DNS 安全扩展 (DNSSEC) 实验

1 实验概述

为了保护 DNS，开发了域名系统安全扩展 (DNSSEC)。DNSSEC 是一组对 DNS 的扩展，旨在为 DNS 数据提供身份验证和完整性检查。通过 DNSSEC，所有来自受保护区域的答案都是数字签名的。通过检查数字签名，DNS 解析器能够检查信息的真实性。通过这种机制，DNS 缓存投毒攻击将被击败，因为任何假数据，无论来自伪造的响应数据包还是来自权威名称服务器，都将被检测到，因为它们将无法通过签名检查。

本实验的目的是帮助学生理解 DNSSEC 的工作原理。学生们将获得一个简化的 DNS 基础设施，他们的任务是配置每个名称服务器，以便它们都支持 DNSSEC。本实验涵盖以下主题：

- DNS 及其工作原理
- DNSSEC 及其工作原理
- 密钥管理、公钥、数字签名

阅读材料和视频 关于 DNS 协议和攻击的详细信息可以在以下内容中找到：

- SEED 书的第 11 章，*Internet Security: A Hands-on Approach*, 3rd Edition, by Wenliang Du. 详情请见 `https://www.handsonsecurity.net`.
- SEED 讲座的第 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 实验环境设置

在实验环境中，我们提供一个迷你 DNS 基础设施，包括一个根服务器、一个顶级域服务器（`edu`）、一个域服务器（`example.edu`）和一个本地 DNS 服务器。这些名称服务器都托管在一个容器内。为了简化，我们将所有容器放在同一个局域网内（见图 1）。这是一个简化的设置，因为在现实世界中，这些名称服务器并不在同一个局域网内。然而，这些名称服务器的配置与现实世界中的配置类似，无论它们是在同一个局域网内还是分散在互联网上。

如果学生在此之前完成了 DNS 基础设施实验，他们可以选择使用 DNS 基础设施实验中的设置，而不是使用本实验中的简化设置，因为该设置使用互联网模拟器作为基础，更加真实。

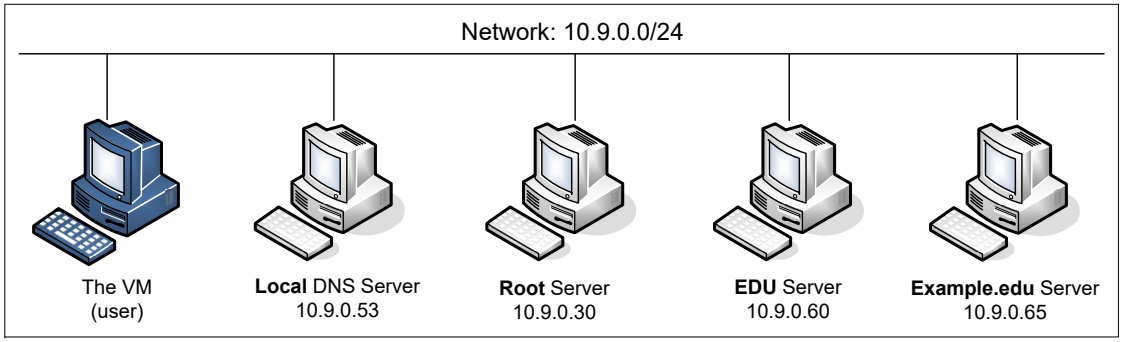

图 1: 实验环境设置

软件安装 我们将使用一些在 SEED 虚拟机中未安装的工具。请使用以下命令安装软件：

```
$ sudo apt-get install bind9utils
```

本实验所需的 DNSSEC 命令包含在上述安装的软件包中。我们只在主机虚拟机中安装它们，而不在容器内安装。然而，命令的结果需要复制到相应的容器中。我们可以使用"docker cp" 命令在主机和容器之间复制文件：

```
// 复制文件xyz到容器的/tmp文件夹
$ docker cp xyz <docker id>:/tmp

// 从容器复制文件/tmp/abc
$ docker cp <docker id>:/tmp/abc .
```

