a MAC adresy na serveri, zobrazovať zotriedený zoznam týchto pravidiel.

```
#!/bin/bash
#
# Tento skript pridáva a zakazuje do firewallu IPTABLES pravidlo pre povolenie IP +  MAC adresy pri forward
# reťazi.
# V kóde je ochrana pred zapísaním zlej IP alebo MAC adresy.
# ***********************************************
# Autor skriptu: Marián Jamrich
# Meno skriptu: allow_deny_access_in_firewall.sh
# ***********************************************
# Skript funguje v spolupráci so skriptom firewall_iptables.sh.

# Premenné
#---------------------------------------
subor=/etc/iptables/ip_mac_adresy.txt
temp="/etc/tmp"
LAN=eth1
INTERNET=eth0

kontrola_prava=`whoami`
if [ $kontrola_prava = root ]; then

clear;
echo
echo '  FIREWALL IPTABLES'
echo
echo
echo "  Zvoľ jednu z možností:"
echo
echo '  [1] povoliť prístup cez firewall (FORWARD CHAIN)'
echo '  [2] odobrať prístup cez firewall (FORWARD CHAIN)'
echo '  [3] zobraziť zotriedený zoznam'
echo
echo -n "Možnosť: "
read moznost

case $moznost in

   1) while true; do

if [ ! -f $subor ]; then
   echo
      echo
      echo '                    UPOZORNENIE !!!'
      echo
      echo '    ================================='
      echo '    | POZOR, súbor s IP a MAC adresami neexistuje !!! |'
      echo '    ================================='
      echo
      echo -n "Prajete si ho vytvoriť ? [a/n]: "
      read odpoved
        case $odpoved in
          ano|a|ANO)
          touch $subor

              if [ $? -ne 0 ]; then
                    echo
                    echo "Chyba: súbor nebol úspešne vytvorený! :("
                    echo
              else
```

```
                echo "Súbor bol úspešne vytvorený. :)"
                echo
                sleep 2
            fi
                ;;
        nie|n|NIE) echo "Bez vytvoreného súbora nemôžete pokračovať! Ukončujem skript..."
                echo
                exit 1
                ;;
        *)
                echo; echo "Musíte vybrať jednu z možností! Ukončujem skript..."; echo
                exit
        esac
fi
clear;
echo ""
echo -n "Zadaj novú MAC adresu [00:11:09:d0:e5:f8]: "
read MAC
echo -n "Zadaj novú IP adresu [10.2.33.44]: "
read IP

   if [[ $MAC == "" ]]; then
      echo; echo "Chyba! Nemôžete zadať prázdnu hodnotu ! Skript sa skončil..."; echo
      exit 2
   elif [[ $IP == "" ]]; then
      echo; echo "Chyba! Nemôžete zadať prázdnu hodnotu ! Skript sa skončil..."; echo
      exit 3
   else
      kontrola_IP=`echo $IP | awk -F"." '$1 <=255 && $2 <= 255 && $3 <= 255 && $4 <= 255'`
      kontrola_MAC=`echo $MAC | sed -n "/^\([0-9A-Z][0-9A-Z]:\)\{5\}[0-9A-Z][0-9A-Z]$/p"`

      if [ -n "$kontrola_IP" ] && [ -n "$kontrola_MAC" ]; then
         echo "$kontrola_IP   $kontrola_MAC" >> $subor
      else
         echo; echo " ** Zlý formát IP alebo MAC adresy! (Neprešiel filtrom)**"; echo
         exit
      fi

      IP=`cat $subor | tail -1 | awk {'print $1'}`
      MAC=`cat $subor | tail -1 | awk {'print $2'}`
 iptables -A FORWARD -i $LAN -o $INTERNET -s $IP -m mac --mac-source $MAC -j ACCEPT &> /dev/null
         if [ $? -gt 0 ]; then echo; echo "** Chyba pridávania pravidla do FW! (iptables chyba) **";echo;
            sleep 2;
            exit;
         fi
fi

if [ $? -ne 0 ]; then
   echo '   ================================================='
   echo '   | Chyba! Nová IP a MAC adresa NEBOLI úspešne povolené vo firewalle !!! |'
   echo '   ================================================='

else
   echo '         ========================================='
   echo '         | Nová IP a MAC adresa bola úspešne povolená vo firewalle !!! |'
   echo '         ========================================='
   echo
   pocet=`cat $subor | wc -l`
   echo "Celkový počet: $pocet IP a MAC adries."
