ISA - Síťové aplikace a správa sítí
Projekt varianta 2.
DNS export pomocí protokolu syslog

Lukáš Kulda (xkulda01)

9. listopadu 2018

Obsah

1 Úvod

1.1 Úvod do problematiky

DNS (Domain Name System) se mimo jiné primárně zabývá překladem tzv. doménových jmen na IP adresy. Existují různé úrovně doménových jmen, jenž jsou hierarchicky rozdělené. Každá úroveň je někým spravována, nejvyšší "root"úroveň je spravována organizací ICANN, následují TLD (Top Level Domain), které jsou spravovány akreditovanými registrátory, za nimi jsou hned domény 3. úrovně, které si spravují už jednotliví uživatelé, organizace apod. DNS je vlastně globální databáze doménových jmen distribuována servery DNS, které mezi sebou komunikují a dotazují se vzájemně na záznamy. Dnes už DNS používá několik desítek typů záznamů a neslouží jen k překladu jmen na adresy, ale i k autorizaci DNS serveru, identifikaci služby a další.

1.2 Popis aplikace

Tato aplikace se zabývá zpracováním různých DNS záznamů a vytvořením statistik z těchto záznamů, které buď vypíše na standardní výstup nebo odešle na centrální logovací server pomocí protokolu Syslog.

2 Implementace

Aplikace je implementována v jazyce C se standardem C99. Jsou implementovány určité stavové výpisy, aby uživatel měl povědomí, že aplikace provádí nějaké akce, například při úspěšném odeslání statistik na centrální logovací server.

2.1 Sockety

Odchytávání packetů na rozhraní je implementováno pomocí `RAW SOCKETŮ`. Pokud je programu korektně předána kombinace parametrů včetně parametru `-i`, vytvoří se `socket(AF_PACKET, SOCK_RAW, htons(0x800));` nabinduje se na interface předaný parametrem `-i` a nastaví se timeout pro obdržení packetu. Následně se zavolá funkce pro zpracování DNS packetů, zde program běží ve smyčce dokud není explicitně ukončen, například signálem `SIGINT`.

Pro odesílání na logovací server je využit `UDP SOCKET`. Před vytvořením tohoto socketu, se nejdříve zjistí, zda byl korektně zadán syslog-server parametrem `-s`, poté se vytvoří `socket(version, SOCK_DGRAM, 0)`, kde *"version"* je verze adresy, buď IPv4 nebo IPv6. K odesílání dochází jen pokud se zpracuje soubor, nebo existují zachycené a zpracované DNS záznamy z rozhraní.

2.2 DNS záznamy

Aplikace umí zpracovávat tyto typy DNS záznamů: `A, AAAA, NS, CNAME, SOA, PTR, MX, TXT, DS, RRSIG, NSEC, DNSKEY`.

Program si dynamicky počítá offset, odkud má přijatá data zpracovávat, je implementováno několik funkcí, které zpracovávají určité části záznamu. Po přijetí packetu se `char* dns_data` dívají na začátek DNS dat, konkrétně tedy za hlavičku specifikovanou *[RFC 2136, 2535]*. Z hlavičky si vezmeme pouze [2] Total Answer RRs a Total Authority RRs, abychom věděli kolik záznamů jsme z daného packetu obdrželi a mohli je postupně zpracovat. Jsme tedy na začátku *"Questions"*, ale z nich nezískáme potřebné informace, proto je přeskočíme, ale obsahují doménové jméno, takže mohou mít různou délku, to řeší funkce `get_offset_to_skip_queries`. Pro každý záznam je vždy alokována dynamicky paměť. Odpověď má vždy na začátku doménové jméno, to zpracuje funkce `get_domain_name` a aktualizuje se offset, tak aby se koukal za zpracovaná data. Dále pomocí funkce `get_rr_type` získáme typ záznamu a délku dat proměnlivou na základě typu a obsahu odpovědi funkcí `get_rr_data_length`. Tyto informace jsou nezbytné a předávají se dál poslední funkci `process_rr_data`, která podle typu záznamu ví co s daty má dělat a vytvoří tak záznam ve tvaru `domain-name rr-type rr-answer count`, jenž nakonec vloží do globální tabulky s rozptýlenými položkami.

2.3 Soubor

Pro zpracování souboru `*.pcap` nejsou použity žádné *"libpcap"* knihovny apod. Formát souboru pcap [1] obsahuje jednou `Global Header` a pak vždy `Packet Header` a `Packet Data` několikrát přímo zasebou. Po standardním otevření souboru se pomocí funkce `fread` přečte potřebný kus dat, `Global Header` se přeskočí, z `Packet Headeru` získáme délku dat a pak se z `Packet Data` zpracovávají DNS záznamy viz výše.

2.4 Tabulka s rozptýlenými položkami

V rámci aplikace je implementována tabulka s rozptýlenými položkami, přejata z vlastní implementace v projektu z předmětu IAL. Nejsou využity všechny funkce, jelikož to není potřeba, některé jsou mírně poupravené, pro správnou funkčnost v rámci této aplikace. Je zde deklarována jedna globální tabulka pro vypsání všech statistik na `stdout`.

2.5 Omezení aplikace

Není implementována fragmentace, zpracovávány jsou pouze DNS packety protokolem UDP a ne TCP. NSEC záznamy nekompletní.

3 Použití aplikace

3.1 Překlad

Překlad se provede příkazem `"make"` nebo `"make all"`. Překlad je proveden ve standardu C99 jazyka C s flagy `"-Wall -Wextra -pedantic"`.

3.2 Spuštění

Spuštění aplikace vyžaduje kvůli implementaci `"RAW_SOCK"` socketů práva **superusera**. Spuštění se tedy provede pomocí příkazu:
`sudo ./dns-export [-r file.pcap] [-i interface] [-s syslog-server] [-t seconds]`.

`[-r file.pcap]` Zpracuje DNS záznamy z pcap souboru.
`[-i interface]` Rozhraní na kterém aplikace bude naslouchat a příjmat packety.
`[-s syslog-server]` Logovací server na který se odešlou statistiky.
`[-t seconds]` Periodické odesílání statistik na logovací server při naslouchání na rozhraní dané parametrem -i, výchozí hodnota je 60 sekund.

Pro podrobnější informace o práci s aplikací je přiložen manuál `dns-export.1`, který lze přečíst pomocí příkazu:
`man -l dns-export.1`

Reference

[1] https://wiki.wireshark.org/Development/LibpcapFileFormat.

[2] https://http://www.networksorcery.com/enp/protocol/dns.htm.