

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



Dokumentace k projektu z předmětu ISA

**Nástroje monitorující a generující zprávy jednoduchých
distance-vector protokolů (Dr. Veselý)**

Autor:

Michal Martinů, xmarti78

17. listopad 2018

Obsah

1	Úvod	1
2	Protokol RIP	1
3	Sniffer RIP a RIPng zpráv	1
3.1	Implementace	1
3.2	Datová část RIP a RIPng paketu	2
3.3	Použití programu	2
3.4	Příklad výstupu programu	3
4	Podvrhávač falešných RIPng Response zpráv	4
4.1	Implementace	4
4.2	Použití programu	4
5	Testování	4
5.1	Sniffer	5
5.2	Podvržení routy směrovači	6
6	Závěr	6
7	Obrázky	7
8	Literatura	7

1 Úvod

Zadáním projektu bylo implementovat sniffer RIPv1, RIPv2 a RIPv6 zpráv, dále také podvrhávač falešných RIPv6 Request zpráv. S pomocí těchto nástrojů se měl poté provést úspěšný útok na cílený směrovač a podvrhnout mu falešnou RIPv6 Request zprávu.

2 Protokol RIP

Routing Information Protocol (RIP) je směrovací protokol, který v počítačové síti umožňuje komunikaci mezi směrovači a reakci na změnu v topologii počítačové sítě. Protokol RIP používá počet hopů jako směrovací metriku. Dále má i ochranu proti zacyklení, ta je dosažena určením maximálního počtu hopů. Největší počet hopů, jaký může RIP protokol mít je 15, dále potom 0 je počátek a 16 znamená nekonečno. Nevýhodou je, že počet hopů omezuje i velikost počítačové sítě, proto nelze používat RIP protokol pro větší počítačové sítě.

Tento protokol vychází z Bellman-Fordova algoritmu, který je určen pro hledání nejkratší cesty v ohodnoceném grafu.

Směrovače aktualizují svou směrovací tabulku každých 30 sekund. V protokolu RIPv1 si směrovače posílají směrovací tabulky všesměrově (broadcast). To může zapříčinit zahlcení sítě. Od verze 2 protokolu RIP je problém částečně vyřešen zasíláním směrovacích tabulek skupinově (multicast).

3 Sniffer RIP a RIPv6 zpráv

Sniffer je aplikace monitorující data putující po síti. Odchycená data poté sniffer zpracuje a interpretuje je do formy čitelné pro uživatele.

3.1 Implementace

Programovacím jazykem pro tento projekt byl použit Jazyk C. Jsou použity standardní knihovny.

Implementace používá knihovnu libpcap, která zajišťuje odchytávání paketů v promiskuitním módu síťové karty. Knihovna libpcap neumí zpracovávat přijaté pakety a pouze předává aplikaci zachycené byty dat. Tyto data obsahují i hlavičky nižších modelových síťových vrstev jako je Ethernet, UDP a IP.

Pro monitorování příchozí komunikace je použit filtr na UDP port 520 (RIP) a 521 (RIPv6). Tímto se předchází zpracování nežádoucí komunikace a knihovna nemusí zpracovávat celou komunikaci po síti.

Knihovna libpcap musí být dobře nastavená, aby mohla odchytávat síť. Bylo nutné otevřít síťový port, zkompileovat sniffer a nastavit filtr dat. Hlavní je funkce `pcap_loop()`, která

při každém přijatém paketu volá callback funkci s právě získanými daty a předává je pomocí parametrů.

Po přijetí paketu dojde k jeho postupné analýze. Ten začíná ethernetovou hlavičkou s pevnou délkou. Ta není pro další vyhodnocování důležitá, proto je možné ji přeskočit a posunout se rovnou na IP hlavičku.

U IP hlavičky se jako první krok musí určit, o jaký protokol se jedná. Jestli IPv4 nebo IPv6. Podle verze protokolu je možné určit způsob, jakým se má zpracovat a o kolik se musí posunout v paměti. U IPv4 je velikost hlavičky uvedena za verzí protokolu. V případě IPv6 je délka fixní.

Po posunu v datech paketu na základě vyhodnocené délky přijde na řadu protokol UDP, ze kterého již není problém určit velikost RIP zprávy. Tato informace je nezbytná pro její zpracování.

