VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Druhý projekt do předmětu IPK

Jednoduchý portový skener

Beran Jan xberan43

21. dubna 2019

Obsah

1 Zadání projektu

Zadáním projektu bylo vytvořit jednoduchý skener TCP/UDP portů pomocí BSD soketů. Zadání dále specifikovalo použitý jazyk; jazyk C. Podrobnější informace (specifikace principu skenování) v sekci „Podrobnější popis principu skenování na základě zadání projektu“.

2 Teorie

V následujících sekcích se nachází stručně shrnuté teoretické informace, na jejichž základě byl projekt vypracován.

2.1 Protokol IP

Protokol IP je protokolem síťové vrstvy. Využívá ho celá rodina protokolů TCP/IP. Sám protokol IP negarantuje doručení paketu a je tedy nespolehlivý. Pro podrobnější informace doporučuji [3].

2.2 Protokol TCP

Protokol TCP je protokolem transportní vrstvy. Vyznačuje se mj. tím, že je spolehlivý; zaručuje, že se data od odesílatele dostanou k příjemci. Při navazování spojení probíhá tzv „three-way handshake“, kdy nejprve klient zasílá TCP paket s příznakem SYN serveru (žádá o spojení), server odpovídá TCP paketem s příznaky SYN a ACK (potvrzuje spojení), RST a ACK (zamítá spojení) nebo neodešle žádnou reakci (filtruje = ignoruje spojení na daném portu). Klient poté zasílá TCP paket s příznakem ACK, kdy potvrzuje ustanovené spojení.
Informace byly získávány z přednášek IPK a následujících zdrojů: [4], [2].

2.3 Protokol UDP

Protokol UDP je stejně jako TCP protokolem transportní vrstvy. Na rozdíl od TCP je ale přenos tímto protokolem nespolehlivý; není garantováno, že informace bude doručena.
Pro podrobnější popis protokolu UDP viz [5].

2.4 Podrobnější popis principu skenování na základě zadání projektu

Při skenování TCP portů se nevyužívá kompletní „three-way handshake“, ale pouze první dvě jeho části, které již stačí k identifikaci stavu portu (při otevřeném portu je zaslán potvrzující paket, při zavřeném odmítací a při filtrovaném portu nepřijde žádná odpověď).
Při skenování UDP portů se využívá jiného principu: zavřený port se detekuje tak, že server (obecně druhá strana komunikace) odešla paket protokolu ICMP typu 3, kódu 3. Pokud ze strany serveru žádná odpověď nepřijde, je znovu odeslán UDP paket na daný port. Pokud ani napodruhé nepřijde žádná reakce, je port prohlášen za otevřený nebo filtrovaný (tyto dva stavy nelze odlišit).

3 Implementace

V následující sekci je popsán průběh implementace a testování projektu.

3.1 Struktura kódu a princip činnosti

Program nejprve zpracuje vstupní argumenty a čísla portů pro UDP a TCP skenování zřetězí do lineárních seznamů. V cyklu dále proběhne skenování TCP portů a nasládně UDP portů (pokud byl pro daný protokol zadán alespoň jeden port).
Pro odchytávání příchozí komunikace slouží knihovna pcap.h (informace čerpány z [1]), konkrétně funkce

`pcap_next`. Zde bylo nutno řešit problém, kdy `pcap_next` donekonečna čeká na odpověď serveru, která ale nikdy přijít nemusí (příkladem budiž otevřený/filtrovaný port při UDP skenování). Tento problém byl vyřešen nastavením alarmu a následným zasláním signálu `SIGALRM`, po jehož vyvolání dojde k vyvolání funkce `pcap_breakloop`, která přeruší vnitřní smyčku uvnitř funkce `pcap_next` a dojde k „odseknutí" programu [8].

3.2 Přejaté části kódu

Pro generování kontrolního součtu byly použité funkce volně dostupné na internetu; konkrétně funkce csum byla přejata z [7], udp_checksum z [6] a checksum2 z [9]. Tímto děkuji všem autorům, kteří tyto skvělé funkce vytvořili.

3.3 Testování a porovnání s programem nmap

Projekt nebyl testován žádným testovacím rámcem, všechno testování probíhalo ručně. Pro vizualizaci přenosu paketů byl použit program Wireshark, pro kontrolu správnosti skenování program nmap.
Při porovnání s programem nmap je ve skenování UDP portů program nmap o něco rychlejší To je pravděpodobně způsobeno jiným principem zachytávání komunikace, kdy můj program čeká jednu vteřinu po každém odeslání UDP paketu a až následně pokračuje v činnosti.
Při TCP skenování odesílá program nmap pakety s nastavenými přepínači v části „TCP Options" v TCP hlavičce, zde popisovaný program toto nedělá, ale funkcionalita je v této oblasti totožná.

3.4 Neimplementované části

Projekt není plně funkční, chybí mu podpora skenování serverů s IP adresou verze 6.

4 Použití

Program byl testován na referenčním stroji a OS Fedora 29 (zde byla nutnost doinstalovat knihovnu pcap.h). Pro přeložení stačí využít přiložený Makefile.
Pogram se spouští s následujícími parametry:

- –i interface: specifikace rozhraní, které bude využito pro skenování. Volitelný parametr, při jeho absenci se využívá první vhodné rozhraní.
- –pu ports: specifikuje UDP porty pro skenování.
- –pt ports: specifikuje TCP porty pro skenování.
- posledním parametrem při spouštění programu je doménové jméno nebo IP adresa. Tento parametr je povinný.

Vždy musí být zadáno doménové jméno nebo IP adresa a alespoň jedne z přepínačů –pt nebo –pu.
Po správném spuštění program oskenuje zadané porty a výsledek skenování vypíše na standardní výstup. V případě chyby se ukončí.

5 Závěr

Byl provedeno teoretické nastudování problematiky vytváření, odesílání a zachytávání paketů při TCP a UDP skenování. Po skončení této fáze byl naimplementován skener TCP a UDP portů, který částečně odpovídá zadání. Podpora IPv6 bohužel není implementována.

Použité zdroje

[1] Carstens, T.: Programming with pcap. [online]. 2002 [cit. 2019-04-18].
URL `https://www.tcpdump.org/pcap.html`

[2] Michael J. Donahoo, K. C.: *TCP/IP sockets in C: Practical Guide for Programmers*. San Francisco: Morgan Kaufmann Publishers, 2001, ISBN 978-1-55860-826-9.

[3] Postel, J.: Internet Protocol. [online]. 1981 [cit. 2019-04-15].
URL `https://tools.ietf.org/html/rfc791`

[4] Postel, J.: Transmission Control Protocol. [online]. 1981 [cit. 2019-04-15].
URL `https://tools.ietf.org/html/rfc793`

[5] Postel, J.: User Datagram Protocol. [online]. 1980 [cit. 2019-04-15].
URL `https://tools.ietf.org/html/rfc768`

[6] Righi, A.: udp.c. [online]. 2003 [cit. 2019-04-19].
URL `https://bit.ly/2Dotrq8`

[7] Unknown: LINUX SOCKET PART 17: Advanced TCP/IP - THE RAW SOCKET PROGRAM EXAMPLES. [online]. [cit. 2019-04-19].
URL `https://www.tenouk.com/Module43a.html`

[8] Unknown: Manpage of PCAP BREAKLOOP. [online]. 2018 [cit. 2019-04-16].
URL `https://bit.ly/2UvJ7gP`

[9] Unknown: The TCP/IP Checksum. [online]. [cit. 2019-04-17].
URL `https://bit.ly/2Uyzs9z`