# Overview

- Last Lecture
  - Scheduled tasks and log management
- This Lecture
  - DNS and BIND
  - Reference: DNS and BIND, 4th Edition, O'Reilly
- Next Lecture
  - Address assignment (DHCP)

## Problem

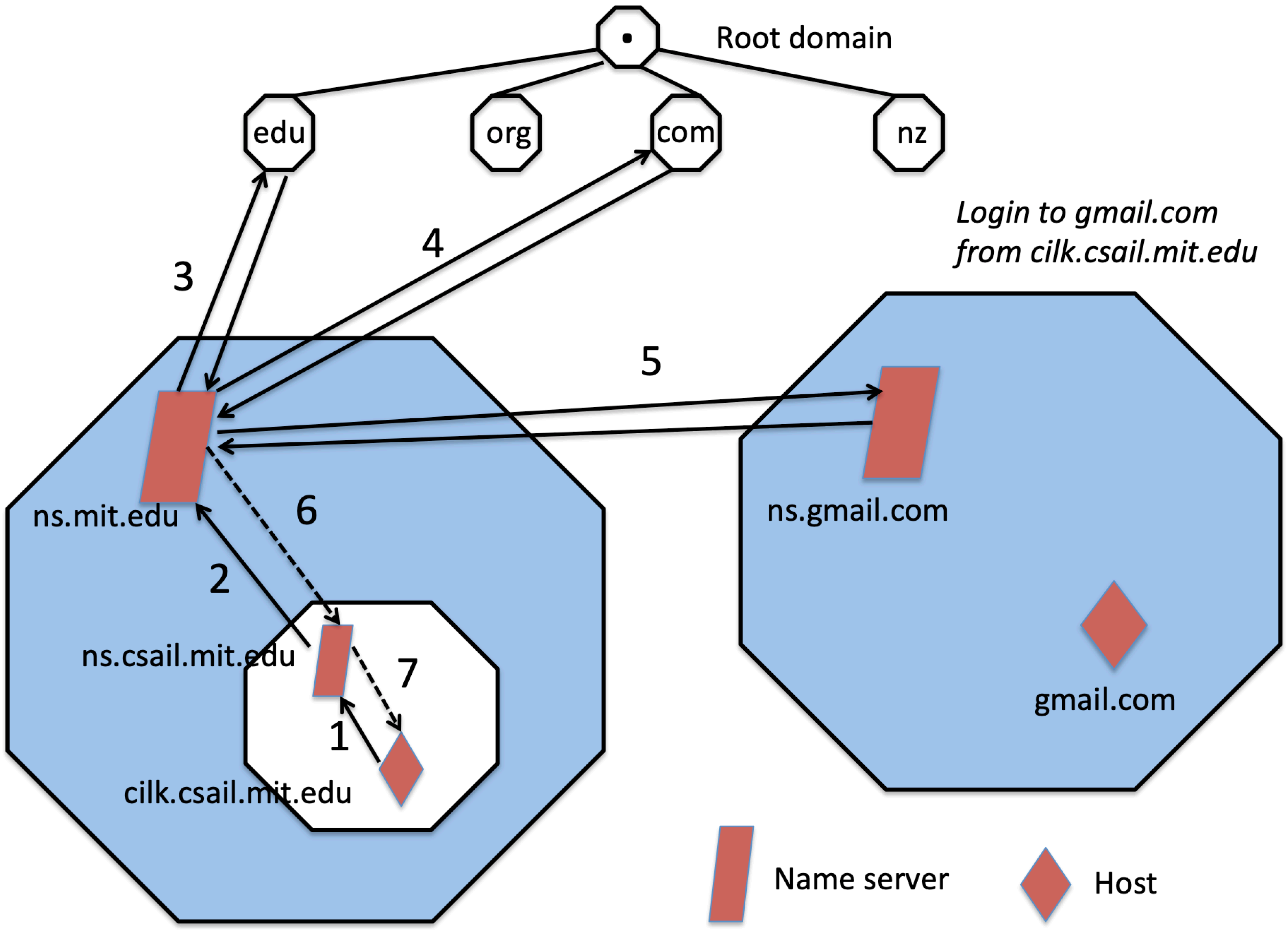
- How to get the IP address with an IP name?
  - Mapping between IP addresses and IP names
- Simple solution
  - Central database, like */etc/hosts* or Sun Microsystems' NIS (Network Information Service) or Windows' WINS for LAN.
- However, keeping billions of such records in a central database used by billions of users is almost impossible.

