Network

Remote login

Domain Name System

The **Domain Name System** (**DNS**) is a *hierarchical and decentralized naming system* for computers, services, or other resources connected to the Internet or a private network. It associates various information with domain names assigned to each of the participating entities. Most prominently, it translates more readily memorized domain names to the numerical IP addresses needed for locating and identifying computer services and devices with the underlying network protocols. By providing a worldwide, distributed directory service, the Domain Name System has been an essential component of the functionality of the Internet since 1985.

Wikipedia ©

DNS work specific

Major record fields : NAME; TYPE; TTL;

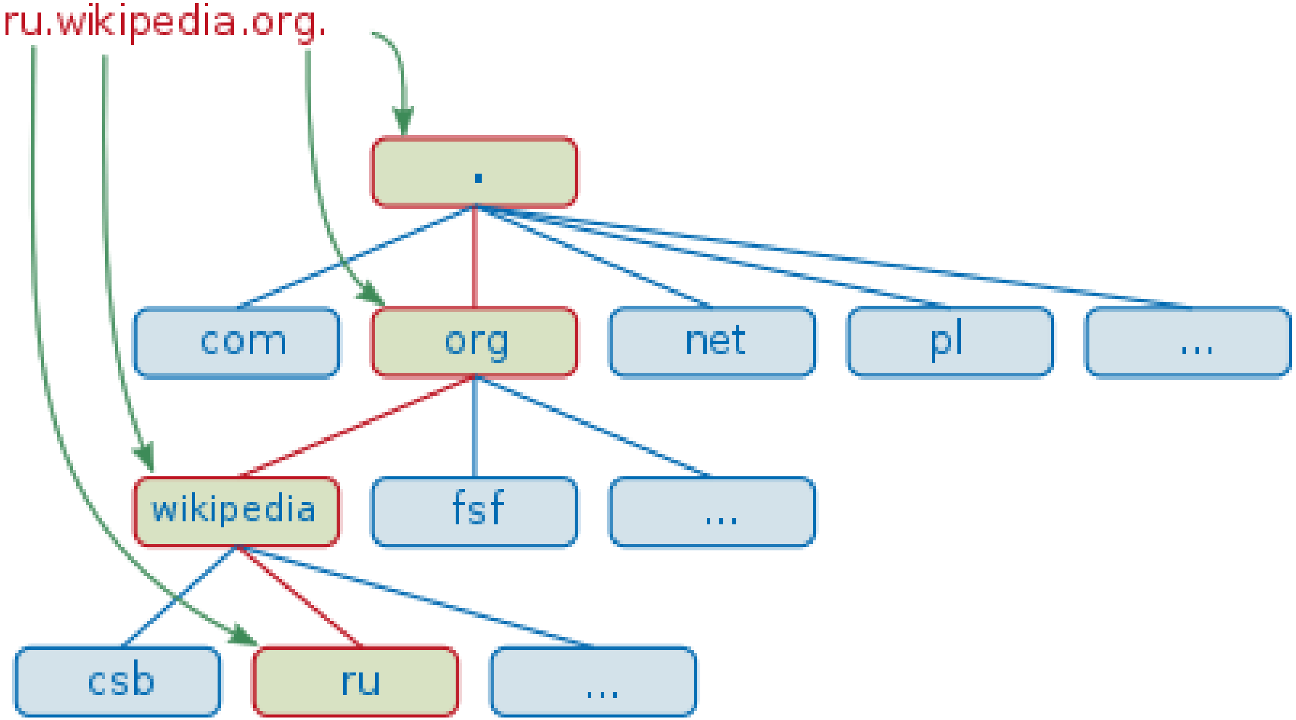
Records type :
A – address record
CNAME – *canonical name record*
MX – *mail exchange*
NS – *name server*
PTR – *pointer*
SOA - *Start of Authority*
SRV - server selection

DNS uses TCP for Zone transfer and UDP for name, and queries either regular (primary) or reverse.

Reverse lookup get name by IP.
For example, assuming the IPv4 address 208.80.152.2 is assigned to Wikimedia, it is represented as a DNS name in reverse order: 2.152.80.208.in-addr.arpa. When the DNS resolver gets a pointer (PTR) request, it begins by querying the root servers, which point to the servers of American Registry for Internet Numbers (ARIN) for the 208.in-addr.arpa zone. ARIN's servers delegate 152.80.208.in-addr.arpa to Wikimedia to which the resolver sends another query for 2.152.80.208.in-addr.arpa, which results in an authoritative response

Domain Name System

.. hierarchical and decentralized ..