   echo
   echo -n "Chcete zapísať ďalšie IP a MAC adresy ? [enter]: "
      read odpoved

      if [[ $odpoved == "" ]]; then
      :
      else
            echo
            echo "Skript sa úspešne skončil..."
            echo
            exit 0
      fi
fi
   done
   ;;
```

```
   2 )
   while true ;  do

   if [ ! -f $subor ]; then
         echo '                  UPOZORNENIE !!!'
         echo '
         echo '   ================================='
         echo '   | POZOR súbor s IP a MAC adresami neexistuje !!! |'
         echo '   ================================='
      exit
   else

      touch $temp
      echo
      echo
      echo -n "Zadaj MAC adresu, ktorú chceš vymazať [00:11:09:d0:e5:f8]: "
      read MAC
      echo -n "Zadaj IP adresu, ktorú chceš vymazať [10.2.33.44]: "
      read IP

if [[ $MAC == "" ]]; then echo "Nezadali ste žiadnu MAC adresu na vymazanie."; exit 4 ;else :;fi

if [[ $IP == "" ]]; then echo "Nezadali si žiadnu IP adresu na vymazanie. Koniec..."; exit 5; else; :; fi

      kontrola_IP=`echo $IP | awk -F. '$1 <=255 && $2 <= 255 && $3 <= 255 && $4 <= 255'`
      kontrola_MAC=`echo $MAC | sed -n "/^\([0-9A-Z][0-9A-Z]:\)\{5\}[0-9A-Z][0-9A-Z]$/p"`

      if [ -n "$kontrola_IP" ] && [ -n "$kontrola_MAC" ]; then
iptables --delete FORWARD -i $LAN -o $INTERNET -s $IP -m mac --mac-source  $MAC -j ACCEPT &> /dev/null
       else
            echo; echo "Máš chybu vo formáte IP alebo MAC adresy."; echo
      fi
      if [ $? = 0 ]; then echo;else echo "Chyba, požadované pravidlo sa vo firewalle nenašlo!"; echo;fi
            cat $subor | grep -iv $IP > $temp
            cat $temp > $subor
            rm -f $temp
            echo
            echo "Požadovanú IP a MAC adresu som vymazal!"
            pocet=`cat $subor | wc -l`
            echo
            echo "Celkový počet: $pocet IP a MAC adries."
       fi
   done
   ;;
   3 )
            echo; cat $subor | sort
            echo '_____________________________'; echo
            pocet=`cat $subor | wc -l`
            echo "Celkový počet záznamov: $pocet"; echo
   ;;

   * )
            echo; echo 'Nerozumiem Vašej voľbe!'; echo
            echo 'Možné odpovede sú: "1", "2" alebo "3".'; echo
            exit 6
   ;;
   esac

else
            echo; echo ' Musíte mať práva root, aby ste mohli spustiť tento skript !!!'
            echo " Ukončujem skript..."; echo; exit 7
fi
exit 0
```

12.2 Príloha B - Implementácia samotného skriptu pre firewall

Samotné pravidlá firewallu som nastavoval podľa požiadaviek. Definoval som povolené porty z/do firewallu (INPUT a OUTPUT reťaz) a v reťazi FORWARD pri routovaní. Celej vnútornej sieti som povolil tieto porty: 80(http), 443(https), 25(smtp), 110(pop). Ďalej som zapol lo-

govanie pre SSH a VNC spojenia, pokiaľ boli z inej IP adresy ako bola tá moja + vypol som porty (okrem portu 25 pre poštu) od 17:00 do 8:00 rána, pretože neboli potrebné.

