The BIND Software

tsaimh (2024, CC-BY)
lwhsu (2020-2023, CC-BY)
? (?-2019)

國立陽明交通大學資工系資訊中心

Information Technology Center of Department of Computer Science, NYCU

BIND

- BIND
  - The Berkeley Internet Name Domain system
  - CSRG, UC Berkeley, 1980s
- Three main versions
  - BIND 4
    - Announced in 1980s
    - Based on RFC 1034, 1035
  - BIND 8
    - Released in 1997
    - Improvements including: efficiency, robustness and security
  - **BIND 9**
    - Released in 2000
    - Enhancements including: multiprocessor support, DNSSEC, IPv6 support, etc
  - BIND 10
    - Released version 1.0 and 1.1 in 2013
    - Released version 1.2 in 2014
      - ISC (Internet Software Consortium) has concluded BIND 10 development with Release 1.2
      - “Bundy” https://bundy-dns.de/

BIND – components

- Four major components
  - named
    - Daemon that answers the DNS query
    - Perform Zone transfer
  - Library routines
    - Routines that used to resolve host by contacting the servers of DNS distributed database
      - Ex: res_query, res_search, …etc.
  - Command-line interfaces to DNS
    - Ex: nslookup, dig, host
    - bind-tools package
  - rndc
    - A program to remotely control named

named in FreeBSD

- Installation
  - /usr/ports/dns/bind918
  - `# pkg install bind918`
- Startup
  - Edit /etc/rc.conf
    - `named_enable="YES"`
  - Manual utility command
    - `# service named start`
    - `$ rndc {stop | reload | flush …}`
- See your BIND version
  - `$ dig @127.0.0.1 version.bind txt chaos`
    - `version.bind.            0       CH      TXT     "9.18.24"`
  - `$ nslookup -debug -class=chaos -query=txt version.bind 127.0.0.1`
    - `version.bind    text = "9.18.24"`
- Good to be put inside of a jail!

BIND – Configuration files

- The complete configuration of named consists of
  - The config file (see named.conf(5) or named(8))
    - /usr/local/etc/namedb/named.conf
  - Zone data file
    - Address mappings for each host
    - Collections of individual DNS data records
  - The root name server hints

BIND Configuration – named.conf

- /usr/local/etc/namedb/named.conf
  - Roles of this host for each zone it serves
    - Master, slave, stub, or caching-only
  - Options
    - Global options (options {};)
      - The overall operation of named and server
    - Zone specific options. (zone {};)
- named.conf is composed of following statements:
  - include, options, server, key, acl, zone, view, controls, logging, trusted-keys, masters

Examples of named configuration

```
// isc.org TLD name server

options {
      directory "/var/named";
      datasize 1000M;
      listen-on { 204.152.184.64; };
      listen-on-v6 { 2001:4f8:0:2::13; };
      recursion no;
      transfer-source 204.152.184.64;
      transfer-source-v6 2001:4f8:0:2::13;
};
 zone "isc.org" {
       type master;
       file "master/isc.org";
       allow-update { none; };
       allow-transfer { none; };
 };

 zone "vix.com" {
       type slave;
       file "secondary/vix.com";
       masters { 204.152.188.234; };
 };
```

```
$TTL 57600
$ORIGIN atrust.com.
@                  SOA    ns1.atrust.com. trent.atrust.com. (
                            2010030400 10800 1200 3600000 3600 )
                   NS     NS1.atrust.com.
                   NS     NS2.atrust.com.
                   MX     10 mailserver.atrust.com.
                   A      66.77.122.161
ns1.atrust.com.    A      206.168.198.209
ns2.atrust.com.    A      66.77.122.161
www                A      66.77.122.161
mailserver         A      206.168.198.209
secure             A      66.77.122.161

; reverse maps
exterior1          A      206.168.198.209
209.198.168.206    PTR    exterior1.atrust.com.
exterior2          A      206.168.198.213
213.198.168.206    PTR    exterior2.atrust.com.
```

DNS Database
– Zone data

The DNS Database

- A set of text files such that
  - Maintained and stored on the domain’s master name server
  - Often called zone files
  - Two types of entries
    - Resource Records (RR)
      - The real data of a DNS database
    - Parser commands
      - Just provide some shorthand ways to create records
      - Influence the way that the parser interprets sequence orders or expand into multiple DNS records themselves

The DNS Database – Parser Commands

- Commands must start from the first column and be on a line by themselves
- `$ORIGIN domain-name`
  - To append to un-fully-qualified name
- `$INCLUDE file-name`
  - Split logical pieces of a zone file
  - Keep sensitive data (e.g., cryptographic keys) with restricted permissions
- `$TTL default-ttl`
  - Default value for time-to-live filed of records
- `$GENERATE start-stop/[step] lhs type rhs`
  - Only in BIND
  - Used to generate a series of similar records
  - Can be used in only CNAME, PTR, NS, A, AAAA, etc. record types

The DNS Database – Resource Record (1)

- Basic format
  - `[name] [ttl] [class] type data`
    - name: the entity that the RR describes
      - Can be relative or absolute
    - ttl: time in second of this RR’s validity in cache
    - class: network type
      - `IN` for Internet
      - `CH` for ChaosNet
      - `HS` for Hesiod
  - Special characters
    - ; (comment)
    - @ (The current domain name)
    - () (allow data to span lines)
    - * (wildcard character, `name` filed only)

The DNS Database – Resource Record (2)

- Types of resource record will be discussed later
    - Zone records: **identify domains and name servers**
        - **SOA**
        - **NS**
    - Basic records: **map names to addresses and route mails**
        - **A**
        - **AAAA**
        - **PTR**
        - **MX**
    - Optional records: **extra information to host or domain**
        - **CNAME**
        - **TXT**
        - **SRV**

The DNS Database – Resource Record (3)

	Type	Name	Function
Zone	SOA	Start Of Authority	Defines a DNS zone
	NS	Name Server	Identifies servers, delegates subdomains
Basic	A	IPv4 Address	Name-to-IPv4-address-translation
	AAAA	IPv6 Address	Name-to-IPv6-address-translation
	PTR	Pointer	Address-to-name translation
	MX	Mail Exchanger	Controls email routing
Security and DNSSEC	DS	Delegation Singer	Hash of singed child zone’s key-signing key
	DNSKEY	Public Key	Public key for a DNS name
	NSEC	Next Secure	Used with DNSSEC for negative answers
	NSEC3	Next Secure v3	Used with DNSSEC for negative answers
	RRSIG	Signature	Singed, authenticated resource record set
	DLV	Lookaside	Nonroot trust anchor for DNSSEC
	CAA	Certification Authority Authorization	Provide information for CA when validating an SSL certificate
	SSHFP	SSH Fingerprint	SSH host key, allows verification via DNS
	SPF	Sender Policy	Identifies mail servers, inhibits forging
	DKIM	Domain Keys	Verify email sender and message integrity
Optional	CNAME	Canonical Name	Nickname or aliases for a host
	SRV	Services	Gives locations for well-known services
	TXT	Text	Comments or untyped information

The DNS Database – Resource Record (4)

- SOA: Start Of Authority
  - Defines a DNS zone of authority, each zone has exactly one SOA record
  - Specify the name of the zone, the technical contact and various timeout information
  - Format
    - [zone] IN SOA [server-name] [administrator’s mail] (serial, refresh, retry, expire, ttl)
  - Ex:

;	means comments
@	means current domain name
( )	allow data to span lines
*	Wildcard character

```
$TTL 3600;
$ORIGIN cs.nctu.edu.tw.
@       IN      SOA    csns.cs.nctu.edu.tw.   root.cs.nctu.edu.tw. (
                       2012050802             ; serial number
                       1D                     ; refresh time for slave server
                       30M                    ; retry
                       1W                     ; expire
                       2H      )              ; minimum
```

The DNS Database – Resource Record (5)

- NS: Name Server
  - Format
    - zone [ttl] [IN] NS hostname
  - Usually follow the SOA record
  - Goal
    - Identify the authoritative server for a zone
    - Delegate subdomains to other organization’s NS

```
$TTL 3600;
$ORIGIN cs.nctu.edu.tw.
@       IN      SOA     dns.cs.nctu.edu.tw.            root.cs.nctu.edu.tw.    (
                                   2012050802          ; serial number
                                   1D                  ; refresh time for slave server
                                   30M                 ; retry
                                   1W                  ; expire
                                   2H      )           ; minimum
        IN      NS      dns.cs.nctu.edu.tw.
        IN      NS      dns2.cs.nctu.edu.tw.
test    IN      NS      dns.test.cs.nctu.edu.tw.       ; delegate test.$ORIGIN
```

The DNS Database – Resource Record (6)

- A record: Address
  - Format
    - hostname [ttl] [IN] A ip4addr
  - Provide mapping from hostname to IPv4 address(es)
  - Load balance (decided by client, not recommended)
  - Ex:

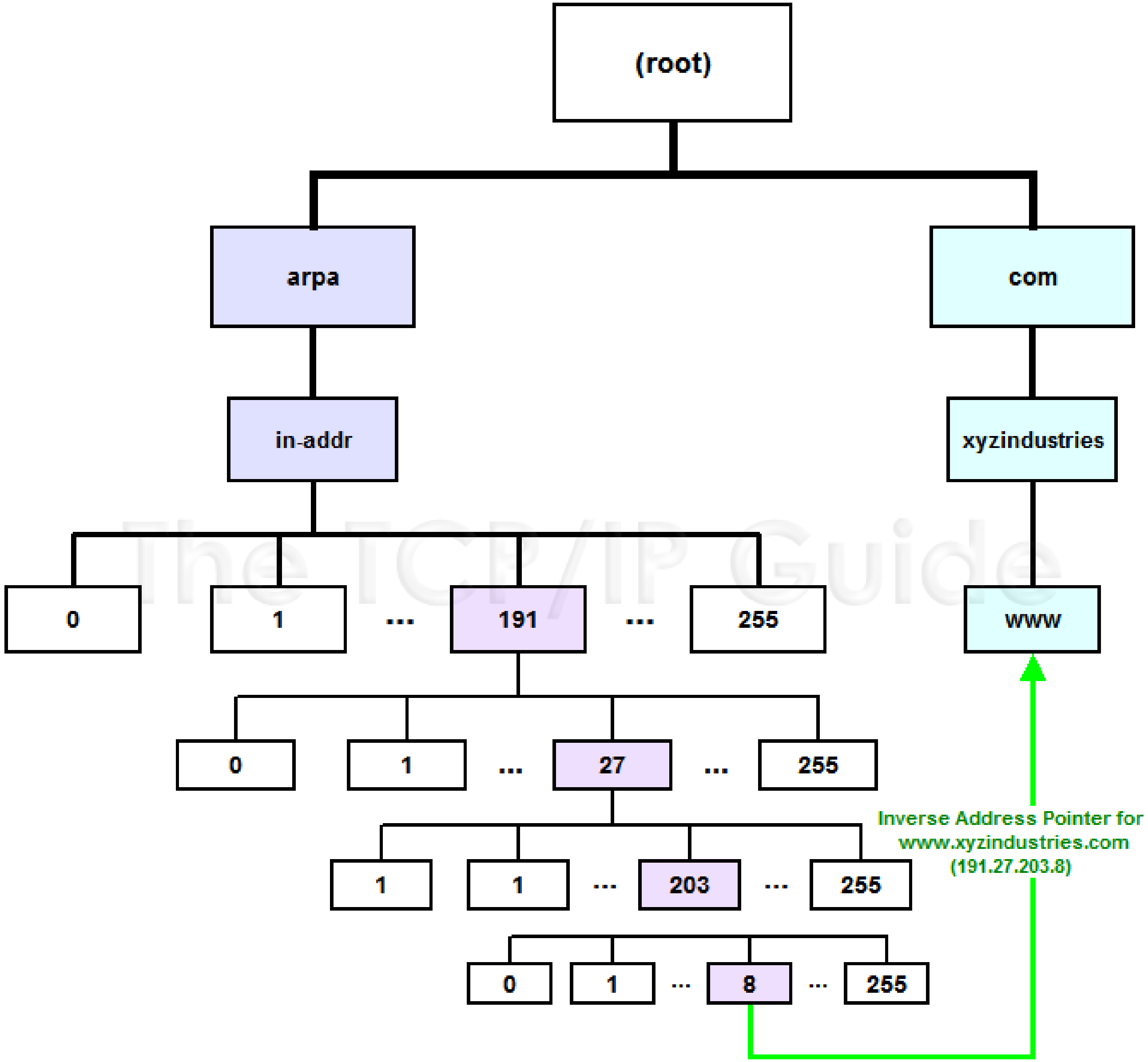
```
$ORIGIN cs.nctu.edu.tw.
@        IN      NS      dns.cs.nctu.edu.tw.
         IN      NS      dns2.cs.nctu.edu.tw.
dns      IN      A       140.113.235.107
dns2     IN      A       140.113.235.103

www      IN      A       140.113.235.111
www      IN      A       140.113.235.112
```

The DNS Database – Resource Record (7)

- PTR: Pointer
  - Perform the reverse mapping from IP address to hostname
  - Special top-level domain: in-addr.arpa
    - Used to create a naming tree from IP address to hostnames
  - Format
    - `addr [ttl] [IN] PTR hostname`

```
$TTL 259200;
$ORIGIN 235.113.140.in-addr.arpa.
@       IN      SOA     csns.cs.nctu.edu.tw.    root.cs.nctu.edu.tw.    (
                              2007052102        ; serial number
                              1D                ; refresh time for secondary server
                              30M               ; retry
                              1W                ; expire
                              2H)               ; minimum
        IN      NS      dns.cs.nctu.edu.tw.
        IN      NS      dns2.cs.nctu.edu.tw.
$ORIGIN in-addr.arpa.
103.235.113.140         IN PTR csmailgate.cs.nctu.edu.tw.
107.235.113.140         IN PTR csns.cs.nctu.edu.tw.
```

The DNS Database – Resource Record (8)



The TCP/IP Guide - DNS Reverse Name Resolution Using the IN-ADDR.ARPA Domain

The DNS Database – Resource Record (9)

- MX: Mail eXchanger
  - Direct mail to mail hubs rather than a single host
  - Format
    - `host [ttl] [IN] MX preference host`
    - No alias allowed
  - Ex:

```
$TTL 3600;
$ORIGIN cs.nctu.edu.tw.
@       IN      SOA     csns.cs.nctu.edu.tw.    root.cs.nctu.edu.tw.    (
                        2007052102              ; serial number
                        1D                      ; refresh time for slave server
                        30M                     ; retry
                        1W                      ; expire
                        2H      )               ; minimum
// ...
        7200    IN    MX    1 csmx1.cs.nctu.edu.tw.
        7200    IN    MX    5 csmx2.cs.nctu.edu.tw.

csmx1   IN      A       140.113.235.104
csmx2   IN      A       140.113.235.105
```

The DNS Database – Resource Record (10)

- CNAME: Canonical name
  - nickname [ttl] IN CNAME hostname
  - Add additional names to a host
    - To associate a function or to shorten a hostname
  - CNAME record can nest eight deep in BIND
  - NOT for load balance (use multiple A/AAAA instead)
    - Multiple CNAME records for one nickname is INVALID
  - Ex:

```
www             IN      A       140.113.209.63
                IN      A       140.113.209.77
penghu-club     IN      CNAME   www
King            IN      CNAME   www

R21601          IN      A       140.113.214.31
superman        IN      CNAME   r21601
```

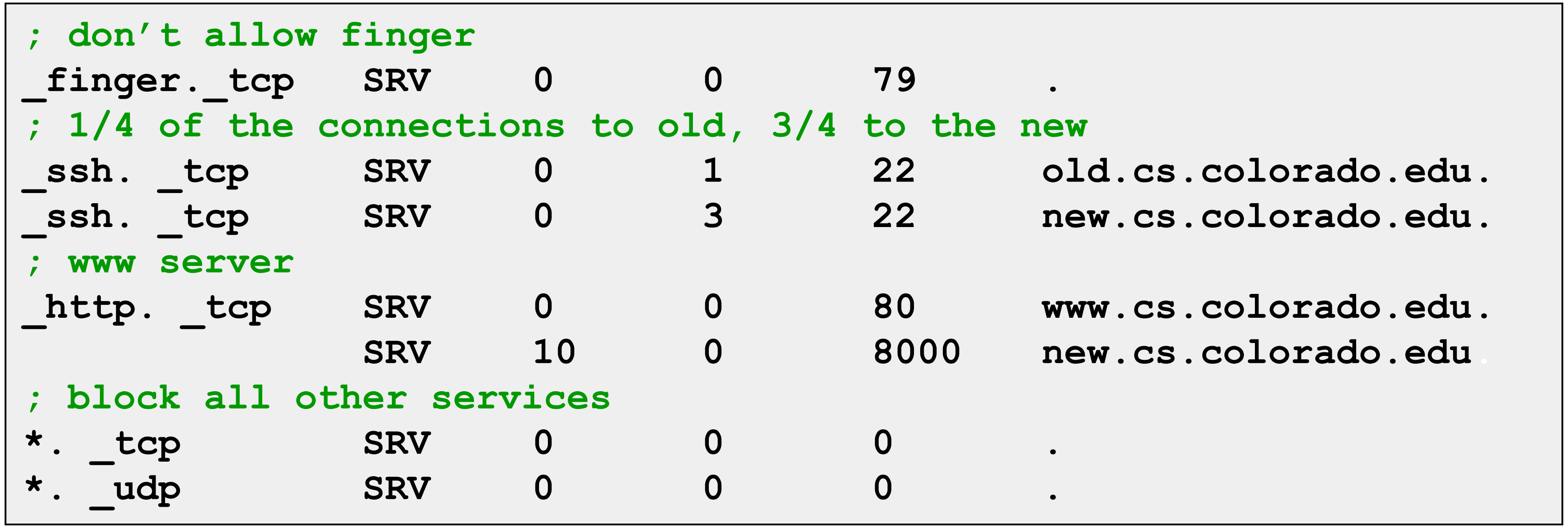
The DNS Database – Resource Record (11)

- TXT: Text
  - Add arbitrary text to a host’s DNS records
  - Format
    - `Name [ttl] [IN] TXT info`
    - All info items should be quoted
  - They are sometimes used to test prospective new types of DNS records
    - SPF records

```
$TTL 3600;
$ORIGIN cs.nctu.edu.tw.
@       IN      SOA     csns.cs.nctu.edu.tw.    root.cs.nctu.edu.tw.    (
                        2007052102              ; serial number
                        1D                      ; refresh time for slave server
                        30M                     ; retry
                        1W                      ; expire
                        2H      )               ; minimum
        IN      NS      dns.cs.nctu.edu.tw.
        IN      NS      dns2.cs.nctu.edu.tw.

        IN      TXT    "Department of Computer Science"
```

The DNS Database – Resource Record (12)

- SRV: Service
  - Specify the location of services within a domain
  - Format:
    - `_<service>._<proto>.name [ttl] IN SRV pri weight port target`
  - Needs application support (client side)
  - Ex:



```
; don’t allow finger
_finger._tcp   SRV     0      0      79     .
; 1/4 of the connections to old, 3/4 to the new
_ssh. _tcp     SRV     0      1      22     old.cs.colorado.edu.
_ssh. _tcp     SRV     0      3      22     new.cs.colorado.edu.
; www server
_http. _tcp    SRV     0      0      80     www.cs.colorado.edu.
               SRV     10     0      8000   new.cs.colorado.edu.
; block all other services
*. _tcp        SRV     0      0      0      .
*. _udp        SRV     0      0      0      .
```

IPv6 Resource Records

- IPv6 forward records
  - Format
    - `Hostname [ttl] [IN] AAAA ip6addr`
  - Example

```
$ dig f.root-servers.net AAAA

;; ANSWER SECTION:
f.root-servers.net.     604795  IN      AAAA    2001:500:2f::f
```

- IPv6 reverse records
  - IPv6 PTR records are in the ip6.arpa top-level domain
  - Example
    - f.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.f.2.0.0.0.0.5.0.1.0.0.2.ip6.arpa. PTR f.root-servers.net.

