Autorzy: Igor Trybański

Łukasz Kądracki

Część Teoretyczna

DNS

Istotą poniższego referatu będzie wyjaśnienie oraz omówienie wybranych zagadnień i istoty działania systemu DNS.

DNS(Domain Name System, pol. system nazw domen) jest to hierarchiczny, rozproszony system nazw sieciowych, który odpowiada na zapytania o nazwy domen. W sieci Internet każdy komputer rozpoznawany jest za pomocą swojego unikalnego numeru IP. Numer IP jest długi i w wielu przypadkach trudny to zapamiętania, zatem system DNS jest niezwykle ważny, gdyż dzięki niemu możemy posługiwać się nazwami mnemonicznymi np.google.com. DNS sprawia właśnie, że nazwy te mogą zostać przetłumaczone na adresy IP, który im odpowiadają. Podstawą systemu DNS są tzw. domeny najwyższego poziomu. Podzielone są one na dwie kategorie: domeny krajowe oraz domeny ogólne. Obecnie istnieje tendencja do rejestrowania domen wyłącznie w obrębie domen krajowych. W obrębie domen tworzone mogą być również subdomeny np. interia.pl oraz jej subdomena poczta.interia.pl. Oprócz domen istnieją także strefy. Strefa jest to obszar jednej lub wielu subdomen oddelegowany podmiotowi pod administrację. Jeśli natomiast chodzi o rodzaje serwerów DNS to możemy podzielić je na dwa typy:

- podstawowy – odczytuje on dane dla strefy domen z pliku przebywającego na serwerze www konta hostingowego oraz wysyła odczytane dane do serwera drugorzędnego
- drugorzędny (podrzędny) – odbiera dane z serwera podstawowego, zapewnia bezpieczeństwo danych (w razie niedostępności serwera podstawowego), zapewnia szybkość przetwarzania informacji.

Warto podać teraz uproszczony schemat działania usługi DNS:

Komputer najpierw sprawdza informacje zawarte w pliku hosts i pamięci podręcznej (uprzednio otrzymane odpowiedzi od serwera), a następnie zachowuje się zależnie od rodzaju zapytania:

- Iteracyjnie: Lokalny serwer DNS odpytuje serwer DNS root, który następnie wskazuje serwer, który powinien zawierać pożądane informacje. Na przykład jeśli poszukiwany adres ma domenę xyz.pl, to DNS obsługujący domenę .pl odeśle nas do serwera subdomeny xyz.

-Rekurencyjnie: Lokalny serwer DNS odpytuje kolejny serwer, który jest wyżej w hierarchii. Ten z kolei, jeśli nie posiada pożądanej informacji robi to samo co swój poprzednik, aż osiągnie serwer poszukiwanej domeny.

Warto w tym miejscu wyjaśnić czym są wspomniane wyżej serwery root. Są one serwerami na szczycie drzewiastej struktury domen. ~~Z reguły~~ posiadają one tylko odwołania do serwerów odpowiedzialnych za domeny ~~niższego rzędu.~~ W Polsce znajduje się 9 instancji serwerów root:

Serwery ROOT zawieraja informacje o tym, kto (jaki serwer) obsluguje domeny najwyzszego rzedu (n

- 7 w Warszawie

- 1 w Poznaniu

- 1 w Gdańsku

Aby odpytywać wspomniane wyżej serwery DNS można korzystać z jednego z kilku programów z linii poleceń, które na to pozwalają. Te programy to na przykład:

- host – prosty program służący do wykonywania zapytań DNS, Najczęściej używany do prostej translacji z IP na nazwę i odwrotnie. Przykład użycia:

  umk.pl has address 158.75.1.96
  umk.pl mail is handled by 0 koala.uci.umk.pl.
  umk.pl mail is handled by 0 outgoing.umk.pl.

- dig (domain information groper) – elastyczne narzędzie służące do przepytywania serwerów DNS. Po wykonaniu wyszukiwania DNS wyświetla odpowiedzi zwrócone z serwerów, które były odpytywane. Narzędzie jest bardzo popularne wśród administratorów z racji na przejrzystość zwracanych danych oraz stosunkowo łatwą obsługę. Gdy w linii poleceń nie podamy nazwy serwera dig spróbuje połączyć się z każdym z serwerów występujących w /etc/resolv.conf. Przykłady:

  - ; <<>> DiG 9.16.8 <<>> umk.pl
  ;; global options: +cmd
  ;; Got answer:
  ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 1909
  ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
  ;; QUESTION SECTION:
  ;umk.pl. IN A
  ;; ANSWER SECTION:
  umk.pl. 5810 IN A 158.75.1.96
  ;; Query time: 0 msec
  ;; SERVER: 158.75.88.5#53(158.75.88.5)
  ;; WHEN: sob paź 31 16:44:10 CET 2020
  ;; MSG SIZE rcvd: 40

