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 separate 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 separate 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 separate 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 resources
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 receiver are configured with it
- RFC2845
TSIG and Message Format

DNS Header

Question

Answer

Authority

Additional & TSIG data

DNS' Original 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:

;; 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 20050914125237 20050914125237 49526 ripe.net.

;; ADDITIONAL SECTION:
ripe.net.               3592    IN      DNSKEY  256 3 5 AQPGmhQPgNllavUXhVoDZZploCBWbHr77kjcLEigi/crt0KCTx45kp+BiX
ripe.net.               3592    IN      DNSKEY  256 3 5 AQPhEMiv80EEjX6gYDc8E7Osfrumf4C/pZxBmTRRi0VL3h60k1CIVCyPl
ripe.net.               3592    IN      DNSKEY  256 3 5 AQPhmQhQPgNllavUXhVoDZZploCBWbHr77kjcLEigi/crt0KCTx45kp+BiX
ripe.net.               3592    IN      DNSKEY  256 3 5 AQPhmQhQPgNllavUXhVoDZZploCBWbHr77kjcLEigi/crt0KCTx45kp+BiX

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