

Dokumentácia projektu pre predmet PDS

DHCP Útoky

29. Apríla 2018

Autor: Patrik Segedy, xseged00@stud.fit.vutbr.cz
Fakulta Informačních Technologíí
Vysoké Učení Technické v Brně

Obsah

1	Úvod	1
2	DHCP Útoky.....	1
2.1	DHCP Starvation	1
2.2	DHCP Spoofing.....	1
3	Implementácia	2
3.1	DHCP Starvation	2
3.2	Rogue DHCP Server	2
4	Demonštrácia činnosti.....	3
4.1	DHCP Starvation	4
4.2	Rogue DHCP Server	5
5	Záver	6
	Literatúra	7

1 Úvod

Táto dokumentácia sa zaoberá popisom DHCP útokov, implementáciou útoku DHCP Starvation a vytvorením Rogue DHCP Servera.

Dokument sa skladá z viacerých kapitol. Kapitola 2 je venovaná možným DHCP útokom. V kapitole 3 je prezentovaná implementácia týchto útokov. Nakoniec je v poslednej, 4. kapitole demonštrovaná činnosť týchto útokov.

2 DHCP Útoky

DHCP (Dynamic Host Configuration Protocol) je protokol pre automatické prideľovanie sieťových parametrov staniciam v sieti. Okrem IP adresy posiela ďalšie parametre, ktoré sú využiteľné pri útokoch. Napríklad nastavenie východzej brány alebo adresy DNS serveru.

2.1 DHCP Starvation

Cieľom útoku DHCP Starvation je vyčerpať adresný pool legitímneho DHCP servera. Útok je posielanie veľkého množstva DHCPDISCOVER správ stále s inou falošnou MAC adresou. To zapríčini odoslanie veľkého počtu DHCPOFFER správ a vytvorenie mnohých výpožičiek, ktoré vyčerpajú celý pool servera.

Podľa rfc [1] však server nie je povinný vytvoriť lease pre adresu už pri odosielaní správy DHCPOFFER ale až pri správe DHCPACK. Preto, ak účelom klienta je naozaj vyčerpať pool servera, je potrebné na správy DHCPOFFER od servera korektne odpovedať správou DHCPREQUEST.

Ak je účelom prinútiť server aby dočasne neodpovedal na nové DHCPDISCOVER správy, tak ho stačí zahliť DHCPDISCOVER správami s rôznymi MAC adresami. To môže byť napríklad prípad následného spustenia Rogue DHCP serveru, ktorý obetiam podstrčí falošné sieťové parametre.

2.2 DHCP Spoofing

Tento útok spôsobí pridelenie falošných údajov klientovi a tým umožní útok man in the middle. DHCP spoofing nadväzuje na DHCP Starvation.

V čase keď legitímny DHCP server nie je schopný odpovedať na nové správy DHCPDISCOVER, spustíme falošný Rogue DHCP server, ktorý na tieto správy zareaguje a nič netušiacim klientom poskytne falošné sieťové údaje, ako napríklad východziu bránu alebo vlastný DNS

server. Takto je možné vytvoriť útok typu man in the middle, pretože všetka komunikácia bude preposielaná cez falošnú bránu.

3 Implementácia

Implementácia oboch útokov je v jazyku C++. Použitý bol raw socket, kvôli možnosti posielania unicastu a broadcastu na L2 adresu, čo je potrebné pre oba útoky. Štruktúra pre DHCP paket je kvôli vyhnutiu sa problémom s paddingom vytvorená s `__attribute__((packed))`. Funkcia pre výpočet kontrolného súčtu IP hlavičky je prebraná z http://www.cs.cmu.edu/afs/cs/academic/class/15213-f00/unpv12e/libfree/in_cksum.c.

3.1 DHCP Starvation

Najprv sa získa index zadaného rozhrania a naplní sa štruktúra `sockaddr_ll`. Ďalším krokom je vytvorenie DHCP paketu, ktorý obsahuje správu typu DHCPDISCOVER a do položky `chaddr` sa vloží vygenerovaná MAC adresa.

Potom sa vytvorí UDP hlavička, IP hlavička a Ethernet hlavička. V Ethernet hlavičke je ako zdrojová MAC adresa uvedená nami vygenerovaná adresa a cieľová adresa je L2 broadcast. Následne je urobený `bind()` socketu na dané rozhranie.