```
#!/bin/bash
#
#
#=================#
#  Definícia premenných
#=================#
INTERNET="eth0"
LAN="eth1"
LAN_IP=`ifconfig $LAN | grep "inet addr" | awk {' print $2'} | cut -c6-`
WAN_IP=`ifconfig $INTERNET | grep "inet addr" | awk {' print $2'} | cut -c6-`
LAN_NET="192.168.200.0/24"
IPT="/usr/sbin/sbin/iptables"
IPT6="/sbin/ip6tables"
SSH_LOG="/var/log/firewall.log"
BLACKLIST="/var/log/ssh_blacklist.txt"
ADMIN_EMAIL="jamrich.majo@gmail.com"

# DEFINOVANIE PORTOV:
ssh="99"
vnc_ssh="5901"
vnc_ssh2="6000"
vnc="5900"

#==============#
#  Admin IP a MAC
#==============#
ADMIN_IP_LOCAL="10.2.33.22"
ADMIN_MAC="00:0b:cd:30:56:1f"
admin_ip_public="188.112.72.212"

#==============#
#  TCP a UDP porty
#==============#
povolene_porty="25,53,67,80,110,443,3128,5190,5191"
povolene_porty_FW="53,80,99,443"

#==============#
#   Definícia funkcií
#==============#
function flush () {
                echo -e "\nMažem všetky pravidlá a nastavujem politiku na ACCEPT..."
                $IPT -X
                $IPT -F -t nat
                $IPT -t nat -F PREROUTING
                $IPT -t nat -F POSTROUTING
                $IPT -t nat -F OUTPUT
                $IPT -t nat -X
                $IPT -F INPUT
                $IPT -F FORWARD
                $IPT -F OUTPUT
                $IPT -P INPUT ACCEPT
                $IPT -P OUTPUT ACCEPT
                $IPT -P FORWARD ACCEPT
                 echo '[OK]'
}

function stop () {
                echo -e "\nMažem všetky pravidlá. Politiku nemením!"
                $IPT -F
                $IPT -F -t nat
                $IPT -t nat -F PREROUTING
                $IPT -t nat -F POSTROUTING
                $IPT -F INPUT
                $IPT -F FORWARD
                 $IPT -F OUTPUT
                 echo '[OK]'
}

function restart () {
```

```
        stop
        sleep 1
        start
}

function overenie_chyby () {
        if [ $? != 0 ];then
           echo; echo; echo "    [  ERROR  ]"
        else
           echo "    [  OK  ]"; echo
        fi
}

#=#=#=#=#=#=#=#=#=#=#=#=#= LAYER 7 SUPPORT #=#=#=#=#=#=#=#=#=#=#=#=#=#=
function layer7 () {
        start
        $IPT -I FORWARD -m layer7 --l7proto bittorrent -j DROP
}
#=#= #=#= #=#= #=#= #=#= #=#= #=#= #=#= #=#= #=#= #=#= #=#= #=#= #=#= #=#=

function start () {
#===============#
#  Zoznami IP adries
#===============#
FILE=/etc/firewall/ip_mac_adresy.txt
POVOLENE_IP=`cat /etc/firewall/povolene_ip.txt`
POVOLENE_MAC=`cat /etc/firewall/povolene_mac.txt`
BLOKOVANE_IP=`cat /var/log/ssh_blacklist.txt`

# Vytvorenie súbora
/bin/mkdir /etc/firewall &> /dev/null

if [ ! -e $FILE ]; then exit 2; fi
if [ -z $POVOLENE_IP ]; then exit 3; fi
if [ -z $POVOLENE_MAC ]; then exit 4; fi

clear;
echo -e "Štartujem firewall skript...\n"
#============================#
#  Nastavenie politiky na DROP pre IPv4
#============================#
echo "[+] nastavujem politiku pre IPv4 na DROP..."
$IPT -P INPUT DROP
$IPT -P FORWARD DROP
$IPT -P OUTPUT DROP
overenie_chyby

#=================#
#  Premazanie pravidiel
#=================#
echo "[+] mažem staré pravidlá..."
$IPT -X
$IPT -F -t nat
$IPT -t nat -F PREROUTING
$IPT -t nat -F POSTROUTING
$IPT -t nat -F OUTPUT
$IPT -t nat -X
$IPT -F INPUT
$IPT -F FORWARD
$IPT -F OUTPUT
overenie_chyby

#============================#
#  Nastavenie politiky na DROP pre IPv6
#============================#
echo "[+] nastavujem politiku pre IPv6 na DROP..."
