

Σύγχρονες Εφαρμογές Ασφάλειας Δικτύων

Άσκηση 6 DNS με Bind9

Αναγνωστόπουλος Άγγελος
Νικόλαος 5^ο Έτος ΗΜΤΥ
up1066593

Πρώτα εγκαθιστούμε την υπηρεσία bind9 με το apt (ή με κάποιον άλλον package manager). Η προεπιλεγμένη του ρύθμιση επιτρέπει τα recursive queries, οπότε μπορούμε να τον χρησιμοποιήσουμε ως έχει, σαν DNS proxy. Αυτό βέβαια δεν έχει και τόση ουσία, μιας και θα έχουμε (προφανώς) τα ίδια αποτελέσματα με το να χρησιμοποιούσαμε απλώς τον δικό μας DNS. Το ενδιαφέρον είναι να στήσουμε τον δικό μας DNS για κάποια συγκεκριμένα domains. Θα πρέπει να κάνουμε configure τα εξής: Γενικές ρυθμίσεις bind, ζώνες και αρχεία για αυτές (κανονικά forward και reverse, θα περιοριστούμε στο ευθύ DNS lookup για την άσκηση).

Για τις γενικές ρυθμίσεις θα κάνουμε edit το αρχείο /etc/bind/named.conf.options. Σε αυτό θα ορίσουμε τις πόρτες και τα IP στα οποία ακούμε, τα queries που δεχόμαστε, το εάν έχουμε dnssec και κυρίως τους forwarders και τα recursive queries. Forwarders είναι οι DNS που ο δικός μας θα ρωτήσει στην περίπτωση που δεν έχει κάποιο A record που μπορεί να μας επιστρέψει απευθείας. Για τις ανάγκες της άσκησης θα κλείσουμε το recursion για να δούμε μόνο τις απαντήσεις του bind9.

```
options {
    directory "/var/cache/bind";
    dump-file "/var/cache/bind/example.com.db";
    listen-on port 53 {any;};
    allow-query {any;};
    dnssec-validation auto;
    recursion no;
};
```

Figure 1: Basic Bind9 options w/o recursion

Εάν θέλουμε να έχουμε recursive queries και να χρησιμοποιήσουμε κάποιους άλλους DNS servers για να απαντήσουν σε ότα αίτημα εμείς δεν έχουμε διαθέσιμο A record, μπορούμε να αλλάξουμε το configuration μας ως εξής για να επιτρέπουμε κάτι τέτοιο στους αντίστοιχους forwarders:

```
options {
    directory "/var/cache/bind";
    dump-file "/var/cache/bind/example.com.db";
    listen-on port 53 {any;};
    allow-query {any;};
    dnssec-validation auto;
    recursion yes;
    allow-recursion {any;};
    allow-query-cache {any;};
    forwarders {1.1.1.1;};
};
```

Figure 2: Bind9 options configuration with recursion

Για τις ζώνες για τις οποίες είμαστε υπεύθυνοι θα κάνουμε edit το αρχείο /etc/bind/named.conf.local. Εδώ θα ορίσουμε το ποιες ζώνες εξυπηρετούμε και το είδος του server μας για αυτές (master/slave). Επίσης δίνουμε το που βρίσκεται το αρχείο με τα δεδομένα για τις ζώνες και το κατά πόσο αυτό μπορεί να αλλάξει εν όψει νέων δεδομένων (πχ. από forwarders). Προφανώς και το allow-update θα είναι κλειστό για την άσκηση.

```

zone "example.com" IN {
    type master;
    file "/var/cache/bind/forward.example.com";
    allow-update {none;};
};