Po týchto krokoch môžeme začať so samotným útokom, ktorý spočíva vo vygenerovaní novej MAC adresy, uložení tejto adresy ako zdrojovej do ethernetovej hlavičky (týmto napríklad zaručíme, aby na rozhranie tohto klienta neprichádzali DHCP OFFER správy od servera) a ako adresa klienta - `chaddr` v DHCP pakete. Prvé 4B MAC adresy sa použijú ako transaction id (xid). Takto vytvorený rámec následne odošleme. Celý tento úsek kódu prebieha v nekonečnom cykle.

3.2 Rogue DHCP Server

DHCP server musí vedieť správne rozoznať prichádzajúcu správu od DHCP klienta a patrične na ňu odpovedať. Keďže sa jedná o falošný DHCP server, nie je potrebné podporovať úplne všetky typy správ, ako napríklad DHCPNAK.

Po prijatí packetu sú vymazané už neplatné výpožičky a zistí sa typ DHCP správy. Ak je to správa typu DHCPDISCOVER, vytvorí sa správa typu DHCP OFFER, ponúkne sa prvá voľná adresa z poolu a dhcp options sa nastaví podľa zadaných parametrov pri spustení programu. Navyše sa nastaví aj maska siete, na základe masky, akú má server. Potom sa správne nastaví UDP hlavička, IP hlavička, Ethernet hlavička a správa sa odošle. Správa sa odosiela L2 unicastom. V prípade ak klient

už má IP adresu a v štruktúre dhcp správy sa nachádza jeho adresa v položke ciaddr, správa sa odošle na jeho aktuálnu IP adresu.

Odpoveď na správu DHCPREQUEST funguje obdobne. Naplní sa štruktúra DHCP správy, vytvorí sa UDP hlavička, IP hlavička, Ethernet hlavička a správa sa odošle. Ak klient už mal pridelenú IP adresu, a teda žiada o obnovenie adresy, tak sa mu ponúkne ním žiadaná adresa a odošle sa na jeho aktuálnu IP adresu. Keď klient nemá pridelenú žiadnu IP adresu, ponúkne sa mu rovnaká adresa ako v DHCPDISCOVER a odošle sa na jeho L2 adresu.

Po obdržaní DHCPRELEASE správy sa v tabuľke výpožičiek vymaže záznam pre MAC adresu daného klienta.

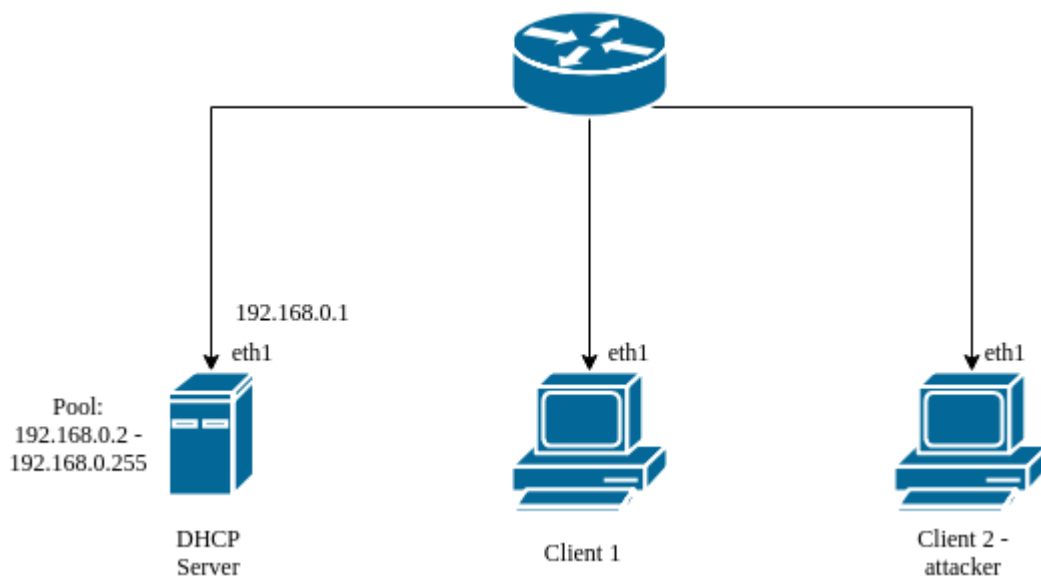
Tabuľka výpožičiek je reprezentovaná štruktúrou `vector<tuple<array<u_char, 16>, uint32_t, time_t, time_t>>`, formát záznamy vyzerá ako `[(MAC, IP address, lease start, lease end), (...), ...]`.

Pri implementácii bol využitý raw socket, kvôli potrebe posielania unicastu na L2 adresu.