Wikipedia ©

Local resolve host names

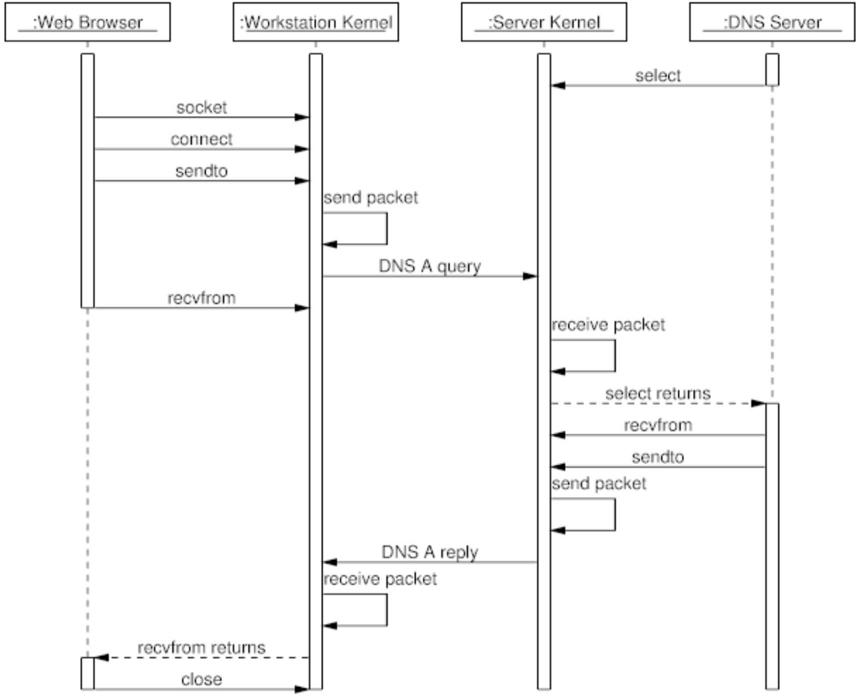
UNIX – /etc/hosts , Sample:

```
# This file was automatically generated by WSL. To stop automatic generation of this file, add the following entry to /etc/wsl.conf:
# [network]
# generateHosts = false
127.0.0.1       localhost
127.0.1.1       workstation.localdomain       workstation

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Windows file - c:\Windows\System32\drivers\etc\hosts , Sample:

```
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      102.54.94.97     rhino.acme.com          # source server
#       38.25.63.10     x.acme.com              # x client host

# localhost name resolution is handled within DNS itself.
#                         127.0.0.1       localhost
#                         ::1             localhost
```

DNS Sequence



DNS Server install

CentOS bind server install :

```
$ sudo yum install bind bind-utils
```

Configure as primary DNS server:

```
$ sudo vi /etc/named.conf
```

If target is:

Host	***Role***	***Private FQDN***	***Private IP Address***
ns1	Primary DNS Server	ns1.nyc3.example.com	10.128.10.11
ns2	Secondary DNS Server	ns2.nyc3.example.com	10.128.20.12

named.config file is:

```
acl "trusted" {
        10.128.10.11;    # ns1 - can be set to localhost
        10.128.20.12;    # ns2
        10.128.100.101;  # host1
        10.128.200.102;  # host2
};
options {
        listen-on port 53 { 127.0.0.1; 10.128.10.11; };
#         listen-on-v6 port 53 { ::1; };
        allow-transfer { 10.128.20.12; };      # disable zone transfers by default
        allow-query { trusted; };                              # allows queries from "trusted" clients
                                    }
include "/etc/named/named.conf.local";
```

named.conf.local file is:

```
zone "nyc3.example.com" {
    type master;
    file "/etc/named/zones/db.nyc3.example.com"; # zone file path
};
zone "128.10.in-addr.arpa" {
    type master;
    file "/etc/named/zones/db.10.128";  # 10.128.0.0/16 subnet
    };
```

Zone file db.nyc3.example.com

```
@       IN      SOA     ns1.nyc3.example.com. admin.nyc3.example.com. (
                          3         ; Serial
           604800     ; Refresh
            86400     ; Retry
          2419200     ; Expire
           604800 )   ; Negative Cache TTL
; name servers - NS records
    IN      NS      ns1.nyc3.example.com.
    IN      NS      ns2.nyc3.example.com.

