



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

**DHCP ÚTOKY**

DHCP ATTACKS

**PROJEKTOVÁ DOKUMENTACE**

PROJECT DOCUMENTATION

**AUTOR PRÁCE**

AUTHOR

**Bc. DAVID KOLEČKÁŘ**

**BRNO 2018**

# Obsah

<b>1</b>	<b>Úvod</b>	<b>2</b>
1.1	DHCP . . . . .	2
<b>2</b>	<b>Implementace</b>	<b>4</b>
<b>3</b>	<b>Demonstrace (testování)</b>	<b>5</b>
	<b>Literatura</b>	<b>7</b>

# Kapitola 1

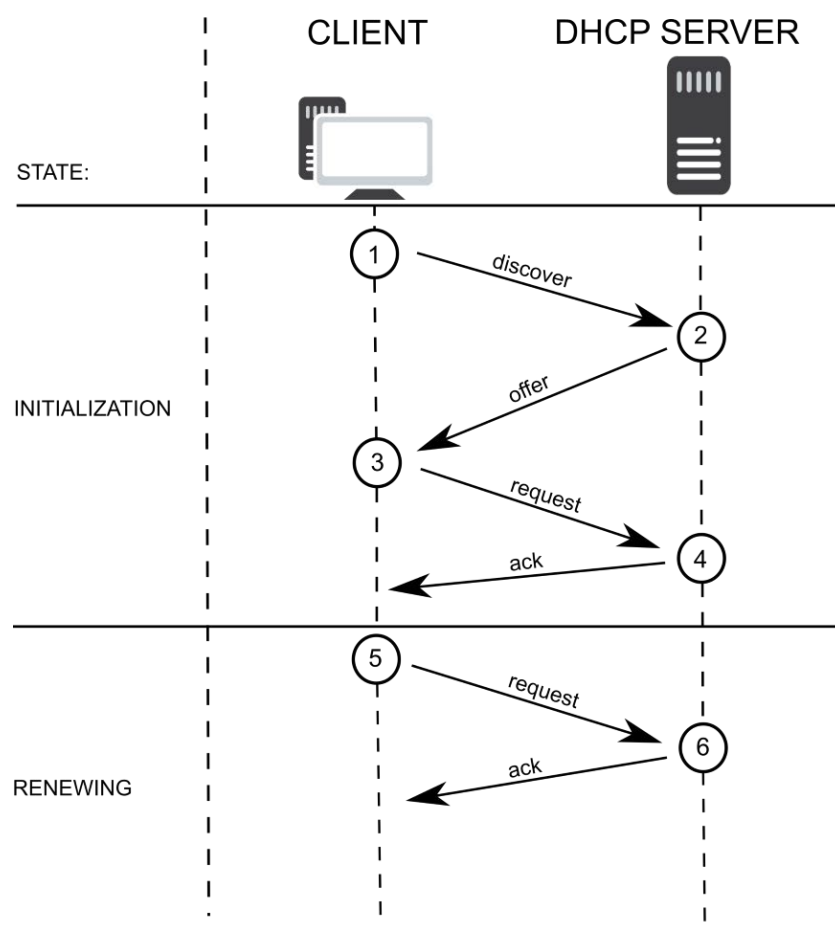
## Úvod

Cílem projektu je nastudovat problematiku DHCP útoků a naprogramovat aplikace realizující DHCP Starvation a Rogue DHCP Server. Úkolem DHCP Starvation je vyčerpat adresní pool legitimního serveru. Rogue DHCP Server provozuje falešný DHCP server poskytující klientům vlastní síťové parametry. V neposlední řadě je nutné demonstrovat aplikace ve vlastní testovací síti.

### 1.1 DHCP

DHCP (Dynamic Host Configuration Protocol) nám automaticky přiděluje následující síťové parametry: IP adresu, masku podsítě, výchozí bránu, primární a sekundární DNS server a další parametry. Komunikace probíhá na portech 68 (klient) a 67 na kterém naslouchá DHCP server.

