Vysoké učení technické v Brne

Fakulta informačních technologií

ISA – Sieťové aplikácie a správa sietí

DHCPv6 relay s podporou vloženia MAC adresy

Meno Priezvisko: Michal Zelenák Login: xzelen24

Obsah

1. Úvod do problematiky

Pre pridelenie IPv6 adresy hostovi dynamicky, si musí DHCPv6 klient vymeniť správy s DHCPv6 serverom. Klient lokalizuje DHCPv6 server posielaním požiadavok na lokálnu multicastovu adresu.

Pre priamu komunikáciu medzi DHCPv6 klientom a DHCPv6 serverom. Obe tieto zariadenia sa musia nachádzať v rovnakej sieti.

Problém nastáva ak v lokálnej sieti nie je prítomný DHCPv6 server. Tento problém sa dá riešiť pridaním DHCPv6 Relay agenta do lokálnej siete, ktorý bude preposielať správy medzi klientom a serverom.

2. Návrh aplikácie

Aplikácia obsahuje tieto časti:

- časť, ktorá spracováva argumenty
- časť, ktorá zisťuje dostupné rozhrania pre počúvanie
- časť, ktorá vytvára proces pre každé rozhranie, ďalšie body patria týmto procesom
- časť, ktorá sa prihlasuje do multicastovej skupiny(každé rozhranie samostatne)
- časť pre odchytávanie a spracovanie paketov od klienta
- časť pre odoslanie týchto nových správ serveru
- časť pre prijatie a spracovanie odpovede od serveru
- časť pre posielanie odpovede od serveru ku klientovi

Jednotlivé časti bližšie budú popísané v popise implementácie

3. Popis implementace

3.1. Časť programu ktorá spracováva argumenty

V tejto časti používame funkciu getopts, a v cykle prechádzame jednotlivé výsledky tejto funkcie a podľa toho nastavujeme rôzne príznaky ktoré budú neskôr v aplikácii použité. Jednotlivé príznaky sú sMark,iMark,dMark,lMark a značia prítomnosť daných argumentov.

3.2 Časť pre zisťovanie dostupných rozhraní pre počúvanie

Keďže pracujeme na DHCPv6 relay agentovi, potrebujeme zistiť ktoré rozhrania majú pridelenú globálnu adresu. Prechádzame zoznam vrátený funkciou pcap_findalldevs a postupne tento zoznam modifikujeme tak aby v ňom ostali len zariadenia, ktoré nie sú loopback rozhraniami a zároveň majú globálnu IPV6 adresu.

3.3. Časť pre vytváranie procesov pre každé rozhranie

Prechádzame už upravený zoznam zariadení a pre každé vytvárame nový proces. Bolo potrebné vyriešiť problém, kedy funkcia fork síce vytvorila nový proces, no proces pokračoval v cykle vytvárania procesov, takže sa vytvorilo oveľa viac procesov ako bolo nutné, a potrebné.

3.4.Časť pre prihlasovanie do multicastovej skupiny

Ak chce relay príjmať správy od klienta, musí byť prihlásený v určitej multicastovej skupine. Táto skupina je určená napr. pre DHCPv6 relay agentov a servery. Toto je možné docieliť pomocou vytvorenia soketu s multicastovou adresou. Toto sa stane v každom procese vytvorenom pomocu funkcie fork(časť vyššie).

3.5. Časť pre odchytávanie a spracovanie paketov od klienta

V tejto časti pomocou knižnice libpcap odchytávame jednotlivé solicit alebo request pakety, zabalíme ich do relay-forward správy spolu s pridanými údajmi. Pomocou práce s ukazateľmi sa pohybujeme po prijatých dátach, z ip a ethernetových hlavičiek vyberáme adresy.

3.6. Časť pre odosielanie správ k serveru

Relay-forward správy vytvorené v predchádzajúcej časti je teraz potrebné odoslať smerom ku serveru. Tento pošleme pomocu knižnice sys/socket.h. Využijeme na to funkciu send(), ktorej predáme argumentom adresu serveru zadanú pomocou argumentu.

3.7. Časť pre príjmanie a spracovanie odpovedí od serveru

Po odoslaní správy k serveru v predošlej časti, ihneď čakáme na odpoveď, ktorú spracujeme a rozbalíme z relay-reply správy. Analyzujeme tento relay paket pomocou práce s ukazatelmi a pohybom po pakete, a ak bol zadaný argument “-d” alebo “-l “, vypíšeme alebo pošleme syslog správy ohľadom pridelenej adresy.

3.8. Časť pre odoslanie prijatých správ od serveru ku klientovi

V tejto časti už len pošleme prijatú správu na interface, ktorý je špecifikovaný v jednej z možností v prijatej správe od serveru, no taktiež ju ešte máme uloženú v správe ktorú sme posielali k serveru(táto správa je stále dostupná).

4. Zaujímavé pasáže implementácie

Jeden zo zaujímavých problémov, bolo ukončenie detských procesov spolu s hlavným procesom. Na toto bola použítá knižnica signal.h, konkrétne funkcia signal, ktorá pri signále SIGINT spustí funkciu closeEverything ktorá v každom procese zavolá funkciu kill(). PID procesu je získané funkciou getpid(), a ako signál bol zvolený SIGKILL. Toto zaručí že každý bežiaci proces je ukončený, a teda deti aj rodič je ukončený práve týmto signálom.