; name servers - A records
ns1.nyc3.example.com.          IN      A       10.128.10.11
ns2.nyc3.example.com.          IN      A       10.128.20.12

; 10.128.0.0/16 - A records
host1.nyc3.example.com.        IN      A      10.128.100.101
host2.nyc3.example.com.        IN      A      10.128.200.102
```

DNS vulnerability

There are three major vulnerabilities with DNS to watch out for, which attackers often exploit to abuse DNS:

1. **Internal DNS servers hold all the server names and IP addresses for their domains and will share them with anyone that asks.** This makes DNS a great source of information for attackers when they're trying to do internal reconnaissance.
2. **DNS caches aren't "authoritative, and they can be manipulated.** If your DNS server is "poisoned" with bad records, computers can be fooled into going to bad places.
3. **DNS relays query information from internal workstations to outside servers,** and attackers have learned how to use this behavior to create "covert channels" to exfiltrate data.

The **Domain Name System Security Extensions** (**DNSSEC**) is a suite of extension specifications by the Internet Engineering Task Force (IETF) for securing data exchanged in the Domain Name System (DNS) in Internet Protocol (IP) networks. The protocol provides cryptographic authentication of data, authenticated denial of existence, and data integrity, but not availability or confidentiality.

Wikipedia ©

Homework task

GOAL:

Get 3 different IP answers on the one host name in 3 requests.

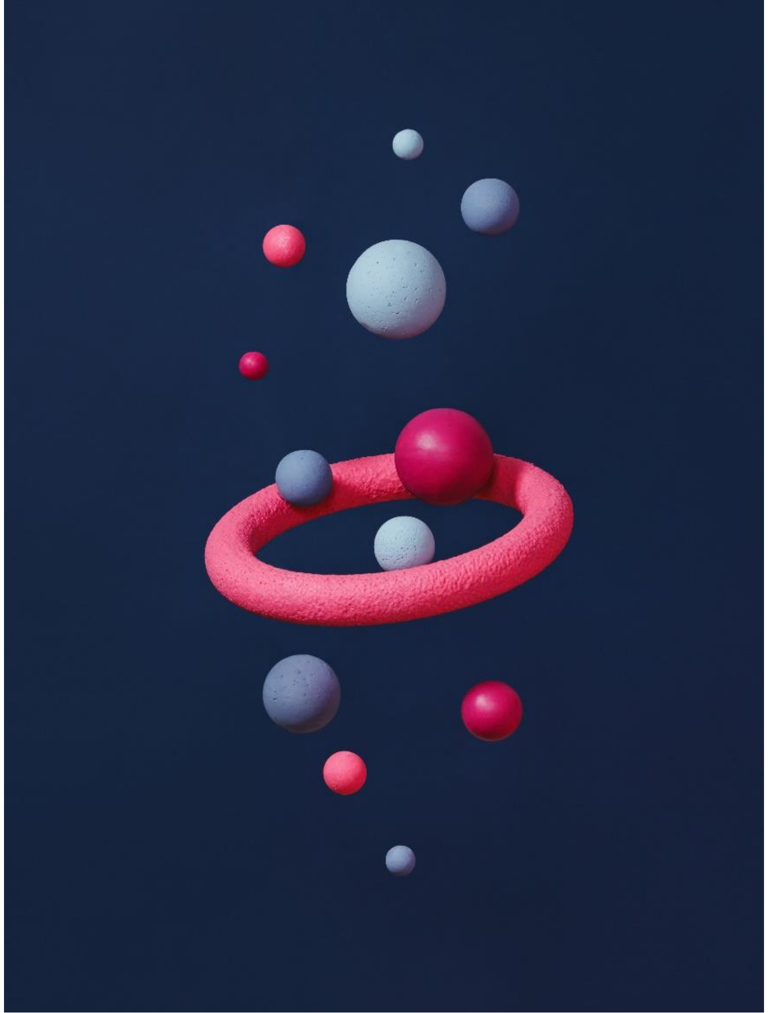
What to do:

- Edit local host definition files
- Set DNS server locally and put record in it.
- Reach original DNS server.

Environment:
1 Virtual Machines (VM) with ethernet adapter in Internet.

Suggest use VirtualBox and CentOS 7 image.

How to check:
use ping or nslookup utility for get answer



THANK YOU