4 Demonštrácia činnosti

Na nasledujúcom obrázku môžeme vidieť testovaciu topológiu vo virtualboxe. Klienti majú dynamicky pridelené IP adresy od DHCP Serveru. Client2 je stroj z ktorého budú spúšťané útoky.

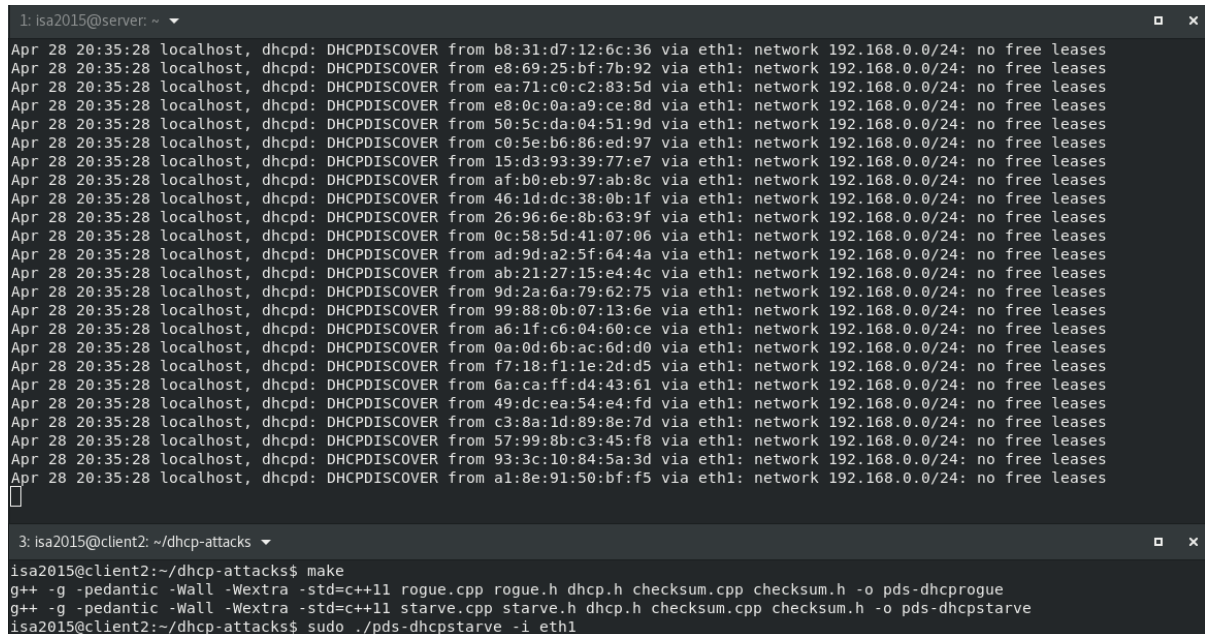
Ako DHCP server je použitý isc-dhcp-server.



Obrázok 1 Topológia siete

4.1 DHCP Starvation

Na klientovi č.2 spustíme DHCP Starvation útok a na serveri kontrolujeme log dhcp serveru vo /var/log/syslog. Z výstupu môžeme vidieť, že server na prichádzajúce DHCPDISCOVERY správy hlási, že má vyčerpaný adresný pool – no free leases.