## Domain Name Service (DNS)

- A distributed solution
- Each organisation, called domain, maintains its own database and answers queries about its domain.
- The domains are organised as a tree so that any domain can find other domains via the parent node in the tree. The root of the tree is a common knowledge and well known.
- BIND is an implementation, an acronym for Berkeley Internet Name Domain

# How DNS works?

- Domain and Zone
  - A leaf node in the domain name space
  - Fully Qualified Domain Name (FQDN), e.g. hextreme.otago.ac.nz
  - Zone is an internal node in the domain name space
- The name space is divided into zones
  - At least one server in each zone
  - Zones are /(root), net, org, nz
  - Can have sub-zones in a zone, e.g. co.nz
- The servers in a zone are responsible for answering queries about the zone
- Zones are organized hierarchically so that servers of a zone can be traced from /(root) servers



Root domain
edu
org
com
nz
Login to gmail.com
from cilk.csail.mit.edu
3
4
5
6
2
7
1
ns.mit.edu
ns.gmail.com
ns.csail.mit.edu
cilk.csail.mit.edu
gmail.com
Name server
Host

# DNS client

- The DNS clients use the resolver library
  - Resolver is a collection of C functions. The central routines are **gethostbyname** and **gethostbyaddr**
- DNS queries are divided into two kinds
  - Recursive: get the final answer of the query
  - Iterative: may get redirected to another server
- There are two important files used by those routines: **host.conf** and **nsswitch.conf**
- **resolv.conf** tells the routines the IP address of the DNS server if DNS is needed.

## Example code

- int main(int argc, char *argv[]) {
- int i;
- struct hostent *he;
- struct in_addr **addr_list;
- struct in_addr addr;
- he = gethostbyname(argv[1]);
- if (he == NULL) { // do some error checking
-     herror("gethostbyname"); // herror(), NOT perror()
-     exit(1);
- }
- // print information about this host:
- printf("Official name is: %s\n", he->h_name);
- printf("IP address: %s\n", inet_ntoa(*(struct in_addr*)he->h_addr));
- printf(“\n”);
- }

## DNS client (cont.)

- **/etc/host.conf**: used by older Linux standard library libc
  - Order: use hosts file, or dns or nis
  - Multi: allow to have several IP addresses for a host in **/etc/hosts**
  - Nospoof: check for spoofing
  - Alert: syslog for spoofing
- **/etc/nsswitch.conf**: used by GNU standard library glibc
  - Can specify the order of how to retrieve each info field such as **hosts** and **networks**
  - The retrieve methods are **dns, nis**, **files,** etc

## DNS server

- Types of servers
  - Primary (master) server: data is stored in authoritative source files and maintained by system administrators
  - Secondary (slave) server: mirrors the data in the primary server and downloads its data second-hand from a primary server at regular intervals
- The name servers are specified in the file **resolv.conf** in a client machine

# DNS server (cont.)

- To configure a name server, there are several important files
  - /etc/named.conf: specify where to find files for lookups and reverse lookups at the local domain, where to forward requests for outside of the domain
  - Database files for each local domain and local subnets
    - Often under **pz** or **sz**, but not required
    - Get a template to create them
  - **named.cache** or **named.root**
    - Contains the names of root servers
    - The name server needs them when a request is out of its knowledge
- To start up name server simply type
  - **/usr/sbin/named**

# RR types

- Types of records in DNS database
  - **SOA**: indicates the start of authority for this domain
  - **NS**: lists a name server for this domain or sub-domain
  - **MX**: lists a mail exchanger for this domain with priority
  - **A**: defines the canonical name of a host with a given IP address; AAAA for IPv6
  - **CNAME**: associates an alias with a canonical name
  - **PTR**: for reverse lookup (IP address to name)
  - **HINFO**: advertises host info, CPU and OS
  - **TXT**: description

## Resource record

- Resource record format for **named** server
  - *Domain [ttl] [class] type data*
  - Domain: to which domain this entry applies
  - ttl: force resolvers to discard info after a certain time
  - Class: IN for IP addresses (Internet), other classes like Hesiod is mostly confined to MIT)
  - Record type: A, AAAA, SOA, PTR, NS, and MX
  - Data: depends on type
- Best practices
  - Set proper ttl field for RR
  - Create PTR records (reverse mapping records) for verifying IP addresses with IP names

## Impact of TTL

- TTL tells a client how long a DNS record should be cached locally.
- If TTL is 3 hours, when a record has been changed, e.g. a server changes to a different IP address, how long should it wait for all the cached records to be flushed?
- What should we do if an server IP address is going to change, but we don't want to disrupt the services?

