VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Dokumentácia k projektu do predmetu IPK

DHCP Starvation útok

9. apríla 2018

Matúš Liščinský (xlisci02)

Obsah

1 Úvod **2**
1.1 DHCP v skratke 2
1.2 DHCP Starvation útok 3
1.3 DHCP snooping 3

2 Zaujiímavé časti implementácie **3**
2.1 Štruktúra DHCP správy 4
2.2 Implementačné detaily 4

3 Demonštrácia činnosti implementovanej aplikácie **5**

1 Úvod

Dokumentácia popisuje implementáciu aplikácie realizujúcej vyčerpanie adresného rozsahu DHCP serveru pomocou DHCP Discover správ tak, aby po jej spustení žiaden nový klient nedostal DHCP pôžičku.

1.1 DHCP v skratke

DHCP *(Dynamic Host Configuration Protocol)* je protokol automatického prideľovania IP adries v sieti, na základe požiadavky. Princíp spočíva v tom, že klienti žiadajú server o IP adresu, ten u každého klienta eviduje vypožičanú IP adresu a čas, do kedy ju klient smie používať. Potom čo vyprší, smie server adresu prideliť iným klientom.

Princíp funkčnosti

Obr. 1: Princíp funkčnosti DHCP protokolu

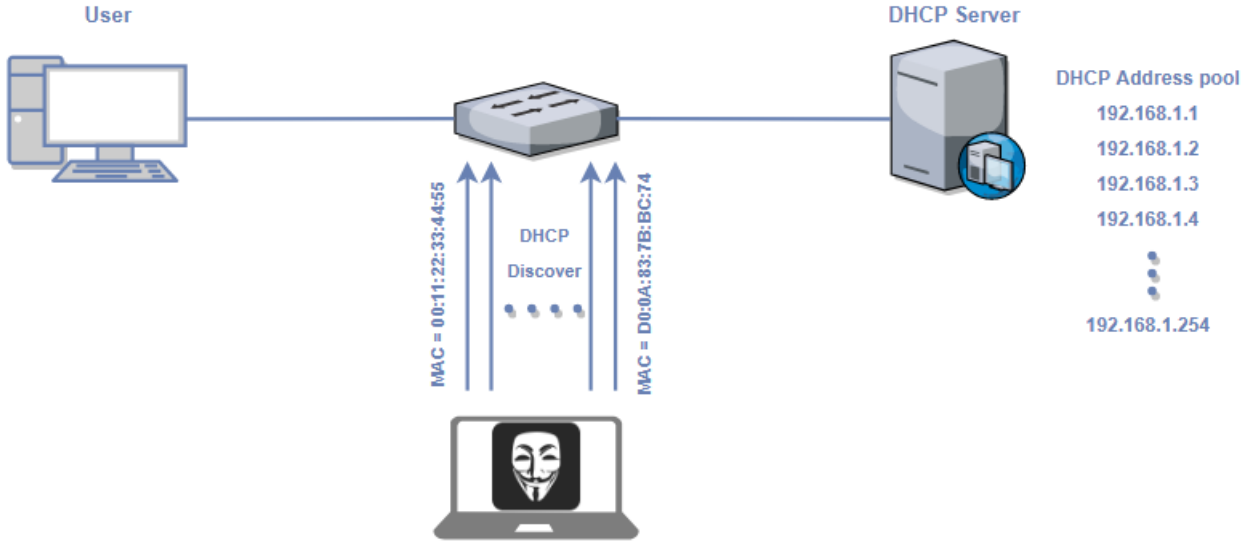
- **DHCP Discover**
  Klient posiela ***DHCP Discover*** správu ako broadcast cez Ethernetovú sieť a hľadá tak dostupný DHCP server v rovnakej podsieti.
- **DHCP Offer**
  DHCP server príjma Discover správu od klienta a posiela ***DHCP Offer*** opäť ako broadcast správu a informuje tým klienta, že je dostupný. ***DHCP Offer*** obsahuje informácie ako IP adresu, masku podsiete, bránu, IP adresy DNS serverov, dobu pôžičky IP adresy, IP adresu DHCP servera atď...
- **DHCP Request**
  Klient môže prijať viacero správ DHCP Offer od viacerých DHCP serverov. Preto po ich prijatí a výbere jedného z nich posiela broadcast ***DHCP Request*** s požiadavkou na sieťové konfiguračné dáta a informáciou ktorý DHCP si vybral. Ostatné DHCP servery, príjmu správu a môžu zmazať uloženú konfiguráciu pre klienta z ich pamäte.
- **DHCP Ack**
  DCHP server, ktorý prijal DHCP Request správu od klienta skontroluje či prijatá IP adresa sedí z uloženou v jeho pamäti. Ak áno, posiela ***DHCP Ack*** správu ako broadcast aby zaručil jej doručenie klientovi. Správa obsahuje všetky konfiguračné dáta a klient tak môže po jej prijatí nakonfigurovať svoje sieťové rozhranie. [3]

