Dokumentace k projektu IPK
Scanner síťových služeb

Petr Křehlík
xkrehl04

21. dubna 2019

Obsah

1 Úvod

Cílem projektu bylo vytvořit funkční scanner síťových služeb. Projekt má za úkol skenovat UDP a TCP porty na zadané IP adrese nebo doménové jméno. Pro implementaci jsem zvolil C++ kvůli práci s řetězci a dalšími vylepšeními, které usnadňují práci.

2 Argumenty

Aplikace je schopná zpracovat jako cíl pro skenování IP adresu nebo doménové jméno. Port se zadávají samostatně pro UDP (`-pu`) a TCP (`-pt`).
Mohou být ve formátu:

- 80
- 80-85
- 80,81,82

Jako poslední lze specifikovat argument `-i`, kterým ručně určujete síťové rozhraní.

3 Síťové rozhraní

3.1 Výběr

Síťové rozhraní může být zadáno pomocí argumentu `-i` nebo nemusí být zadáno. V tom případě se prochází všechna dostupná rozhraní a vybere se první dostupné, které není loopback.

3.2 Získání IP adresy

IP adresa rohraní se získává otevřením socketu s argumentem `AF_INET` a poté použitím funkce `ioctl` s argumentem `SIOCGIFADDR`, která naplní strukturu `ifreq` infomracemi o rohraní. Poté stačí přistoupit k IP adrese.

4 Překlad doménového jména

Pro překlad doménového jménu slouží funkce `getaddrinfo`, poté se získaná IP adresa převede na standardní formát (čísla a tečky). Funkcí prochází i IP adresy, čímž se ověřuje jejich správnost.

5 Skenování portů

5.1 UDP

UDP skenování probíhá v následujících krocích:

- Vytvoření za sebe navazujících struktur IP a UDP hlaviček.
- Vytvoření a nastavení socketu (RAW, UDP)
- Vyplnění IP hlavičky (verze protokolu IP, TTL, protokol, ...)
- Vyplnění UDP hlavičky (port, cílová IP)
- Výpočet kontrolního součtu IP hlavičky pro zajištění integrity dat
- Vytvoření a nastavení filtru PCAP (filtr zajišťuje zachycení jen paketů, které jsou poslané z cílové na zdrojovou IP adresu se správnými porty a jsou typu ICMP)
- Odeslání paketu
- Vyhodnocení přijatého paketu ICMP
  - Pokud má paket typ 3 a kód 3 tak se port prohlásí za uzavřený
  - Pokud nepřijde žádná odpověď v určitém čase, tak se paket pošle znovu. Jestli ani poté nepřijde žádná odpověď tak se port prohlásí za otevřený.

Problémem UDP je, že reálně nelze rozlišit otevřený port od filtrovaného, jelikož se ani v jednom případě nevrací žádná odpověď.

5.2 TCP

TCP skenování probíhá v následujících krocích:

- Vytvoření za sebe navazujících struktur IP a TCP hlaviček.
- Vytvoření a nastavení socketu (RAW, TCP)
- Vyplnění IP hlavičky (verze protokolu IP, TTL, protokol, ...)
- Vyplnění TCP hlavičky (port, cílová IP, SYN, SEQ (náhodné číslo), ...)

- Vytvoření a vyplnění pseudo-TCP hlavičky (cílová a zdrojová IP, protokol, délka)
- Vytvoření pseudo-paketu, který se naplní pseudo-TCP hlavičkou a TCP hlavičkou (pseudo-paket se používá jen pro výpočet kontrolního součtu, ve skutečnosti se nikam neposílá)
- Výpočet kontrolního součtu z pseudo-paketu a uložení do TCP hlavičky
- Výpočet kontrolního součtu IP hlavičky pro zajištění integrity dat
- Vytvoření a nastavení filtru PCAP (filtr zajišťuje zachycení jen paketů, které jsou poslané z cílové na zdrojovou IP adresu se správnými porty)
- Odeslání paketu
- Vyhodnocení přijatých paketů
  - Pokud má paket nastavené příznaky SYN a ACK, tak je port otevřený
  - Pokud má paket nastavené příznaky ACK a RST, tak je port zavřený
  - Pokud nepřijde žádná odpověď v určitém čase, tak se paket pošle znovu. Jestli ani poté nepřijde žádná odpověď tak se port prohlásí za filtrovaný.