Klient nejdříve pošle DHCPDISCOVER zprávu na broadcast rozhraní a čeká na odpověď od serveru. Ten by měl poslat zprávu typu DHCPOFFER, ve které je umístěna adresa, kterou klientovi nabízí. Klient ji zpracuje a může na ni odpovědět zprávou DHCPREQUEST, ve které pošle adresu o kterou žádá. Server mu ji vzápětí potvrdí odpovědí DHCPACK. Jakmile klient obdrží DHCPACK, může síťové parametry používat. Po vypršení doby výpůjčky, může klient znovu zaslat DHCPREQUEST, ve které žádá o stejnou adresu. Ovšem adresa mu nemusí být přidělena pokud je již zabrána [2]. Celou komunikaci lze vidět na obrázku 1.1



Obrázek 1.1: Ilustrace komunikace mezi klientem a serverem

## Kapitola 2

# Implementace

Aplikace je napsána v jazyce C++. Pro kontrolu parametrů aplikace využívají funkci *getopt*. Jednotlivé programy používají RAW sockety [3], který slouží ke komunikaci v síti. Raw socket umožňuje programátorovi v aplikaci přímé odesílání a přijímání síťových paketů s možností obejít standardní síťová zapouzdření. Pro nastavení jednotlivých DHCP Options jsem využil informace z [1]. V případě DHCP Starvation se program rozdělí na 2 procesy. Každý z nich obsahuje nekonečný cyklus. První odesílá každou vteřinu DHCPDISCOVER, druhý přijme DHCPOFFER a obratem posílá DHCPREQUEST.

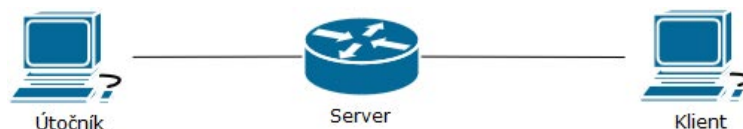
Základem Rogue DHCP Serveru je nekonečná smyčka, ve které se odchyťávají jednotlivé zprávy. V případě příchozího DHCPDISCOVER odešle falešný server DHCPOFFER s volnou adresou. Jestliže se jedná o DHCPREQUEST zkontroluje zda požadovaná adresa již není použita či není v jiné podsíti. V takovém případě server zareaguje zprávou DHCPNAK. Jinak server odpoví zprávou DHCPACK a klient může síťové parametry začít používat.

## Kapitola 3

# Demonstrace (testování)

### Topologie sítě:

V prostředí VirtualBox jsem nainstaloval 3 virtuální počítače (Klient, Útočník, Server). Tyto počítače běží pod operačním systémem Ubuntu, jehož obraz (ISA2015) nám byl poskytnut jako doporučený systém pro testování aplikace. Všechny počítače jsem propojil vnitřní sítí (internal network) na rozhraní eth1. Na počítači nazvaný Server jsem nainstaloval DHCP server, nakonfiguroval ho a nastavil mu statickou IP adresu.



Obrázek 3.1: Topologie sítě

Aplikace je možné přeložit příkazem *make*, který vygeneruje dva spustitelné soubory *pds-dhcpstarve* a *pds-dhcprogue*. Jednotlivé parametry programu jsou popsány v souboru *readme*, případně je lze zobrazit pomocí nápovědy u jednotlivého programu. Překlad programu byl testován i na školním serveru Merlin, který proběhl bez problémů.

### Simulace DHCP Starvation:

Na počítači Server jsem zapnul DHCP server a na počítači Útočník jsem spustil program, který jsem nechal běžet. Program lze ukončit zasláním signálu SIGINT (CRL + C).

```
sudo ./pds-dhcpstarve -i eth1
```

