

The BIND Software

tsaimh (2024-2025, 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 **B**erkeley **I**nternet **N**ame **D**omain 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

Install 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 ...}`

Version of BIND

- 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"`
- See BIND version of a DNS server
 - `$ dig chaos @140.113.6.1 txt version.bind`
 - ;; ANSWER SECTION:
`version.bind. 5 CH TXT "PowerDNS Authoritative Server 4.9.2 (built Oct 25 2024 15:29:05 by root@ns1.nycu.edu.tw)"`
- 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 (`named.root`)

BIND Configuration – named.conf

- /usr/local/etc/namedb/named.conf
 - Roles of this host for each zone it serves
 - primary, secondary,, 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, primaries

Examples of named configuration

named.conf

```
options {
    directory "/usr/local/etc/namedb";
    datasize 1000M;
    listen-on {140.113.100.100;};
    listen-on-v6 {2001:4f8:0:2::13;};
    recursion no;
};
zone "nasa.nycu" {
    type primary;
    file "primary/nasa.nycu";
    allow-update {none; };
    allow-transfer {none; };
};
zone "nasa.nthu" {
    type secondary;
    file "secondary/nasa.nthu";
    primaries { 140.114.1.1; 140.114.2.2; };
};
```

primary/nasa.nycu (Zone file)

```
$TTL 57600
$ORIGIN nasa.nycu
@      SOA ns1.nasa.nycu root.nasa.nycu (
        2025032000 10800 1200 360000 3600)
      NS ns1.nasa.nycu
      NS ns2.nasa.nycu
      MX 10 mx.nasa.nycu
      A  140.113.100.100
ns1    A  140.113.100.100
ns2    A  140.113.200.200
www    A  140.113.100.101
mx     A  140.113.100.102
```


DNS Database

– Zone data

The DNS Database

- A set of **text files** such that
 - Maintained and stored on the domain's **primary** name server
 - Often called **zone files**
 - Two types of entries
 - Resource Records (RR; e.g., A, NS, SOA)
 - The real data of a DNS database
 - Parser commands (e.g., \$TTL)
 - 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 – Format of Resource Record

- 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 – Types of Resource Records

- 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 Records

	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 signed 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	Signed, 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 – SOA