Glue Record (1/2)

- Glue record – Link between domains
  - DNS referrals occur only from parent domains to child domains
  - The servers of a parent domain must know the IP of the name servers for all of its subdomains
    - Parent zone needs to contain the NS records for each delegated zone
    - Making a normal DNS query
    - Having copies of the appropriate A records
    - The foreign A records are called glue records

```
; subdomain information
booklab           IN NS ns1.astust.com.
                  IN NS ubuntu.booklab.astust.com.
testlab           IN NS ns1.astust.com.
                  IN NS ns.testlab.astust.com.

; glue records
ubuntu.booklab    IN A  63.173.189.194
ns.testlab        IN A  63.173.189.17
```

Glue Record (2/2)

- There are two ways to link between zones
  - By including the necessary records directly
  - By using stub zone
    - Only contains SOA, NS, A (of NS)

- Lame delegation
  - DNS subdomain administration has delegate to you, but you never use the domain or parent domain’s glue record is not updated

Statements of named.conf

Examples of named configuration

```
// isc.org TLD name server

options {
        directory "/var/named";
        datasize 1000M;
        listen-on { 204.152.184.64; };
        listen-on-v6 { 2001:4f8:0:2::13; };
        recursion no;
        transfer-source 204.152.184.64;
        transfer-source-v6 2001:4f8:0:2::13;
};

zone "isc.org" {
        type master;
        file "master/isc.org";
        allow-update { none; };
        allow-transfer { none; };
};

zone "vix.com" {
        type slave;
        file "secondary/vix.com";
        masters { 204.152.188.234; };
};
```

```
$TTL 57600
$ORIGIN atrust.com.
@                SOA   ns1.atrust.com. trent.atrust.com. (
                         2010030400 10800 1200 3600000 3600 )
                 NS    NS1.atrust.com.
                 NS    NS2.atrust.com.
                 MX    10 mailserver.atrust.com.
                 A     66.77.122.161
ns1.atrust.com.  A     206.168.198.209
ns2.atrust.com.  A     66.77.122.161
www              A     66.77.122.161
mailserver       A     206.168.198.209
secure           A     66.77.122.161

; reverse maps
exterior1        A     206.168.198.209
209.198.168.206  PTR   exterior1.atrust.com.
exterior2        A     206.168.198.213
213.198.168.206  PTR   exterior2.atrust.com.
```

BIND Configuration
– named.conf address match list

- Address Match List
  - A generalization of an IP address that can include:
    - An IP address
      - Ex. 140.113.17.1
    - An IP network with CIDR netmask
      - Ex. 140.113/16
    - The name of a previously defined ACL
    - A cryptographic authentication key
    - The ! character to negate things
  - First match
  - Examples:
    - {!1.2.3.4; 1.2.3/24;};
    - {128.138/16; 198.11.16/24; 204.228.69/24; 127.0.0.1;};

BIND Configuration – named.conf acl

- The “acl” statement
  - Define a class of access control
  - Define before they are used
  - Syntax

```
acl acl_name {
    address_match_list
};
```

  - Predefined acl classes
    - any, localnets, localhost, none
  - Example

```
acl CSnets {
    140.113.235/24; 140.113.17/24; 140.113.209/24; 140.113.24/24;
};
acl NCTUnets {
    140.113/16; 10.113/16; 140.126.237/24;
};
allow-transfer {localhost; CSnets; NCTUnets};
```

BIND Configuration – named.conf key

- The “key” statement
  - Define an encryption key used for authentication with a particular server
  - Syntax

```
key key-id {
    algorithm string;
    secret string;
}
```

  - Example:

```
key serv1-serv2 {
    algorithm hmac-md5;
    secret "ibkAlUA0XXAXDxWRTGeY+d4CGbOgOIr7n63eizJFHQo="
}
```

  - This key is used to
    - Sign DNS request before sending to target
    - Validate DNS response after receiving from target

BIND Configuration – named.conf include

- The “include” statement
  - Used to separate large configuration file
  - Another usage is used to separate cryptographic keys into a restricted permission file
  - Ex:

```
include "/etc/namedb/rndc.key";

-rw-r--r--  1 root  wheel  4947 Mar  3  2006 named.conf
-rw-r-----  1 bind  wheel    92 Aug 15  2005 rndc.key
```

  - If the path is relative
    - Relative to the directory option

BIND Configuration
– named.conf option (1/3)

- The “option” statement
  - Specify global options
  - Some options may be overridden later for specific zone or server
  - Syntax:

```
options {
    option;
    option;
};
```

  - There are more than 150 options in BIND 9
    - version "There is no version.";   [real version num]
      - version.bind.   0   CH   TXT   "9.3.3"
      - version.bind.   0   CH   TXT   "There is no version."
    - directory "/etc/namedb/db";
      - Base directory for relative path and path to put zone data files

BIND Configuration – named.conf option (2/3)

- notify yes | no [yes]
  - Whether notify slave sever when relative zone data is changed
- also-notify {140.113.235.101;}; [empty]
  - Also notify this non-advertised NS server
- recursion yes | no [yes]
  - Recursive name server
  - Open resolver
- allow-recursion {address_match_list }; [all]
  - Finer granularity recursion setting
- recursive-clients number; [1000]
- max-cache-size number; [unlimited]
  - Limited memory

BIND Configuration – named.conf option (3/3)

- query-source address ip_addr port ip_port; [random]
  - NIC and port to send DNS query
  - DO NOT use port
- use-v4-udp-ports { range beg end; }; [range 1024 65535]
- avoid-v6-udp-ports { port_list }; [empty]
- forwarders {in_addr; …}; [empty]
  - Often used in cache name server
  - Forward DNS query if there is no answer in cache
- forward only | first; [first]
  - If forwarder does not response, queries for forward only server will fail
- allow-query { address_match_list }; [all]
  - Specify who can send DNS query to you
- allow-transfer address_match_list; [all]
  - Specify who can request zone transfer of your zone data
- allow-update address_match_list; [none]
- blackhole address_match_list; [empty]
  - Reject queries and would never ask them for answers

BIND Configuration
– named.conf zone (1/5)

- The “zone” statement
  - Heart of the named.conf that tells named about the zones that it is authoritative
  - zone statement format varies depending on roles of named
    - master, slave, hint, forward, stub
  - The zone file is just a collection of DNS resource records
  - Basically

```
Syntax:
zone "domain_name" {
        type master | slave| stub;
        file "path”;
        masters {ip_addr; ip_addr;};
        allow-query {address_match_list};           [all]
        allow-transfer { address_match_list};       [all]
        allow-update {address_match_list};          [empty]
};
```

allow-update cannot be used for a slave zone

BIND Configuration
– named.conf zone (2/5)

- Master server zone configuration

```
zone "cs.nctu.edu.tw" IN {
    type master;
    file "named.hosts";
    allow-query { any; };
    allow-transfer { localhost; CS-DNS-Servers; };
    allow-update { none; };
};
```

- Slave server zone configuration

```
zone "cs.nctu.edu.tw" IN {
    type slave;
    file "cs.hosts";
    masters { 140.113.235.107; };
    allow-query { any; };
    allow-transfer { localhost; CS-DNS-Servers; };
};
```

BIND Configuration
– named.conf zone (3/5)

- Forward zone and reverse zone

```
zone "cs.nctu.edu.tw" IN {
    type forward;
    forwarders { CS-DNS-Servers; };
    allow-query { any; };
};
```

```
zone "235.113.140.in-addr.arpa" IN {
    type master;
    file "named.235.rev";
    allow-query { any; };
    allow-transfer { localhost; CS-DNS-Servers; };
    allow-update { none; };
};
```

BIND Configuration
– named.conf zone (4/5)

- Example
  - In named.hosts, there are plenty of A or CNAME records

```
…
bsd1            IN      A       140.113.235.131
csbsd1          IN      CNAME   bsd1
bsd2            IN      A       140.113.235.132
bsd3            IN      A       140.113.235.133
bsd4            IN      A       140.113.235.134
bsd5            IN      A       140.113.235.135
…
```

- In named.235.rev, there are plenty of PTR records

```
…
131.235.113.140    IN    PTR   bsd1.cs.nctu.edu.tw.
132.235.113.140    IN    PTR   bsd2.cs.nctu.edu.tw.
133.235.113.140    IN    PTR   bsd3.cs.nctu.edu.tw.
134.235.113.140    IN    PTR   bsd4.cs.nctu.edu.tw.
135.235.113.140    IN    PTR   bsd5.cs.nctu.edu.tw.
…
```

BIND Configuration
– named.conf zone (5/5)

- Setting up root hint
  - ○ A cache of where are the DNS root servers

```
zone "." IN {
    type hint;
    file "named.root";
};
```

- Setting up forwarding zone
  - ○ Forward DNS query to specific name server, bypassing the standard query path

```
zone "nctu.edu.tw" IN {
    type forward;
    forward first;
    forwarders { 140.113.250.135; 140.113.1.1; };
};

zone "113.140.in-addr.arpa" IN {
    type forward;
    forward first;
    forwarders { 140.113.250.135; 140.113.1.1; };
};
```

BIND Configuration – named.conf server

- The “server” statement
  - Tell named about the characteristics of its remote peers
  - Syntax

```
server ip_addr {
    bogus no|yes;
    provide-ixfr yes|no;     (for master)
    request-ixfr yes|no;     (for slave)
    transfer-format many-answers|one-answer;
    keys { key-id; key-id};
};
```