  - dig .com NS +noall +answer

- nslookup – narzędzie wykorzystywane do diagnostyki stref DNS. nslookup pozwala na wysłanie do dowolnego wskazanego serwera DNS, prośby o zwrócenie wskazanego typu rekordu dla wskazanej nazwy DNS, co mimo tego, iż może wydawać się mało imponujące, jest największą zaletą tego narzędzia. Przykład:

  dig rowniez posiada taka mozliwoscMG

  Server: 158.75.88.5
  Address: 158.75.88.5#53
  Non-authoritative answer:
  Name:umk.pl
  Address: 158.75.1.96

Popularnym serwerem DNS jest BIND((*Berkeley Internet Name Domain*), który jest ważnym składnikiem zapewniającym poprawne działanie systemu nazw w Internecie. Głównym plikiem konfiguracyjnym BIND jest `/etc/named.conf.`
Natomiast sam plik named.conf składa się z:

- bloku options { … }; w którym zawarte są opcje serwera.
- bloku logging
- definicji stref
- dołączeń innych plików

Wszystkie pliki powiązane z strefami znajdują się w /var/named/.
Rekordy używane w plikach stref:

- Format Of BIND – przechowuje pola zawierające informacje np. o nazwie, czasie przechowywania danych w bazie, klasę adresową.
- SOA(Start of Authority Entry) – określa on początek strefy. Może występować maksymalnie jeden rekord SOA dla danej strefy
- Name Server Entry – rekord ten określa, że dany komputer jest name server'em dla danej strefy
- Adress Entry – wylicza wszystkie adresy poszczególnych komputerów
- Domain Name Pointer Entry – pozwala on na odwoływanie się przez specjalne nazwy do innych lokalizacji w domenie
- Canonical Name Entry – specyfikuje alias dla pełnych nazw BINDA
- Well Known Services Entry – określa usługi prowadzone przez poszczególne protokoły na podanym adresie
- Host Information Entry – umożliwia zapisanie informacji o komputerze
- Mail Exchanger Entry – pole określa, która z maszyn w systemie lokalnym będzie pełnić funkcję gateway'a

Warto również wyjaśnić działanie kilku podstawowych konstrukcji z plików stref:

- $TTL – czas przez jaki serwer DNS będzie przetrzymywał wynik wykonanego zapytania. Po upływie czasu wykona ponowne zapytanie.
- $ORIGIN – jeśli nie zdefiniujemy tej dyrektywy, będzie miała ona wartość nazwy strefy z pliku named.rfc1912.zones lub named.conf.
- $GENERATE – dyrektywa pozwalająca na wygenerowanie dużej ilości linii w pliku (działa jak pętla). Przykład użycia:

  $GENERATE 1-254 $ IN PTR domlab$.studmat.uni.torun.pl.
  Przez $ będą oznaczone kolejno liczby od 1 do 254

Programem do zarządzania serwerem BIND jest *rndc.* Komunikuje się on z serwerem DNS przez połączenie TCP, wysyłając polecenia podpisane cyfrowo.

Konfiguracja *rndc*:
I. wpisujemy: *rndc-confgen*

Zostaje utworzony plik rndc.conf, znajduje się w nim metoda szyfrowania, klucz oraz opcje. Plik musi znajdować się w /etc/.

II. W pliku named.conf muszą znaleźć się:

```
include "/etc/rndc.key";
controls {
inet 127.0.0.1 allow { localhost; } keys { "rndc-key"; };
};
```

III. Restartujemy usługę DNS.

W pliku rndc.key znajdują się informacje o kluczu.

Widać zatem, iż DNS jest bardzo potrzebny w codziennym użytkowaniu. Translacja adresów sieciowych na nazwy znacznie ułatwia obsługę i komunikację po stronie człowieka, natomiast nazwy przetłumaczone na adresy sieciowe są znacznie prostsze do przetworzenia po stronie sprzętowej. Z racji na ważną rolę jaką odgrywają serwery DNS w codziennym ruchu sieciowym, narażone są one na ataki. Spowodować mogą one np. opóźnienia w wczytywaniu serwisów, oszustwa mające na celu wyłudzenie pieniędzy na przykład poprzez ładowanie fałszywych stron internetowych, a także ataki mogą spowodować zepsucie struktury Internetu rzeczy.