- 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 secondary server  
                                30M             ; retry  
                                1W             ; expire  
                                2H              ; negative caching TTL (since BIND 8.2)  
                                )
```

The DNS Database – NS

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


The DNS Database – A

- 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:

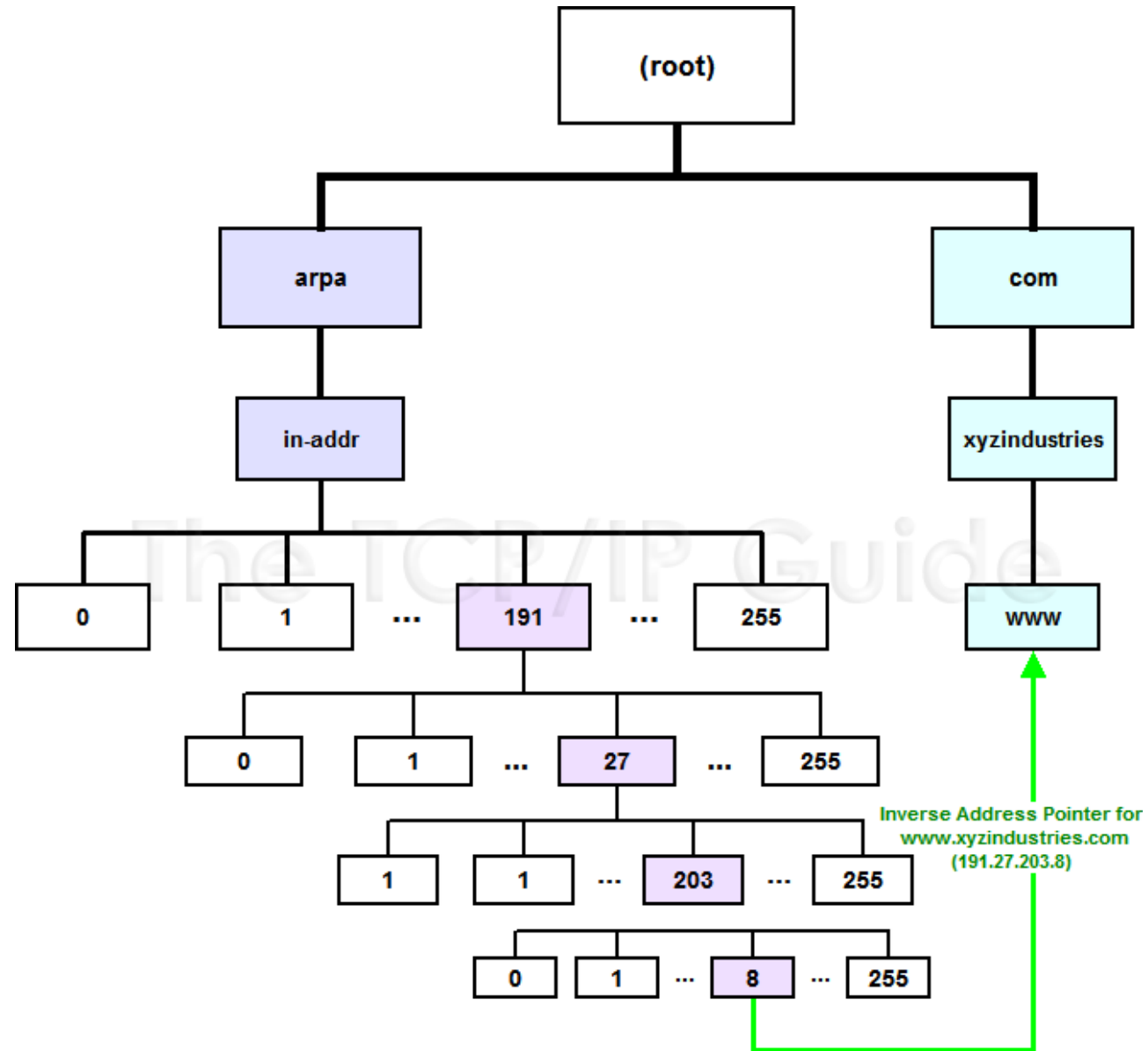
```
$ORIGIN cs.nctu.edu.tw.  
@      IN      NS      dns.cs.nctu.edu.tw.  
      IN      NS      dns2.cs.nctu.edu.tw.  
dns    IN      A       140.113.235.107  
dns2   IN      A       140.113.235.103  
  
www    IN      A       140.113.235.111  
www    IN      A       140.113.235.112
```

The DNS Database – PTR

- 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 – A and PTR Hierarchy



The DNS Database – MX

- 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 secondary
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 – CNAME

- 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:

<code>www</code>	<code>IN</code>	<code>A</code>	<code>140.113.209.63</code>
	<code>IN</code>	<code>A</code>	<code>140.113.209.77</code>
<code>penghu-club</code>	<code>IN</code>	<code>CNAME</code>	<code>www</code>
<code>King</code>	<code>IN</code>	<code>CNAME</code>	<code>www</code>
<code>R21601</code>	<code>IN</code>	<code>A</code>	<code>140.113.214.31</code>
<code>superman</code>	<code>IN</code>	<code>CNAME</code>	<code>r21601</code>

The DNS Database – TXT

- 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 (Sender Policy Framework) records (RFC 4408 -> 7208)

```
$TTL 3600;
$ORIGIN cs.nctu.edu.tw.
@      IN      SOA      csns.cs.nctu.edu.tw.      root.cs.nctu.edu.tw.      (
                                2007052102
                                1D      ; serial number
                                ; 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.
```

SPF records in TXT

