SEED Lab: DNS In a Box

1 实验概述

DNS（域名系统）是互联网的电话簿，它将主机名转换为 IP 地址（反之亦然）。这个转换过程通过 DNS 解析完成，这个过程在后台进行。该解析过程涉及多个名称服务器，包括根服务器、顶级域（TLD）服务器和最终的域名服务器。这些名称服务器组成了整个 DNS 系统，是互联网的重要基础设施。

为了帮助学生理解这些名称服务器如何一起工作以形成基础设施，我们将创建一个名为“DNS In a Box”的小型 DNS 系统。正如其名称所示，该 DNS 系统中的多个名称服务器运行在一个单一机器上。这是通过容器技术实现的。

尽管这个系统很小，但它包含了真实 DNS 基础设施的所有基本元素。通过构建这样的系统，学生将更深入地了解 DNS 是如何工作的。虽然这不是一个安全实验，但它是几个 SEED 实验的基础。本实验涵盖以下主题：

- DNS 及其工作原理
- DNS 查询过程
- 根服务器和顶级域（TLD）服务器
- Docker 容器、Docker Compose

参考资料。 关于 DNS 协议的详细内容可以在以下资源中找到：

- 《SEED 书》第 18 章，*Computer & Internet Security: A Hands-on Approach*, 3rd Edition, by Wenliang Du. 详情请见 `https://www.handsonsecurity.net`.
- 《SEED 讲座》第 7 节，*Internet Security: A Hands-on Approach*, by Wenliang Du. 详情请见 `https://www.handsonsecurity.net/video.html`.

实验环境。 本实验在我们预先构建好的 Ubuntu 20.04 VM（可以从我们的 SEED 网站当中下载）当中测试可行。既然我们使用容器来建立实验环境，本实验不太依赖 SEED VM。您可以在其他 VM、物理机器以及云端 VM 上进行此实验。

2 实验准备

在本实验中，我们将构建一个简化的 DNS 基础设施。我们将从小规模开始构建，然后逐渐扩大。我们建立的第一个 DNS 系统由四个名称服务器组成，每个名称服务器代表 DNS 基础设施中的特定角色。在现实世界中，这些名称服务器分布在不同的网络上，但在本实验中，为了简化操作，我们将它们放置在同一网络中。图 1描述了该系统的配置。我们将使用容器来运行这些名称服务器。

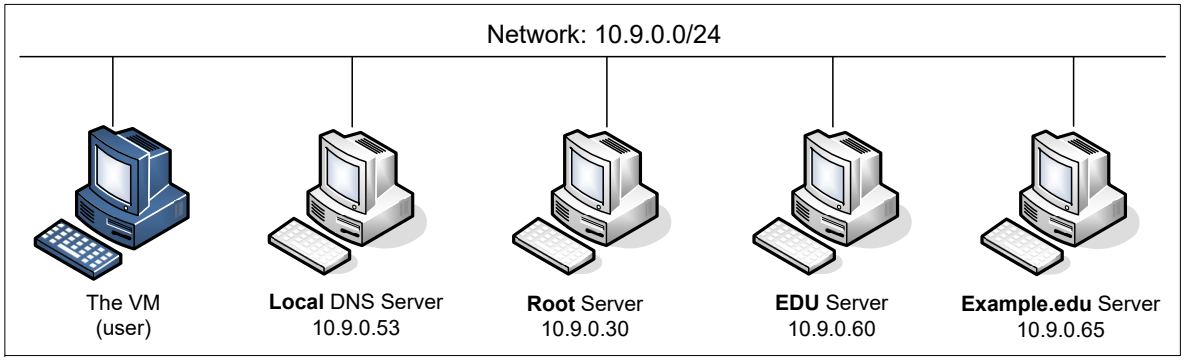

图 1: 简化版的 DNS 基础设施

2.1 容器的设置和命令

“‘ 请从实验的网站下载 Labsetup.zip 文件到你的 VM 中，解压它，进入 Labsetup 文件夹，然后用 docker-compose.yml 文件安装实验环境。对这个文件及其包含的所有 Dockerfile 文件中的内容的详细解释都可以在链接到本实验网站的用户手册 [1] 中找到。如果这是您第一次使用容器设置 SEED 实验环境，那么阅读用户手册非常重要。