Potom je nutné rozlišit verzi protokolu. Mezi verzí RIPv1 nebo RIPv2 není velký rozdíl. V případě verze RIPv2 je potřeba k vyhodnocení přistupovat lehce odlišně. RIPv2 má navíc oproti ostatním verzím heslo. To může být jednoduché nebo u verze RIPv2 MD5 šifrované. V tomto případě se vypíše pouze jednotlivá data v hexadecimální podobě.

3.2 Datová část RIP a RIPv2 paketu

Pro práci s protokolem RIP a RIPv2 bylo nutné definovat struktury, které budou interpretovat jednotlivé hodnoty v přijatých datech.

Protokol RIP a RIPv2 je rozdělen na hlavičku a záznamy. Záznam bývá v případě RIPv2 autentizační nebo nosný. U protokolu RIPv1 autentizační záznam není. Protokol RIPv2 vedle nosného záznamu může obsahovat navíc také nex-hop. Ten je platný pro všechny následující záznamy.

Pro bližší nahlédnutí jsou struktury paketů definovány v souboru *packet_headers.h*.

3.3 Použití programu

Program se spouští pomocí příkazu `./myriprsniffer -i <rozhraní>`, kde povinný parametr `-i <rozhraní>` udává, na kterém rozhraní bude odchyt paketů prováděn. Také lze místo rozhraní zadat soubor s příponou *.pcap* nebo *.pcapng* pro čtení komunikace ze záznamu v souboru. Je doporučeno program spouštět s uživatelskými právy správce.

3.4 Příklad výstupu programu

```
=====
No. 1 [18:55:53]
Protocol: RIPv2
Recieved Request from [10.0.0.1]
```

IP	Mask	Next hop	Tag	Metric
0.0.0.0	0.0.0.0	0.0.0.0	0	16

```
=====
No. 2 [18:55:53]
Protocol: RIPvng
Recieved Request from [fe80::a00:27ff:fede:1c5d]
```

IP/Prefix-length	Tag	Metric
::/0	0	16

```
=====
No. 3 [18:55:54]
Protocol: RIPv2
Recieved Response from [10.0.0.1]
```

```
Authentication
Authentication type: Simple Password (2)
Password: ISA>29b16c1914d
```

IP	Mask	Next hop	Tag	Metric
10.55.56.0	255.255.255.0	0.0.0.0	0	1
10.97.116.0	255.255.255.0	0.0.0.0	0	1
10.114.223.0	255.255.255.0	0.0.0.0	0	1
10.213.105.0	255.255.255.0	0.0.0.0	0	1

```
=====
No. 4 [18:55:54]
Protocol: RIPvng
Recieved Response from [fe80::a00:27ff:fede:1c5d]
```

IP/Prefix-length	Tag	Metric
fd00::/64	0	1
fd00:db:3138::/64	0	1
fd00:10c:3164::/64	0	1
fd00:4e6:6d::/64	0	1
fd00:c08:1538::/64	0	1

4 Podvrhávač falešných RIPng Response zpráv

Jedná se o útočný program, který napadá směrovače a podvrhává jim falešné RIPng zprávy. Kontrola na straně směrovačů je nízká, a tak směrovač vezme jakoukoliv zprávu, která splňuje formální požadavky.

4.1 Implementace

Program nejdříve vytvoří falešný RIPng paket na základě vložených argumentů. Využívají se zde struktury vytvořené pro sniffer. Po sestavení je RIPng paket odeslán použitím BSD soketů pomocí UDP portu číslo 521.

4.2 Použití programu

Spuštění programu probíhá pomocí příkazu

```
sudo ./myripresponse -i <rozhraní> -r <IPv6>/[16-128] {-n <IPv6>} {-m [0-16]} {-t [0-65535]}
```

- *-i <rozhraní>*: udává, na které rozhraní bude útočný paket odeslán.
- *-r <IPv6>/[16-128]*: je IP adresa podvrhované sítě za lomítkem číselná adresa masky sítě.
- *-m*: následující číslo udává RIP Metriku, tedy počet hopů, implicitně.
- *-n <IPv6>*: za tímto parametrem je adresa next-hopu pro podvrhávanou routu, implicitně ::.
- *-t*: číslo udává hodnotu Router Tagu, implicitně 0.

U této syntaxe vstupních parametrů složené závorky {} znamenají nepovinné argumenty. Při jejich nepoužití je nastavena implicitní hodnota definovaná výše.