```
$ dig +noall +answer txt nycu.edu.tw
nycu.edu.tw. 604800 IN TXT "_ag9zdjfv8ng0uiwp5y2omyq2g162e0o"
nycu.edu.tw. 604800 IN TXT "tDB2udtYmEh6tJvyRcm7NAA83_AsWe-
sfX_QYGF'NMF8"
nycu.edu.tw. 604800 IN TXT "google-site-
verification=NE6TgoobrSXGeGc-3nCIkeNW2NY_OBxj9QY4Wfo7Czw"
nycu.edu.tw. 604800 IN TXT "v=spf1 ip4:140.113.2.64/26
ip4:211.76.241.6 ip4:140.113.98.162 ip4:140.113.98.175
ip4:140.113.9.141 ip4:120.126.43.22 ip4:140.113.7.200
ip4:140.113.40.99 include:_spf.google.com ~all"

$ dig +noall +answer txt cs.nycu.edu.tw
cs.nycu.edu.tw. 3600 IN TXT "google-site-
verification=rxtQWSb_R1PsXyKX14dkC0PjaGR09n391Cv_SqJjWTQ"
cs.nycu.edu.tw. 3600 IN TXT "google-site-
verification=9Up-zAP5iwKawIBeUX4cPyja-K7BCXxXiIzbJ8QVcMM"
cs.nycu.edu.tw. 3600 IN TXT "v=spf1
redirect=_spf.cs.nycu.edu.tw"
```

SPF records in TXT (cont.)

```
$ dig +noall +answer txt _spf.cs.nycu.edu.tw
_spf.cs.nycu.edu.tw.      86400 IN      TXT      "v=spf1
include:legacy._spf.cs.nycu.edu.tw include:mailer-
cluster._spf.cs.nycu.edu.tw ~all"

$ dig +noall +answer txt legacy._spf.cs.nycu.edu.tw
legacy._spf.cs.nycu.edu.tw. 86400 IN      TXT      "v=spf1
mx:cs.nctu.edu.tw a:csmailer.cs.nycu.edu.tw
a:tcsmailer.cs.nycu.edu.tw ~all"

$ dig +noall +answer txt mailer-cluster._spf.cs.nycu.edu.tw
mailer-cluster._spf.cs.nycu.edu.tw. 86400 IN TXT "v=spf1
ip4:140.113.235.121 ip4:140.113.235.122 ip4:140.113.166.8
ip4:140.113.166.9 ~all"
```


The DNS Database – SRV

- 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:

`$ ssh tsaimh@cs.nycu.edu.tw`

```
; 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.nycu.edu.tw.
_ssh._tcp         SRV      0      3      22      new.cs.nycu.edu.tw.
; www server
_http._tcp        SRV      0      0      80      www.cs.nycu.edu.tw.
                  SRV      10     0      8000    new.cs.nycu.edu.tw.
; 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.f.2.0.0.0.5.0.1.0.0.2.ip6.arpa.
PTR f.root-servers.net.

Glue Record

- 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
 - **Parent zone also need to have 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
```

Trust between two forests

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

Lame delegation

- Lame delegation occurs when the **nameservers** responsible for providing an **authoritative** answer for a domain **fail to respond to DNS queries** or **respond improperly** in some way.
- Lame delegations can have a variety of adverse impacts.
 - Longer lookup times (e.g., delays in loading web pages)
 - Resolution failures
 - Degraded user experiences
 - Security threats
 - Poor organic search performance (e.g., poor ranking in search engine)
 - Damage to online reputation

Statements of named.conf

國立陽明交通大學資工系資訊中心

Information Technology Center of Department of Computer Science, NYCU

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
A ns1.atrust.com. 206.168.198.209
A ns2.atrust.com. 66.77.122.161
A www 66.77.122.161
A mailserver 206.168.198.209
A secure 66.77.122.161
; reverse maps
A exterior1 206.168.198.209
PTR 209.198.168.206 exterior1.atrust.com.
A exterior2 206.168.198.213
PTR 213.198.168.206 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 secondary 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 statement

- 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
 - primary, secondary, hint, forward, stub
 - The zone file is just a collection of DNS resource records
 - Basically