在下面，我们列出了一些与 Docker 和 Compose 相关的常用命令。由于我们将非常频繁地使用这些命令，因此我们在 .bashrc 文件（在我们提供的 SEED Ubuntu 20.04 虚拟机中）中为它们创建了别名。

```
$ docker-compose build  # 建立容器镜像
$ docker-compose up     # 启动容器
$ docker-compose down   # 关闭容器

// 上述 Compose 命令的别名
$ dcbuild       # docker-compose build 的别名
$ dcup          # docker-compose up 的别名
$ dcdown        # docker-compose down 的别名
```

所有容器都在后台运行。要在容器上运行命令，我们通常需要获得容器里的 Shell 。首先需要使用 docker ps 命令找出容器的 ID ，然后使用 docker exec 在该容器上启动 Shell 。我们已经在 .bashrc 文件中为这两个命令创建了别名。

```
$ dockps        // docker ps --format "{{.ID}}  {{.Names}}" 的别名
$ docksh <id>   // docker exec -it <id> /bin/bash 的别名

// 下面的例子展示了如何在主机 C 内部得到 Shell
$ dockps
b1004832e275  hostA-10.9.0.5
0af4ea7a3e2e  hostB-10.9.0.6
```

[1] 如果你在部署容器的过程中发现从官方源下载容器镜像非常慢，可以参考手册中的说明使用当地的镜像服务器

```
9652715c8e0a  hostC-10.9.0.7

$ docksh 96
root@9652715c8e0a:/#

// 注: 如果一条 docker 命令需要容器 ID，你不需要
//     输入整个 ID 字符串。只要它们在所有容器当中
//     是独一无二的，那只输入前几个字符就足够了。
```

如果你在设置实验环境时遇到问题，可以尝试从手册的 “Miscellaneous Problems” 部分中寻找解决方案。““

2.2 Docker Compose 文件

我们的 DNS 系统包括四个名称服务器。这些名称服务器的作用总结如下：

- 本地 DNS 服务器：为其他计算机进行 DNS 解析。
- 根服务器：提供根区域的服务的名称服务器。
- edu 服务器：提供.edu 区域的服务，这是一个顶级域（TLD）。
- example.edu 服务器：提供 example.edu 区域的服务。

我们为每个名称服务器都使用了一个容器。这些容器是如何构建和运行的将在 docker-compose.yml 文件中描述，该文件包含在实验配置文件中。在这个文件中，我们创建了一个网络 10.9.0.0/24 以及四个容器：每个 services 部分中的条目代表一个容器，并且为这些容器分配了 IP 地址。以下是一个示例片段：

```
services:
  seed_base_router:                        ☆
    build: ./base_image
    image: seed-base-image-bind
    container_name: seed-base-bind
    command: " echo 'exiting ...' "

  example_edu_server:
    build:
        context: ./nameserver
        args:
            BIND_CONF_DIR: edu.example
    image: example-edu-server
    container_name: example-edu-10.9.0.65
    tty: true
    networks:
      seed-net:
        ipv4_address: 10.9.0.65

  edu_server:        ... (省略，与example_edu_server相似) ...
        ipv4_address: 10.9.0.60
```

```
  root_server:       ... (省略, 与example_edu_server相似) ...
        ipv4_address: 10.9.0.30
  local_dns_server:  ... (省略, 与example_edu_server相似) ...
        ipv4_address: 10.9.0.53
```

行 ☆ 指定了一个基础镜像,基本上是基于我们放置在 Docker Hub 上的handsonsecurity/seed-server:bind 镜像构建的（见 base_image 文件夹中的 Dockerfile）。虽然所有名称服务器可以直接从 Docker Hub 中使用这个基础镜像来构建它们自己的容器，但为了减少对 Docker Hub 的访问次数（公司规定每六小时只能进行一定数量的访问），我们首先在本地构建一个基于 Docker Hub 上的该基础镜像的本地基础镜像，然后使用该本地基础镜像来构建本实验中的其他容器镜像。

2.3 容器镜像

除了本地 DNS 服务器外，所有名称服务器都使用相同的文件夹（nameserver）来构建 Docker 镜像。以下是 Dockerfile 的内容：

```
FROM seed-base-image-bind                          ❶

ARG BIND_CONF_DIR

# Copy the BIND configuration files
COPY named.conf named.conf.options  /etc/bind/     ❷
COPY ${BIND_CONF_DIR}     /etc/bind/               ❸

# Start the nameserver
CMD service named start  && tail -f /dev/null
```

