VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ
DEPARTMENT OF INFORMATION SYSTEMS

MONITOROVÁNÍ DNS KOMUNIKACE

MONITORING OF DNS COMMUNICATION

PROJEKT DO PŘEDMĚTU ISA
PROJECT FOR THE SUBJECT ISA

AUTOR PRÁCE **MICHAL BLAŽEK**
AUTHOR

VEDOUCÍ PRÁCE **Ing. RADEK HRANICKÝ, Ph.D.**
SUPERVISOR

BRNO 2024

Obsah

Kapitola 1

Úvod

1.1 Uvedení do problematiky

Hlavním cílem programu je zachytávání a výpis DNS zpráv. DNS zprávy slouží pro překlad doménových jmen, která jsou pro člověka lehce zapamatovatelná, na IP adresy, se kterými zase lépe pracují počítače.

Při monitorování této komunikace můžeme nalézt záznamy, které bychom neočekávali nebo obráceně očekáváme záznam, který třeba ani nepřijde. Díky tomu lze odhalit škodlivou aktivitu, nebo třeba problémy s připojením.

1.2 Návrh aplikace

Původní návrh aplikace odpovídá vlastně funkci `main()`, a to je zpracování vstupních parametrů programu a následné naslouchání a zachytávání DNS záznamů v protokolu UDP.

`DnsMonitor` zpracuje parametry programu a předá řízení `DnsCapturer`, který následně na rozhraní začne naslouchat nebo zpracuje soubor formátu PCAP. `DnsCapturer` vytvoří pro každý záznam instanci třídy `Packet` a ta si pomocí DnsHeader pomůže pro zpracování informací z DNS hlaviček. Instance třídy `Packet` pak mají možnost vypsat data o daném zachyceném záznamu.

1.3 Popis implementace

Jelikož návrh aplikace byl objektově orientovaný, bylo vhodné zvolit i objektově orientovaný jazyk, tudíž je program napsaný v jazyce C++.

Počáteční funkce programu `main()` vytvoří instanci třídy `DnsMonitor`, která zpracuje vstupní parametry a vytvoří instanci třídy `PacketCapturer`. `PacketCapturer` otevře rozhraní nebo PCAP soubor pro naslouchání, nastaví filtr pouze na DNS záznamy na portu 53 a pouze na protokol UDP. Tyto záznamy poté zachytává a každý paket ukládá do nové instance třídy `Packet`.

Třída `Packet` se stará o zpracování dat do čitelné struktury, kde si uloží zdrojovou a cílovou IP adresu a port. Při vypsání informací z paketů si instance třídy `Packet` pomůže pomocí `DnsHeader`. `DnsHeader` zpracovává informace v DNS záznamů a případně jeho doplňkových záznamech v sekcích Questions, Answer, Authority a Additional, které jsou blíže popsané v [1].

Největších oříškem je zpracování doménových jmen v sekcích Answer, Authority a Additional, protože z důvodu ušetření velikosti paketů se zde může místo celého doménového jména nacházet jen část a byte mající první 2 bity v 1 (0xC0 až 0xFF). Lze nalézt v [3]. Toto značí skok na jiný byte v DNS záznamu a těchto skoků může být několik před přečtením celého doménového jména. Řešeno je to rekurzivně voláním metody pro dekódování doménového jména, kde se pak jednotlivé části pospojují.

1.3.1 Popis výstupních souborů

Výstupní soubory doménových jmen a překladů doménových jmen na IP adresy nesmí mít duplicitní záznamy. Toto je vyřešeno pomocí množiny (std::set) ze standardní knihovny šablon. Jelikož množina může mít pouze unikátní hodnoty, je každý záznam v ní uchován pouze jednou jako píší zde [2]. Toho se využívá tak, že se nejdříve načtou veškeré záznamy z výstupních souborů do těchto množin, poté se přidají záznamy z aktuálně zpracovávaného paketu a výsledná množina unikátních záznamů se uloží zpět do výstupních souborů. Díky tomuto také docílíme alfabetického uspořádání doménových jmen a překladů doménových jmen.

Kapitola 2

Program

2.1 Základní informace o programu

Program zachycuje veškerou DNS komunikaci na daném rozhraní přes protokol UDP, nebo může program zpracovat tuto komunikaci v souboru ve formátu PCAP.

Program vypisuje informace z DNS zpráv buď v zjednodušeném, nebo v kompletním výpisu na standardní výstup. V případě potřeby může vypisovat nalezená doménová jména do zvoleného souboru a stejným způsobem může vypisovat překlady doménových jmen na IP adresy do dalšího zvoleného souboru.

V kompletním výpisu DNS zpráv nalezneme pouze informace o A, AAAA, NS, MX, SOA, CNAME a SRV typech záznamů. Ostatní typy záznamů program ignoruje a nevypisuje.

2.2 Návod na použití

V archivu je přiložen soubor `Makefile`, tudíž pro přeložení programu stačí mít nainstalovaný balíček make a použít stejnojmenný příkaz `make`.