  - ixfr
    - Incremental zone transfer
  - transfers
    - Limit of number of concurrent inbound zone transfers from that server
    - Server-specific transfers-in
  - keys
    - Any request sent to the remote server is signed with this key

BIND Configuration – named.conf view (1/2)

- The “view” statement
  - Create a different view of DNS naming hierarchy for internal machines
    - Restrict the external view to few well-known servers
    - Supply additional records to internal users
  - Also called “split DNS”
  - In-order processing
    - Put the most restrictive view first
  - All-or-nothing
    - All zone statements in your named.conf file must appear in the content of view

BIND Configuration – named.conf view (2/2)

- Syntax

```
view view-name {
        match_clients {address_match_list};
        view_options;
        zone_statement;
};
```

- Example

```
view "internal" {
        match-clients {our_nets;};
        recursion yes;
        zone "cs.nctu.edu.tw" {
         type master;
         file "named-internal-cs";
        };
};
view "external" {
 match-clients {any;};
        recursion no;
        zone "cs.nctu.edu.tw" {
         type master;
         file "named-external-cs";
        };
};
```

BIND Configuration – named.conf controls

- The “controls” statement
  - Limit the interaction between the running named process and rndc
  - Syntax

```
controls {
      inet ip_addr port ip-port allow {address_match_list} keys {key-id};
};
```

  - Example:

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

```
key "rndc_key" {
    algorithm       hmac-md5;
    secret "GKnELuie/G99NpOC2/AXwA==";
};
```

BIND Configuration – rndc

- RNDC – remote name daemon control
    - reload, restart, status, dumpdb, …..
    - rndc-confgen -b 256

```
# Start of rndc.conf
key "rndc-key" {
        algorithm hmac-md5;
        secret "qOfQFtH1nvdRmTn6gLXldm6lqRJBEDbeK43R8Om7wlg=";
};

options {
        default-key "rndc-key";
        default-server 127.0.0.1;
        default-port 953;
};
# End of rndc.conf
```

```
SYNOPSIS
       rndc [-c config-file] [-k key-file] [-s server] [-p port] [-V]
            [-y key_id] {command}
```

Updating zone files

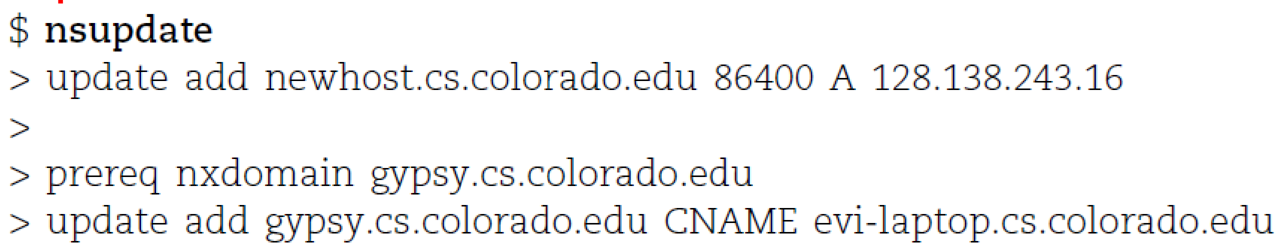
- Master
  - Edit zone files
    - Serial number
    - Forward and reverse zone files for single IP
  - Do “rndc reload”
    - “notify” is on, slave will be notified about the change
    - “notify” is off, refresh timeout, or do “rndc reload” in slave
- Zone transfer
  - DNS zone data synchronization between master and slave servers
  - AXFR (all zone data are transferred at once, before BIND8.2)
  - IXFR (incremental updates zone transfer)
    - provide-ixfr
    - request-ixfr
  - TCP port 53

Dynamic Updates

- The mappings of name-to-address are relatively stable
- DHCP will dynamically assign IP addresses to the hosts
  - Hostname-based logging or security measures become very difficult

```
dhcp-host1.domain              IN      A       192.168.0.1
dhcp-host2.domain              IN      A       192.168.0.2
```

- Dynamic updates
  - RFC 2136
  - BIND allows the DHCP daemon to notify the updating RR contents
  - nsupdate



```
$ nsupdate
> update add newhost.cs.colorado.edu 86400 A 128.138.243.16
>
> prereq nxdomain gypsy.cs.colorado.edu
> update add gypsy.cs.colorado.edu CNAME evi-laptop.cs.colorado.edu
```

  - Using allow-update, or allow-policy
    - rndc frozen zone, rndc thaw zone
    - allow-policy (grant | deny) identity nametype name [types]

Non-byte boundary (1/5)

- In normal reverse configuration:
  - named.conf defines a zone statement for each reverse subnet zone and
  - Your reverse db contains lots of PTR records
  - Example:

```
$TTL    3600
$ORIGIN 1.168.192.in-addr.arpa.
@       IN      SOA     chwong.csie.net chwong.chwong.csie.net.  (
                        2007050401      ; Serial
                        3600            ; Refresh
                        900             ; Retry
                        7D              ; Expire
                        2H )            ; Minimum
        IN      NS      ns.chwong.csie.net.
254     IN      PTR     ns.chwong.csie.net.
1       IN      PTR     www.chwong.csie.net.
2       IN      PTR     ftp.chwong.csie.net.
…
```

```
zone "1.168.192.in-addr.arpa." {
    type master;
    file "named.rev.1";
    allow-query {any;};
    allow-update {none;};
    allow-transfer {localhost;};
};
```

Non-byte boundary (2/5)

- What if you want to delegate 192.168.2.0 to another sub-domain
  - Parent
    - **Remove** forward db about 192.168.2.0/24 network
    - Ex:
      - pc1.chwong.csie.net.   IN   A   192.168.2.35
      - pc2.chwong.csie.net.   IN   A   192.168.2.222
      - …
    - **Remove** reverse db about 2.168.192.in-addr.arpa
      - Ex:
        - 35.2.168.192.in-addr.arpa.   IN   PTR   pc1.chwong.csie.net.
        - 222.2.168.192.in-addr.arpa. IN   PTR   pc2.chwong.csie.net.
        - …
    - Add glue records about the name servers of sub-domain
      - Ex: in zone db of "chwong.csie.net"
        - sub1 IN   NS   ns.sub1.chwong.csie.net.
        - ns.sub1   IN   A   192.168.2.1
      - Ex: in zone db of "168.192.in-addr.arpa."
        - 2   IN   NS   ns.sub1.chwong.csie.net.
        - 1.2   IN   PTR   ns.sub1.chwong.csie.net

Non-byte boundary (3/5)

- What if you want to delegate 192.168.3.0 to four sub-domains (a /26 network)
  - 192.168.3.0 ~ 192.168.3.63
    - ns.sub1.chwong.csie.net.
  - 192.168.3.64 ~ 192.168.3.127
    - ns.sub2.chwong.csie.net.
  - 192.168.3.128 ~ 192.168.3.191
    - ns.sub3.chwong.csie.net.
  - 192.168.3.192 ~ 192.168.3.255
    - ns.sub4.chwong.csie.net.
- It is easy for forward setting
  - In zone db of chwong.csie.net
    - sub1 IN NS ns.sub1.chwong.csie.net.
    - ns.sub1 IN A 192.168.3.1
    - sub2 IN NS ns.sub2.chwong.csie.net.
    - ns.sub2 IN A 192.168.3.65
    - …

Non-byte boundary (4/5)

- Non-byte boundary reverse setting
  - Method1

```
$GENERATE 0-63      $.3.168.192.in-addr.arpa.   IN   NS ns.sub1.chwong.csie.net.
$GENERATE 64-127    $.3.168.192.in-addr.arpa.   IN   NS ns.sub2.chwong.csie.net.
$GENERATE 128-191   $.3.168.192.in-addr.arpa.   IN   NS ns.sub3.chwong.csie.net.
$GENERATE 192-255   $.3.168.192.in-addr.arpa.   IN   NS ns.sub4.chwong.csie.net.
```

And

```
zone "1.3.168.192.in-addr.arpa. " {
    type master;
    file "named.rev.192.168.3.1";
};

; named.rev.192.168.3.1
@     IN     SOA sub1.chwong.csie.net. root.sub1.chwong.csie.net. (1;3h;1h;1w;1h)
      IN     NS  ns.sub1.chwong.csie.net.
```

Non-byte boundary (5/5)

- Method2

```
$ORIGIN  3.168.192.in-addr.arpa.
$GENERATE  1-63                 $        IN   CNAME     $.0-63.3.168.192.in-addr.arpa.
0-63.3.168.192.in-addr.arpa.             IN   NS        ns.sub1.chwong.csie.net.
$GENERATE  65-127               $        IN   CNAME     $.64-127.3.168.192.in-addr.arpa.
64-127.3.168.192.in-addr.arpa.           IN   NS        ns.sub2.chwong.csie.net.
$GENERATE  129-191              $        IN   CNAME     $.128-191.3.168.192.in-addr.arpa.
128-191.3.168.192.in-addr.arpa.          IN   NS        ns.sub3.chwong.csie.net.
$GENERATE  193-255              $        IN   CNAME     $.192-255.3.168.192.in-addr.arpa.
192-255.3.168.192.in-addr.arpa.          IN   NS        ns.sub4.chwong.csie.net.

zone "0-63.3.168.192.in-addr.arpa." {
    type master;
    file “named.rev.192.168.3.0-63”;
};
```

