Authoritative-only server & TSIG

cctld-workshop

Nairobi,12-15 october 2005

aalain@trstech.net

Different type of servers

Several types of name servers

- Authoritative servers
  - master (primary)
  - slave (secondary)
- (Caching) recursive servers
  - also caching forwarders
- Mixture of functionality

Why to seperate functionality ?

Authoritative and non-authoritative data are served to different sets of clients

- In order to serve authoritative data to the Internet,the nameserver must be outside any firewalls.
- Caching nameservers should generally be placed inside firewalls to protect them from outside abuse.

Serving authoritative data is more critical than serving cached data.

Why to seperate functionality ?

Caching nameservers are subject to poisoning

- if an attacker can trick your caching nameserver into accepting a forged RR with high TTL, invalid data may be used when serving authoritative data.

Certain denial-of-service and buffer overrun attacks are more likely to be successful in caching nameservers.

Why to seperate functionality ?

Authoritative server may serve authoritative data (constant in size) more efficiently when cached data does not compete for system resources.

- Recursing client uses memory (up to 20kb)
- Caching server uses memory to cache data
- Answering recursive queries needs processing time and system ressources

How to run an Authoritative-only Name server

Stop recursion

- With bind9

  options { recursion no ; };

  and restart named

Check dns response from server for non “ra” flag

dig @196.216.0.X xxxx.cctld.or.ke soa

Check if your server is now authoritative-only

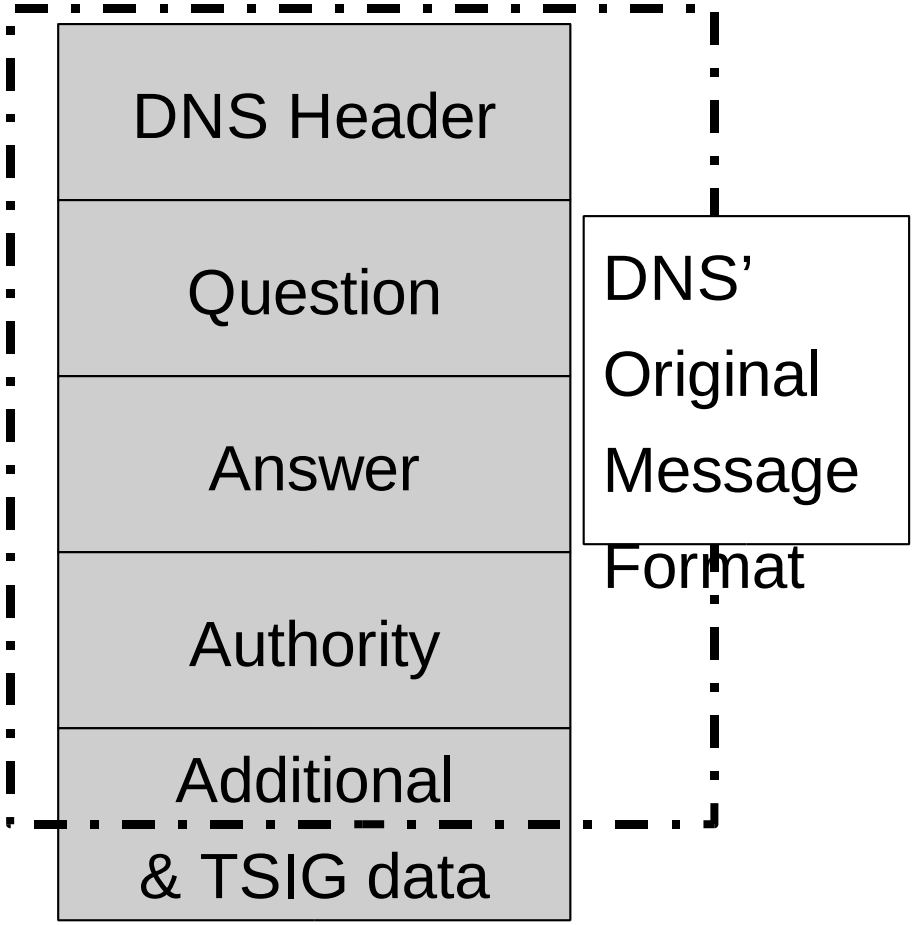
dig @196.216.0.X noc.cctld.or.ke A

You should get referrals to root servers

What is TSIG?