3 任务 1：设置 example.edu 域

在此任务中，我们将进入 example.edu 名称服务器的容器文件夹。此容器托管 example.edu 域。我们将修改此文件夹中的文件，以便名称服务器可以支持 DNSSEC 查询。

3.1 任务 1.a：为 example.edu 服务器生成密钥

首先，我们需要生成两个密钥对，一个称为区域签名密钥（ZSK），另一个称为密钥签名密钥（KSK）。每个密钥由公钥和私钥组成。ZSK 用于签名区域记录，而 KSK 用于签名 ZSK。

这种双密钥方案在密钥管理中相当常见。实际签署记录的密钥（即 ZSK）可以频繁更换以实现更好的安全性。如果只使用一个密钥，当该密钥发生更改时，密钥管理系统中存储的数据需要更新。在 DNSSEC 中，父区域存储有关此密钥的信息，因此需要更新父区域文件。

为了避免这种开销，引入了第二个密钥。这个密钥称为密钥签名密钥，即它仅签署区域文件中的 ZSK 记录，而其他记录仍由 ZSK 签名。KSK 的信息提供给父区域，但 KSK 不经常更改。当 ZSK 更改时，我们所需做的就是使用 KSK 为 ZSK 生成新的签名。

ZSK 和 KSK 都是公钥对，但由于它们的角色和生命周期不同，通常 KSK 密钥设置得比 ZSK 密

钥更强，即密钥大小更长。以下命令生成一个使用 RSASHA256 算法、密钥大小为 1024 的 ZSK 密钥。第二个命令使用相同算法生成 KSK 密钥，但将密钥大小加倍。如果您的操作系统找不到该命令，说明您忘记安装所需的软件（请参见实验设置部分）。

```
dnssec-keygen -a RSASHA256 -b 1024 example.edu
dnssec-keygen -a RSASHA256 -b 2048 -f KSK example.edu
```

第二个命令生成 KSK。-f 选项在密钥文件中放置 257，表示这是 KSK。如果没有此选项，则默认值 256 会放入密钥文件，表示这是 ZSK。

3.2 任务 1.b：签名 example.edu 域的区域文件

我们将使用以下 dnssec-signzone 命令签署区域文件。

```
dnssec-signzone -e 20501231000000 -S -o example.edu example.edu.db
```

使用-S 选项，该命令将查找指定文件夹中的区域签名密钥和密钥签名密钥（我们没有使用-K 选项指定密钥文件夹，因此默认是当前文件夹）。如果您在指定文件夹中有多个密钥对，该命令将使用每个密钥生成签名，因此您将为每条记录生成多个签名。

-e 选项指定签名的过期时间。默认情况下，过期时间仅为一个月。因此，如果您不使用-e 选项，当您一个月后检查设置时，它将不再有效。对于学生来说，一个月应该足够，但对于可能在未来重用答案的教师来说，设置较长的过期时间可以避免不必要的麻烦。过期时间的格式为 YYYYMMDDHHMMSS。在例子中，我们将过期时间设置为 2050 年。

一旦区域文件签名完成，将生成一个以.signed 结尾的新区域文件。查看该文件并描述新条目是如何添加到原始区域文件中的。请解释它们的目的。

这个新区域文件将由我们的名称服务器使用，因此我们将修改 named.conf.seedlabs 文件，以告诉名称服务器使用此文件作为其区域文件。见以下示例：

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

3.3 任务 1.c：测试

使用 docker-compose 命令，我们可以构建并启动所有容器。然后使用以下 dig 命令进行测试。该命令允许我们直接向服务器（@server）请求特定域或主机 name 的特定 type DNS 记录。通过使用 +dnssec 标志，我们在查询的额外部分的 OPT 记录中设置 DNSSEC OK 位（DO），请求服务器返回相关的 DNSSEC 记录，例如签名。

```
一般格式:
$ dig @server name type +dnssec

示例:
```

```
$ dig @10.9.0.65 example.edu DNSKEY +dnssec
```