行 ❶ 指明这是一个本地镜像（基础镜像）。行 ❷ 将 BIND 9 的配置文件复制到容器内的 /etc/bind 文件夹中。这两个文件对所有名称服务器都是一样的，但每个名称服务器都有自己的区域文件。这就是为什么我们使用了 BIND_CONF_DIR 参数（行 ❸）来指定应该使用的配置文件夹（每个名称服务器都有一个独立的目录）。BIND_CONF_DIR 的值在 Compose 文件中设置。

本地 DNS 服务器。 本地 DNS 服务器的容器镜像与名称服务器的非常相似。我们稍后会解释其中的不同，因为构建该镜像是一个任务的一部分。

3 任务 1：为 example.edu 域建立名称服务器

在这个任务中，我们将为名为 example.edu 的域建立一个名称服务器。容器文件位于 nameserver 文件夹内。特定域的配置文件位于 edu.example 子文件夹中。

步骤 1. 添加区域条目 . 要为主机提供 example.edu 域服务，我们需要在 BIND 9 的配置文件 named.conf 中添加一个区域条目，以便名称服务器知道自己将要管理哪些域。为了方便起见，我们将这个条目添加到我们修改后的 named.conf 文件中的 named.conf.seedlabs。

此条目表示当前名称服务器是该域的主服务器，并且指定区域文件在 `file` 条目中。

```
zone "example.edu" {
        type master;
        file "/etc/bind/example.edu.db";
};
```

步骤 2. 创建区域文件 . 在构建 Docker 镜像时，区域文件 `example.edu.db` 将被复制到容器内的 `/etc/bind` 文件夹中。为了让学生开始操作，我们在该区域文件中提供了一些信息，但学生需要进行所有必要的更改。

```
$TTL 3D
@       IN      SOA   ns.example.edu. admin.example.edu. (
                2008111001
                8H
                2H
                4W
                1D)

; Records for this nameserver (you need to make changes)
@               IN   NS    ns.example.edu.
ns.example.edu  IN   A     1.2.3.4


; IP addresses for the hostnames in the example.du domain
@       IN      A     1.2.3.5
www     IN      A     1.2.3.5
xyz     IN      A     1.2.3.6
*       IN      A     1.2.3.7
```

步骤 3. 测试 . 使用 `docker-compose` 命令来构建并启动所有容器。一旦容器运行，在你的虚拟机（即容器外部）运行以下命令。我们使用 `@10.9.0.65` 选项将 DNS 查询直接发送到文本中指定的 IP 地址 `10.9.0.65`，这是我们在设置中的 `example.edu` 名称服务器的 IP 地址。如果一切操作正确，则可以获取你在区域文件中指定的 IP 地址。请报告你的观察结果。

```
$ dig @10.9.0.65 www.example.edu
...
;; ANSWER SECTION:
www.example.edu.    259200   IN  A   1.2.3.5
...
```

4 任务 2：建立 edu TLD 名称服务器

在这个任务中，我们将为顶级域（TLD）域名构建一个名称服务器，即 edu 域名。我们需要修改 nameserver/edu 文件夹中的文件。首先，我们需向 named.conf.seedlabs 文件添加以下区域条目。此条目表示当前的名称服务器是该域的主服务器，并且指定区域文件在 file 条目中。

```
zone "edu" {
        type master;
        file "/etc/bind/edu.db";
};
```

创建区域文件 . 接下来，我们需要向区域文件添加记录。我们已经创建了区域文件 edu.db，但它不完整。学生需要进行所有必要的更改。

所有属于 edu 域的名称服务器都必须将其自己的名称服务器注册到这个顶级域（TLD）名称服务器中；否则，没有人能够找到它们。例如，我们为 syr.edu 域添加了两个记录：一个 NS 记录和一个 A 记录。NS 记录指定了 syr.edu 域的名称服务器，而 A 记录则指出了名称服务器的 IP 地址。这些记录中的数据是真实的，因为它们模拟了 syr.edu 域已经将其自身注册到了我们的 edu TLD 名称服务器上的事实。

```
; Real records for syr.edu
syr.edu.                IN      NS      ns1.syr.edu.
ns1.syr.edu.            IN      A       128.230.12.8
```