- A mechanism for protecting a message from a resolver to server and vice versa
- A keyed-hash is applied (like a digital signature) so recipient can verify message
- Based on a shared secret - both sender and reciever are configured with it
- RFC2845

TSIG and Message Format



TSIG and Message Format

```
; <<>> DiG 9.3.0 <<>> @localhost www.rfi.fr a -k /var/named/keys/Khost1-host2.+157+50032.key

;; QUESTION SECTION:
;www.rfi.fr.                IN      A

;; ANSWER SECTION:
www.rfi.fr.          86400   IN      A       194.117.210.38

;; AUTHORITY SECTION:
rfi.fr.              86400   IN      NS      ns1.mgn.net.
rfi.fr.              86400   IN      NS      ns2.mgn.net.
rfi.fr.              86400   IN      NS      ns3.mgn.net.

;; ADDITIONAL SECTION:
ns1.mgn.net.         172800  IN      A       195.46.193.86
ns2.mgn.net.         172800  IN      A       195.46.193.87
ns3.mgn.net.         172800  IN      A       195.46.214.178

;; TSIG PSEUDOSECTION:
host1-host2.         0       ANY     TSIG    hmac-md5.sig-alg.reg.int. 1126708829 300 16 jfqapw+5tnpqKceNaf5RnQ==  31634
NOERROR 0
```

```
; <<>> DiG 9.3.0 <<>> @localhost ripe.net a -k /var/named/keys/Khost1-host2.+157+50032.key +dnssec
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 7

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;ripe.net.                      IN      A

;; ANSWER SECTION:
ripe.net.               592     IN      A       193.0.0.214
ripe.net.               592     IN      RRSIG   A 5 2 600 20051014125237 20050914125237 49526 ripe.net.
GFWL87C+fcxPkFQ93ifF3SS0eq523Ktv92p0QPGcRs2q4t9pMVy8qjHN oTXEmthamwGdIy90wW5lcUtcfZMTarhx+0Q7zJwO76sXcjjNqMB4nbEb
i2D/596k23DghZ/+Wg/zy/u0yRoYm0LfmbKIZE4WHnb7AeSadKjEz+Ts iuS5wdk5F7SkxginC2JfYRmgxQOQ9NaY

;; ADDITIONAL SECTION:
ripe.net.               3592    IN      DNSKEY  257 3 5 AQOTT7bx7N38sPgDWniKnHnSnTxYxdMpEq7dyrDHDaRQgq7DULPWX6ZY
0U1XKKMuNloHRP7H8r17IBhgXcPZjZhtSGYagtPe22mhAMjZ4e8KGgP9 kJTTcpgzoYulvSiETBxjQ42EZWJG+6bxK+vyrwTbqEScmdZfqQz3ltVw
k6Sos0UuSmTeb2C6RSkgHaTpKCu5yIrncVer1gvyvXGv3HOel8jiDGuj 8peNByiaSRD4OIJUxu1jUqLvfDH6Anq6ZeNohxsYVUVajiRl1T+3x5+8
acgwat3V55Z1Nm4O4Z1BKECdPO65EXIEC7/pqs5XvpsiJbafdj03uqLS a3aScpy3
ripe.net.               3592    IN      DNSKEY  256 3 5 AQPGmhQPgNllavUXhVoDZZploCWbHr7lqcIGEiR/ct0KCTx4Skp+tBfX
qnq8fz8/UpK4B+s6xIk5FPdjNnFXltdSx81bcM+BadLHL5iuBdQdkH8e yJq2Fk1LAUiP2AB8RAFBd4WQMAklw5z/91jw6aMXSfAo6sSxUFSS1WY8
ChesKvwefNcqglSswlFwxjWHo9XNkFsx0u8=
ripe.net.               3592    IN      DNSKEY  256 3 5 AQPhEMiv80EEjX6gYDc8E7Osfumf4C/pZxBmTRRiOVL3h6lx1CIVCyP1
V34WuVUkqqpID2fxGzmUFTG0f61x9lzRapX0lIdlo2AtRCYWpkPY+D3F IrMYukWiC8pyeHF/a/Wk6HZNLVYko2dcLwUpfiDrK7zCFgR9DLcZkOmj
N5xB9CBzVrZDkd1lsGCC9hytndbLuZ3VtpE=
ripe.net.               3592    IN      RRSIG   DNSKEY 5 2 3600 20051014125237 20050914125237 10908
ripe.net.IF1rvW4DS5t4VhtZ6dQcA7tvcHtU+6qSwM8IoPh7v+2TqS8SLDBHnEeT qsMEE3oNihtXhwxVPZu3K8p1PCch399b/1z1dJCNLDtq4QVu5add0ygU
qQkZ028ARUVQ1QVR/FN/zhcAouMpCBG0kdijtGJv3AUS3OJFM/obubQt pZYGTuHS5Sqh54zu/R65veApd0dc3m6CE+bG+udn3XPLO/LAEbbetqcA
h4PWHcrkHcugSEv6UEk63c/jXysd23AgmBu2sPrj+DByokJaX7IBFLlj ySvr046CdQrkP0A5jvpVoiUXTwd6P7oNRi1Mz3SeT5PwVHWM+L3wJ8SE krrhtQ==
ripe.net.               3592    IN      RRSIG   DNSKEY 5 2 3600 20051014125237 20050914125237 49526
ripe.net.1+p9pI1h7+4EK0eiB91P2RX2Os0zhEkWR7cvgu61UqRVY5ziVP7txDPv eWMBISOgLgtE4L4zjEHpSc28uo33X0k/e3eiH6f8Hqa/IqrSdIrYpOxc
ob5eQUylqYbkKSadpOP1oi+1zPXdCgcPxyaBuIG7wMM5cY5pHFwPfLUA cw0hlkJDp8uHTau1K8iytocNGSOE3Ys4

;; TSIG PSEUDOSECTION:
host1-host2.            0       ANY     TSIG    hmac-md5.sig-alg.reg.int. 1126710236 300 16 eaDNJtJXavAjVqDZSANllA== 39
NOERROR 0
```

