

Vysoké učení technické v Brně

Fakulta informačních technologií

Počítačové komunikace a sítě
2017/2018

Projekt 2 Varianta 2
DHCP Starvation Attack

Obsah

| | |
|-------------------------------------|----------|
| Zadanie..... | 3 |
| Problematika | 3 |
| Implementácia..... | 3 |
| Testovanie | 4 |
| Kompilácia a spustenie | 4 |
| Príloha | 5 |
| Zdroje | 6 |

Zadanie

Naštudovať problematiku DHCP útoku a relevantné informácie uviesť v projektovej dokumentácii. Naprogramovať aplikáciu DHCP Starvation útok, ktorý by vyčerpá adresný pool legitímneho DHCP serveru. Demonštrovať činnosť aplikácie v podmienkach Vašej testovacej siete.

Problematika

Našou úlohou bolo implementovať v jazyku C/C++ *DHCP Starvation útok*, ktorý v zjednodušenej forme spamuje DHCP server takzvanými DHCP DISCOVERY správami, ktorým je pridelená falošná *MAC adresa*, a taktiež falošné *transaction ID*. DHCP server pre dané správy dočasne alokuje miesto, a teda ich berie ako obsadené. Tento proces sa opakuje, čím sa vyčerpá *adresný pool*, a teda nie je možné dostať od DHCP serveru pridelenú adresu siete a získať prístup ku sieti. V kompletnom útoku by náš program mal reagovať aj na spätné DHCP OFFER správy, z ktorých potrebujeme vyčítať adresu serveru a nám pridelenú IP adresu, následne odošleme na server *DHCP REQUEST*, ktorý obsahuje *DHCP_SERVER_IDENTIFIER* a *REQUIRED_IP*, tým nám server zaalokuje po dobu lease time (často býva až do 24 hodín) pridelenú IP adresu, ktorú nebude môcť využiť iný host. Týmto opakovaným procesom vyčerpáme adresný pool a router nebude schopný prideľovať novým užívateľom adresy aj po ukončení nášho útoku. Proti tomuto rozšírenému útoku sa však na základe našich skúseností väčšina moderných zariadení dokáže efektívne brániť, a však, jednoduché zasielanie DISCOVERY správ sa nám overilo ako funkčné.

Implementácia

Vrámci našej implementácie sme sa rozhodli využívať takzvaný RAW paket, ktorý manuálne napĺňame potrebnými parametrami ktoré musí daný DHCP packet spĺňať v závislosti od typu nami posielaného paketu (DISCOVERY, REQUEST). Na pripojenie na DHCP server sme využívali knižnicu **pcap.h** pre jazyk C/C++, z nej sme použili funkcie *pcap_open_live* pre spojenie serveru s daným interface-om (uvedený v argumente programu) a *pcap_inject* pre odoslanie nami vytvoreného RAW paketu na server, a aj niekoľko ďalších potrebných štruktúr. Ďalej sme potrebovali vytvoriť podľa nájdeného vzoru štruktúru pre DHCP ktorú sme v pakete odosieli, štruktúru IP hlavičky, atď. Program po spustení overí argumenty a v prípade ich korektnosti sa pokúsi napojiť na daný interface, v prípade úspešného napojenia delíme program pomocou príkazu `fork()` na dve vlákna. Prvé vlákno vytvára pakety ktoré potom podľa našej potreby napĺňa a neustále zasiela na DHCP server pomocou broadcastu vyššie spomínané DISCOVERY správy, ktorým generuje náhodnú MAC adresu a náhodné transaction ID. Druhé vlákno čaká na správy z DHCP serveru, na ktoré reaguje vypísaním IP adresy a následných zaslaním vyššie spomínanej REQUEST správy na server, ktorá by nám mala zabezpečiť dlhšie trvajúce pridelenie IP adresy. Táto implementácia sa nám však neoverila, pretože ako sme skôr spomínali, väčšina moderných zariadení sa proti takto ľahkému útoku dokáže brániť. Príkladom mechanizmu bránenia ktorý sa nám overil aj na našej domácej testovacej sieti bolo, že router je schopný overiť MAC adresu ktorá mu je doručená v pakete s MAC adresou odosielateľa. A však aj na základe zabezpečenia sa nám podarilo pomocou našej implementácie server odstaviť po dobu behu programu, čo je tiež forma Starvation útoku.