实验室任务 . 学生需要完成以下任务：（1）将你的 example.edu 名称服务器注册到此顶级域名称服务器中；（2）选择三个真实的域名，并将其注册到这个顶级域名称服务器。你可以通过运行 "dig <name> NS" 命令来查找一个域的名称服务器（例如，"dig syr.edu NS"）。

使用 docker-compose 命令，学生可以构建并启动所有容器。一旦容器运行，从你的虚拟机发送 DNS 查询到 edu 名称服务器（其 IP 地址在我们的设置中为 10.9.0.60）。请报告你的观察结果。

```
$ dig @10.9.0.60 www.example.edu
$ dig @10.9.0.60 www.syr.edu
$ dig @10.9.0.60 <你选择的其他三个域名>
$ dig @10.9.0.60 <不在你的区域文件中的某个域名>
```

需要注意的是，edu TLD 名称服务器配置为不执行递归查询，因此它只会告诉你查询中域名所对应的名称服务器；它不会为你解决这个查询。在最后一个命令中，你应该尝试一些以 edu 结尾但不在你的区域文件中的域名。

5 任务 4：建立根名称服务器

在这个任务中，我们将为根域建立一个名称服务器。所有顶级域名（TLD）名称服务器都必须向根名称服务器注册，以便它们能在 DNS 查询过程中被找到。在我们的容器中，根域的定义位于

bind.conf.seedlabs 文件中，并从区域文件 root.db 中加载记录。

对于我们希望包含在微型 DNS 系统中的每个顶级域（TLD），我们需要在区域文件中至少添加两条记录，包括一个 NS 记录和一个 A 记录。以下是一个例子，在该示例中，我们为 com 和 net 域添加了名称服务器信息（这些记录中的数据是真实的）。这两个域由同一家公司管理，因此它们共享相同的名称服务器集（我们仅包含了一个名称服务器）。

```
com.                 IN   NS   a.gtld-servers.net.
net.                 IN   NS   a.gtld-servers.net.
a.gtld-servers.net.  IN   A    192.5.6.30
```

实验室任务 . 学生需要向 root.db 区域文件中添加记录，以满足以下要求：

- 将你的 edu 名称服务器注册到这个根名称服务器。- 选择两个其他真实的顶级域名，并将它们也注册到此 TLD 名称服务器。其中一个顶级域（TLD）应是一个国家代码顶级域（ccTLD），它是某个国家的根。你可以通过运行 "dig <tld> NS" 命令来查找一个顶级域名称服务器的信息（例如，"dig com NS"）。你也可以从以下网址找到所有顶级域名信息：https://www.internic.net/domain/root.zone。

使用 docker-compose 命令，学生可以构建并启动所有容器。一旦容器运行，从你的虚拟机发送 DNS 查询到根名称服务器（通过 "dig @10.9.0.30" 命令）来测试是否你的 DNS 设置按预期工作。

6 任务 5：建立本地 DNS 服务器

当计算机需要将主机名转换为 IP 地址（反之亦然）时，它会向其辅助器发送一个请求。这个辅助器可以不在附近。本地 DNS 服务器将会进行整个 DNS 解析过程，并将结果反馈给计算机。在这个任务中，我们将设置这个本地 DNS 服务器。

根提示文件 . 如果本地 DNS 服务器无法从缓存中找到答案，则它将继续通过迭代过程从外部名称服务器获取答案。这一过程始于根服务器。因此，本地 DNS 服务器需要知道根服务器的 IP 地址。

在 /etc/bind/named.conf.default-zones 中（这是 BIND9 的配置文件 /etc/bind/named.conf 中的一部分），有一个针对根域的条目。这个条目为根区域指定了一个提示文件，本地 DNS 服务器通过这个文件知道根服务器的 IP 地址。

```
zone "." {
    type hint;
    file "/usr/share/dns/root.hints";
};
```

根提示文件指定了根区中名称服务器的信息（根域有 13 个名称服务器）。这些名称服务器的 IPv4 和 IPv6 地址都在这个文件中提供。以下是一个示例片段。

```
.                        3600000      NS    A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.      3600000      A     198.41.0.4
A.ROOT-SERVERS.NET.      3600000      AAAA  2001:503:ba3e::2:30
...
```