```
Syntax:
zone "domain_name" {
    type primary | secondary | stub;
    file "path";
    primaries {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 secondary zone

BIND Configuration

– named.conf primary and secondary zones

- primary server zone configuration

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

- secondary server zone configuration

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


BIND Configuration

– named.conf forward zone

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

BIND Configuration

– named.conf reverse zone

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

```
$ORIGIN in-addr.arpa.
```

```
...
```

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.

```
...
```

```
named.235.rev
```

BIND Configuration

– named.conf root servers

- 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 primary)  
    request-ixfr yes|no;     (for secondary)  
    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;  
};
```

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

- RNDCC – 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

- primary
 - Edit zone files
 - Serial number
 - Forward and reverse zone files for single IP
 - Do “rndc reload”
 - “notify” is on, secondary will be notified about the change
 - “notify” is off, refresh timeout, or do “rndc reload” in secondary
- Zone transfer
 - DNS zone data synchronization between primary and secondary 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

<code>dhcp-host1.domain</code>	<code>IN</code>	<code>A</code>	<code>192.168.0.1</code>
<code>dhcp-host2.domain</code>	<code>IN</code>	<code>A</code>	<code>192.168.0.2</code>

- 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
                                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 primary;
    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 primary;  
    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 primary;
    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.
...
```

