IPK 2019

Projekt č. 2

Scanner síťových služeb

Martin Litwora

Datum: 21.4.2019

1. Úvod

Cílem projektu bylo vytvořit program, který oskenuje sítové porty zadané uživatelem, včetně protokolu a vypíše, zda je port otevřený, uzavřený nebo filtrovaný. Uživatel zadá cílovou IP adresu, nebo doménové jméno skenovaného zařízení. Rovněž může zadat odchozí interface ze kterého se budou posílat pakety. Pakety musí být odeslány pomocí takzvaných raw socketů, které můžou obejít standartní síťová zapouzdření TCP/IP implementovaného typicky uvnitř operačního systému.

2. Základní informace o fungovaní scanneru

TCP skenovaní

Posílají se pakety, které mají v TCP hlavičce nastaven příznak SYN. Pokud ze skenovaného zařízení přijde odpověď s nastaveným příznakem SYN a ACK je port označen jako otevřený. Dorazí-li paket s příznakem ACK a RST je port označen jako uzavřený. Nedojde-li žádná odpověď, paket se zašle znovu a poté se při nedostání odpovědi může označit jako filtrovaný. Neprobíhá klasický 3-way handshake mezi komunikujícími zařízeními.

Při samotné implementaci byla nejobtížnější pasáž, kdy se musí vypočítat správně kontrolní součet pro TCP hlavičku. Nejdůležitější bylo nutné si vytvořit pseudohlavičku, která musí obsahovat mimo jiné také cílovou a zdrojovou IP adresu, která však u TCP hlavičky chybí. Z této pseudohlavičky a samotné TCP hlavičky je poté teprve možné vypočítat správný kontrolní součet.
Pro odchytávaní paketů přijatých z cílové stanice bylo nutné použít funkce z knihovny libcap. Konkrétně funkce *pcap_loop()* s nástavným filtrem. Možné bylo též použít funkce *pcap_next(), pcap_dispatch()*. Aplikace (zdrojový počítač) poslouchá na portu 43345. Pro časový interval, jak dlouho se má na paket čekat (v mém případě 2s) bylo nutné nastavit alarm, který vyvolá signál SIGALARM, jež po obsloužení zavolá funkci *pcap_breakloop()*. Odchycený paket je pak nutné namapovat na odpovídající IP, TCP hlavičky. Teprve poté je možné získat informace o nastavených příznacích.

UDP skenování
UDP je benevolentnější k vyplnění hlavičky. Ačkoliv kontrolní součet není nutné počítat, tak při špatné hodnotě může cílová stanice paket tiše zahodit. Proto počítám kontrolní součet pro UDP obdobně jako pro TCP. Při skenování portů UDP paketem však nejsme schopni zjistit rozdíl mezi filtrovaným a otevřeným portem. Nazpět totiž žádná odpověď nepřijde. Pouze pokud je port uzavřen cílová stanice generuje ICMP paket s kódem 3 (port unreachable). Tehdy je možné port označit jako uzavřený. Proto abychom označili port jako otevřený, je nutné UDP paket na daný port poslat vícekrát, jelikož nemáme jistotu toho, že UDP nebo ICMP paket dorazil bez problému. Aplikace naslouchá na portu 43346 pro UDP skenování.

3. Poznámky k implementaci

- Program je napsán v jazyce C++, pro překlad je přiložen Makefile (příkaz make)
- Příklad spuštění:
  ./ipk-scan {-i <interface>} -pu <port-ranges> -pt <port-ranges> [<domain-name> | <IP-address>]
  ./ipk-scan -i wlp2s0 -pu 19-21 -pt 1-30 wis.fit.vutbr.cz
- Aplikace nedokáže pracovat s IPv6 adresou
- Při úspěšném proběhnutí aplikace vrací hodnotu 0. V opačném případě hodnoty 1-5, detailněji popsané v úvodu zdrojového kódu

4. Bibliografie

www.tenouk.com, 2003 [online] [cit. 21.4.2019]. Dostupné z: https://www.tenouk.com/Module43a.html
www.cplusplus.com, 6.5.2013 [online] [cit. 21.4.2019]. Dostupné z: http://www.cplusplus.com/articles/DEN36Up4/
Wikipedia, 23.10.2018 [online] [cit. 21.4.2019]. Dostupné z: https://cs.wikipedia.org/wiki/IPv4
<devdungeon>, 14.8.2015 [online] [cit. 21.4.2019]. Dostupné z: https://www.devdungeon.com/content/using-libpcap-c
Stackoverflow, 6.1.2015 [online] [cit. 21.4.2019]. Dostupné z: https://stackoverflow.com/questions/14088274/raw-socket-linux-send-receive-a-packet
TcpDump, 2010 [online] [cit. 21.4.2019]. Dostupné z: https://www.tcpdump.org/pcap.html
BinaryTides, 17.3.2012 [online] [cit. 21.4.2019]. Dostupné z: https://www.binarytides.com/c-program-to-get-ip-address-from-interface-name-on-linux/