在我们实验配置文件中的 local_dns_server 文件夹内，我们创建了一个名为 root.hints 的空文件。当我们为本地 DNS 服务器构建容器镜像时，这个文件将被复制到 /usr/share/dns/ 文件夹中。你需要在该文件中添加记录，以便你的本地 DNS 服务器可以使用你的根服务器，而不是使用真实世界中的服务器。

测试 . 使用 docker-compose 命令来构建并启动所有容器。一旦容器运行，从你的虚拟机发送以下 DNS 查询命令（其 IP 地址在我们的设置中为 10.9.0.53）到本地 DNS 名称服务器。

```
$ dig @10.9.0.53 www.example.edu
$ dig @10.9.0.53 www.example.com
$ dig @10.9.0.53 www.example.net
$ dig @10.9.0.53 www.syr.edu
$ dig @10.9.0.53 www.xyz.<注册在你的根服务器中的TLD>
$ dig @10.9.0.53 www.xyz.<未注册在你的根服务器中的TLD>
```

7 任务 6：配置虚拟机以使用这个本地 DNS 服务器

到目前为止，我们在 dig 命令中需要使用 @<ip> 来指定 dig 应该与哪个 DNS 服务器通信。虽然这对 dig 没有问题，但对于依赖 DNS 的其他软件可能会出现问题。我们需要告诉操作系统我们在这个任务中构建的 DNS 服务器容器是系统的本地 DNS 服务器。

这可以通过更改用户机器上的解析器配置文件（/etc/resolv.conf），将该容器的 IP 地址作为第一个 nameserver 条目添加到文件中来实现，即这个服务器将被用作首选 DNS 服务器。不幸的是，我们提供的虚拟机使用动态主机配置协议（DHCP）获取网络参数，如 IP 地址、本地 DNS 服务器等。DHCP 客户端会覆盖 /etc/resolv.conf 文件以包含来自 DHCP 服务器的信息。

为了在不担心 DHCP 的情况下将我们的信息添加到 /etc/resolv.conf 中，可以在路径 /etc/resolvconf/resolv.conf.d/head 添加以下条目：

```
Add the following entry to /etc/resolvconf/resolv.conf.d/head
  nameserver 10.9.0.53

Run the following command for the change to take effect
$ sudo resolvconf -u
```

文件的内容将被添加到动态生成的解析器配置文件的开头。通常，这是只是一个注释行（在 /etc/resolv.conf 中的注释来自于这个头文件）。

完成用户机器的配置后，请重复任务 5 中的测试命令，但这次不需要包含 @<ip>。注意高亮显示的 IP 地址以检查响应是否来自你的容器。如果设置不成功，则 IP 地址会不同。

```
$ dig abc.example.edu

...
;; ANSWER SECTION:
abc.example.edu.  259200   IN A  1.2.3.7
```

```
;; Query time: 3 msec
;; SERVER: 10.9.0.53#53(10.9.0.53)
...
```

★★注意（非常重要）★★. 完成实验后，请务必从 /etc/resolvconf/resolv.conf.d/head 中删除添加的条目，否则这将对未来实验造成许多问题。如果你将来在连接互联网时遇到麻烦，但你的网络是正常的，则可能是你忘记了删除这个条目，它破坏了 DNS 查询过程。我经常遇到这种情况。

8 任务 7：使其更接近现实

在这个任务中，我们将通过为我们的微型 DNS 系统添加以下名称服务器或记录来使其更加真实：

- 添加 2 个更多的根服务器。实际世界中的 DNS 系统有 13 个不同的根服务器的 IP 地址，在这个实验室中我们创建 3 个。

- 选择你 root 服务器注册的两个 TLD，并分别为它们建立二级域名，使用你的姓氏作为域名称。例如，如果你的姓氏是 Smith，你需要为 smith.<tld1> 和 smith.<tld2> 域构建名称服务器。

将这两个域名都托管在同一名称服务器上。

9 提交

你需要提交一份带有截图的详细实验报告来描述你所做的工作和你观察到的现象。你还需要对一些有趣或令人惊讶的观察结果进行解释。请同时列出重要的代码段并附上解释。只是简单地附上代码不加以解释不会获得学分。

清理 . 这是另一个关于从 /etc/resolvconf/resolv.conf.d/head 中删除添加条目的提醒。