## Advanced features

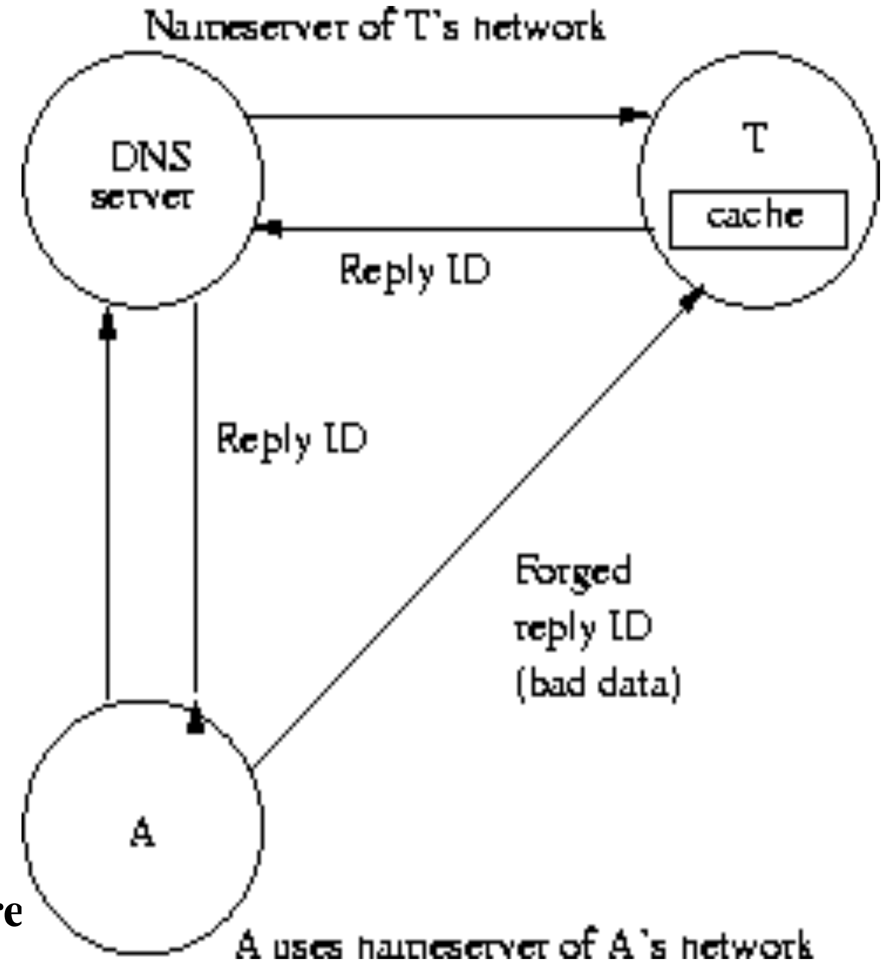
- Caching makes DNS more efficient
  - Store what the server has learnt
  - TTL is important parameter for caching
- Delegation of sub-domains
  - Forward requests to servers in sub-domains
  - Needs to specify a server for each sub-domain
  - The delegated sub-domain is called a zone in the domain. The domain itself is a zone as well.
- Forwarders
  - Reduce offsite DNS traffic
- Reverse mapping: address to name mapping
- Remote control of name server
  - rndc: talk with named through a TCP connection

# Security issues

- DNS cache poisoning
  - The attacker sends DNS queries to the victim DNS server and collects the query of the server
  - If the server uses static port for sending query, the vulnerability can be exploited.
  - The attacker causes the victim to send a query. Then it send a forged answer to the victim, which makes the forged answer into its cache. Therefore, the cache is poisoned
  - The attacker can take advantage of the poisoned cache to impersonate a trusted host.
  - Solution: don't allow recursive query outside your network; use secure DNS

# DNS cache poisoning

http://www.youtube.com/watch?v=1d1tUefYn4U


Nameserver of T's network
DNS server
T
cache
Reply ID
Reply ID
Forged reply ID (bad data)
A
A uses nameserver of A's network

# Other DNS cache poisoning

- Request: what are the address records for subdomain.attacker.example?
- Attacker's response:
  - Answer:
    - (no response)
    - Authority section: attacker.example. 3600 IN NS ns.kiwi_bank.co.nz.
    - Additional section: ns.kiwi_bank.co.nz. IN A w.x.y.z (an IP address controlled by the attacker)
- A vulnerable server would cache the additional A-record (IP address) for ns.kiwi_bank.co.nz, allowing the attacker to resolve queries to the entire kiwi_bank.co.nz domain.

## Bit-squatting

At Black Hat 2011, Artem Dinaburg introduced a "bit-squatting" attack against DNS where an attacker registers new domain names that differ by one bit from popular ones, in order to collect traffic intended for those names when bit errors occur during communications or storage.

DNSSEC has begun to be established as a mechanism for securing information distributed in DNS records. The new DANE protocol being developed in the IETF leverages DNSSEC to enable a domain name owner to inform relying parties which certificates and certificate authorities it trusts.

## Secure DNS Protocols

- DNS-SEC: DNS Security Extensions
  - origin authentication of DNS data
  - data integrity
  - authenticated denial of existence
  - No confidentiality of data (no encryption)
  - No protection against DDoS (Distributed Denial of Service)
- DNS-TSIG
  - Secret Key Transaction Authentication for DNS

## Other issue

- Should the names be English?
  - Internationalised Domain Names (IDNs)

## Summary

- The understanding of how DNS works in a distributed system.
- Recursive and iterative queries
- The impact of the TTL field of a DNS record
- The security issues of DNS