Można zatem zadać sobie pytanie czemu warto używać tego rozwiązania i jakie są jego minus oraz niedociągnięcia . DNS jest warty użycia gdyż, jest najbardziej rozpowszechnioną usługą, zatem jego użycie jest po prostu proste oraz wygodne. Nie jest to jednak rozwiązanie idealne. DNS nie zapewnia np. mechanizmów autoryzujących. Na szczęście problem ten rozwiązuje DNSSEC(DNS Secruity Extensions) zapewniając uwierzytelniania serwerów DNS przy użyciu kryptografii asymetrycznej i podpisów cyfowych.

Bibliografia:

https://pl.wikipedia.org/wiki/Domain_Name_System
https://pl.wikipedia.org/wiki/DNSSEC
http://slow7.pl/sieci-komputerowe/item/152-co-w-sieci-siedzi-protokol-dns
https://root-servers.org/
https://domenomania.pl/centrum-wiedzy/jakie-sa-typy-i-rodzaje-serwerow-dns
https://www.serverpronto.com/kb/page.php?id=Configure+your+DNS+Server+%28CentOS%2FFedora%29

https://www.tech-recipes.com/rx/312/dnsbind-resource-record-using-generate-to-makemany-records/

https://www.zytrax.com/books/dns/ch8/origin.html
https://tecadmin.net/configure-rndc-for-bind9/

https://en.wikipedia.org/wiki/DNS_zone

http://www.cs.put.poznan.pl/ddwornikowski/sieci/sieci2/dns.html

https://mail.pk.edu.pl/~pmj/dns/dns_2.html

https://www.pcworld.pl/porada/Zagrozenia-DNS-jakie-sa-i-jak-ich-unikac,414721.html

https://www.tutorialspoint.com/unix_commands/host.htm

http://71.61.4.1/linuxdocs/rhl-rg-en-7.3/s1-bind-configuration.html

https://www.tutorialspoint.com/unix_commands/dig.htm

https://www.tutorialspoint.com/unix_commands/nslookup.htm

Część Praktyczna

Instrukcja do zadania:

I. Zakładam, że jesteśmy cały czas na zalogowani na root.
dnf install bind bind-utils -y
II. Następnie edytujemy następujące pliki (wszystkie wystąpienia liczby 227 zamieniamy na
ostatni oktet ze swojego ip):

/etc/named.conf: (muszą znaleźć się tu takie linijki)

W bloku options:

```
listen-on port 53 { 127.0.0.1; 192.168.134.227; };
allow-query { localhost; 192.168.134.0/24; };
allow-query-cache { localhost; 192.168.134.0/24; };
allow-transfer {localhost; 192.168.134.185;};  //tutaj podajemy ostatni oktet z ip naszego serwera zapasowego
```

Na końcu pliku:

```
zone ''domlab227.studmat.uni.torun.pl'' IN{
        type master;
        file ''forward.fedora.local'';
        allow-update {none;};
        allow-transfer {192.168.134.185;}; //tutaj podajemy ostatni oktet z ip naszego serwera zapasowego
};
zone ''134.168.192.id-addr.arpa'' IN {
        type master;
        file ''reverse.fedora.local''
        allow-update {none;};
        allow-transfer {192.168.134.185;}; //tutaj podajemy ostatni oktet z ip naszego serwera zapasowego
};
zone ''domlab185.studmat.uni.torun.pl'' IN{    //tutaj podajemy ostatni oktet z ip mastera
        type slave;
        masters {192.168.134.185;};
        file ''slaves/forward.fedora.local'';
};
zone ''185.168.192.id-addr.arpa'' IN{    //tutaj podajemy ostatni oktet z ip naszego mastera
        type slave;
        masters {192.168.134.185;};
        file ''slaves/reverse.fedora.local'';
};
```

/var/named/forward.fedora.local:

```
$TTL 86400
@ IN    SOA     ns1.domlab227.studmat.uni.torun.pl. root.domlab227.studmat.uni.torun.pl. (
        2020110100 ;Serial
        480     ;Refresh
        120     ;Retry
        1800    ;Expire
```

```
            600         ;Minimum TTL
)
IN          NS          ns1.domlab227.studmat.uni.torun.pl.

IN          NS          ns2.domlab227.studmat.uni.torun.pl.
IN          MX          20          domlab227.studmat.uni.torun.pl.
IN          A           192.168.134.227
@ IN TXT "v=spf1 mx ip4:192.168.134.277 ~all"
ns1         IN          A           192.168.134.227
ns2         IN          A           192.168.134.185 ; //tutaj podajemy ostatni oktet z ip naszego serwera zapasowego
poczta      IN          CNAME       ns1
www         IN          CNAME       ns1
$GENERATE 1-254 lab$ IN A 192.168.134.$
```

