IPK - Počítačové komunikace a sítě

24. března 2022
22:05:33

Projekt 2 » zpět na seznam variant

Zpět na termíny

Varianta termínu - Varianta ZETA: Sniffer paketů - přihlášen

» **Odevzdané soubory**

Popis:

ZADÁNÍ:
1) Navrhněte a implementujte síťový analyzátor v C/C++/C#, který bude schopný na určitém síťovém rozhraním zachytávat a filtrovat pakety (13 b)
2) Vytvořte relevantní manuál/dokumentaci k projektu (7b)

UPŘESNĚNÍ ZADÁNÍ:
Ad 1)
Volání programu:

./ipk-sniffer [-i *rozhraní* | --interface *rozhraní*] {-p *port*} {[--tcp|-t] [--udp|-u] [--arp] [--icmp] } {-n *num*}

kde

- -i *eth0* (právě jedno rozhraní, na kterém se bude poslouchat. Nebude-li tento parametr uveden, či bude-li uvedené jen -i bez hodnoty, vypíše se seznam aktivních rozhraní)
- -p *23* (bude filtrování paketů na daném rozhraní podle portu; nebude-li tento parametr uveden, uvažují se všechny porty; pokud je parametr uveden, může se daný port vyskytnout jak v source, tak v destination části)
- -t nebo --tcp (bude zobrazovat pouze TCP pakety)
- -u nebo --udp (bude zobrazovat pouze UDP pakety)
- --icmp (bude zobrazovat pouze ICMPv4 a ICMPv6 pakety)
- --arp (bude zobrazovat pouze ARP rámce)
- Pokud nebudou konkrétní protokoly specifikovány, uvažují se k tisknutí všechny (tj. veškerý obsah, nehledě na protokol)
- -n *10* (určuje počet paketů, které se mají zobrazit, tj. i "dobu" běhu programu; pokud není uvedeno, uvažujte zobrazení pouze jednoho paketu, tedy jakoby -n 1)
- argumenty mohou být v libovolném pořadí

Formát výstupu (odchylky jako bílé znaky a další zajímavé informace z rámců/paketů/datagramů/segmentů jsou připouštěny):

timestamp: *čas*
src MAC: *MAC adresa s : jako oddělovačem*
dst MAC: *MAC adresa s : jako oddělovačem*
frame length: *délka*
src IP: *pokud je tak IP adresa (podpora v4 ale i v6 dle RFC5952)*
dst IP: *pokud je tak IP adresa (podpora v4 ale i v6 dle RFC5952)*
src port: *pokud je tak portové číslo*
dst port: *pokud je tak portové číslo*

offset_vypsaných_bajtů: *výpis_bajtů_hexa výpis_bajtů_ASCII*

přičemž:

- *čas* je ve formátu dle RFC3339
- *délka* je v bytech

(takto vypíšete úplně celý paket)

Příklady volání:

./ipk-sniffer -i eth0 -p 23 --tcp -n 2
./ipk-sniffer -i eth0 --udp
./ipk-sniffer -i eth0 -n 10
./ipk-sniffer -i eth0 -p 22 --tcp --udp --icmp --arp .... stejné jako:
./ipk-sniffer -i eth0 -p 22
./ipk-sniffer -i eth0

Příklady výstupu:

./ipk-sniffer
./ipk-sniffer -i

lo0
eth0

./ipk-sniffer -i eth0

timestamp: 2021-03-19T18:42:52.362+01:00
src MAC: 00:1c:2e:92:03:80
dst MAC: 00:1b:3f:56:8a:00
frame length: 512 bytes
src IP: 147.229.13.223
dst IP: 10.10.10.56
src port: 4093
dst port: 80

0x0000: 00 19 d1 f7 be e5 00 04 96 1d 34 20 08 00 45 00 ....4 ..
0x0010: 05 a0 52 5b 40 00 36 06 5b db d9 43 16 8c 93 e5 ..R[@.6. [..C....
0x0020: 0d 6d 00 50 0d fb 3d cd 0a ed 41 d1 a4 ff 50 18 .m.P..=. ..A...P.
0x0030: 19 20 c7 cd 00 00 99 17 f1 60 7a bc 1f 97 2e b7 . ...`z.....
0x0040: a1 18 f4 0b 5a ff 5f ac 07 71 a8 ac 54 67 3b 39 ....Z._. .q..Tg;9
0x0050: 4e 31 c5 5c 5f b5 37 ed bd 66 ee ea b1 2b 0c 26 N1._.7. .f...+.&
0x0060: 98 9d b8 c8 00 80 0c 57 61 87 b0 cd 08 80 00 a1 .....W a.....

Netisknutelné znaky budou nahrazeny tečkou, vypisovat můžete i případný padding.

Ad 2)
V dobré dokumentaci se OČEKÁVÁ následující: titulní strana, obsah, logické strukturování textu, výcuc relevantních informací z nastudované literatury, popis zajímavějších pasáží implementace, sekce o testování (ve které kromě vlastního programu otestujete nějaký obecně známý open-source nástroj), bibliografie, popisy k řešení bonusových zadání.

DOPORUČENÍ:

- Při implementaci použijte knihoven pcap / libnet
  Pcap: http://www.tcpdump.org/pcap3_man.html
  Libnet: http://www.packetfactory.net/projects/libnet/
- U syntaxe vstupních voleb jednotlivým programům složené závorky {} znamenají, že volba je nepovinná (pokud není přítomna, tak se použíje implicitní hodnota), oproti tomu [] znamená povinnou volbu. Přičemž pořadí jednotlivých voleb a jejich parametrů může být libovolné. Pro jejich snadné parsování můžete použít např. funkci getopt().
- Výsledky vaší implementace by měly být co možná nejvíce multiplatformní mezi OS založenými na unixu, ovšem samotné přeložení projektu a funkčnost vaší aplikace budou testovány na referenčním Linux image (viz obecné pokyny k zadání) pro síťové předměty (přihlašovací údaje student / student).
- Očekává se použití promiskuitního módu síťové karty.
- Program by se měl dát v kterémkoli okamžiku korektně ukončit pomocí Ctrl+C.
- K testování IPv6 funkcionality lze využít např. síťový stack na vývojářské virtuálce.
- Podpora IPv4 Options, IPv6 Extension Headers či jiných link-type než LINKTYPE_ETHERNET je vítaná, nikoli však nutná.

ODEVZDÁNÍ:
Součástí projektu budou zdrojové soubory přeložitelné na referenčním operačním systému, funkční Makefile (či pomocné provozy C#), soubor manual.pdf a README (viz obecné pokyny). Projekt odevzdejte jako jeden soubor xlogin00.tar, který vytvoříte programem tar.

LITERATURA:

- Wikipedia, the free encyclopedia: http://en.wikipedia.org/wiki/Pcap
- TCPDUMP/LIBPCAP public repository: http://www.tcpdump.org/
- Odkazy na knihovnu http://packetfactory.openwall.net/projects/libnet/
- RFC 792 - Internet Control Message Protocol a RFC 4443 - ICMPv6
- RFC 826 - ARP
- RFC 5952 - A Recommendation for IPv6 Address Text Representation

PATCH-NOTES:

- bude doplněno na základě případné diskuze na fóru