```

Figure 3: Bind9 zone configuration

Για τα αρχεία των ζωνών και τους ορισμούς των παραμέτρων στα queries, καθώς και τα NS και A records, θα φτιάξουμε το forward.example.com. Κάνω την άσκηση 2 μήνες πριν την προθεσμία της και δεν έχω το αρχείο από το eclass οπότε θα γράψουμε ένα custom. Εδώ τα πράγματα είναι κάπως περίπλοκα και υπάρχει όλη η ουσία του DNS. Εδώ θα ορίσουμε τους name servers που μας απαντούν στα requests, τις διευθύνσεις τους, τις παραμέτρους ενός DNS query (id, timeout κλπ.) και τα A records, δηλαδή το σε ποιά IP γίνεται resolve ένα url. Να σημειωθεί ότι τα ονόματα πρέπει να τελειώνουν με τον χαρακτήρα της τελείας! Σώζουμε το αρχείο και ελέγχουμε εάν το bind9 service τρέχει κανονικά.

```

;
;
$TTL 604800
;
@      IN      SOA      ns1.example.com. ns2.example.com. (
    8      ;      Serial
    604800 ;      Refresh
    86400  ;      Retry
    2419200 ;      Expire
    604800 )      ;      Negative Cache TTL
;
;
; name servers - NS records
    IN      NS      ns1.example.com.
    IN      NS      ns2.example.com.
; name servers - A records
ns1.example.com.      IN      A      83.212.80.118
ns2.example.com.      IN      A      83.212.80.118

example.com.          IN      A      83.212.80.118
super.example.com.    IN      A      66.254.114.41
;client               IN      A      85.75.98.135

```

Figure 4: Bind9 records for example.com zone

```

debian@snf-33179:~$ sudo systemctl status bind9
● bind9.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/bind9.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2022-11-17 21:36:34 UTC; 4min 59s ago
     Docs: man:named(8)
  Process: 20447 ExecStop=/usr/sbin/rndc stop (code=exited, status=1/FAILURE)
 Main PID: 20478 (named)
    Tasks: 5 (limit: 4915)
   CGroup: /system.slice/bind9.service
           └─20478 /usr/sbin/named -f -u bind

Noé 17 21:36:34 snf-33179 named[20478]: managed-keys-zone: loaded serial 25
Noé 17 21:36:34 snf-33179 named[20478]: zone 0.in-addr.arpa/IN: loaded serial 1
Noé 17 21:36:34 snf-33179 named[20478]: zone localhost/IN: loaded serial 2
Noé 17 21:36:34 snf-33179 named[20478]: zone 255.in-addr.arpa/IN: loaded serial 1
Noé 17 21:36:34 snf-33179 named[20478]: zone 127.in-addr.arpa/IN: loaded serial 1
Noé 17 21:36:34 snf-33179 named[20478]: zone example.com/IN: loaded serial 8
Noé 17 21:36:34 snf-33179 named[20478]: all zones loaded
Noé 17 21:36:34 snf-33179 named[20478]: running
Noé 17 21:36:34 snf-33179 named[20478]: zone example.com/IN: sending notifies (serial 8)
Noé 17 21:40:14 snf-33179 named[20478]: client 139.19.117.8#52046 (LOSBS_53d45076.public-
lines 1-20/20 (END)

```

Figure 5: Bind9 service status

Αφού στήσουμε τον bind9 DNS μας, μπορούμε να τροποποιήσουμε το αρχείο `/etc/resolv.conf` και να τον θέσουμε ως τον DNS του υπολογιστή μας. Να σημειωθεί εδώ ότι με το πρώτο restart της υπηρεσίας NetworkManager, αυτό θα αναστραφεί. Εναλλακτικά, μπορούμε να κάνουμε dig με χρήση του flag `@nameserver` για να χρησιμοποιήσουμε το VM μας. Έχοντας πλέον το VM του okeanos σαν DNS, μπορούμε να ζητήσουμε τους servers τους οποίους αυτό εξυπηρετεί. Αυτοί ανήκουν στο zone `example.com`, δηλαδή αφορά τα URLs τύπου `xyz.example.com`. Έχουμε ήδη ορίσει τα A records για αυτά, οπότε το μόνο που μένει είναι να χρησιμοποιήσουμε τις εντολές dig για να κάνουμε ένα DNS query. Στην προκειμένη το `example.com` μας επιστρέφει το IP του ίδιου του DNS μας. Εάν δε, το κάνουμε curl (είχα ανοίξει apache server από πριν), θα λάβουμε το welcome page του Apache στα debian.

```

→ ~ dig @83.212.80.118 example.com

; <<>> DiG 9.18.1-1ubuntu1.2-Ubuntu <<>> @83.212.80.118 example.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 24145
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 3
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;example.com.                IN      A

;; ANSWER SECTION:
example.com.                 604800  IN      A      83.212.80.118

;; AUTHORITY SECTION:
example.com.                 604800  IN      NS      ns1.example.com.
example.com.                 604800  IN      NS      ns2.example.com.

;; ADDITIONAL SECTION:
ns1.example.com.             604800  IN      A      83.212.80.118
ns2.example.com.             604800  IN      A      83.212.80.118

;; Query time: 16 msec
;; SERVER: 83.212.80.118#53(83.212.80.118) (UDP)
;; WHEN: Thu Nov 17 23:37:31 EET 2022
;; MSG SIZE rcvd: 124

