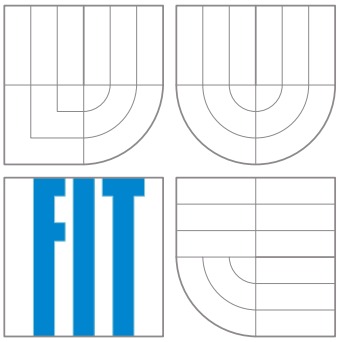
Fakulta informačních technologií
Vysoké učení technické v Brně



Dokumentace do předmětu ISA k projektu

Monitorování HTTP hlaviček

30. listopadu 2014

Radek Pištělák xpiste04@stud.fit.vutbr.cz

Obsah

1 Úvod

Práce se zabývá problematikou zachytávání HTTP hlaviček v síťovém provozu nebo jeho záznamu za pomoci jazyka `Python` a knihovny `scapy`. Nejsou zde tedy uvedeny žádné informace typu: *"Na prvních 4 bitech IP paketu naleznete verzi IP protokolu atd."* ..., které v případě využití uvedených nástrojů nejsou nutné.

Dále v úvodní kapitole je uvedeno zadání problému, který se práce snaží řešit a menší modifikace zadání. Návrh programu spolu s přehledem nastudovaných informací se nachází v kapitole 2. Třetí kapitola se pak zabývá samotnou implementací řešení (3). Shrnutí práce je uvedeno v kapitole 4.

1.1 Zadání

Napište program *httphdrs*, který bude monitorovat hlavičky HTTP. Program bude umět monitorovat jak provoz na zadaném síťovém rozhraní (specifikovaném jeho jménem), tak procházet uložený provoz ve formátu pcap. Program při spuštění získá seznam hlaviček, které bude vyhledávat v dotazech klienta (HTTP Request). Nalezené hlavičky bude program ukládat do XML souboru, jehož formát je specifikován níže. Program bude ukládat pouze hlavičky zasílané klientem.

Při vytváření programu je povoleno použít hlavičkové soubory pro práci se schránkami a další obvyklé funkce používané v síťovém prostředí (jako je `netinet/*`, `sys/*`, `arpa/*` apod.), knihovnu pro práci s vlákny (`pthread`), signály, časem, stejně jako standardní knihovnu jazyka `C`, `C++` a `STL`. Z knihoven třetích stran je možné použít knihovnu pro práci s XML `libxml2` (`http://www.xmlsoft.org/`) a pro práci se síťovými provozem libpcap (`http://www.tcpdump.org/`). Jiné knihovny jazyka `C/C++` nejsou povoleny.

Program je možné implementovat v jazyce `python` včetně The Python Standard Library. Je povoleno využívat knihovny `scapy` (`http://www.secdev.org/projects/scapy/`) a `IPy` (`https://pypi.python.org/pypi/IPy/`).

1.1.1 Změny zadání

Práce rozšiřuje zadání o možnost vyhledávat hlavičky i v odpovědích od serveru.

2 Návrh programu

Pro správný návrh programu bylo nejprve nutné nastudovat formát HTTP hlaviček. K tomu dobře posloužily vybrané kapitoly z (R. Fielding – J. Reschke, 2014a) a (R. Fielding – J. Reschke, 2014b) (`RFC7230` a `RFC7231`). Následně se bylo potřeba seznámit s knihovnou `scapy` pro jazyk `python`, ve kterém je práce implementována. Pro knihovnu `scapy` existuje oficiální dokumentace (Biondi), která není příliš obsáhlá, proto bylo potřeba hledat informace jinde. Požadované znalosti doplnily www stránky `http://thepacketgeek.com` zejména seriály (Mat, 2013a) a (Mat, 2013b).