$IPT6 -P INPUT DROP
$IPT6 -P FORWARD DROP
$IPT6 -P OUTPUT DROP
overenie_chyby

#=========================#
#  Zavedenie modulov do jadra + NAT
#=========================#
echo "[+] zavádzam moduly do jadra a povoľujem NAT..."
```

```
# Load iptables NAT module when required
/sbin/modprobe iptable_nat
# Modul potrebný pre aktívne FTP spojenie pri NAT
/sbin/modprobe ip_nat_ftp
# Touto možnosťou zapneme ochranu voči SYN flood- pri tomto útoku by mal byť server chránený natoľko, že bude stíhať obsluhovať legitímne požiadavky
echo 1 > /proc/sys/net/ipv4/tcp_syncookies
# Zapnutie ignorovania pingu na broadcast
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
# Povolenie forwardu priamo v jadre kernelu
echo 1 > /proc/sys/net/ipv4/ip_forward
# Zapnutie IP spoofing ochrany
for i in /proc/sys/net/ipv4/conf/*/rp_filter; do echo 1 > $i; done
# Ochrana proti paketom s príznakom SYN (flood útoky)
echo 1 > /proc/sys/net/ipv4/tcp_syncookies
# Ignorovať ICMP echo požiadavky
echo 0 > /proc/sys/net/ipv4/icmp_echo_ignore_all
# Logovať packety z nemožných adries
for i in /proc/sys/net/ipv4/conf/*/log_martians; do echo 1 > $i; done
# Nelogovať neplatné odpovede z broadcastu.
echo 1 > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses
# Neakceptovať alebo neposielať ICMP presmerovania.
for i in /proc/sys/net/ipv4/conf/*/accept_redirects; do echo 0 > $i; done
for i in /proc/sys/net/ipv4/conf/*/send_redirects; do echo 0 > $i; done
# Don't accept source routed packets.
for i in /proc/sys/net/ipv4/conf/*/accept_source_route; do echo 0 > $i; done
# Vypnutie proxy_arp.
for i in /proc/sys/net/ipv4/conf/*/proxy_arp; do echo 0 > $i; done

# Enable secure redirects, i.e. only accept ICMP redirects for gateways
# Helps against MITM attacks.
for i in /proc/sys/net/ipv4/conf/*/secure_redirects; do echo 1 > $i; done

# Vypnutie bootp_relay
for i in /proc/sys/net/ipv4/conf/*/bootp_relay; do echo 0 > $i; done
overenie_chyby

#===============#
#    FORWARD reťaz
#===============#
echo "[+] nastavujem FORWARD reťaz a zapínam NAT..."

# Zahodi neplatne (invalid) forward NAT spojenia
$IPT -A FORWARD -m state --state INVALID -j DROP
$IPT -A FORWARD -i $LAN -o $INTERNET -m state --state NEW -j ACCEPT

# Všetky požiadavky z vnútornej siete, ktoré nevyhovujú povoleným portom TCP/UDP sa zamietnu!
$IPT -A FORWARD -i $LAN -o $INTERNET -m multiport -p tcp --dports $povolene_porty -j LOG --log-level 7 --log-prefix " BLOKOVANE PORTY: "
$IPT -A FORWARD -i $LAN -o $INTERNET -m multiport -p udp --dports $povolene_porty -j LOG --log-level 7 --log-prefix " BLOKOVANE PORTY: "
$IPT -A FORWARD -i $LAN -o $INTERNET -m multiport -p tcp ! --dports $povolene_porty -j DROP
$IPT -A FORWARD -i $LAN -o $INTERNET -m multiport -p udp ! --dports $povolene_porty -j DROP

# Načíta v cykle WHILE všetky zviazane IP a MAC adresy, ktoré majú povolený prístup do Internetu
while read line; do
IP=`echo $line | awk {'print $1'}`
MAC=`echo $line | awk {'print $2'}`
   $IPT -A FORWARD -i $LAN -o $INTERNET -s $IP -m mac --mac-source $MAC -j ACCEPT
done < $FILE

# Zabalí všetky pakety z vnútornej siete do verejnej IP
$IPT -t nat -A POSTROUTING -s $LAN_NET -j SNAT --to $WAN_IP

# Všetky TCP pakety smerované cez tento server s príznakom SYN zakážeme
$IPT -A FORWARD -p tcp --tcp-flags ALL SYN -j DROP

# Prepustí iba už nadviazané spojenia smerom do vnútornej siete
$IPT -A FORWARD -i $INTERNET -o $LAN -m state --state ESTABLISHED,RELATED -j ACCEPT
#overenie_chyby

#==================#
#    INPUT, OUTPUT reťaz
#==================#
```

```
echo "[+] nastavujem INPUT a OUTPUT reťaz..."

