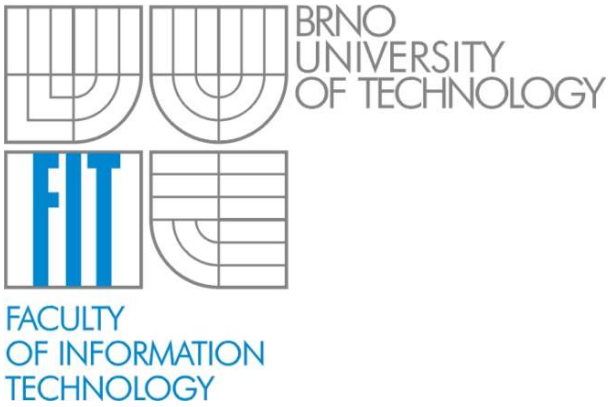

Dokumentácia k projektu pre predmet ISA

Monitorovanie hlavičiek HTTP

Programovanie sieťovej služby

25.11.2014

Autor: Ladislav Šulák, xsulak04@stud.fit.vutbr.cz, 3BIT

Fakulta informačných technológií

Vysoké Učení Technické v Brne

Obsah

Úvod

Úlohou bolo vytvoriť program, ktorý bude monitorovať *HTTP* hlavičky na zadanom sieťovom rozhraní alebo zo vstupného súboru formátu *.pcap* ktorý obsahuje zachytenú komunikáciu.

Vďaka monitorovaniu *HTTP* hlavičiek ktoré sú odoslané zo strany klienta je možné zistiť aké stránky klient navštívil, aký ma prehliadač, operačný systém, IP adresu a mnoho ďalších informácií. Táto dokumentácia popisuje návrh, implementáciu a použitie výslednej aplikácie.

Ako implementačný nástroj bol zvolený jazyk *C++,* s použitím knižníc pre spracovanie a zachytenie paketov *libpcap*, a pre prácu s formátom *XML libxml2*.

Návrh programu

Návrh aplikácie nevyužíva prístup objektovo-orientovaného programovania, no využíva niektorých objektov zo štandardných knižníc jazyka C++, hlavne pre prácu s reťazcami.

Pre filtrovanie paketov sa používa funkcia pre iniciovanie filtru ***pcap_compile(),*** ktorá má v argumente zadaný výraz, ktorý sa aplikuje, a pakety sú filtrované ešte pred ich samotným spracúvaním. Následne sa pakety po jednom spracúvajú a priebežne zapisujú do súboru.

Implementácia

Spracovanie argumentov

Spracovanie argumentov vykonáva funkcia **arg_parsing(),** ktorá tiež ukladá hodnoty argumentov ktoré sa budú ďalej využívať do globálnej štruktúry **p_struct parsed_struct**. Ak dôjde k chybe, tak sa zavolá funkcia **help()** ktorá vypíše správny formát očakávaných vstupných parametrov.

Príprava pre odchytávanie paketov

Po spracovaní argumentov sa stanoví výraz, ktorý bude použitý pre filtrovanie paketov. Samotný filter sa spúšťa funkciou **pcap_setfilter().** Vďaka tomuto výrazu je možné nastaviť filter tak, aby sa spracovali pakety iba na *TCP* spojení a pre určitý port .

Pre monitorovanie paketov na určitom rozhraní je stanovený smer ktorý v tomto prípade určuje, že má spracovať pakety smerujúce od klienta. To zaisťuje funkcia **pcap_setdirection().**

Následne sa vytvorí výstupný súbor typu XML a zapíšu sa počiatočné elementy pomocou funkcie **xmlStart(),** ktorá obsahuje funkcie z knižnice *libxml2* **xmlNewTextWriterFilename()**,**xmlTextWriterStartDocument** a **xmlTextWriterStartElement().**

Kvôli prehľadnosti, testovaniu a čitateľnosti XML výstupného súboru je indentácia explicitne nastavená. Ak sa bude s XML súborom ďalej pracovať – spracovanie nejakým nástrojom pre porovnávanie alebo ukladanie XML súborov, je nutné toto brať do úvahy.

Pracovanie s filtrom *pcap* bolo inšpirované príkladmi v dokumentácií k funkcii **pcap_filter().**

Spracovanie paketov

Pre každý prijatý paket sa využíva funkcia **pcap_loop(),** ktorá pri monitorovaní na určitom rozhraní končí iba ak je program ukončený signálmi *SIGINT*, *SIGQUIT* alebo *SIGTERM*.

Táto funkcia obsahuje ako parameter callback funkciu **handle_packet(),** ktorá zisťuje zdrojovú *IP* adresu a *port* zariadenia, ktoré následne zapíše do *XML* súboru.

Funkcia ďalej prepočítava kde sa dáta *(payload)* v pakete nachádzajú, ich veľkosť, a zavolá funkciu **payload_parse(),** ktorá v prvom kroku vyjme prvý riadok, ten spracuje funkcia **payload_continue()**, ktorá rozpozná, či na začiatku (v podstate prvý riadok dát, ukončený *CR*, *NL*) je prítomná jedna z možných *HTTP* požiadaviek (*OPTIONS*, *GET*, *HEAD*, *PUT*, *POST*, *DELETE*, *TRACE* alebo *CONNECT*). Ak nie, paket sa preskočí, v opačnom prípade sa pokračuje vo funkcié **payload_parse(),** ktorá rozdelí jednotlivé *HTTP* hlavičky a pre každú zavolá funkciu **header_parse()**. Samotné rozdelenie hlavičiek je realizované vďaka tomu, že medzi hlavičkami sa nachádzajú znaky *CARRIAGE RETURN* (ASCII 13) a *NEWLINE* (ASCII 10).