```

Figure 6: DNS response for example.com

Εάν πειράξουμε τις διευθύνσεις στον DNS μας με κάποια ήδη υπάρχουσα, τότε θα λάβουμε ως απάντηση για κάποιο άσχετο site, αυτή την IP. Τώρα μπορούμε να στέλνουμε όποιον χρησιμοποιεί τον DNS μας σε όποιο site θέλουμε και αυτός να μην έχει ιδέα. Για του λόγου το αληθές

```
→ ~ dig super.example.com

; <<>> DiG 9.18.1-1ubuntu1.2-Ubuntu <<>> super.example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 25818
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 3
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
super.example.com.          IN      A

;; ANSWER SECTION:
super.example.com.          604800  IN      A      66.254.114.41

;; AUTHORITY SECTION:
example.com.                 604800  IN      NS      ns2.example.com.
example.com.                 604800  IN      NS      ns1.example.com.

;; ADDITIONAL SECTION:
ns1.example.com.             604800  IN      A      83.212.80.118
ns2.example.com.             604800  IN      A      83.212.80.118

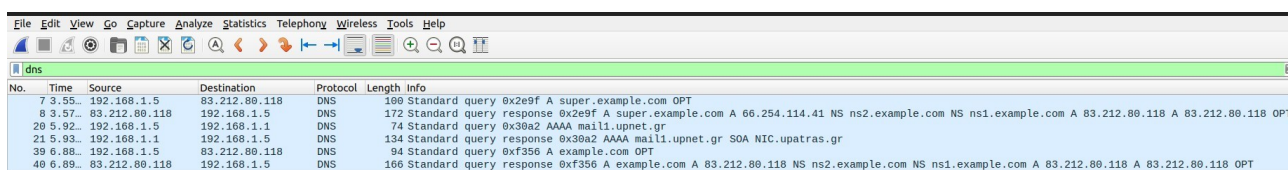
;; Query time: 20 msec
;; SERVER: 83.212.80.118#53(83.212.80.118) (UDP)
;; WHEN: Thu Nov 17 23:08:10 EET 2022
;; MSG SIZE rcvd: 130

→ ~ curl super.example.com
<html>
<head><title>301 Moved Permanently</title></head>
<body>
<center><h1>301 Moved Permanently</h1></center>
<hr><center>openresty</center>
</body>
</html>
→ ~
```

Figure 7: DNS response for super.example.com, replying with a different IP

Βλέπουμε ότι μας απαντούν οι ns1 και ns2, ότι είμαστε υπεύθυνοι για τα zones example.com. Το super.example.com (που πραγματικά δεν υπάρχει), μας επιστρέφει κανονικότητα μία IP. Εάν μάλιστα κάνουμε curl στην IP αυτή, παίρνουμε το HTTP status code 301, το οποίο στο browser μας θα έκανε redirect σε κάποια ιστοσελίδα.

Με ένα Wireshark scan μπορούμε να δούμε και τη συνομιλία με τους DNS μας.



No.	Time	Source	Destination	Protocol	Length	Info
7	3.55.	192.168.1.5	83.212.80.118	DNS	180	Standard query 0x2e9f A super.example.com OPT
8	3.57.	83.212.80.118	192.168.1.5	DNS	172	Standard query response 0x2e9f A super.example.com A 66.254.114.41 NS ns2.example.com NS ns1.example.com A 83.212.80.118 A 83.212.80.118 OPT
20	5.92.	192.168.1.5	192.168.1.1	DNS	74	Standard query 0x30a2 AAAA mail1.upnet.gr
21	5.93.	192.168.1.1	192.168.1.5	DNS	134	Standard query response 0x30a2 AAAA mail1.upnet.gr SOA NIC.upatras.gr
39	6.88.	192.168.1.5	83.212.80.118	DNS	94	Standard query 0xf356 A example.com OPT
40	6.89.	83.212.80.118	192.168.1.5	DNS	166	Standard query response 0xf356 A example.com A 83.212.80.118 NS ns2.example.com NS ns1.example.com A 83.212.80.118 A 83.212.80.118 OPT

Figure 8: Wireshark output for DNS conversation