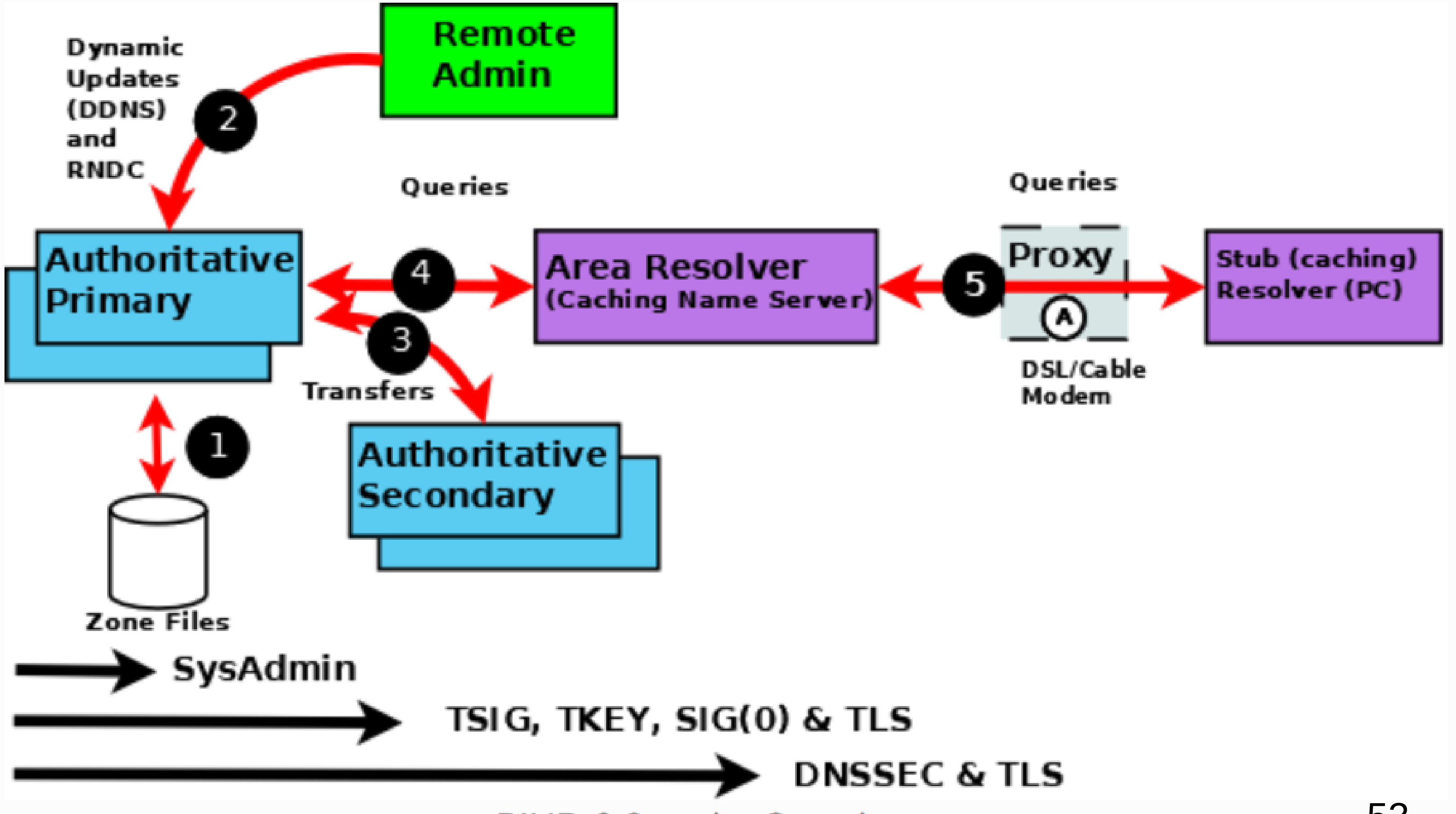
```
; named.rev.192.168.3.0-63
@   IN     SOA     sub1.chwong.csie.net. root.sub1.chwong.csie.net. (1;3h;1h;1w;1h)
    IN     NS      ns.sub1.chwong.csie.net.
1   IN     PTR     www.sub1.chwong.csie.net.
    IN     PTR     abc.sub1.chwong.csie.net.
…
```

BIND Security

Source: https://downloads.isc.org/isc/bind9/9.18.25/doc/arm/html/chapter1.html#dns-security-overview

Security Consideration

1. System administration techniques: file permission, ACL, jail, etc.
2. rndc / nsupdate uses TSIG or SIG(0) cryptographic methods.
3. Zone transfer can be secured through TSIG or TLS.
4. DNSSEC is the only solution.
5. DNS over TLS may be configured.



BIND 9 Security Overview

Security
– named.conf security configuration

- Security configuration

Feature	Config. Statement	comment
allow-query	options, zone	Who can query
allow-transfer	options, zone	Who can request zone transfer
allow-update	zone	Who can make dynamic updates
blackhole	options	Which server to completely ignore
bogus	server	Which servers should never be queried

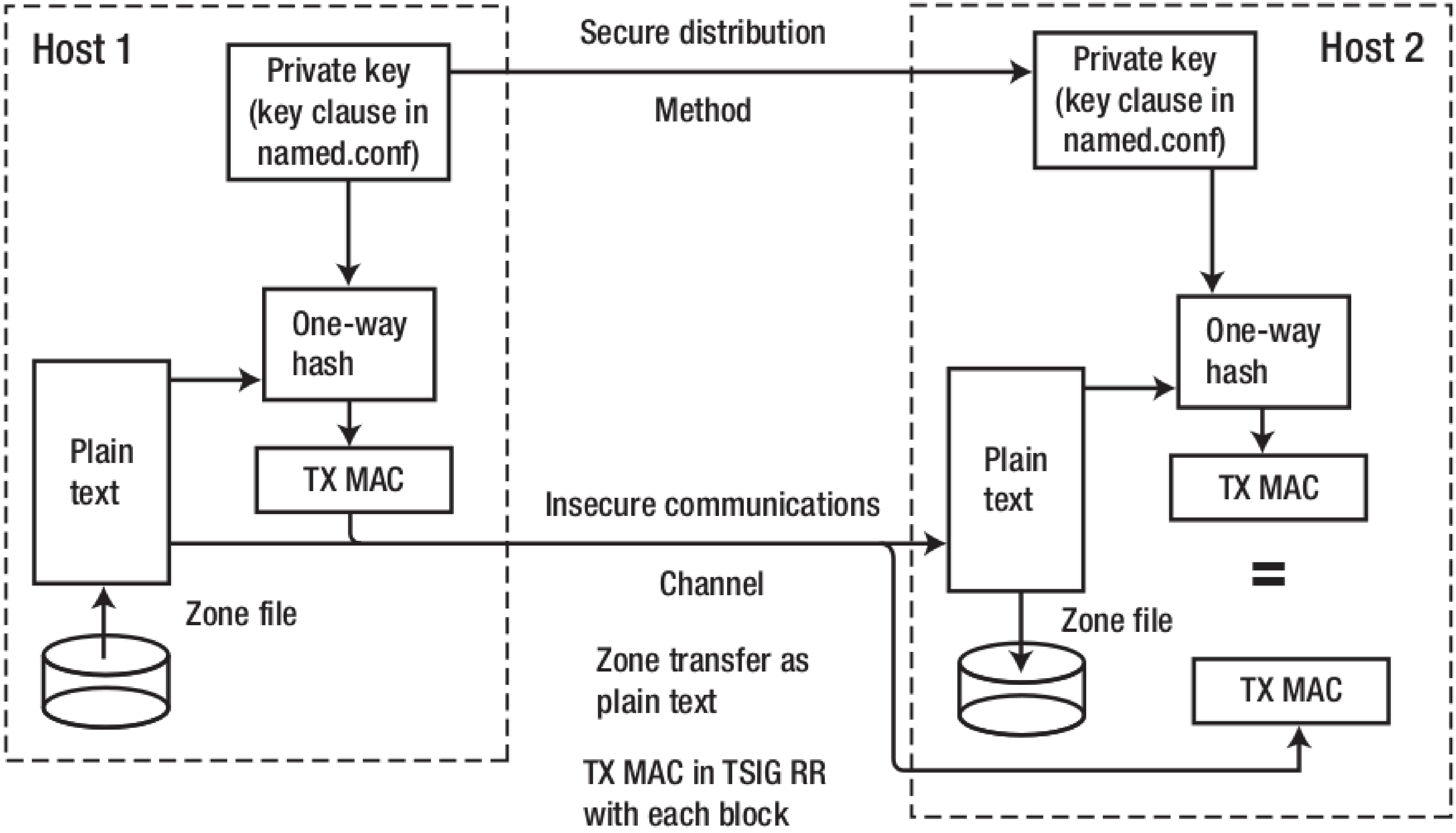
```
acl bogusnet {
        0.0.0.0/8 ;      // Default, wild card addresses
        1.0.0.0/8 ;      // Reserved addresses
        2.0.0.0/8 ;      // Reserved addresses
        169.254.0.0/16 ; // Link-local delegated addresses
        192.0.2.0/24 ;   // Sample addresses, like example.com
        224.0.0.0/3 ;    // Multicast address space
        10.0.0.0/8 ;     // Private address space (RFC1918)25
        172.16.0.0/12 ;  // Private address space (RFC1918)
        192.168.0.0/16 ; // Private address space (RFC1918)
};
```

```
allow-recursion { ournets; };
blackhole { bogusnet; };
allow-transfer { myslaves; };
```

Security – With TSIG (1)

- TSIG (Transaction SIGnature)
  - Developed by IETF (RFC2845)
  - Symmetric encryption scheme to sign and validate DNS requests and responses between servers
  - Algorithm in BIND9
    - DH (Diffie Hellman), HMAC-MD5, HMAC-SHA1, HMAC-SHA224, HMAC-SHA256, HMAC-SHA384, HMAC-SHA512
  - Usage
    - Prepare the shared key with dnssec-keygen
    - Edit “key” statement
    - Edit “server” statement to use that key
    - Edit “zone” statement to use that key with:
      - allow-query
      - allow-transfer
      - allow-update

Security – With TSIG (2)



Security – With TSIG (3)

- TSIG example (dns1 with dns2)