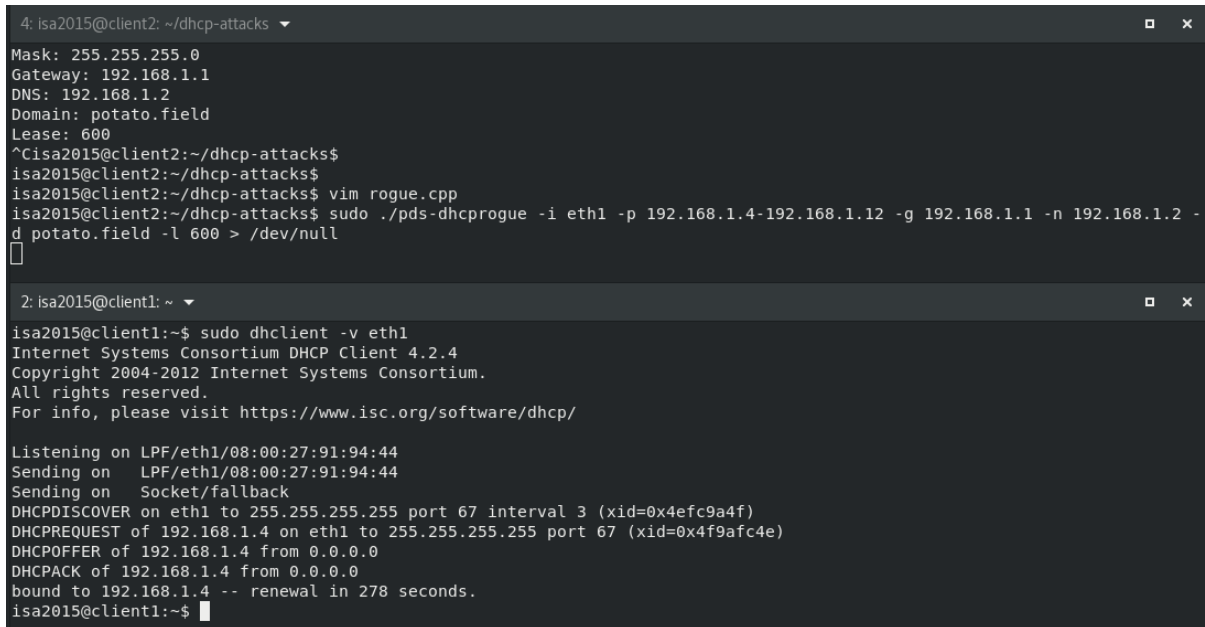


```
1: isa2015@server: ~  
Apr 28 20:35:28 localhost, dhcpd: DHCPDISCOVER from b8:31:d7:12:6c:36 via eth1: network 192.168.0.0/24: no free leases  
Apr 28 20:35:28 localhost, dhcpd: DHCPDISCOVER from e8:69:25:bf:7b:92 via eth1: network 192.168.0.0/24: no free leases  
Apr 28 20:35:28 localhost, dhcpd: DHCPDISCOVER from ea:71:c0:c2:83:5d via eth1: network 192.168.0.0/24: no free leases  
Apr 28 20:35:28 localhost, dhcpd: DHCPDISCOVER from e8:0c:0a:a9:ce:8d via eth1: network 192.168.0.0/24: no free leases  
Apr 28 20:35:28 localhost, dhcpd: DHCPDISCOVER from 50:5c:d4:04:51:9d via eth1: network 192.168.0.0/24: no free leases  
Apr 28 20:35:28 localhost, dhcpd: DHCPDISCOVER from c0:5e:b6:86:ed:97 via eth1: network 192.168.0.0/24: no free leases  
Apr 28 20:35:28 localhost, dhcpd: DHCPDISCOVER from 15:d3:93:39:77:e7 via eth1: network 192.168.0.0/24: no free leases  
Apr 28 20:35:28 localhost, dhcpd: DHCPDISCOVER from af:b0:eb:97:ab:8c via eth1: network 192.168.0.0/24: no free leases  
Apr 28 20:35:28 localhost, dhcpd: DHCPDISCOVER from 46:1d:dc:38:0b:1f via eth1: network 192.168.0.0/24: no free leases  
Apr 28 20:35:28 localhost, dhcpd: DHCPDISCOVER from 26:96:6e:8b:63:9f via eth1: network 192.168.0.0/24: no free leases  
Apr 28 20:35:28 localhost, dhcpd: DHCPDISCOVER from 0c:58:5d:41:07:06 via eth1: network 192.168.0.0/24: no free leases  
Apr 28 20:35:28 localhost, dhcpd: DHCPDISCOVER from ad:9d:a2:5f:64:4a via eth1: network 192.168.0.0/24: no free leases  
Apr 28 20:35:28 localhost, dhcpd: DHCPDISCOVER from ab:21:27:15:e4:4c via eth1: network 192.168.0.0/24: no free leases  
Apr 28 20:35:28 localhost, dhcpd: DHCPDISCOVER from 9d:2a:6a:79:62:75 via eth1: network 192.168.0.0/24: no free leases  
Apr 28 20:35:28 localhost, dhcpd: DHCPDISCOVER from 99:88:0b:07:13:6e via eth1: network 192.168.0.0/24: no free leases  
Apr 28 20:35:28 localhost, dhcpd: DHCPDISCOVER from a6:1f:c6:04:60:ce via eth1: network 192.168.0.0/24: no free leases  
Apr 28 20:35:28 localhost, dhcpd: DHCPDISCOVER from 0a:0d:6b:ac:6d:d0 via eth1: network 192.168.0.0/24: no free leases  
Apr 28 20:35:28 localhost, dhcpd: DHCPDISCOVER from f7:18:f1:1e:2d:d5 via eth1: network 192.168.0.0/24: no free leases  
Apr 28 20:35:28 localhost, dhcpd: DHCPDISCOVER from 6a:ca:ff:d4:43:61 via eth1: network 192.168.0.0/24: no free leases  
Apr 28 20:35:28 localhost, dhcpd: DHCPDISCOVER from 49:dc:ea:54:e4:fd via eth1: network 192.168.0.0/24: no free leases  
Apr 28 20:35:28 localhost, dhcpd: DHCPDISCOVER from c3:8a:1d:89:8e:7d via eth1: network 192.168.0.0/24: no free leases  
Apr 28 20:35:28 localhost, dhcpd: DHCPDISCOVER from 57:99:8b:c3:45:f8 via eth1: network 192.168.0.0/24: no free leases  
Apr 28 20:35:28 localhost, dhcpd: DHCPDISCOVER from 93:3c:10:84:5a:3d via eth1: network 192.168.0.0/24: no free leases  
Apr 28 20:35:28 localhost, dhcpd: DHCPDISCOVER from a1:8e:91:50:bf:f5 via eth1: network 192.168.0.0/24: no free leases  
[  
3: isa2015@client2: ~/dhcp-attacks  
isa2015@client2:~/dhcp-attacks$ make  
g++ -g -pedantic -Wall -Wextra -std=c++11 rogue.cpp rogue.h dhcp.h checksum.cpp checksum.h -o pds-dhcprogue  
g++ -g -pedantic -Wall -Wextra -std=c++11 starve.cpp starve.h dhcp.h checksum.cpp checksum.h -o pds-dhcpstarve  
isa2015@client2:~/dhcp-attacks$ sudo ./pds-dhcpstarve -i eth1
```