2.1 Scapy a pakety

```
###[ Ethernet ]###
  dst= 98:fc:11:e5:90:39
  src= 10:40:f3:91:8f:d4
  type= 0x800
###[ IP ]###
     version= 4L
     ihl= 5L
     tos= 0x0
     len= 238
     id= 24592
     flags= DF
     frag= 0L
     ttl= 64
     proto= tcp
     hksum= 0x8279
     src= 192.168.1.138
     dst= 77.75.72.3
     \options\
###[ TCP ]###
        sport= 54126
        dport= http
        seq= 2724963881
        ack= 681313927
        dataofs= 8L
        reserved= 0L
        flags= PA
        window= 4140
        chksum= 0x9f68
        urgptr= 0
        options= [('NOP', None), ('NOP', None), ('Timestamp', (692919670,
            383190542))]
###[ Raw ]##\#
          load= 'GET / HTTP/1.0\r\nUser-Agent: w3m/0.5.3\r\nAccept: text/html,
              text/*;q=0.5, image/*\r\nAccept-Encoding: gzip, compress, bzip, bzip2
              , deflate\r\nAccept-Language: en;q=1.0\r\nHost: www.seznam.cz\r\n\r\n
              '
```

Ukázka vnitřní struktury paketu v knihovně `Scapy`

2.2 Návrh

Tato práce řeší filtrování paketů na základě jejich vlastního obsahu (vrsta `Raw` v úkázce paketu), kde podle čtvrté kapitoly `RFC7231` (R. Fielding – J. Reschke, 2014b) "Request Methods" (případně

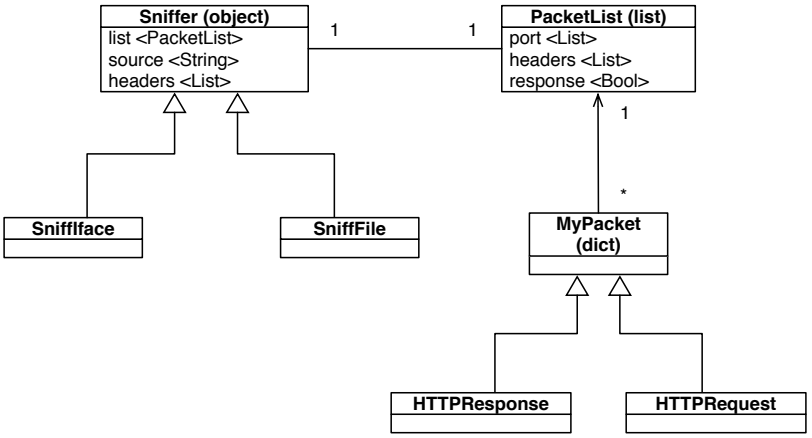
RFC2616 kapitola 5 (Fielding et al., 1999)) víme, jak je definována sémantika HTTP hlavičky (na straně požadavku) a poté již není obtížné sestavit konečný automat nebo regulární výraz, který v případě, že tento text přijme prohlasí paket za dotaz (= `HTTP Request`).

U paketu s `HTTP Response` je situace obdobná liší se pouze v samotném regulárním výrazu.

Informace typu zdrojové/cílové IP adresy a zdrojového/cílového portu jsou již za pomoci knihovny `scapy` lehce dosažitelné a to přes IP respektive TCP vrstu.

Samotnému získání hlaviček z `Raw` vrstvy napomáhá fakt, že každá hlavička je na samostatném řádků ve formátu *hlavička: obsah hlavičky.*

2.3 Návrh tříd



Obrázek 1: Návrh tříd

3 Implementace

3.1 `HTTPpacket.py`

V uvedeném modulu je implementován seznam `PacketList`, u kterého lze pomocí metody `append` vyvolat pokus o přidání paketu (typ `Packet` definovaný knihovnou `scapy`) do tohoto seznamu. Volání této metody spustí proces ověření, zda se v paketu nachází HTTP požadavek nebo odpověď (definováno parametrem viz. 3.3). V případě kladného výsledku je do seznamu vložen objekt typu `HTTPRequest` nebo `HTTPResponse` oddděděný od třídy `MyPacket`, která má za předka slovník.

3.2 `utils.py`

Hlavním obsahem souboru je obecná třída `Sniffer`, která implementuje uložení paketů ze seznamu `PacketList` do výsledného XML souboru. Třída `Sniffer` je pak předkem třídám `SniffFile` a `SniffIface`, ve kterých jsou implementovány metody pro získání vstupních dat. V případě `SniffFile` se budou data načítat ze souboru formátu `.pcap` a ve druhém případě budou data odposlechnuty ze zadaného rozhraní (viz. 3.3).