1. % dnssec-keygen -a HMAC-MD5 -b 128 -n HOST cs

```
% dnssec-keygen -a HMAC-MD5 -b 128 -n HOST cs
Kcs.+157+35993
% cat Kcs.+157+35993.key
cs. IN DNSKEY 512 3 157 oQRab/QqXHVhkyXi9uu8hg==
```

```
% cat Kcs.+157+35993.private
Private-key-format: v1.2
Algorithm: 157 (HMAC_MD5)
Key: oQRab/QqXHVhkyXi9uu8hg==
```

2. Edit /etc/named/dns1-dns2.key

```
key dns1-dns2 {
    algorithm hmac-md5;
    secret “oQRab/QqXHVhkyXi9uu8hg==”
};
```

3. Edit both named.conf of dns1 and dns2
   - Suppose dns1 = 140.113.235.107 dns2 = 140.113.235.103

```
include "dns1-dns2.key"
server 140.113.235.103 {
    keys {dns1-dns2;};
};
```

```
include "dns1-dns2.key"
server 140.113.235.107 {
    keys {dns1-dns2;};
};
```

Security – With DNSSEC (1)

- DNSSEC (Domain Name System SECurity Extensions)
  - Using public-key cryptography (asymmetric)
  - Follow the delegation of authority model
  - Provide data authenticity and integrity
    - Signing the RRsets with private key
    - Public DNSKEYs are published, used to verify RRSIGs
    - Children sign their zones with private key
      - The private key is authenticated by parent’s signing hash (DS) of the child zone’s key

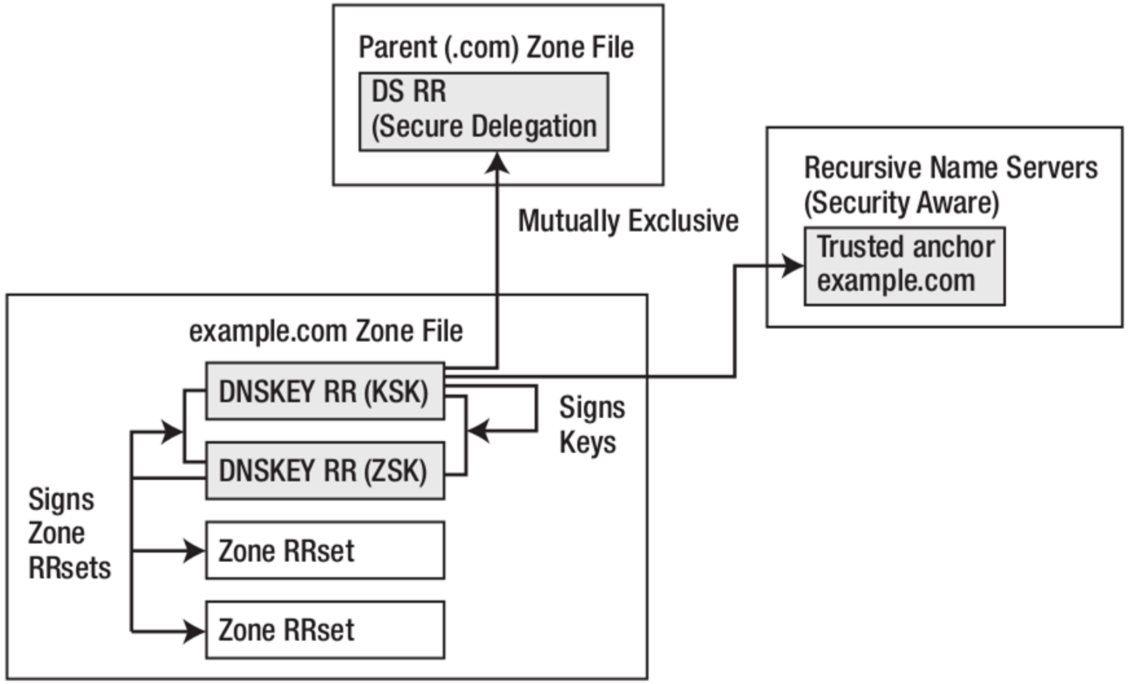
RRset: Resource Record Set
RRSIG: Resource Record Signature
DS: Delegation of Signing

Security – With DNSSEC (2)

- Types of Resource Record for DNSSEC
  - RRSIG (Resource Record Signature)
    - Crypto signatures for A, AAAA, NS, etc.
    - Tracks the type and number at each node.
  - NSEC (Next Secure)/NSEC3
    - Confirms the NXDOMAIN response
  - DNSKEY
    - Public keys for the entire zone
    - Private side is used generate RRSIGs
  - DS (Delegation Signer) Record
    - Handed up to parent zone to authenticate the NS record

Security – With DNSSEC (3)

- KSK (Key Signing Key)
  - The private key is used to generate a digital signature for the ZSK
  - The public key is stored in the DNS to be used to authenticate the ZSK
- ZSK (Zone Signing Key)
  - The private key is used to generate a digital signature (RRSIG) for each RRset in a zone
  - The public key is stored in the DNS to authenticate an RRSIG



BIND Debugging and Logging

Logging (1)

- Logging configuration
  - Using a *logging* statement
  - Define what are the channels
  - Specify where each message category should go
- Terms
  - Channel
    - A place where messages can go
    - Ex: syslog, file or /dev/null
  - Category
    - A class of messages that named can generate
    - Ex: answering queries or dynamic updates
  - Module
    - The name of the source module that generates the message
  - Facility
    - syslog facility name
  - Severity
    - Priority in syslog
- When a message is generated
  - It is assigned a “category”, a “module”, a “severity”
  - It is distributed to all channels associated with its category

Logging (2)

- Channels
  - Either "file" or "syslog" in channel sub-statement
    - size:
      - ex: 2048, 100k, 20m, 15g, unlimited, default
    - facility:
      - Daemon and local0 ~ local7 are reasonable choices
    - severity:
      - critical, error, warning, notice, info, debug (with an optional numeric level), dynamic
      - Dynamic is recognized and matches the server’s current debug level

```
logging {
    channel_def;
    channel_def;
    …
    category category_name {
         channel_name;
        channel_name;
         …
    };
};
```

```
channel channel_name {
    file path [versions num|unlimited] [size siznum];
    syslog facility;

    severity severity;
    print-category yes|no;
    print-severity yes|no;
    print-time yes|no;
};
```

Logging (3)

- Predefined channels

default_syslog	Sends severity info and higher to syslog with facility daemon
default_debug	Logs to file “named.run”, severity set to dynamic
default_stderr	Sends messages to stderr or named, severity info
null	Discards all messages

- Available categories

default	Categories with no explicit channel assignment
general	Unclassified messages
config	Configuration file parsing and processing
queries/client	A short log message for every query the server receives
dnssec	DNSSEC messages
update	Messages about dynamic updates
xfer-in/xfer-out	zone transfers that the server is receiving/sending
db/database	Messages about database operations
notify	Messages about the “zone changed” notification protocol
security	Approved/unapproved requests
resolver	Recursive lookups for clients

Logging (4)

- Example of logging statement

```
logging {
    channel security-log {
        file "/var/named/security.log" versions 5 size 10m;
        severity info;
        print-severity yes;
        print-time yes;
    };
    channel query-log {
        file "/var/named/query.log" versions 20 size 50m;
        severity info;
        print-severity yes;
        print-time yes;
    };
    category default        { default_syslog; default_debug; };
    category general        { default_syslog; };
    category security       { security-log; };
    category client         { query-log; };
    category queries        { query-log; };
    category dnssec         { security-log; };
};
```

Debug

- Named debug level
  - From 0 (debugging off) ~ 11 (most verbose output)
  - % named -d2 (start named at level 2)
  - % rndc trace (increase debugging level by 1)
  - % rndc trace 3 (change debugging level to 3)
  - % rndc notrace (turn off debugging)
- Debug with “logging” statement
  - Define a channel that include a severity with “debug” keyword
    - Ex: severity debug 3
    - All debugging messages up to level 3 will be sent to that particular channel

Tools

Tools – nslookup

- Interactive and Non-interactive
  - Non-Interactive
    - $ nslookup cs.nctu.edu.tw.
    - $ nslookup -type=mx cs.nctu.edu.tw.
    - $ nslookup -type=ns cs.nctu.edu.tw. 140.113.1.1
  - Interactive
    - $ nslookup
    - > set all
    - > set type=any
    - > server host
    - > lserver host
    - > set debug
    - > set d2

```
$ nslookup
> set all
Default server: 140.113.235.107
Address: 140.113.235.107#53
Default server: 140.113.235.103
Address: 140.113.235.103#53

Set options:
  novc                      nodebug          nod2
  search                    recurse
  timeout = 0               retry = 3        port = 53
  querytype = A             class = IN
  srchlist = cs.nctu.edu.tw/csie.nctu.edu.tw
>
```

Tools – host

- host command
  - $ host cs.nctu.edu.tw.
  - $ host -t mx cs.nctu.edu.tw.
  - $ host 140.113.1.1
  - $ host -v 140.113.1.1

Tools – dig

- Usage
  - $ dig cs.nctu.edu.tw
  - $ dig cs.nctu.edu.tw mx
  - $ dig @ns.nctu.edu.tw cs.nctu.edu.tw mx
  - $ dig -x 140.113.209.3
    - Reverse query
- Find out the root servers
  - $ dig @a.root-servers.net . ns
- drill

Tools – drill

- Usage
  - $ drill cs.nctu.edu.tw
  - $ drill cs.nctu.edu.tw mx
  - $ drill @ns.nctu.edu.tw cs.nctu.edu.tw mx
  - $ drill -x 140.113.209.3
- DNSSEC (-D) & Trace (-T)
  - `$ drill -DT www.cs.nctu.edu.tw`