# localhost
$IPT -A INPUT -i lo -j ACCEPT
$IPT -A OUTPUT -o lo -j ACCEPT
$IPT -P INPUT  DROP
$IPT -P OUTPUT DROP

# Zahodí neplatne (invalid) prichádzajúce spojenia
$IPT -A INPUT -m state --state INVALID -j DROP

# Zahodí neplatné (invalid) odchádzajúce spojenia
$IPT -A OUTPUT -m state --state INVALID -j DROP

# port-scanner-limitation
$IPT -A INPUT -p tcp -i $INTERNET --tcp-flags SYN,ACK,FIN,RST RST -m limit --limit 5/s -j ACCEPT
$IPT -A INPUT -p tcp -i $INTERNET --tcp-flags SYN,ACK,FIN,RST RST -j DROP
$IPT -A INPUT -m state --state NEW -p tcp --tcp-flags ALL ALL -j DROP
$IPT -A INPUT -m state --state NEW -p tcp --tcp-flags ALL NONE -j DROP

# Antispoofing pravidla
$IPT -A INPUT -p tcp --tcp-flags ALL FIN,URG,PSH -j DROP
$IPT -A INPUT -p tcp --tcp-flags SYN,RST SYN,RST -j DROP
$IPT -A INPUT -p tcp --tcp-flags SYN,FIN SYN,FIN -j DROP
# Time-based rules (od polnoci do 8 rána sa uzavrú všetky porty. Výnimkou je iba IP adresa admina)
$IPT -A INPUT -s $admin_ip_public -p tcp -m time --timestart 00:00 --timestop 08:00 --weekdays Mon,Tue,Wed,Thu,Fri,Sat,Sun -j ACCEPT
$IPT -A INPUT -s $admin_ip_public -p udp -m time --timestart 00:00 --timestop 08:00 --weekdays Mon,Tue,Wed,Thu,Fri,Sat,Sun -j ACCEPT
$IPT -A INPUT -p tcp -m time --timestart 00:00 --timestop 08:00 --weekdays Mon,Tue,Wed,Thu,Fri,Sat,Sun -j DROP
$IPT -A INPUT -p udp -m time --timestart 00:00 --timestop 08:00 --weekdays Mon,Tue,Wed,Thu,Fri,Sat,Sun -j DROP

# VNC SERVER
$IPT -A INPUT -p tcp --dport $vnc -i $INTERNET -m state --state NEW -m recent --set
$IPT -A INPUT -p tcp --dport $vnc -i $INTERNET -m state --state NEW -m recent --update --seconds 20 --hitcount 2 -j DROP
$IPT -A INPUT -i $INTERNET ! -s $admin_ip_public -m multiport -p tcp --dport $vnc,$vnc_ssh -j LOG --log-prefix "** VNC pokus **" --log-level 4
$IPT -A INPUT -i $INTERNET -s $admin_ip_public -m multiport -p tcp --dport $vnc,$vnc_ssh -j ACCEPT
$IPT -A INPUT -i $INTERNET -m multiport -p tcp --dport $vnc -j DROP

# SSH s ochranou pred brute force útokom. Po neúspešnom zadaní hesla som musel čakať 5 minút ☺
$IPT -A INPUT -p tcp --dport $ssh -i $INTERNET -m state --state NEW -m recent --set
$IPT -A INPUT -p tcp --dport $ssh -i $INTERNET -m state --state NEW -m recent --update --seconds 300 --hitcount 2 -j DROP
$IPT -A INPUT -i $INTERNET ! -s $admin_ip_public -p tcp --dport $ssh -j LOG --log-prefix "** SSH pripojenie **: " --log-level 7