Po přeložení programu vznikne binární soubor `dns-monitor`. Tento binární soubor lze spustit s několika argumenty:

- *-i <interface>* je argument udávající rozhraní, na kterém bude program poslouchat,
- *-p <pcapfile>* určuje soubor, který program zpracuje a pokusí se v něm nalézt DNS komunikaci,
- *-v* zapíná kompletní výpis detailů o DNS zprávách,
- *-d <domainsfile>* slouží pro výběr souboru, do kterého se případně uloží nalezená doménová jména,
- *-t <translationsfile>* je poslední argument, který určuje kam se mají uložit případné překlady adres, které se během komunikace naleznou.

Argumenty *-v*, *-d <domainsfile>* a *-t <translationsfile>* jsou volitelné, ale program vyžaduje použití *-i <interface>* pro naslouchání na daném rozhraní, případně *-p <pcapfile>* pro zpracování souboru, avšak program nesmí dostat oba zároveň. Může buď naslouchat na rozhraní, nebo zpracovat soubor.

V případě potřeby, může uživatel ukončit naslouchání na rozhraní pomocí zaslání signálu SIGINT skrz klávesovou zkratku Ctrl+C.

Kapitola 3

Testování

3.1 Způsob testování

Testování probíhalo několika způsoby. Nejdříve byl program testován pomocí souborů formátu PCAP, které dostával na vstup a standardní výstup z programu se porovnával s předpokládaným výstupem. Tento předpokládaný výstup se dal zjistit jednoduše pomocí nástroje wireshark, jenž slouží i na vytvoření PCAP souborů, které se pak mohou otestovat.

Abychom měli v síti vůbec nějaké DNS záznamy, potřebujeme alespoň zavolat příkaz ping na nějakou doménu. Případně lze otevřít webový prohlížeč a otevřít nějakou webovou stránku za pomocí doménového jména.

Dalším způsobem testování probíhalo na rozhraní. Toto vyžaduje práva k přístupu a naslouchání na daném rozhraní. Na lokálním zařízení stačí spustit program jako root. Pro zjištění dostupných rozhraní byl využit nástroj netstat.

Poslední věc na otestování byly výstupní soubory s doménovými jmény a s překlady. Tyto soubory se dají porovnat s daty v sekcích Question, Answer, Authority a Additional, které lze získat z kompletního výpisu.

3.2 Výsledky testů

3.2.1 Test: Zjednodušený výpis ze souboru

Použitý příkaz:

```
./dns-monitor -p tests/dns.pcap
```

Standardní výstup:

```
2024-09-29 10:52:17 192.168.170.20 -> 192.168.170.8 (R 1/1/0/0)
2024-09-29 10:52:17 192.168.170.8 -> 192.168.170.20 (Q 1/0/0/0)
2024-09-29 10:52:17 192.168.170.56 -> 217.13.4.24 (Q 1/0/0/0)
2024-09-29 10:52:17 192.168.170.20 -> 192.168.170.8 (R 1/4/0/0)
2024-09-29 10:52:17 217.13.4.24 -> 192.168.170.56 (R 1/0/0/0)
2024-09-29 10:52:17 192.168.170.56 -> 217.13.4.24 (Q 1/0/0/0)
2024-09-29 10:52:17 217.13.4.24 -> 192.168.170.56 (R 1/0/0/0)
2024-09-29 10:52:17 192.168.170.56 -> 217.13.4.24 (Q 1/0/0/0)
2024-09-29 10:52:17 217.13.4.24 -> 192.168.170.56 (R 1/0/0/0)
```

3.2.2 Test: Kompletní výpis ze souboru

Použitý příkaz:

```
./dns-monitor -p tests/dns.pcap -v
```

Standardní výstup:

```
Timestamp: 2024-09-29 10:52:17
SrcIP: 192.168.170.8
DstIP: 192.168.170.20
SrcPort: UDP/32797
DstPort: UDP/53
Identifier: 0x208A
Flags: QR=0, OPCODE=0, AA=0, TC=0, RD=1, RA=0, AD=0, CD=0, RCODE=0

[Question Section]
isc.org. IN NS
====================

Timestamp: 2024-09-29 10:52:17
SrcIP: 192.168.170.20
DstIP: 192.168.170.8
SrcPort: UDP/53
DstPort: UDP/32797
Identifier: 0x208A
Flags: QR=1, OPCODE=0, AA=0, TC=0, RD=1, RA=1, AD=0, CD=0, RCODE=0

[Question Section]
isc.org. IN NS

[Answer Section]
isc.org. 3600 IN NS ns-ext.nrt1.isc.org.
isc.org. 3600 IN NS ns-ext.sth1.isc.org.
isc.org. 3600 IN NS ns-ext.isc.org.
isc.org. 3600 IN NS ns-ext.lga1.isc.org.
====================
```

3.2.3 Test: Kompletní výpis z rozhraní se soubory pro doménová jména a překlady

Použitý příkaz:

```
./dns-monitor -i wlp9s0 -v -d domains.txt -t translations.txt
```