References

- BIND 9 Administrator Reference Manual
  - https://downloads.isc.org/isc/bind9/9.18.25/doc/arm/html/
- ISC Concludes BIND 10 Development with Release 1.2
  - https://www.isc.org/blogs/isc-concludes-bind-10-development-with-release-1-2-project-renamed-bundy/
- A Short History of Chaosnet
  - https://twobithistory.org/2018/09/30/chaosnet.html
  - https://linux.cn/article-10674-1.html
- 用 dig 查瑞士的 top domain 剛好會遇到的 "feature"
  - https://blog.gslin.org/archives/tag/chaosnet/

Appendix

Security – Configuring DNSSEC (1)

- Creating DNS Keys for a Zone
  - Generate KSK (Key signing key)

    ```
    $ dnssec-keygen -a RSASHA256 -b 2048 -f KSK -n ZONE example.com
    Kexample.com.+008+34957
    ```

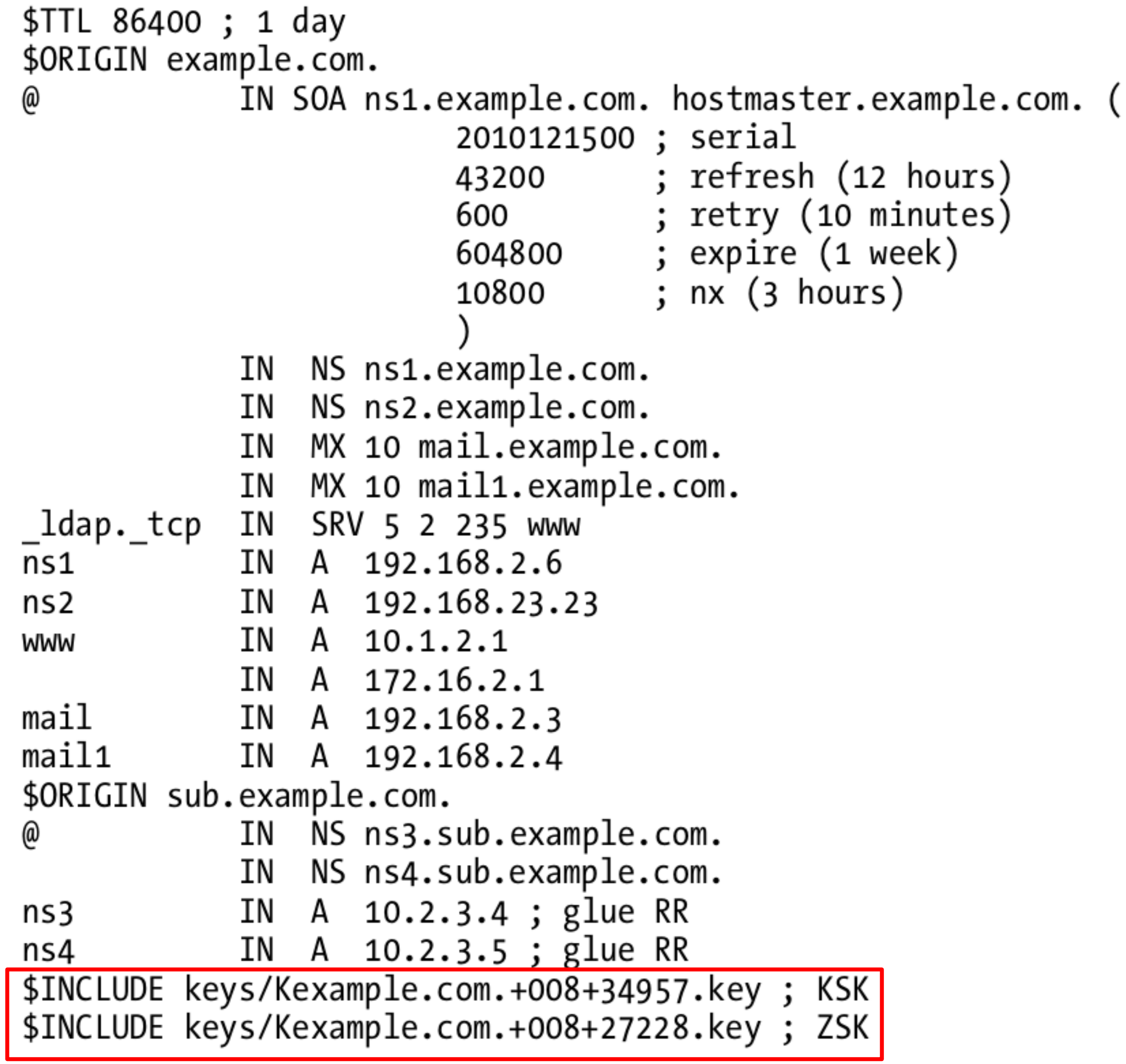
  - Generate ZSK (Zone signing key)

    ```
    $ dnssec-keygen -a RSASHA256 -b 2048 -n ZONE example.com
    Kexample.com.+008+27228
    ```

  - -P : publish
  - -A : activate
  - -I : inactive
  - -D : delete
  - YYYYMMDDHHMMSS (GMT timezone)

Security – Configuring DNSSEC (2)

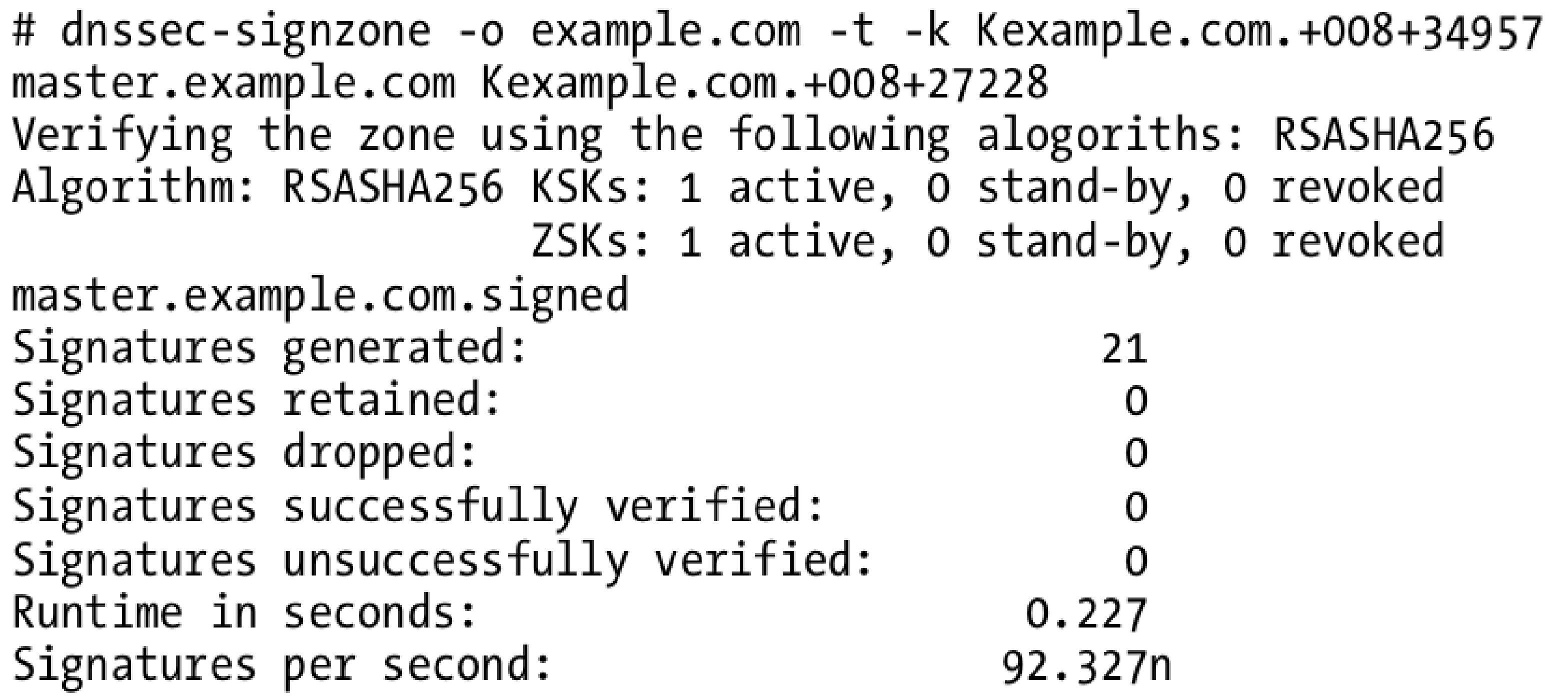
- Publishing DNS Keys (public keys) in a Zone



```
$TTL 86400 ; 1 day
$ORIGIN example.com.
@           IN SOA ns1.example.com. hostmaster.example.com. (
                        2010121500 ; serial
                        43200      ; refresh (12 hours)
                        600        ; retry (10 minutes)
                        604800     ; expire (1 week)
                        10800      ; nx (3 hours)
                        )
            IN  NS ns1.example.com.
            IN  NS ns2.example.com.
            IN  MX 10 mail.example.com.
            IN  MX 10 mail1.example.com.
_ldap._tcp  IN  SRV 5 2 235 www
ns1         IN  A  192.168.2.6
ns2         IN  A  192.168.23.23
www         IN  A  10.1.2.1
            IN  A  172.16.2.1
mail        IN  A  192.168.2.3
mail1       IN  A  192.168.2.4
$ORIGIN sub.example.com.
@           IN  NS ns3.sub.example.com.
            IN  NS ns4.sub.example.com.
ns3         IN  A  10.2.3.4 ; glue RR
ns4         IN  A  10.2.3.5 ; glue RR
$INCLUDE keys/Kexample.com.+008+34957.key ; KSK
$INCLUDE keys/Kexample.com.+008+27228.key ; ZSK
```

Security – Configuring DNSSEC (3)