Testovanie

V rámci testovania sme použili našu domácu sieť, v ktorej máme router ASUS RT-N14U na ktorý sme pripojili počítač ktorý realizoval útok podľa zadania. Zasielanie DISCOVERY správ sme overili pomocou nástroju Wireshark (viz. Obr. 1-2), následne sme sa pokúšali o opakované pripojenie do siete pomocou smartphonu a taktiež ďalšieho počítača, podľa očakávania neúspešne. Server však neodpovedal OFFER správami, takže sme neboli schopný overiť zasielanie REQUEST správ na server, a však projekt spĺňa zadanie aj bez tohto zasielania. V prípade že by to testovacie prvky v laboratóriu umožňovali a teda nami vytvorené REQUEST správy by boli zasielané, nezaručujeme úplnú korektnosť danej implementácie. V prípade problémov s REQUEST správami postačí v rámci súboru *ipk-dhcpstarve.cpp* zakomentovať riadok 391, a implementácia bude fungovať pomocou už spomínaného spamovania.

Kompilácia a spustenie

K projektu je vytvorený jednoduchý Makefile, obsahujúci cieľ all ktorý sa pomocou prekladaču gcc postará o korektné preloženie nami vytvoreného súboru. Spustiteľný súbor je terminálová aplikácia ktorá sa spúšťa pomocou zadania

```
./ipk-dhcpstarve -i <interface>
```

Kde interface udáva interface pomocou ktorého sme napojený na DHCP server, v Linuxových a Unixových distribúciách je možné zistiť dostupné rozhrania pomocou príkazu *ifconfig*, program je kompatibilný s Linuxom (funguje aj na nami testovanom macOS).

Príloha

bootp
Expression...

| No. | Time | Source | Destination | Protocol | Length | Info |
|------|-----------|---------|-----------------|----------|--------|---|
| 3755 | 14.296652 | 0.0.0.0 | 255.255.255.255 | DHCP | 590 | DHCP Discover - Transaction ID 0x4d5cbb48 |
| 3756 | 14.296653 | 0.0.0.0 | 255.255.255.255 | DHCP | 590 | DHCP Discover - Transaction ID 0x122ba97c |
| 3757 | 14.296653 | 0.0.0.0 | 255.255.255.255 | DHCP | 590 | DHCP Discover - Transaction ID 0x45209e8d |
| 3758 | 14.296653 | 0.0.0.0 | 255.255.255.255 | DHCP | 590 | DHCP Discover - Transaction ID 0x557984d1 |
| 3759 | 14.296654 | 0.0.0.0 | 255.255.255.255 | DHCP | 590 | DHCP Discover - Transaction ID 0x84e6b64 |
| 3760 | 14.296654 | 0.0.0.0 | 255.255.255.255 | DHCP | 590 | DHCP Discover - Transaction ID 0x39068905 |
| 3761 | 14.296654 | 0.0.0.0 | 255.255.255.255 | DHCP | 590 | DHCP Discover - Transaction ID 0xb4c2a89 |
| 3762 | 14.296655 | 0.0.0.0 | 255.255.255.255 | DHCP | 590 | DHCP Discover - Transaction ID 0x531f12b |
| 3763 | 14.296656 | 0.0.0.0 | 255.255.255.255 | DHCP | 590 | DHCP Discover - Transaction ID 0x37d0bcf2 |
| 3764 | 14.296656 | 0.0.0.0 | 255.255.255.255 | DHCP | 590 | DHCP Discover - Transaction ID 0x4047dd7c |
| 3765 | 14.296656 | 0.0.0.0 | 255.255.255.255 | DHCP | 590 | DHCP Discover - Transaction ID 0x57ef1d8c |
| 3766 | 14.296657 | 0.0.0.0 | 255.255.255.255 | DHCP | 590 | DHCP Discover - Transaction ID 0x1c738573 |
| 3767 | 14.296657 | 0.0.0.0 | 255.255.255.255 | DHCP | 590 | DHCP Discover - Transaction ID 0x6414098c |
| 3768 | 14.296657 | 0.0.0.0 | 255.255.255.255 | DHCP | 590 | DHCP Discover - Transaction ID 0xaa4ff0a |
| 3769 | 14.296657 | 0.0.0.0 | 255.255.255.255 | DHCP | 590 | DHCP Discover - Transaction ID 0x3b6bee7b |
| 3770 | 14.296658 | 0.0.0.0 | 255.255.255.255 | DHCP | 590 | DHCP Discover - Transaction ID 0x465cc4dc |
| 3771 | 14.296658 | 0.0.0.0 | 255.255.255.255 | DHCP | 590 | DHCP Discover - Transaction ID 0x44f0444e |
| 3772 | 14.296658 | 0.0.0.0 | 255.255.255.255 | DHCP | 590 | DHCP Discover - Transaction ID 0x2f4a4199 |
| 3773 | 14.296659 | 0.0.0.0 | 255.255.255.255 | DHCP | 590 | DHCP Discover - Transaction ID 0x7740bf47 |