# Zistí aké IP adresy sú v blackliste pre SSH spojenie a zapíše do súbora pre neskoršie použitie
grep -ir "SSH pripojenie" $SSH_LOG | awk -F"SRC=" '{ print $2 }' | awk -F"DST=" '{ print $1 }' | sed 's/ //g' \ | sort | uniq | grep -v $admin_ip_public | grep -v $peta_ip | grep -v $sanko_ip  > $BLACKLIST
echo "Správa s SSH servera jmfvos.sorea.sk" | /bin/mail -s "Na SSH serveri boli zamietnuté ip adresy, ktoré sú v prílohe." -a  $BLACKLIST -r server@jmfvos.sorea.sk $ADMIN_EMAIL

# V cykle for sa načítajú všetky IP adresu v logu ssh, ktoré sú vyhodnotené v blackliste
for IP_BLOCK in `cat $BLACKLIST`; do
$IPT -I INPUT -i $INTERNET -p tcp -s $IP_BLOCK --dport $ssh -j DROP
done

# V poslednom kroku sa zamietne všetká komunikácia pre SSH
$IPT -A INPUT -i $INTERNET -p tcp --dport $ssh -j DROP
$IPT -A INPUT -p tcp -i $INTERNET --tcp-flags SYN,FIN SYN,FIN -j LOG -m limit --limit 10/m --log-prefix="** bogus packet **: " --log-level 7
$IPT -A INPUT -p tcp -i $INTERNET --tcp-flags SYN,FIN SYN,FIN -j DROP

# Odfiltruje pokusy o zahltenie ICMP, prepustí iba PING
$IPT -A INPUT -p icmp --icmp-type echo-request -m limit --limit 1/s --limit-burst 5 -j ACCEPT

# Zahodí všetky pakety, ktoré nepatria nasej vonkajšej sieťovke
$IPT -A INPUT -i $INTERNET ! -s $WAN_IP -j LOG --log-prefix "** DROP non WAN IP paket **: " --log-level 4
$IPT -A INPUT -i $INTERNET ! -s $WAN_IP -j DROP

# ESTABLISHED,RELATED pre reťaz OUTPUT, spôsobí, že prepustí už nadviazané spojenia
```

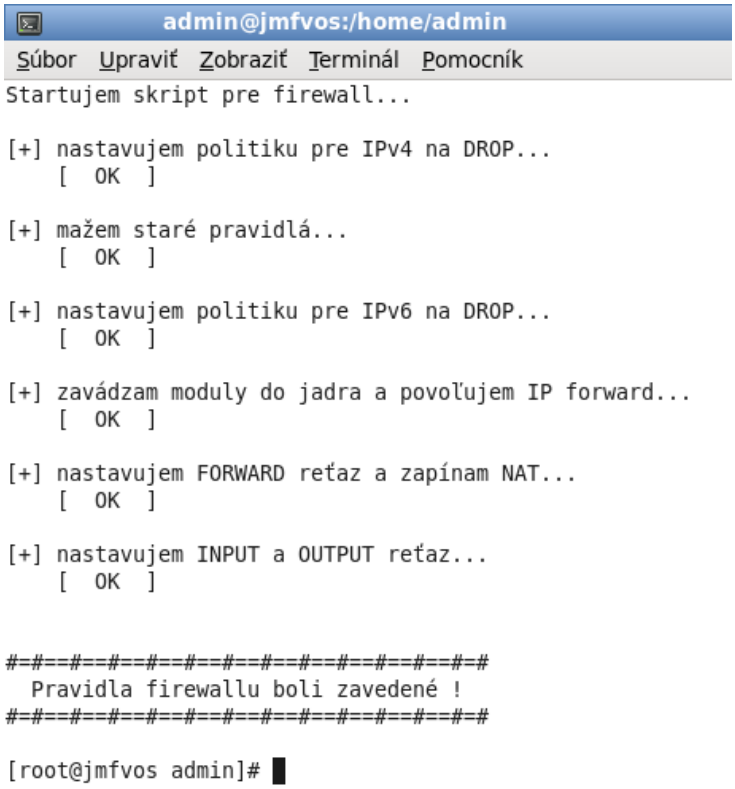
```
$IPT -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# Definícia odchádzajúcich portov
$IPT -A OUTPUT -o $INTERNET -m multiport -p tcp --dports $povolene_porty_FW -j ACCEPT
$IPT -A OUTPUT -o $INTERNET -m multiport -p udp --dports $povolene_porty_FW -j ACCEPT
$IPT -A INPUT  -i $INTERNET -m state --state ESTABLISHED,RELATED -j ACCEPT

# Povolí ping smerom von na vzdialené serveri
$IPT -A OUTPUT -p icmp -o $INTERNET -j ACCEPT

# Spoofed IP adresy
$IPT -A INPUT -i $INTERNET -p tcp -m tcp --tcp-flags FIN,SYN,RST,PSH,ACK,URG NONE -j DROP
$IPT -A INPUT -i $INTERNET -p tcp -m tcp --tcp-flags FIN,SYN FIN,SYN -j DROP
$IPT -A INPUT -i $INTERNET -p tcp -m tcp --tcp-flags SYN,RST SYN,RST -j DROP
$IPT -A INPUT -i $INTERNET -p tcp -m tcp --tcp-flags FIN,RST FIN,RST -j DROP