Program je nutný spouštět jako správce z důvodu odesílání přes UDP port 521, proto je zde doporučeno použít příkaz *sudo*.

5 Testování

Pro účely testování byla vytvořena testovací síť s jedním virtuálním síťovým adaptérem, na který byl připojen virtuální počítač s Linuxem isa2015 s programem *myripresponse* a fyzický počítač se systémem Mac OS, který používal *myripsniffer* pro odchyt RIP zpráv. Jako softwarový směrovač sloužil upravený virtuální počítač FreeBSD s aktivní konfigurací pro uživatele xmarti78.



Obrázek 1 – Testovací síť

5.1 Sniffer

Výpis obdržených zpráv od směrovače vypadal následovně:

```
=====
No. 7 [12:49:38]
Protocol: RIPv2
Recieved Response from [10.0.0.1]

Authentication
Authentication type: Simple Password (2)
Password: ISA>29b16c1914d

-----
IP                Mask            Next hop          Tag      Metric
-----
10.55.56.0        255.255.255.0    0.0.0.0          0        1
10.97.116.0       255.255.255.0    0.0.0.0          0        1
10.114.223.0      255.255.255.0    0.0.0.0          0        1
10.213.105.0      255.255.255.0    0.0.0.0          0        1
-----

=====
No. 8 [12:49:38]
Protocol: RIPv2
Recieved Response from [10.0.0.1]

Authentication
Authentication type: Simple Password (2)
Password: ISA>29b16c1914d

-----
IP                Mask            Next hop          Tag      Metric
-----
10.55.56.0        255.255.255.0    0.0.0.0          0        1
10.97.116.0       255.255.255.0    0.0.0.0          0        1
10.114.223.0      255.255.255.0    0.0.0.0          0        1
10.213.105.0      255.255.255.0    0.0.0.0          0        1
-----

=====
No. 9 [12:49:38]
Protocol: RIPv2
Recieved Response from [192.168.88.154]

Authentication
Authentication type: Simple Password (2)
Password: ISA>29b16c1914d

-----
IP                Mask            Next hop          Tag      Metric
-----
10.0.0.0          255.255.255.0    0.0.0.0          0        1
10.55.56.0        255.255.255.0    0.0.0.0          0        1
10.97.116.0       255.255.255.0    0.0.0.0          0        1
10.114.223.0      255.255.255.0    0.0.0.0          0        1
10.213.105.0      255.255.255.0    0.0.0.0          0        1
-----

=====
No. 10 [12:49:38]
Protocol: RIPv2
Recieved Response from [192.168.88.154]

Authentication
Authentication type: Simple Password (2)
Password: ISA>29b16c1914d

-----
IP                Mask            Next hop          Tag      Metric
-----
10.0.0.0          255.255.255.0    0.0.0.0          0        1
10.55.56.0        255.255.255.0    0.0.0.0          0        1
10.97.116.0       255.255.255.0    0.0.0.0          0        1
10.114.223.0      255.255.255.0    0.0.0.0          0        1
10.213.105.0      255.255.255.0    0.0.0.0          0        1
-----

=====
No. 11 [12:49:38]
Protocol: RIPvng
Recieved Response from [fe80::a00:27ff:fede:1c5d]

-----
IP/Prefix-length          Tag      Metric
-----
fd00::/64                0        1
fd00:db:3138::/64        0        1
fd00:10c:3164::/64       0        1
fd00:4e6:6d::/64         0        1
fd00:c08:1538::/64       0        1
-----

=====
No. 12 [12:49:38]
Protocol: RIPvng
Recieved Response from [fe80::a00:27ff:fede:1c5d]

-----
IP/Prefix-length          Tag      Metric
-----
fd00::/64                0        1
fd00:db:3138::/64        0        1
fd00:10c:3164::/64       0        1
fd00:4e6:6d::/64         0        1
fd00:c08:1538::/64       0        1
-----
```