1.2 DHCP Starvation útok

Útok zvaný DHCP starvation attack spočíva v tom, že každý DHCP server má presne definovaný a konečný počet IP adries, ktoré vie zariadeniam prideliť. Môže ísť napríklad o rozsah `192.168.1.0/24` , čo predstavuje maximálne 253 použiteľných adries pre klientov.

Keď si však útočník jedným počítačom vypýta všetky voľné IP adresy, ktoré má server k dispozícii, tým, že podvrhne veľké množstvo **falošných MAC adries**, nebude mať server voľné IP adresy pre nové zariadenia.

Po vyradení legitímneho DHCP servera môže jeho úlohy prevziať na seba. Môže tak podvrhnúť falošné konfiguračné dáta a tým sa dostávame už k tzv. *DHCP Spoofing útoku.*



Obr. 2: DHCP Starvation útok

1.3 DHCP snooping

Jedným z obranných mechanizmov proti DHCP Starvation útoku je odpočúvanie protokolu DHCP - **DHCP snooping**. Prepínač v tomto prípade skúma správy protokolu DHCP a tie, ktoré vyhodnotí, ako rizikové blokuje. Pri odpočúvaní je vytváraná *väzobná tabuľka*, ktorá mapuje **IP adresy** a **porty**. Zostavuje ju podľa právoplatných DHCP správ. Prepínač podľa zadefinovania dôveryhodných portov, za ktorými predpokladá existenciu DHCP servera prijíma tomu zodpovedajúce správy. Ak by prišla správa, ktorá má pochádzať iba od DHCP servera z portu označeného ako nedôveryhodný, teda nachádza sa za ním bežná stanica, prepínač packet zablokuje. [4]

2 Zaujímavé časti implementácie

Aplikácia realizuje útok pomocou DHCP Discover správ prostredníctvom tzv. *Raw Ethernet Packetov* posielaných z definovaného sieťového rozhrania. Pri vytváraní packetu využíva už vytvorené štruktúry z dostupných knižníc jazyka C. [1]

Packet je tvorený z týchto častí:

- Ethernet hlavička (`struct ether_header`)
- IP hlavička (`struct iphdr`)
- UDP hlavička (`struct udphdr`)
- Dáta DHCP správy

2.1 Štruktúra DHCP správy

```
#define CHADDR_LEN 16
#define SNAME_LEN 64
#define FILE_LEN 128
#define OPTIONS_LEN 8

typedef struct dhcp_packet_struct{
    uint8_t  op;
    uint8_t  htype;
    uint8_t  hlen;
    uint8_t  hops;
    uint32_t xid;
    uint16_t secs;
    uint16_t flags;
    struct in_addr ciaddr;
    struct in_addr yiaddr;
    struct in_addr siaddr;
    struct in_addr giaddr;
    unsigned char chaddr [CHADDR_LEN];
    char sname [SNAME_LEN];
    char file [FILE_LEN];
    char options[OPTIONS_LEN];
}dhcp_packet;
```