var/named/reverse.fedora.local:

```
$TTL 86400
@           IN          SOA         domlab227.studmat.uni.torun.pl. root.studmat.uni.torun.pl. (
            2020110100 ;serial
            480         ;refresh
            120         ;retry
            1800        ;expire
            600         ;minimum ttl
)
@           IN          NS          ns1.domlab227.studmat.uni.torun.pl.
@           IN          NS          ns2.domlab227.studmat.uni.torun.pl.

ns1         IN          A           192.168.134.227
ns2         IN          A           192.168.134.185 //tutaj podajemy ostatni oktet z ip naszego serwera zapasowego

@           IN          PTR         studmat.uni.torun.pl.
$GENERATE 1-254 lab$.domlab227.studmat.uni.torun.pl          IN          A           192.168.134.$
$GENERATE 1-254 $           IN          PTR         lab$.domlab227.studmat.uni.torun.pl.
```

III. Zajmijmy się udostępnieniem portu. W pliku /etc/firewalld/zones/FedoraServer.xml w bloku zone dodajemy następującą linijkę:

```
<service name=”dns”/>
```

Następnie wpisujemy komendę:

```
systemctl reload firewalld
```

IV. Sprawdzamy poprawność składniową plików:

named-checkconf /etc/named.conf
named-checkzone forward.fedora.local /var/named/forward.fedora.local
named-checkzone reverse.fedora.local /var/named/reverse.fedora.local

V. Jeśli sprawdzenie wyszło ok, to włączamy usługę:

systemctl enable named
systemctl start named

VI. Ustawiamy nasz serwer DNS jako domyślny na naszej maszynie. W pliku /etc/resolv.conf
zakomentowujemy wszystkie linie „nameserver” używając #. Dodajemy nasz:

nameserver 192.168.134.227

Plik resolv.conf jest automatycznie updatowany, dlatego nasza zmiana nie jest trwała, aby to

zmienić:

chattr +i /etc/resolv.conf

VII. Testujemy działanie. Wpisujemy w konsoli (jeśli dig nie jest zainstalowany to instalujemy
tak jak binda):

dig umk.pl

Wszystko działa jeśli w pola QUERY, ANSWER, AUTHORITY są nie zerowe, oraz gdy w jednej z
ostatnich linijek:

;; SERVER: 192.168.134.227

pojawia się nasze ip.

Bibliografia:

https://www.serverpronto.com/kb/page.php?id=Configure+your+DNS+Server+%28CentOS%2FFedora%29
https://fedoramagazine.org/how-to-setup-a-dns-server-with-bind/
https://docs.fedoraproject.org/en-US/Fedora/15/html/Deployment_Guide/s2-bind-zone.html?fbclid=IwAR3Y-J2nrR2JbD6kAmKLM3Y9T7qrk-S2K6eT7KDvuepYy6j8pM4F4Msh9Ro

https://www.itsmarttricks.com/how-to-configure-slave-dns-server-with-bind-secondary-dns-server-in-linux/

Weryfikacja zadania:

1.

dig @192.168.134.227 umk.pl MX
dig @8.8.8.8 umk.pl MX

dig @192.168.134.227 mat.umk.pl NS
dig @8.8.8.8 mat.umk.pl NS

2. Domyślny serwer ustawiamy w pliku /etc/resolv.conf, zakomentowujemy obecne w nim
linijki zaczynające się na nameserver i dodajemy:

nameserver 192.168.134.227

za www odpowiada linia (plik forward):

www IN CNAME ns1

3. dig -x 192.168.134.227 powinno zwrócić taką linijkę:
;; ANSWER SECTION:
227.134.168.192.in-addr.arpa. 86400 IN PTR domlab227.studmat.uni.torun.pl.

4.
ping lab227.domlab227.studmat.uni.torun.pl //ten działa ciekawe jak inne
ping lab254.domlab227.studmat.uni.torun.pl //działa

5.

Aby sprawdzić czy odbył się transfer plików sprawdzamy czy w katalogu /var/named/slaves/ pojawiły się pliki, które miały zostać przesłane. Sprawdzenie czy

odbył się transfer odbywa się poprzez sprawdzenie pola serial;. Jeśli jest ono takie samie to wszystko zakończyło się sukcesem.

Logi można znaleźć w pliku *//var/named/data/named.run*

Powinny znaleźć się tam podobne linijki:

zone domlab227.studmat.uni.torun.pl/IN: sending notifies (serial 2020110100)

zone domlab185.studmat.uni.torun.pl/IN: sending notifies (serial 2020110100)

6.
dig domlab227.studmat.uni.torun.pl MX