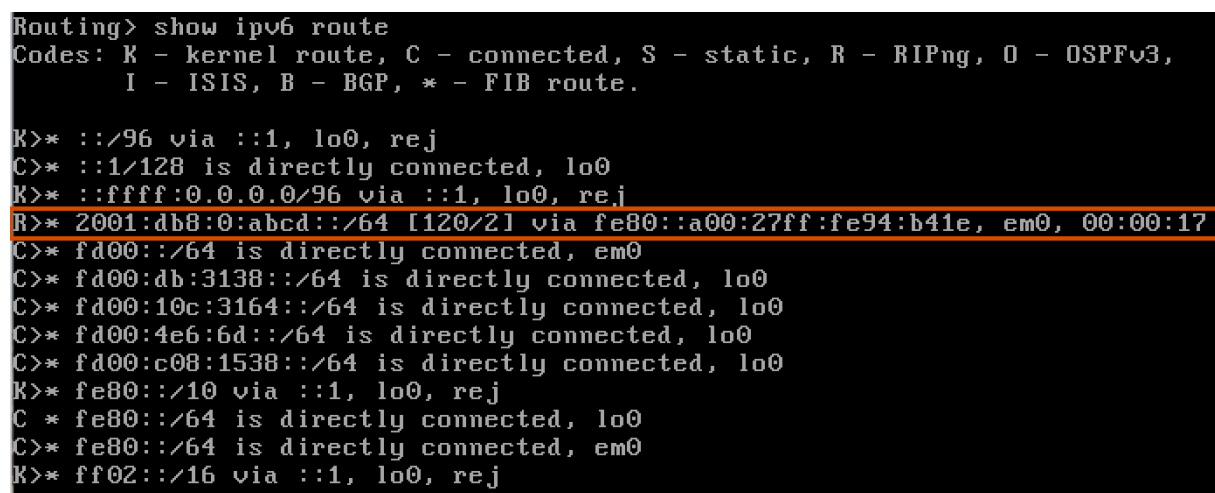
Úkolem bylo tuto komunikaci zpracovat a následně v ní najít heslo. Po inspekci obdržených zpráv se podařilo zachytit heslo, které je ISA>29b16c1914d.

5.2 Podvržení routy směrovači

Jako druhý krok testování bylo nutné na příslušný softwarový směrovač zaútočit a podvrhnout mu falešnou RIPng Response routu s adresou 2001:db8:0:abcd::/64. Bylo tedy nutné ve virtuálním počítači isa2015 spustit program příkazem:

```
sudo ./myripresponse -i eth0 -r 2001:db8:0:abcd::/64.
```

Ověření úspěšnosti útoku se pak již provedlo ověřením routy ve směrovací tabulce na sw směrovači. Útok byl úspěšný a podvržená routa je označena červeným ohrazením v obrázku 2.



```
Routing> show ipv6 route
Codes: K - kernel route, C - connected, S - static, R - RIPng, O - OSPFv3,
       I - ISIS, B - BGP, * - FIB route.

K>* ::/96 via ::1, lo0, rej
C>* ::1/128 is directly connected, lo0
K>* ::ffff:0.0.0.0/96 via ::1, lo0, rej
R>* 2001:db8:0:abcd::/64 [120/2] via fe80::a00:27ff:fe94:b41e, em0, 00:00:17
C>* fd00::/64 is directly connected, em0
C>* fd00:db:3138::/64 is directly connected, lo0
C>* fd00:10c:3164::/64 is directly connected, lo0
C>* fd00:4e6:6d::/64 is directly connected, lo0
C>* fd00:c08:1538::/64 is directly connected, lo0
K>* fe80::/10 via ::1, lo0, rej
C * fe80::/64 is directly connected, lo0
C>* fe80::/64 is directly connected, em0
K>* ff02::/16 via ::1, lo0, rej
```

Obrázek 2 – Záznam směrovací tabulky

6 Závěr

Tento dokument se věnuje teoretické části, které popisuje princip RIP protokolů dále se zaměřuje na popis implementace, fungování RIP-RIPng snifferu a také programu na podvrhování falešných RIPng zpráv.

7 Obrázky

Obrázek 1 – Testovací síť	4
Obrázek 2 – Záznam směrovací tabulky	6

8 Literatura

- [1] Heidrick, C. *RFC1058, Routing Information Protocol*. June 1998, <https://tools.ietf.org/html/rfc1058>.
- [2] Malkin, G. *RFC2453, RIP Version 2*. November 1998, <https://tools.ietf.org/html/rfc2453>.
- [3] Malkin, G., R. Minnear. *RFC2080, RIPng for IPv6*. January 1997, <https://tools.ietf.org/html/rfc2080>.
- [4] Carstens, T. *Programming with pcap*. 2002, <https://www.tcpdump.org/pcap.html>.