Standardní výstup:

```
Timestamp: 2024-10-04 20:04:18
SrcIP: 10.0.0.138
DstIP: 10.0.0.11
SrcPort: UDP/53
```

```
DstPort: UDP/37628
Identifier: 0x781F
Flags: QR=1, OPCODE=0, AA=0, TC=0, RD=1, RA=1, AD=0, CD=0, RCODE=0

[Question Section]
vscode-sync.trafficmanager.net. IN AAAA

[Answer Section]
vscode-sync.trafficmanager.net. 48 IN CNAME
↪  vscode-sync-euw-01.azurewebsites.net.
vscode-sync-euw-01.azurewebsites.net. 24 IN CNAME
↪  waws-prod-am2-325.sip.azurewebsites.windows.net.
waws-prod-am2-325.sip.azurewebsites.windows.net. 1130 IN CNAME
↪  waws-prod-am2-325.westeurope.cloudapp.azure.com.

[Authority Section]
westeurope.cloudapp.azure.com. 60 IN SOA ns1-201.azure-dns.com.
====================
...

Timestamp: 2024-10-04 20:04:18
SrcIP: 10.0.0.138
DstIP: 10.0.0.11
SrcPort: UDP/53
DstPort: UDP/39262
Identifier: 0x3B6D
Flags: QR=1, OPCODE=0, AA=0, TC=0, RD=1, RA=1, AD=0, CD=0, RCODE=0

[Question Section]
vscode-sync.trafficmanager.net. IN A

[Answer Section]
vscode-sync.trafficmanager.net. 48 IN CNAME
↪  vscode-sync-euw-01.azurewebsites.net.
vscode-sync-euw-01.azurewebsites.net. 24 IN CNAME
↪  waws-prod-am2-325.sip.azurewebsites.windows.net.
waws-prod-am2-325.sip.azurewebsites.windows.net. 1130 IN CNAME
↪  waws-prod-am2-325.westeurope.cloudapp.azure.com.
waws-prod-am2-325.westeurope.cloudapp.azure.com. 9 IN A 13.69.68.64
====================
```

Obsah souboru `domains.txt`:

```
ns1-201.azure-dns.com
vscode-sync-euw-01.azurewebsites.net
vscode-sync.trafficmanager.net
waws-prod-am2-325.sip.azurewebsites.windows.net
waws-prod-am2-325.westeurope.cloudapp.azure.com
westeurope.cloudapp.azure.com
```

Obsah souboru `translations.txt`:

```
waws-prod-am2-325.westeurope.cloudapp.azure.com 13.69.68.64
```

3.2.4 Test: Zjednodušený výpis z rozhraní s výpisem nalezených překladů

Použitý příkaz:

```
./dns-monitor -i wlp9s0 -t translations.txt
```

Standardní výstup:

```
2024-10-04 20:21:00 10.0.0.11 -> 10.0.0.138 (Q 1/0/0/0)
2024-10-04 20:21:00 10.0.0.138 -> 10.0.0.11 (R 1/12/0/0)
2024-10-04 20:21:05 10.0.0.11 -> 10.0.0.138 (Q 1/0/0/0)
2024-10-04 20:21:05 10.0.0.11 -> 10.0.0.138 (Q 1/0/0/0)
2024-10-04 20:21:05 10.0.0.138 -> 10.0.0.11 (R 1/1/0/0)
2024-10-04 20:21:06 10.0.0.138 -> 10.0.0.11 (R 1/1/0/0)
2024-10-04 20:21:06 10.0.0.11 -> 10.0.0.138 (Q 1/0/0/0)
2024-10-04 20:21:06 10.0.0.138 -> 10.0.0.11 (R 1/0/1/0)
```

Obsah souboru `translations.txt`:

```
connectivity-check.ubuntu.com 185.125.190.17
connectivity-check.ubuntu.com 185.125.190.18
connectivity-check.ubuntu.com 185.125.190.48
connectivity-check.ubuntu.com 185.125.190.49
connectivity-check.ubuntu.com 185.125.190.96
connectivity-check.ubuntu.com 185.125.190.97
connectivity-check.ubuntu.com 185.125.190.98
connectivity-check.ubuntu.com 91.189.91.48
connectivity-check.ubuntu.com 91.189.91.49
connectivity-check.ubuntu.com 91.189.91.96
connectivity-check.ubuntu.com 91.189.91.97
connectivity-check.ubuntu.com 91.189.91.98
www.example.com 2606:2800:21f:cb07:6820:80da:af6b:8b2c
www.example.com 93.184.215.14
```

Literatura

[1] *Domain names - implementation and specification* RFC 1035. RFC Editor, listopad 1987. Dostupné z: https://doi.org/10.17487/RFC1035.

[2] Robson, R. *Using the STL: The C++ Standard Template Library*. Springer New York, 2012. ISBN 9781461213123. Dostupné z: https://books.google.cz/books?id=Wfj2BwAAQBAJ.

[3] Satrapa, P. a Filip, O. *Domain Name System: Principy fungování DNS a praktické otázky spojené s jeho používáním*. CZ.NIC, 2023. ISBN 9788088168720. Dostupné z: https://books.google.cz/books?id=RFfgEAAAQBAJ.