Names and Secrets

- TSIG name
  - A name is given to the key, the name is what is transmitted in the message (so receiver knows what key the sender used)
- TSIG secret value
  - A value determined during key generation
  - Usually seen in Base64 encoding
- 'Looks' like the rndc key
  - BIND uses same interface for TSIG and RNDC keys

Using TSIG to protect AXFR

- Deriving a secret
  - `dnssec-keygen -a ... -b ... -n... name`
- Configuring the key
  - in named.conf file, same syntax as for rndc
  - `key { algorithm ...; secret ...;}`
- Making use of the key
  - in named.conf file
  - `server x { keys ...; }`
  - where 'x' is an IP number of the other server

Configuration Example

Primary server

10.33.40.46

```
key ns1-ns2.zone. {
        algorithm hmac-md5;
        secret "APlaceToBe";
};
server 10.33.40.35 {
        keys {ns1-ns2.zone.;};
};
zone "my.zone.test." {
        type master;
        file...;
        allow-transfer {
                key ns1-ns2.zone.;
                key ns1-ns3.zone.;};
};
```

Secondary server

10.33.40.35

```
key ns1-ns2.zone. {
        algorithm hmac-md5;
        secret "APlaceToBe";
};
server 10.33.40.46 {
  keys {ns1-ns2.zone.;};
};
zone "my.zone.test." {
        type slave;
        file...;
        masters {10.33.40.46;};
        allow-transfer {
                key ns1-ns2.zone.;};
};
```

Again, the secret looks okay, but is purposely invalid

TIME!!!

- TSIG is time sensitive - to stop replays
  - Message protection expires in 5 minutes
  - Make sure time is synchronized
  - For testing, set the time
  - In operations, (secure) NTP is needed

Other uses of TSIG

- TSIG was designed for other purposes
  - Protecting sensitive stub resolvers
    - This has proven hard to accomplish
  - Dynamic Update
    - Discussed later, securing this relies on TSIG