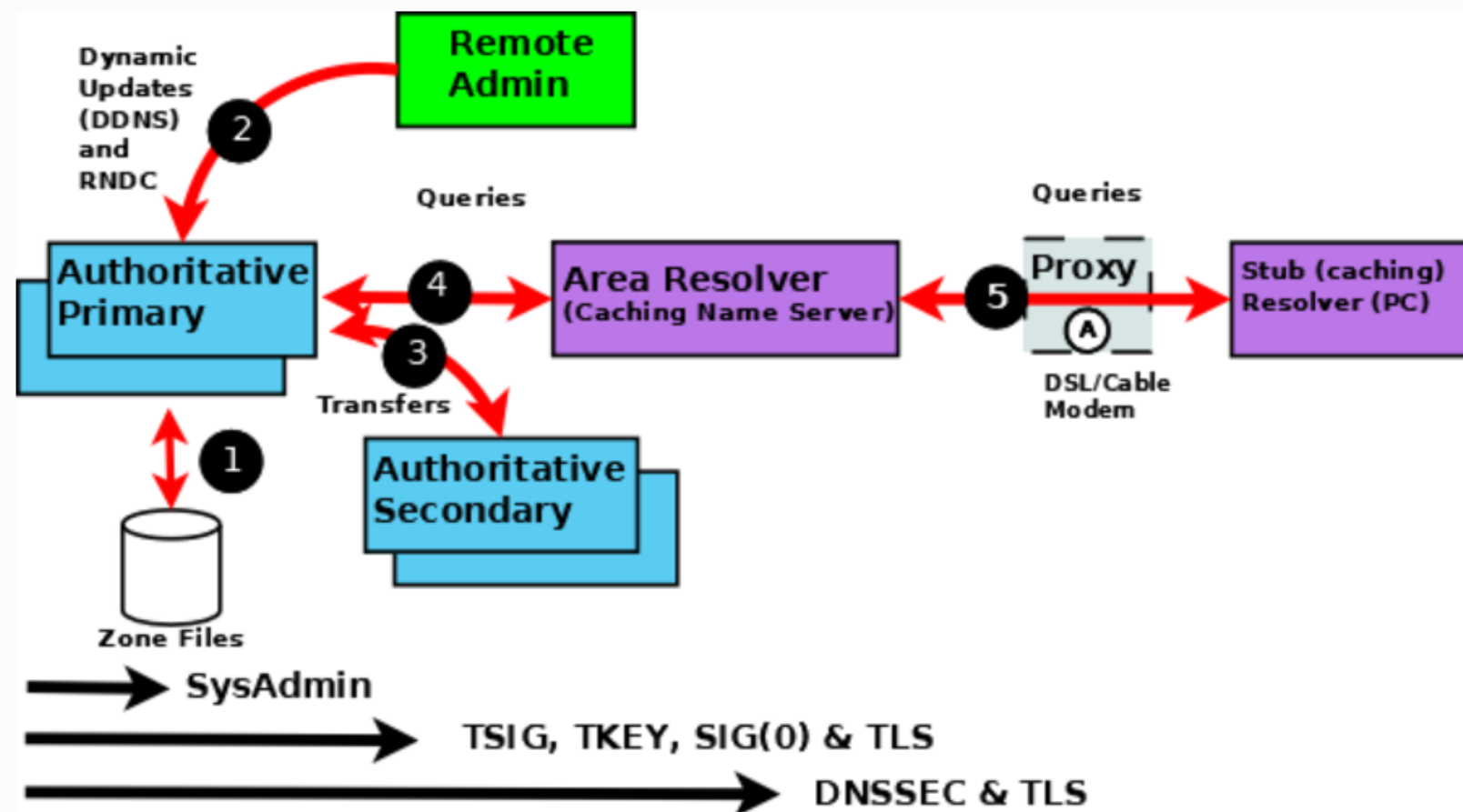
BIND Security

國立陽明交通大學資工系資訊中心

Information Technology Center of Department of Computer Science, NYCU

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.



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

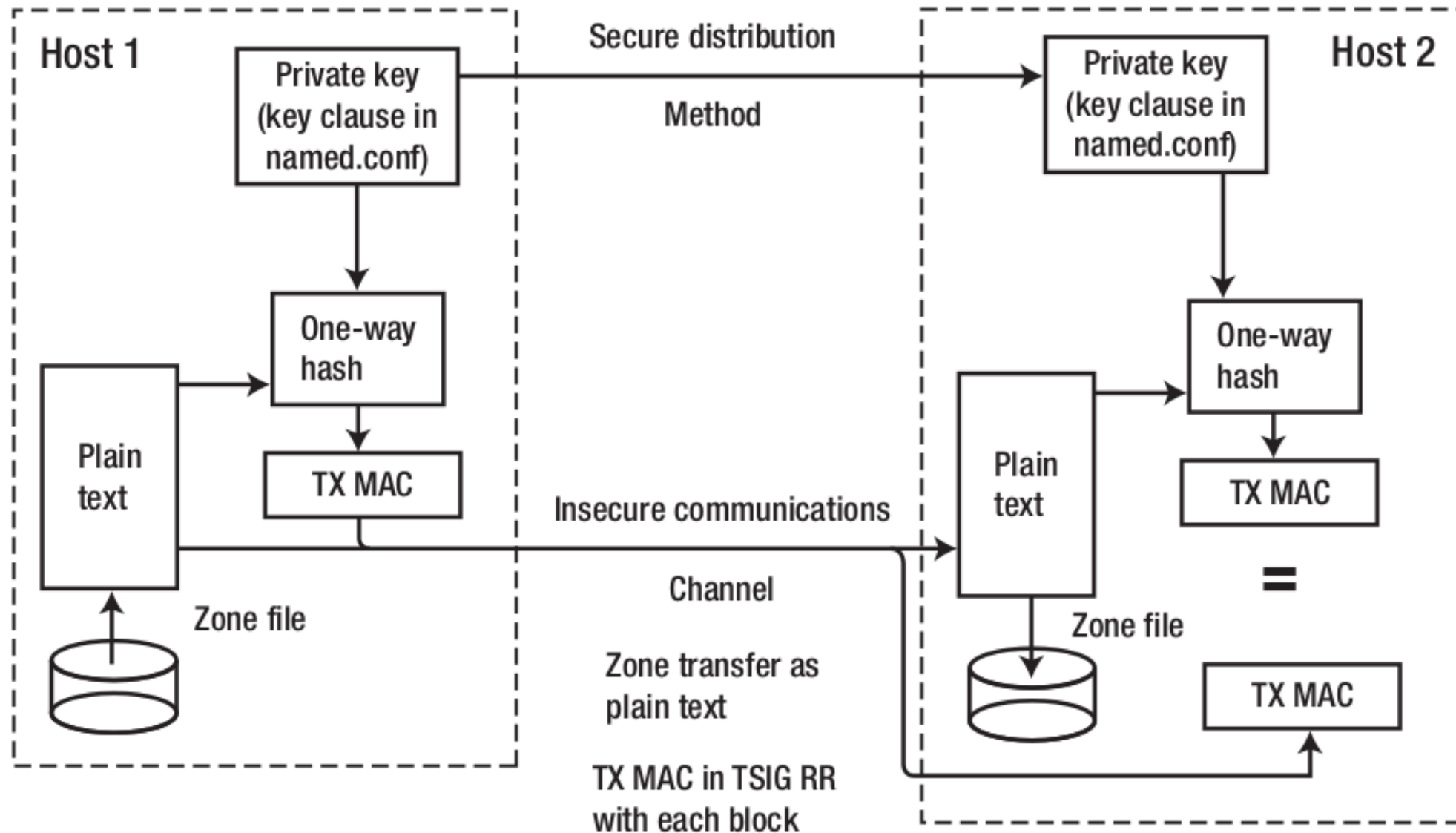
```
acl bogusnet {  
    0.0.0.0/8 ;           // Default, wild card addresses  
    1.0.0.0/8 ;           // Reserved addresses  
    2.0.0.0/8 ;           // Reserved addresses  
    169.254.0.0/16 ;      // Link-local delegated addresses  
    192.0.2.0/24 ;        // Sample addresses, like example.com  
    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 { mysecondaries; }
```