Zdrojový kód aplikácie je v **jazyku C**, rozdelený na:

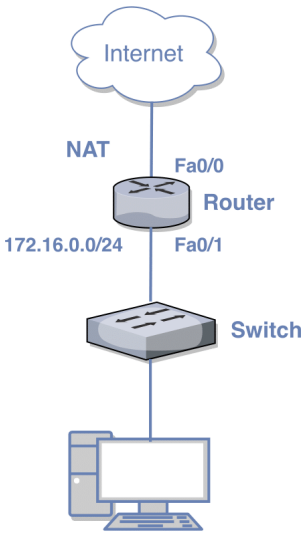
- `ipk-dhcpstarve.c` (obsahujúci implementáciu použitých funkcií a telom hlavnej funkcie main)
- `ipk-dhcpstarve.h` (hlavičkovým súborom s deklaráciami funkcií a definíciami konštánt).

2.2 Implementačné detaily

Jednou zo zaujímavostí implementácie je pole `options`, kde prvé štyri byty tvoria tzv. **magic cookie** a ďalšie byty identifikujú typ správy. [2] Navyše každý packet obsahuje náhodne vygenerovanú MAC adresu uloženú v poli `chaddr`, čo je kľúčom k úspešnému útoku. Ešte pred odoslaním packetu sa spočíta a uloží do IP hlavičky kontrolný súčet (*IPv4 header checksum*).

3 Demonštrácia činnosti implementovanej aplikácie

Demonštrácia činnosti aplikácie prebehla na nasledujúcej topológii, kde Router predstavuje DHCP server:

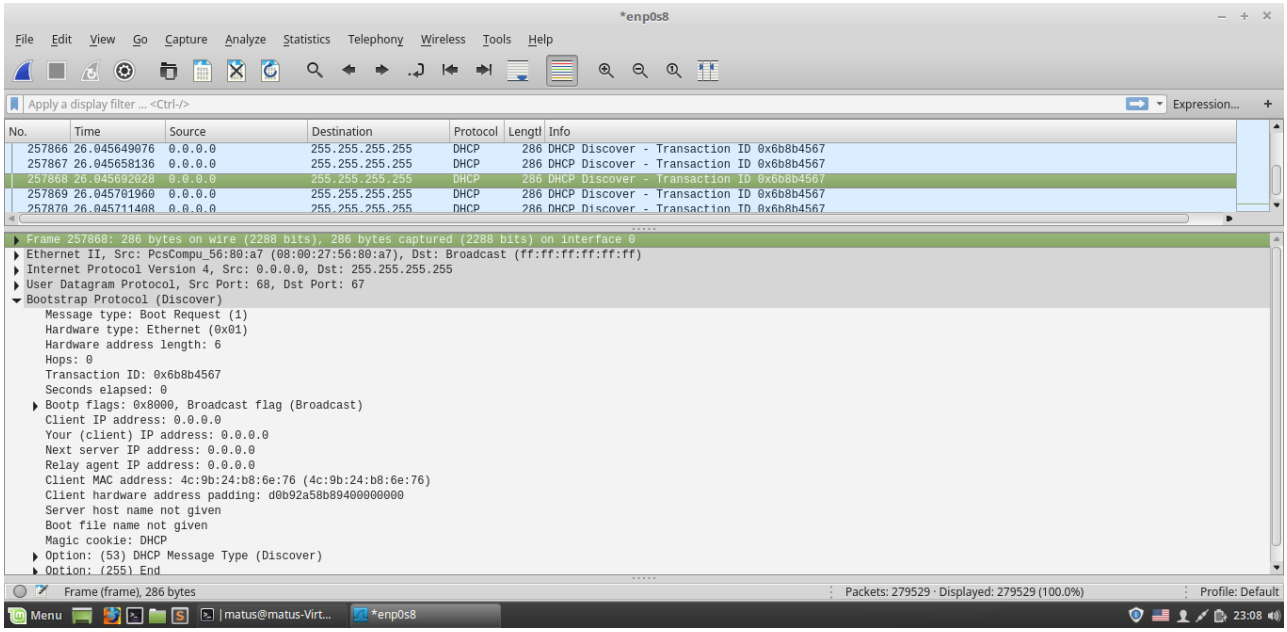


Obr. 3: Testovacia topológia

```
172.16.0.196        2af4.e937.0a0e          Apr 06 2018 12:30 PM    Automatic
172.16.0.197        35a4.58d8.509f          Apr 06 2018 12:30 PM    Automatic
172.16.0.198        b89e.8b22.ef49          Apr 06 2018 12:30 PM    Automatic
172.16.0.199        9af2.5c42.7e48          Apr 06 2018 12:30 PM    Automatic
172.16.0.200        c239.1747.d3cc          Apr 06 2018 12:30 PM    Automatic
172.16.0.201        c6fd.c1b0.35cb          Apr 06 2018 12:30 PM    Automatic
172.16.0.202        be6a.6f17.42bf          Apr 06 2018 12:30 PM    Automatic
172.16.0.203        b6fb.5d41.1d4c          Apr 06 2018 12:30 PM    Automatic
172.16.0.204        8bb7.3ee7.f9bc          Apr 06 2018 12:30 PM    Automatic
172.16.0.205        be3e.4bbe.e109          Apr 06 2018 12:30 PM    Automatic
172.16.0.206        0544.401e.202e          Apr 06 2018 12:30 PM    Automatic
172.16.0.207        f06d.90aa.e75a          Apr 06 2018 12:30 PM    Automatic
172.16.0.208        9392.56bd.374a          Apr 06 2018 12:30 PM    Automatic
172.16.0.209        fc50.30f1.146e          Apr 06 2018 12:30 PM    Automatic
172.16.0.210        a9d2.acf4.918d          Apr 06 2018 12:30 PM    Automatic
172.16.0.211        fe96.d13e.b5f1          Apr 06 2018 12:30 PM    Automatic
172.16.0.212        6da5.5efd.5046          Apr 06 2018 12:30 PM    Automatic
172.16.0.213        57e3.d8ae.a010          Apr 06 2018 12:30 PM    Automatic
172.16.0.214        f89c.6028.8d74          Apr 06 2018 12:30 PM    Automatic
172.16.0.215        9637.4642.2bd7          Apr 06 2018 12:30 PM    Automatic
172.16.0.216        cf29.6ea0.6823          Apr 06 2018 12:30 PM    Automatic
172.16.0.217        92d5.c8f0.d218          Apr 06 2018 12:30 PM    Automatic
172.16.0.218        bff4.bcfc.f3b4          Apr 06 2018 12:30 PM    Automatic
172.16.0.219        35d3.5f87.b5ca          Apr 06 2018 12:30 PM    Automatic
172.16.0.220        82ef.cb73.183c          Apr 06 2018 12:30 PM    Automatic
172.16.0.221        8714.bd41.ef56          Apr 06 2018 12:30 PM    Automatic
172.16.0.222        c028.75a7.69ae          Apr 06 2018 12:30 PM    Automatic
172.16.0.223        2b28.a2e7.2496          Apr 06 2018 12:30 PM    Automatic
172.16.0.224        9b59.69fa.e11e          Apr 06 2018 12:30 PM    Automatic
172.16.0.225        c563.0d90.d625          Apr 06 2018 12:30 PM    Automatic
172.16.0.226        cc5d.398a.9f29          Apr 06 2018 12:30 PM    Automatic
172.16.0.227        e05f.5156.06bb          Apr 06 2018 12:30 PM    Automatic
172.16.0.228        0431.e3a7.1908          Apr 06 2018 12:30 PM    Automatic
172.16.0.229        3db4.61a6.af42          Apr 06 2018 12:30 PM    Automatic
172.16.0.230        c474.a5d1.047b          Apr 06 2018 12:30 PM    Automatic