Funkcia **header_parse()** rozdelí hlavičky na názvy a ich hodnoty, čo je realizované tým, že názov hlavičky a jej hodnota sú oddelené znakom ‘:’ nasledovaným 0-n mezerami, no v praxi sa prevažne vyskytuje 1 medzera. Následne sa hlavičky zapíšu do súboru. To vykonáva funkcia **xmlAdd2()**.

Spracovanie paketu bolo inšpirované ukážkovými príkladmi z projektu pre analýzu paketov *tcdump* a aj ukážkovým programom *sniffex* pre odchytávanie paketov. Ako aj príkladmi v dokumentácii k tomuto nástroju.

Ďalšie informácie

Program spracuje aj také spojenia, ktoré sú dotazmi klienta, no majú prázdnu hlavičku. V tomto prípade sa zapíše prázdny element s názvom *connection* nasledovaný číslom portu zdrojového rozhrania.

UDP pakety sa nespracovávajú (sú hneď odfiltrované), program pracuje s *IP* verziami *IPv4*.

IPv6 komunikácia bola úspešne implementovaná, no testovaná bola iba na jednoduchých príkladoch (napríklad príklady komunikácie z *Wiresharku* - v6-http.cap). Okrem .pcap súboru, teda na sieťovom rozhraní k testovaniu nedošlo.

Program končí s hodnotou 0 ak nedošlo k žiadnej chybe, alebo 1 ak chyba nastala, s príslušnou chybovou hláškou.

Pri zachytení signálu pre ukončenie cyklu ktorý prijíma pakety na danom rozhraní sa zavolá funkcia **sig_handler(),** ktorá vypíše všetky koncové tagy pre neukončené elementy a ukončí *XML* súbor a uvoľní všetky alokované zdroje.

Rovnaká činnosť pre ukončenie programu prebieha, ak je program spustený tak, aby sa spracovávala komunikácia zo vstupného súboru.

Použitie programu

Preklad:

Pomocou *Makefile*, príkaz *make* preloží program nasledovne:

g++ -Wall -Wextra -pedantic httphdrs.cpp -o httphdrs -lpcap -lxml2 -I/usr/include/libxml2

Spustenie:

$ ./httphdrs {-f | -i } source [-H header1,header2,headern] [-p 80,8080] -o output --help

Argumenty:

Povinný argument *-f | -i* - práve 1 musí byt zadaný, za nim musí nasledovať: *source*

Povinný argument *source* - odchytáva pakety na danom zariadení(-i) alebo zo súboru (-f)

Voliteľný argument *-H [...]* - argument -H určuje, že sa program má spustiť s inými než implicitnými hodnotami *HTTP* hlavičiek, ktoré sú*: User-Agent,Accept,Accept-Encoding,Accept-Language*.

Zoznam hlavičiek nasleduje hneď za parametrom -H, musí byt bez medzier a oddelený čiarkami.

Voliteľný argument *-p [...]* - argument -p určuje, že sa program ma spustiť s inou než implicitnou hodnotou portu 80. Číslo (celočíselný rozsah validnych portov je 0-65535) portu nasleduje hneď za parametrom -p, musí byt bez medzier a oddelený čiarkami.

Povinný argument *-o* -špecifikuje, že nasledujúci argument bude udávať názov výstupného súboru

Voliteľný argument *--help* - zobrazí nápovedu k programu

Záver

Program je spustiteľný na operačných systémoch *Linux*, bol vyvíjaný a testovaný na operačnom systéme *ISA2014*, postavenom na operačnom systéme *Ubuntu 14.04.*

Program nie je kompatibilný s *OS Windows*, no ku knižniciam *libpcap* existuje alternatíva, a to knižnice *WinDump*. Príklad nástroja podobného *tcpdumpu*, teda nástroja pre analýzu paketov je *WinPcap*.

Metriky

Počet zdrojových súborov:	1 súbor
Počet riadkov zdrojového kódu:	860 riadkov
Veľkosť statických dát:	28653B
Veľkosť spustiteľného súboru:	34360B

Literatúra

Pri tvorbe programu boli použité manuálové stránky, nižšie uvedené RFC, dokumentácia niektorých knižníc C++, dokumentácia knižnice libpcap a libxml2 aj s príkladmi.

Štandardy RFC pre HTTP/1.1

RFC7230: Semantic and Content http://tools.ietf.org/html/rfc7230

RFC7231: Message Syntax and Routing http://tools.ietf.org/html/rfc7231

Knižnice pre prácu s paketmi a formátom XML

libpcap http://www.tcpdump.org/

Jednoduhý analyzátor *sniffex* http://www.tcpdump.org/sniffex.c

Dokumentácia *pcap-filter* http://www.tcpdump.org/manpages/pcap-filter.7.html

libxml2	http://www.xmlsoft.org/
Príklad práce s knižnicou *libxml2*	http://www.xmlsoft.org/examples/testWriter.c
Indentácia v *XML*	http://www.w3.org/TR/2008/REC-xml-20081126/#sec-white-space

Nástroj, ktorý zisťuje vlastnosti webového prehliadača

Panopclick	https://panopticlick.eff.org