- Signing a Zone

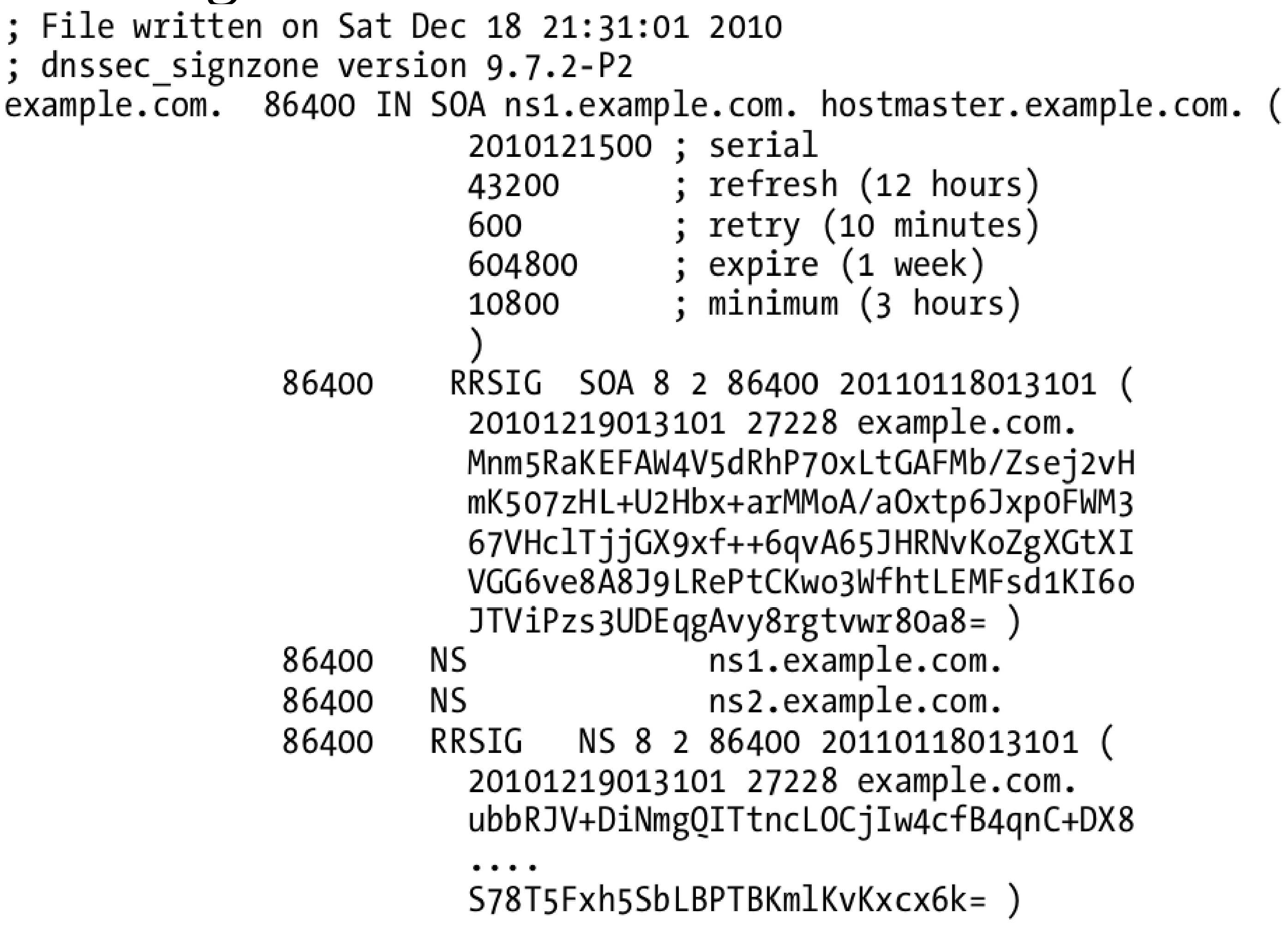


```
# dnssec-signzone -o example.com -t -k Kexample.com.+008+34957
master.example.com Kexample.com.+008+27228
Verifying the zone using the following alogoriths: RSASHA256
Algorithm: RSASHA256 KSKs: 1 active, 0 stand-by, 0 revoked
                     ZSKs: 1 active, 0 stand-by, 0 revoked
master.example.com.signed
Signatures generated:                       21
Signatures retained:                         0
Signatures dropped:                          0
Signatures successfully verified:            0
Signatures unsuccessfully verified:          0
Runtime in seconds:                      0.227
Signatures per second:                  92.327n
```

  - When signing the zone with only ZSK, just omit the -k parameter

Security – Configuring DNSSEC (4)

- Signing a Zone (Cont.)
  - example.com.signed



```
; File written on Sat Dec 18 21:31:01 2010
; dnssec_signzone version 9.7.2-P2
example.com.  86400 IN SOA ns1.example.com. hostmaster.example.com. (
                         2010121500 ; serial
                         43200      ; refresh (12 hours)
                         600        ; retry (10 minutes)
                         604800     ; expire (1 week)
                         10800      ; minimum (3 hours)
                         )
               86400    RRSIG  SOA 8 2 86400 20110118013101 (
                         20101219013101 27228 example.com.
                         Mnm5RaKEFAW4V5dRhP7OxLtGAFMb/Zsej2vH
                         mK5O7zHL+U2Hbx+arMMoA/aOxtp6JxpOFWM3
                         67VHclTjjGX9xf++6qvA65JHRNvKoZgXGtXI
                         VGG6ve8A8J9LRePtCKwo3WfhtLEMFsd1KI6o
                         JTViPzs3UDEqgAvy8rgtvwr8Oa8= )
               86400   NS             ns1.example.com.
               86400   NS             ns2.example.com.
               86400   RRSIG   NS 8 2 86400 20110118013101 (
                         20101219013101 27228 example.com.
                         ubbRJV+DiNmgQITtncLOCjIw4cfB4qnC+DX8
                         ....
                         S78T5Fxh5SbLBPTBKmlKvKxcx6k= )
```

Security – Configuring DNSSEC (5)

- Updating the Zone file
    - Edit the zone file

```
zone "example.com" {
        type master;
        file "example.com.signed";
        masters {ip_addr; ip_addr;};
        allow-query {address_match_list};
        allow-transfer { address_match_list};
        allow-update {address_match_list};
};
```

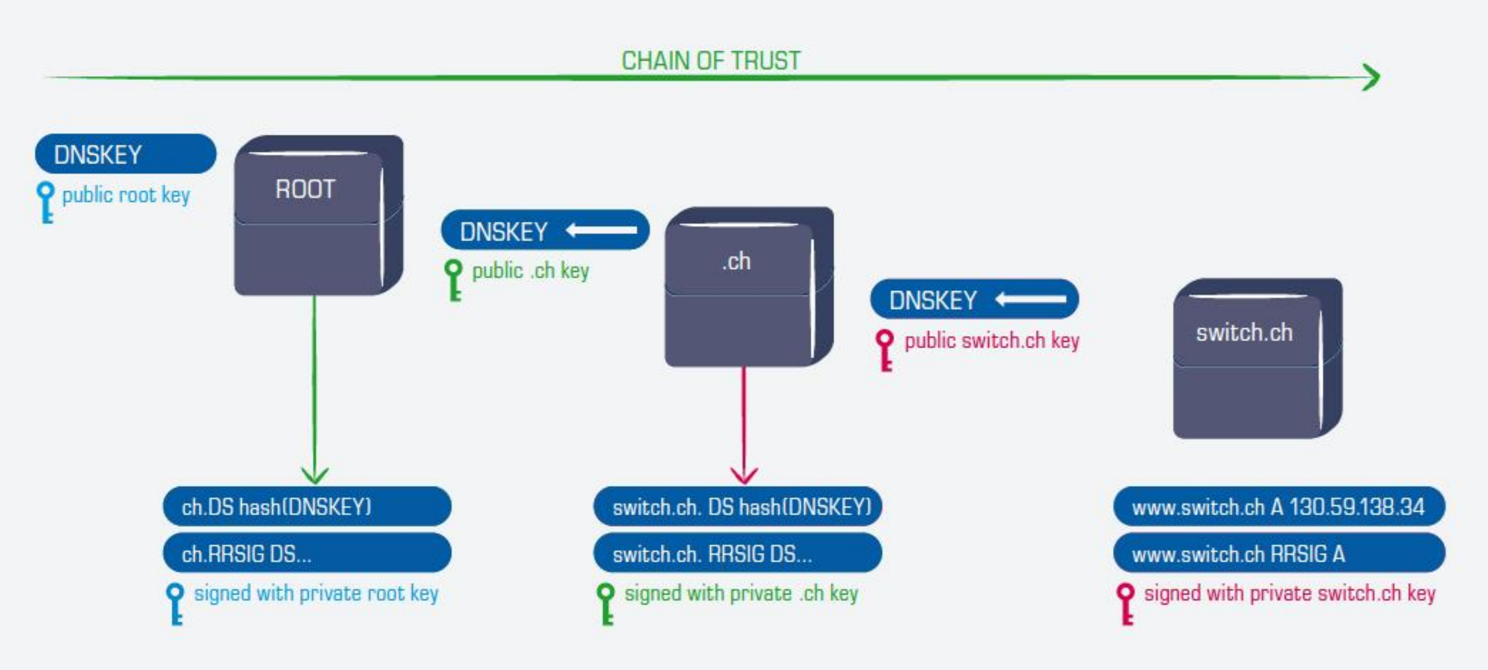
- Load the new zone file
    - rndc reload

Security – Configuring DNSSEC (6)

- Create Chain of Trust
  - Extract DNSKEY RR and use dnssec-dsfromkey
  - Add -g parameter when signing zone using dnssec-signzone

```
$ dnssec-signzone -g …
```

    - A file named ds-set.example.com was also created, which contains DS record
    - DS records have to be entered in your parent domain



Security –DNSSEC maintenance

- Modify zone
  - nsupdate(1)
    - bind-tools
  - By hand
    - Freeze zone
      - rndc freeze
    - Edit zone file
    - Sign zone file
      - dnssec-signzone
    - Reload zone file
      - rndc reload
    - Unfreeze zone
      - rndc thaw