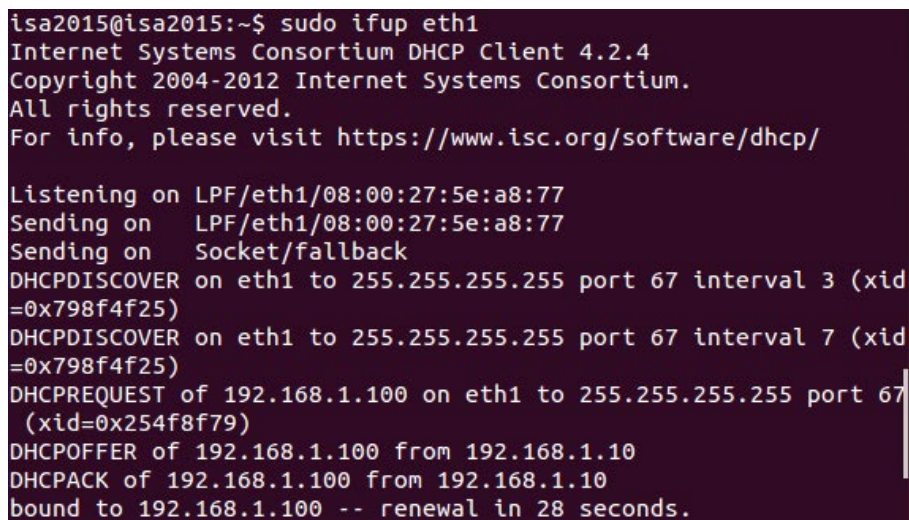
V závislosti na nastavení DHCP serveru (možnosti rozsahu) se za pár minut adresní pool vyčerpá. To jsem si ověřil zapnutím počítače Klient, který se již nemohl připojit. Dále jsem se také podíval do log souboru DHCP serveru (*dhcpd.leases*), kde lze vidět jednotlivé zaregistrované adresy. Na počítači Útočník jsem měl také spuštěn nástroj Wireshark, kde jsem sledoval jednotlivé příchozí a odchozí DHCP zprávy.

## Simulace Rogue DHCP Serveru:

Stejně jako v předešlém případě, na počítači Server jsem zapnul DHCP server a na počítači Útočník jsem spustil program, který jsem nechal běžet. Program lze ukončit zasláním signálu SIGINT (CRL + C).

```
./pds-dhcprogue -i eth1 -p 192.168.1.100-192.168.1.110 -g 192.168.1.1  
-n 8.8.8.8 -d fit.vutbr.cz -l 3600
```

Poté jsem na počítači Klient požádal o přiřazení síťových parametrů od serveru a to pomocí příkazu `sudo ifup eth1`. Jak lze vidět na obrázku 3.2, falešný server poskytl vlastní síťové parametry klientovi. V nástroji Wireshark jsem současně sledoval celou komunikaci a kontroloval jednotlivé hlavičky protokolů.



```
isa2015@isa2015:~$ sudo ifup 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/eth1/08:00:27:5e:a8:77  
Sending on   LPF/eth1/08:00:27:5e:a8:77  
Sending on   Socket/fallback  
DHCPDISCOVER on eth1 to 255.255.255.255 port 67 interval 3 (xid  
=0x798f4f25)  
DHCPDISCOVER on eth1 to 255.255.255.255 port 67 interval 7 (xid  
=0x798f4f25)  
DHCPPREQUEST of 192.168.1.100 on eth1 to 255.255.255.255 port 67  
(xid=0x254f8f79)  
DHCPOFFER of 192.168.1.100 from 192.168.1.10  
DHCPACK of 192.168.1.100 from 192.168.1.10  
bound to 192.168.1.100 -- renewal in 28 seconds.
```

Obrázek 3.2: Výpis z procesu přidělení adresy

# Literatura

- [1] *Dynamic Host Configuration Protocol*. 2018, [Online; navštíveno 16.04.2018].  
URL [https://en.wikipedia.org/wiki/Dynamic\\_Host\\_Configuration\\_Protocol](https://en.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol)
- [2] Droms, R.: *RFC 2131: Dynamic Host Configuration Protocol*. 1997, [Online; navštíveno 16.04.2018].  
URL <https://tools.ietf.org/html/rfc2131>
- [3] Saxena, S.: *A Guide to Using Raw Sockets*. 2015, [Online; navštíveno 17.04.2018].  
URL <https://opensourceforu.com/2015/03/a-guide-to-using-raw-sockets/>