5.3 Problém endianity

Při plnění hlaviček v paketu je nutné u vícebajtových polí (více jak 8 bitů) provést konverzi čísla pomocí funkce `htons`(16 bitů) nebo `htonl` (32 bitů) na tzv. network byte order. Network byte order totiž používá big endian a pokud stanice používá little endian, je nutné provést konverzi. Při čtení údajů z paketu je nutné provést navíc `byte swap`.

6 Použití aplikace

Aplikaci je nutné spouštět s root přístupem (sudo) kvůli vytváření socketu. Použití aplikace je možné zobrazit spuštěním aplikace s argumente `--help` nebo `-h\`.

7 Testování

Testování jsem prováděl ručně, tedy spouštěním s různými vstupy a porovnával s výstupy open-source aplikace `nmap` se stejnými vstupy.

8 Chybové kódy

- 0..Žádná chyba
- 1..Chyba počtu nebo formátu argumentů
- 2..Chyba socketu
- 3..Chyba pcap
- 4..Chyba síťových funkcí
- 99..Neznámá chyba

9 Závěr

Projekt se mi podařilo dotáhnout do funkční podoby, nicméně po několika nezdařených pokusech jsem vynechal podporu IPv6, což mě mrzí. Co se týče IPv4 tak po testování je program spolehlivý a použitelný. Ačkoliv nedopadl projekt dle mých představ tak mi přinesl mnoho znalostí z oblasti síťové komunikace.

10 Zdroje použité v projektu

TheComet. C++ cross platform resolve hostname to ip library. Gamedev [online]. 2015 [cit. 2019-04-21]. Dostupné z: https://www.gamedev.net/forums/topic/671428-c-cross-platform-resolve-hostname-to-ip-library/?fbclid=IwAR0LJNmyYsQB6-JZGreHUGv4a_8wizdOBYAjywJhpq-IM6SvHcDFmK6Kiw0

Get the IP address of a network interface in C using SIOCGIFADDR. MicroHOWTO [online]. [cit. 2019-04-21]. Dostupné z: http://www.microhowto.info/howto/get_the_ip_address_of_a_network_interface_in_c_using_siocgifaddr.html

LEBEAU, Remy. Client in C++, use gethostbyname or getaddrinfo. Stackoverflow [online]. 2018 [cit. 2019-04-21]. Dostupné z: https://stackoverflow.com/questions/52727565/client-in-c-use-gethostbyname-or-getaddrinfo

Using libpcap in C. DEV Dungeon [online]. 2017 [cit. 2019-04-21]. Dostupné z:https://www.devdungeon.com/content/using-libpcap-c

czokl. Jednoduchý TCP/UDP scanner v C++. Security portal [online]. 2011 [cit. 2019-04-21]. Dostupné z: https://www.security-portal.cz/clanky/jednoduch%C3%BD-tcpudp-scanner-v-c

How to Find Out Whether a Machine is Big Endian or Little Endian in C/C++?. TECHNOLOGY OF COMPUTING [online]. 2016 [cit. 2019-04-21]. Dostupné z:https://helloacm.com/how-to-find-out-whether-a-machine-is-big-endian-or-little-endian-in-cc/

rbaron. Raw TCP Socket. GitHub [online]. 2012 [cit. 2019-04-21]. Dostupné z:https://github.com/rbaron/raw_tcp_socket/blob/master/raw_tcp_socket.c?fbclid=IwAR1rlNNONqg5qHdfT4ue1NQcMV6-ChRjWNzGDf5DEhbTYqeCVcXgMIDQHBM

LINUX SOCKET PART 17: Advanced TCP/IP - THE RAW SOCKET PROGRAM EXAMPLES. Tenouk [online]. [cit. 2019-04-21]. Dostupné z: https://www.tenouk.com/Module43a.html