5. Potrebné knižnice pre správne fungovanie programu

- stdio.h
- stdlib.h
- unistd.h
- pcap/pcap.h
- string.h
- netinet/in.h
- netinet/ether.h
- net/if.h
- arpa/inet.h
- sys/types.h
- sys/socket.h
- time.h
- math.h
- errno.h
- error.h
- syslog.h
- signal.h

6. Problémy

Počas riešenia projektu sa objavilo veľa problémov, no niektoré ostali kvôli časovej tiesni nevyriešené, alebo si vyžadovali riešenie ktoré bolo nemožné v poskytnutom čase. Jedným z nich je kontrola adresy rozhraní kde počúvame. Užívateľ ktorý spúšťa tento program by sa preto mal v prípade problémov uistiť že je táto adresa správna. Taktiež ostal nevyriešený problém kedy rozhranie nemá žiadnu adresu a relay by mal prideliť do správy meno rozhrania. Bolo by taktiež vhodné pridať semafor počas používania funkcii z knižnice libpcap.

7. Návod na použitie

Pre spustenie relay-agenta treba program d6r.c skompilovať pomocou priloženého makefile súboru, spustením príkazu make. Potom môžme spustiť samotný program d6r. Povinným parametrom je adresa serveru, ktorú treba zadať v tvare "-s adresa". A teda program je spustitelný príkazom ./d6r -s 2001:db8:15a:8c:192:5(vymyslená adresa).

Jedným z ďalších možných parametrov programu je "-d", ktorý zapína vypisovanie pridelených adries k jednotlivým mac adresám a to v tvare "pridelena adresa,Mac adresa", alebo v tvare "pridelena adresa/prefix pridelenej adresy,mac adresa", podľa toho o akú adresu klient požiada.

Rovnaký formát výstupu používa ďalší parameter "-l", ktorý ukladá rovnaké informácie do syslog súboru.

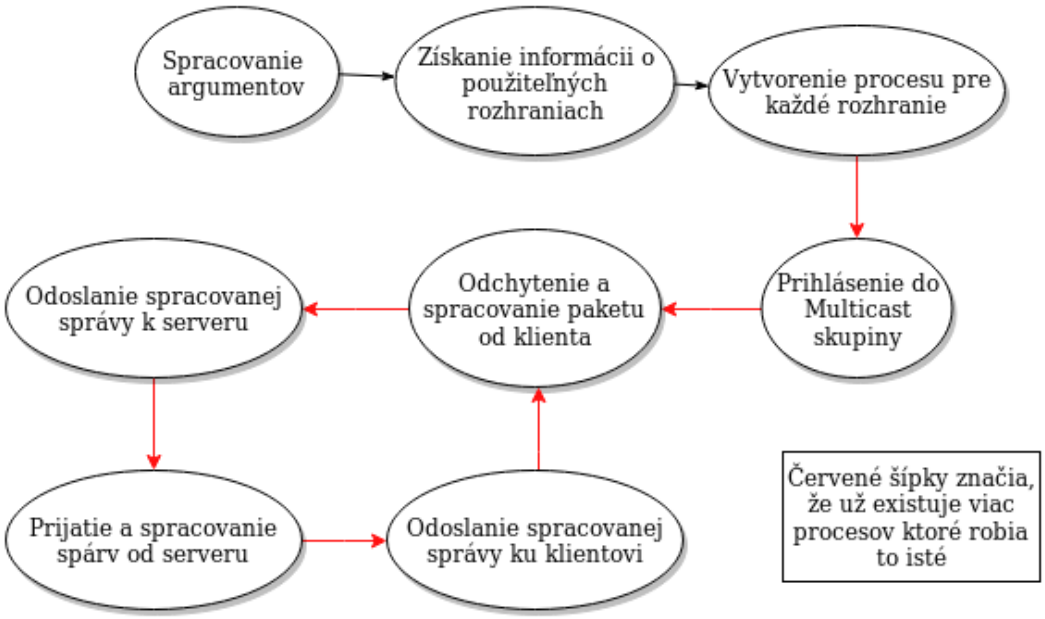
Posledný možný parameter je "-i interface", ktorý spôsobí že program bude počúvať len na rozhraní, ktoré má rovnaký názov ako parameter interface. Čiže za interface treba dosadiť názov rozhrania ak túto možnosť chceme použiť. Počúvanie zariadenia sa netýka komunikácie s DHCPv6 serverom.

8. Súbory

Projekt obsahuje tieto súbory,

- Dokumentácia – manual.pdf
- Manuál(README/ - d6r.1 (spustenie pomocou man -l d6r.1)
- d6r.c (spustenie pomocou prikazu make)
- makefile

9. Prílohy


Spracovanie argumentov
Získanie informácii o použiteľných rozhraniach
Vytvorenie procesu pre každé rozhranie
Prihlásenie do Multicast skupiny
Odchytenie a spracovanie paketu od klienta
Odoslanie spracovanej správy k serveru
Prijatie a spracovanie spárv od serveru
Odoslanie spracovanej správy ku klientovi
Červené šípky značia, že už existuje viac procesov ktoré robia to isté

10. Referenčná literatúra a zdroje

[1]

[RFC8415] T. Mrugalski, M. Siodelski, B. Volz, A. Yourtchenko, ISC, Cisco, M. Richardson, SSW, S. Jiang, Huawei, T. Lemon, Nibbhaya Consulting, T. Winters, UNH-IOL, "*Dynamic Host Configuration Protocol for IPv6 (DHCPv6)",RFC8415.November 2018, https://ietf.org/rfc/rfc8415.txt*

[2]

[RFC6339] Josefsson, S. and L. Hornquist Astrand, "Context Token Encapsulate/Decapsulate and OID Comparison Functions for the Generic Security Service Application Program Interface (GSS-API)", RFC 6339, August 2011. http://www.ietf.org/rfc/rfc6339.txt