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

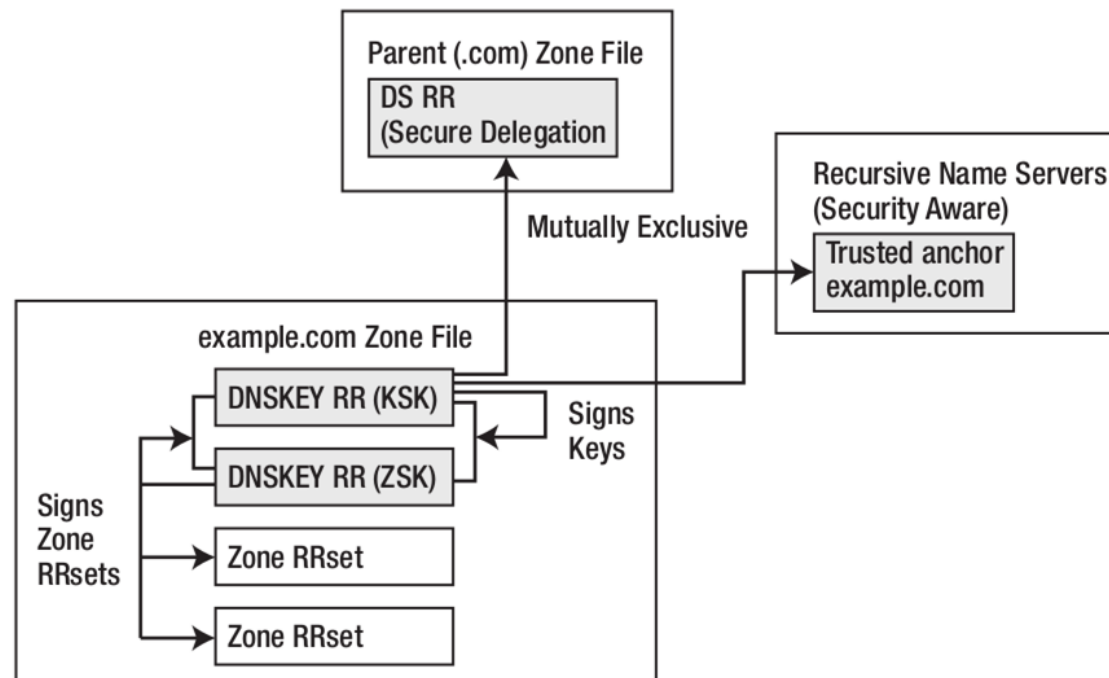
DS: Delegation of Signing

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



Verification of RRSIG

```
> dig +dnssec www.cs.nycu.edu.tw @8.8.8.8
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 2932
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1

;; ANSWER SECTION:
www.cs.nycu.edu.tw.      1696      IN        CNAME     csproxy-itsc.cs.nycu.edu.tw.
www.cs.nycu.edu.tw.      1696      IN        RRSIG     CNAME 13 5 3600 20250403000000 20250313000000
65052 cs.nycu.edu.tw. uNfavtR6yzmiYYKGRsMyY7exC0BC7KE1GSUS2PynNgq6eRbJQGiJaR6Y
sa4NXO0yXZnBRQneDW6GUIJFgaSVvQ==
csproxy-itsc.cs.nycu.edu.tw. 1696 IN      A         140.113.96.55
csproxy-itsc.cs.nycu.edu.tw. 1696 IN      RRSIG     A 13 5 3600 20250403000000 20250313000000
65052 cs.nycu.edu.tw. Yh8mZqU98SVfkc4kJ7NGSPpE2Dmg9GGPdh2Hy/UIf7TyVv94+fHhgrTP
QUPVydZ4hW7EnzVKXY6gx6Tf69Y3wA==

> delv www.cs.nycu.edu.tw 8.8.8.8
; fully validated
www.cs.nycu.edu.tw.      15        IN        CNAME     csproxy-itsc.cs.nycu.edu.tw.
www.cs.nycu.edu.tw.      15        IN        RRSIG     CNAME 13 5 3600 20250403000000 20250313000000
65052 cs.nycu.edu.tw. uNfavtR6yzmiYYKGRsMyY7exC0BC7KE1GSUS2PynNgq6eRbJQGiJaR6Y
sa4NXO0yXZnBRQneDW6GUIJFgaSVvQ==
csproxy-itsc.cs.nycu.edu.tw. 15 IN      A         140.113.96.55
csproxy-itsc.cs.nycu.edu.tw. 15 IN      RRSIG     A 13 5 3600 20250403000000 20250313000000
65052 cs.nycu.edu.tw. Yh8mZqU98SVfkc4kJ7NGSPpE2Dmg9GGPdh2Hy/UIf7TyVv94+fHhgrTP
QUPVydZ4hW7EnzVKXY6gx6Tf69Y3wA==
```

BIND Debugging and Logging

國立陽明交通大學資工系資訊中心

Information Technology Center of Department of Computer Science, NYCU

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

國立陽明交通大學資工系資訊中心

Information Technology Center of Department of Computer Science, NYCU

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

國立陽明交通大學資工系資訊中心

Information Technology Center of Department of Computer Science, NYCU

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 algorithms: 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.
    Mnm5RaKEFAW4V5dRhP70xLtGAFMb/Zsej2vH
    mK507zHL+U2Hbx+arMMoA/a0xtp6Jxp0FWM3
    67VHclTjjGX9xf++6qvA65JHRNvKoZgXGtXI
    VGG6ve8A8J9LRePtCKwo3WfhtLEMFsd1KI6o
    JTViPzs3UDEqgAvy8rgtvwr80a8= )
86400 NS ns1.example.com.
86400 NS ns2.example.com.
86400 RRSIG NS 8 2 86400 20110118013101 (
    20101219013101 27228 example.com.
    ubbRJV+DiNmgQITtncLOCjIw4cfB4qnC+DX8
    ....
    S78T5Fhx5SbLBPTBKm1KvKxcx6k= )
```

Security – Configuring DNSSEC (5)

- Updating the Zone file
 - Edit the zone file