▶ Frame 3759: 590 bytes on wire (4720 bits), 590 bytes captured (4720 bits) on interface 0

▶ Ethernet II, Src: Ze:el:ee:6e:c0:20 (2e:el:ee:6e:c0:20), Dst: Broadcast (ff:ff:ff:ff:ff:ff)

▶ Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255

▶ User Datagram Protocol, Src Port: 68, Dst Port: 67

▶ Bootstrap Protocol (Discover)

```

0000  ff ff ff ff ff ff 2e e1  ee 6e c0 20 08 00 45 10  .....n..E.
0010  02 40 ff ff 00 00 10 11  a8 9e 00 00 00 00 ff ff  .@.....
0020  ff ff 00 44 00 43 02 2c  00 00 01 01 06 00 08 4e  ...D.C....N
0030  66 b4 00 00 80 00 00 00  00 00 00 00 00 00 00 00  f.....
0040  00 00 00 00 00 00 2e e1  ee 6e c0 20 00 00 00 00  .....n.....
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  .....
0060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  .....
0070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  .....
0080  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  .....
0090  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  .....
00a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  .....
00b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  .....
00c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  .....
00d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  .....
00e0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  .....
00f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  .....
0100  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  .....
0110  00 00 00 00 00 63 82  53 63 35 01 01 ff 00 00  .....c.Sc5....

```

Bootstrap Protocol: Protocol
Packets: 3966 · Displayed: 3647 (92.0%)
Profile: Default

Obr. 1

Wi-Fi: en0
Wireshark - Packet 3759 - wireshark_en0_20180409211950_rhY9VNZ

```

Hardware address length: 6
Hops: 0
Transaction ID: 0x084e66b4
Seconds elapsed: 0
▶ Bootp flags: 0x8000, Broadcast flag (Broadcast)
Client IP address: 0.0.0.0
Your (client) IP address: 0.0.0.0
Next server IP address: 0.0.0.0
Relay agent IP address: 0.0.0.0
Client MAC address: 2e:e1:ee:6e:c0:20 (2e:e1:ee:6e:c0:20)
Client hardware address padding: 000000000000000000000000
Server host name not given
Boot file name not given
Magic cookie: DHCP
▼ Option: (53) DHCP Message Type (Discover)
    Length: 1
    DHCP: Discover (1)
▶ Option: (255) End
    Padding: 0000000000000000000000000000000000000000000000000000000000000000...
0000 ff ff ff ff ff ff 2e e1 ee 6e c0 20 08 00 45 10 .....n..E.
0010 02 40 ff ff 00 00 10 11 a8 9e 00 00 00 00 ff ff @.....
0020 ff ff 00 44 00 43 02 2c 00 00 01 01 05 00 08 4e ...D.C.....N
0030 56 b4 00 00 80 00 00 00 00 00 00 00 00 00 00 00 f.....
0040 00 00 00 00 00 00 2e e1 ee 6e c0 20 00 00 00 00 .....n.....
0050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0080 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00a0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00b0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00c0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00d0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00e0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00f0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0100 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0110 00 00 00 00 00 00 63 82 53 63 35 01 01 ff 00 00 .....C. Sc5.....
0120 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0130 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0140 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0150 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0160 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0170 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0180 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....

```

No. 3759 · Time: 14.296654 · Source: 0.0.0.0 · Destination: 255.255.255.255 · Protocol: DHCP · Length: 590 · Info: DHCP Discover - Transaction ID 0x84e66b4

Help
Close

Obr. 2

Zdroje

- [1] Cprogramming.com forum - <https://cboard.cprogramming.com/c-programming/124445-dhcp-discover.html>
- [2] Udpdphdr structure reference - http://www.cse.scu.edu/~dclark/am_256_graph_theory/linux_2_6_stack/structudphdr.html
- [3] DHCP Demo project - <http://ftp.icpdas.com.tw/pub/cd/8000cd/napdos/8000/843x883x/tcp/other/dhcp/client.c>
- [4] TCP/IP Guide, DHCP message format - http://www.tcpipguide.com/free/t_DHCPMessageFormat.htm
- [5] RFC 2131 DHCP - <http://www.faqs.org/rfcs/rfc2131.html>
- [6] RFC 2131 - <https://www.ietf.org/rfc/rfc2131.txt>
- [7] RFC 2132 - <https://tools.ietf.org/html/rfc2132>
- [8] DHCP plugin for Nagios - <https://cs.uwaterloo.ca/twiki/pub/CF/DhcpDebug/dhcp.c>
- [9] Wiki/DHCP - https://en.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol
- [10] Preventing DHCP Starvation attack by Cheers - <http://www.revolutionwifi.net/revolutionwifi/2011/03/preventing-dhcp-starvation-attacks.html>