172.16.0.231        f6d0.d92f.5a78          Apr 06 2018 12:30 PM    Automatic
172.16.0.232        583b.d7aa.91dd          Apr 06 2018 12:30 PM    Automatic
172.16.0.233        6595.0e48.3c27          Apr 06 2018 12:30 PM    Automatic
172.16.0.234        5079.dcb2.1f8b          Apr 06 2018 12:30 PM    Automatic
172.16.0.235        f4e3.ff9a.b403          Apr 06 2018 12:30 PM    Automatic
172.16.0.236        15aa.d3ee.da2e          Apr 06 2018 12:30 PM    Automatic
172.16.0.237        6632.693d.dcfa          Apr 06 2018 12:30 PM    Automatic
172.16.0.238        1a41.8f29.8acc          Apr 06 2018 12:30 PM    Automatic
172.16.0.239        50da.452c.8c65          Apr 06 2018 12:30 PM    Automatic
172.16.0.240        b781.48b6.1bfd          Apr 06 2018 12:30 PM    Automatic
172.16.0.241        b930.a78d.1f81          Apr 06 2018 12:30 PM    Automatic
172.16.0.242        bb85.b424.c390          Apr 06 2018 12:30 PM    Automatic
172.16.0.243        1edd.d2ad.065c          Apr 06 2018 12:30 PM    Automatic
172.16.0.244        7957.36bf.83c3          Apr 06 2018 12:30 PM    Automatic
172.16.0.245        243b.446c.f15f          Apr 06 2018 12:30 PM    Automatic
172.16.0.246        69ab.8f11.38ae          Apr 06 2018 12:30 PM    Automatic
172.16.0.247        92f3.3446.17f7          Apr 06 2018 12:30 PM    Automatic
172.16.0.248        d735.d4a9.e2db          Apr 06 2018 12:30 PM    Automatic
172.16.0.249        055c.323b.1bb5          Apr 06 2018 12:30 PM    Automatic
172.16.0.250        fe3f.f042.abe2          Apr 06 2018 12:30 PM    Automatic
172.16.0.251        a115.8d31.26c5          Apr 06 2018 12:30 PM    Automatic
172.16.0.252        dfb8.b813.ffcf          Apr 06 2018 12:30 PM    Automatic
172.16.0.253        0ad6.04df.7fe6          Apr 06 2018 12:30 PM    Automatic
172.16.0.254        ba84.42ec.bf5d          Apr 06 2018 12:30 PM    Automatic
Router#
```

Obr. 4: Množina IP adries, ktoré sú rezervované obsahuje všetky možné IP adresy prideliteľné klientom

Správnosť vyplnenia packetu bola overená aj pomocou programu Wireshark:



Obr. 5: Štruktúra DHCP packetu odchyteného programom Wireshark

Literatúra

[1] Austinmarton: Sending raw Ethernet packets from a specific interface in C on Linux. online, 2011.
URL https://austinmarton.wordpress.com/2011/09/14/sending-raw-ethernet-packets-from-a-specific-interface-in-c-on-linux/

[2] Droms, R.: Dynamic Host Configuration Protocol. online, 1997.
URL https://tools.ietf.org/html/rfc2131

[3] Netmanias: Understanding the basic operations of DHCP. online, 2013.
URL https://www.netmanias.com/en/post/techdocs/5998/dhcp-network-protocol/understanding-the-basic-operations-of-dhcp

[4] Zuzčák, M.: Bezpečnosť na LAN pod lupou: DHCP spoofing. online, 2011.
URL https://secit.sk/sk/content/bezpecnost-na-lan-pod-lupou-dhcp-spoofing