$IPT -A INPUT -i $INTERNET -p tcp -m tcp --tcp-flags FIN,ACK FIN -j DROP
$IPT -A INPUT -i $INTERNET -p tcp -m tcp --tcp-flags ACK,URG URG -j DROP
# DROP všetko ostatné (čo nie je povolené je zakázané!)
$IPT -A INPUT -j DROP
$IPT -A OUTPUT -j DROP
overenie_chyby
echo -e "\n#=#==#==#==#==#==#==#"
echo "  Pravidla firewallu boli zavedené !"
echo -e "#=#==#==#==#==#==#==#=\n"
}
case "$1" in
        start|-s)
            start
    ;;
    flush|-f)
        flush
    ;;
    help|--help|-h)
        echo -e "\n----------------------------"
        echo -e "Možnosti IPTABLES layer4:\n"
        echo "    --help|help|-h          - zobrazí túto nápovedu"
        echo "    flush|-f                - vymaže všetky predošlé pravidlá v reťazi: INPUT, OUTPUT, FORWARD,
NAT a nastaví politiku na ACCEPT"
        echo "    start|-s                - spustí skript pre všetky reťaze"
        echo '    stop                    - to isté ako "flush", len NENASTAVÍ politiku na ACCEPT'
        echo '    restart|-r              - zastaví ("stop") a následne spustí ("start") skript'
        echo "    status                  - skontroluje či je firewall spustený"
        echo "    layer7|-l7              - zapne iptables layer4 + podporu aj pre iptables layer7"
        echo "    show                    - zobrazí tabulky pre: INPUT, OUTPUT a FORWARD"
        echo -e "   show-nat             -zobrazí tabulky pre: PREROUTING, POSTROUTING a OUTPUT\n"
        echo -e "Exit kódy:\n\n\t0 - OK"
        echo -e "\t1 - Nastala všeobecná chyba"
        echo -e "\t2 - Neexistuje konfiguračný súbor pre IP a MAC"
        echo -e "\t3 - Súbor s povolenými IP adresami je prázdny"
        echo -e "\t4 - Súbor s povolenými MAC adresami je prázdny"
        echo
    ;;
    layer7|-l7)
        layer7
    ;;
    restart|-r)
        restart
    ;;
    stop)
        stop
    ;;
    show)
        clear;
        $IPT -nvL |less
    ;;
    show-nat)
            clear;
            $IPT -t nat -nvL
    ;;
    status)
        rules=`$IPT -nvL | wc -l`
        if [ $rules -gt 15 ]; then
            echo -e "\n[ OK ] Firewall je spustený!"
        else
            echo -e "\n[ STOP ] Firewall je zastavený."
```

```
    fi
  ;;
  *)
    echo -e "\nPouži: $0 {start|stop|flush|restart|layer7|show|show-nat|status|help}"
  ;;
esac
exit 0
```



Obrázok 9. – Obrazovka s výsledkom skriptu firewallu