Výsledný XML soubor je generován vždy i v případě, že nedojde k žádné shodě mezi získanými hlavičkami ze vstupního zdroje a hlavičkami specifikovanými parametrem případně výchozím seznamem. Minimální kostra je specifikována zadáním. Jelikož není jasné, zda zadavatel nezamýšlí využívat výstupní data programu jako vstupní data jiného programu, tak výsledný soubor neobsahuje XML hlavičku (chybí v příkladech výstupu aplikace i její specifikaci).

3.3 `httphdrs.py`

V hlavním modulu programu jsou zpracovány parametry a za zmínku stojí funkce `get_input()`, která dle zadaných parametrů vrátí objekt typu `SniffFile` nebo `SniffIface`, nicméně oba objekty mají stejného předka a i rozhraní, lze s nimi tedy pracovat pomocí stejných metod (viz. funkce `main`).

3.4 Použití

```
ISA xpiste04$ python2.7 httphdrs.py -h
usage: httphdrs.py [-h] (-f FILE | -i IFACE) [-H HEADERS] [-p PORT] -o OUTPUT
                   [--extra]

arguments:
  -h, --help  show this help message and exit
  -f FILE     Pouzije jako vstup soubor ve formatu pcap.
  -i IFACE    Pouzije jako vstup rozhrani iface.
  -H HEADERS  Slouzi ke specifikaci sledovanych HTTP hlavicek. Case insensitive.
  -p PORT     Slouzi ke specifikaci portu (muze jich byt vice).
  -o OUTPUT   Povinny parametr specifikuje nazev vyst. souboru.
  --extra     Rozsireni, ktere nebude ukladat HTTPRequest, ale HTTPResponse
```

Při použití parametru `--extra` je funkcionalita ostatních parametrů zachována. Výchozí seznam hlaviček je pak změněn na `"Host,Location,Date,Status-Line,Server"`.

Ukončení běhu programu – v případně načítání hlaviček přes rozhraní stisk kláves `CTRL+C` ukončí odposlouchávání rozhraní a provede zápis dat do XML souboru. V jiných případech, ale tato kombinace kláves program okamžitě ukončí! Přerušení běhu programu není nijak ošetřeno záměrně, protože program není cílen na běžné uživatele, ale na pokročilé uživatele, kteří pokud danou klávesovou kombinaci volí, tak k tomu mají důvod.

3.5 Použité knihovny

- `Scapy` pro získání vstupních dat.
- `Argparse` pro práci zpracování parametrů.
- `Xml` pro výstup do XML souboru.
- `Os` pro ověření možnosti práce se soubory.

3.6 Metriky kódu

- Počet řádků celkem: 341
- Počet řádků komentářů: 58
- Počet řádků zdojového kódu: 191

4 Závěr

Výsledná aplikace by měla splňovat zadání v plném rozsahu. Aplikace by mohla po rozšíření sloužit například pro jednoduchou firemní analýzu internetového provozu (hlavička `Host` v požadavcích klienta).

Literatura

BIONDI, P. *Scapy documentation* [online]. [cit. 28. 11. 2014]. Dostupné z: `http://www.secdev.org/projects/scapy/files/scapydoc.pdf`.

FIELDING, R. et al. *RFC 2616: Hypertext Transfer Protocol – HTTP/1.1* [online]. 1999.

MAT. *Looking at Packets* [online]. 2013a. [cit. 28. 11. 2014]. Dostupné z: `http://thepacketgeek.com/scapy-p-04-looking-at-packets/`.

MAT. *Scapy Sniffing with Custom Actions* [online]. 2013b. [cit. 28. 11. 2014]. Dostupné z: `http://thepacketgeek.com/series/scapy-sniffing-with-custom-actions/`.

R. FIELDING, E. – J. RESCHKE, E. *RFC 7230: Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing* [online]. 2014a. [cit. 30. 11. 2014]. Dostupné z: `http://www.rfc-editor.org/rfc/rfc7230.txt`.

R. FIELDING, E. – J. RESCHKE, E. *RFC 7231: Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content* [online]. 2014b. [cit. 30. 11. 2014]. Dostupné z: `http://www.rfc-editor.org/rfc/rfc7231.txt`.