```
zone "example.com" {  
    type primary;  
    file "example.com.signed";  
    primaries {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

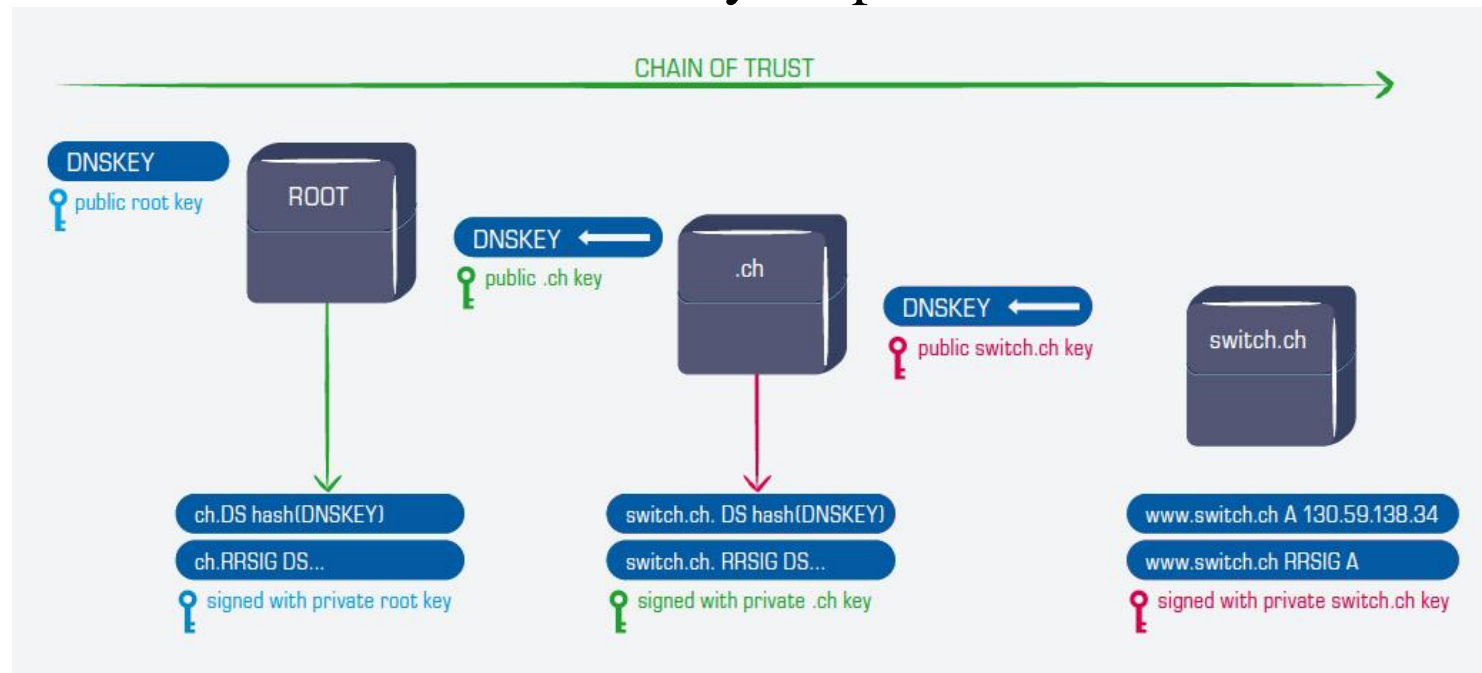
Security – Configuring DNSSEC (6)

- 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