Obrázok 2 DHCP Starvation

4.2 Rogue DHCP Server

Na klientovi číslo 2 spustíme DHCP Server a na klientovi č. 1 požiadame o novú IP adresu pomocou `dhclient -v eth1`. Ako vidíme na obrázku, klient má korektne pridelenú adresu od serveru, ktorý beží na klientovi č.2.



```
4: isa2015@client2: ~/dhcp-attacks
Mask: 255.255.255.0
Gateway: 192.168.1.1
DNS: 192.168.1.2
Domain: potato.field
Lease: 600
^Cisa2015@client2:~/dhcp-attacks$
isa2015@client2:~/dhcp-attacks$
isa2015@client2:~/dhcp-attacks$ vim rogue.cpp
isa2015@client2:~/dhcp-attacks$ sudo ./pds-dhcprogue -i eth1 -p 192.168.1.4-192.168.1.12 -g 192.168.1.1 -n 192.168.1.2 -
d potato.field -l 600 > /dev/null
[]

2: isa2015@client1: ~
isa2015@client1:~$ 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/eth1/08:00:27:91:94:44
Sending on   LPF/eth1/08:00:27:91:94:44
Sending on   Socket/fallback
DHCPDISCOVER on eth1 to 255.255.255.255 port 67 interval 3 (xid=0x4efc9a4f)
DHCPPREREQUEST of 192.168.1.4 on eth1 to 255.255.255.255 port 67 (xid=0x4f9afc4e)
DHCPOFFER of 192.168.1.4 from 0.0.0.0
DHCPACK of 192.168.1.4 from 0.0.0.0
bound to 192.168.1.4 -- renewal in 278 seconds.
isa2015@client1:~$
```

Obrázok 3 Rogue DHCP Server

5 Záver

Rogue DHCP Server je schopný správne odpovedať na správy DHCPDISCOVER, DHCPREQUEST, DHCPRELEASE a nastaviť obeť IP adresu zo zadaného rozsahu. Útok DHCP Starvation prebieha iba ako DISCOVER flood, čo spôsobí na strane legitímneho klienta hlášku „No free leases“ v logu dhcpd.

Možným rozšírením je pri útoku DHCP Starvation posielat' aj správy DHCPREQUEST, pre skutočné vyčerpanie adresného rozsahu legitímneho serveru.

Pri implementácii boli využité niektoré štruktúry a princíp funkcií ako v mojom projekte do predmetu ISA¹, kde bolo úlohou implementovať DHCP server.

Projekt bol vypracovaný a otestovaný na zadanom virtuálnom stroji s operačnom systéme Ubuntu 14.04.1 a prekladačom GCC vo verzii 4.8.4.

¹ Dostupné z <https://github.com/psegedy/dserver>

Literatúra

- [1] RFC 2131[online]. [cit. 2018-04-28]. Dostupné z: <https://tools.ietf.org/pdf/rfc2131.pdf>