运行以下命令以获取来自 example.edu 名称服务器的不同类型的记录。为每个响应中的记录提供解释。在我们的设置中，10.9.0.65 是分配给 example.edu 名称服务器的 IP 地址。

```
$ dig @10.9.0.65 example.edu DNSKEY +dnssec
$ dig @10.9.0.65 example.edu NS +dnssec
$ dig @10.9.0.65 www.example.edu A +dnssec
```

4 任务 2：设置 edu 服务器

在此任务中，我们将设置 edu 服务器。说明与任务 1 中的类似。

4.1 任务 2.a：查找并添加 DS 记录

在签署 example.edu 的区域文件后，为密钥签名密钥生成 DS 记录。默认情况下，它放置在名为 dsset-<xyz> 的文件中，其中 <xyz> 是区域名称。在我们的案例中，dsset-example.edu. 是文件名。

DS（Delegation Signer）记录保存已委派区域的名称。它引用子委派区域中的 DNSKEY 记录。DS 记录应与委派 NS 记录一起放置在父区域中。以下是 DS 记录的示例：

```
example.edu.   IN DS 10246   8  2   563D...(省略)...1D59D1
                      ①              ②
```

DS 记录包含一个密钥标签（标记为①），这是一个短的数值，标识引用的密钥。该密钥是域的密钥签名密钥（KSK）。DS 记录中最重要的元素是摘要（标记为②），这是 KSK 的一种单向哈希值。通过将此摘要值放入父区域（即 edu 区域），可以验证子委派区域 KSK 的完整性。这是 DS 记录的主要目的。

4.2 任务 2.b：设置 edu 服务器

遵循任务 1 中的说明，为该域生成 ZSK 和 KSK 密钥，然后使用这些密钥签署区域文件。在签署区域文件之前，我们需要向其中添加一个条目。该条目是子委派区域的 DS 记录。在我们的案例中，example.edu 区域被委派给另一个名称服务器，因此其密钥签名密钥的摘要必须包含在 edu 区域中；这样可以验证密钥的完整性。

我们可以将 DS 记录的内容复制并粘贴到 edu 的区域文件中，或使用以下 INCLUDE 条目来包含该文件。后一种方法对实验更方便，因为即使我们更改区域的密钥签名密钥，DS 记录的文件名也保持不变，因此无需更改父区域的区域文件。

```
$INCLUDE ../edu.example/dsset-example.edu.
```

4.3 任务 2.c：测试

构建并运行所有容器。然后运行以下命令以从 edu 名称服务器获取不同类型的记录。在我们的设置中，10.9.0.60 是分配给 edu 名称服务器的 IP 地址。

```
$ dig @10.9.0.60 edu DNSKEY +dnssec
$ dig @10.9.0.60 edu NS +dnssec
$ dig @10.9.0.60 example.edu +dnssec
```

5 任务 3：设置根服务器

设置根服务器的过程与前一个任务相同。记得在签署区域之前，将 edu 区域的 DS 记录添加到区域文件中。

测试。 运行以下命令以从根名称服务器获取不同类型的记录。提供对响应中每个记录的解释。在我们的设置中，10.9.0.30 是分配给根名称服务器的 IP 地址。

```
$ dig @10.9.0.30 . DNSKEY +dnssec
$ dig @10.9.0.30 . NS +dnssec
$ dig @10.9.0.30 edu +dnssec
$ dig @10.9.0.30 example.edu +dnssec
```

6 任务 4：设置本地 DNS 服务器

当计算机需要从主机名（或反之）解析 IP 地址时，它会向其助手发送请求，该助手称为本地 DNS 服务器（它不必是本地的）。本地 DNS 服务器将进行整个 DNS 解析过程，然后将结果发送回计算机。

在 DNSSEC 中，每个名称服务器向客户端提供自己的公钥（包括 ZSK 和 KSK），以便客户端可以使用 KSK 验证 ZSK 上的签名，然后使用 ZSK 验证每条记录上的签名。那还有一个不确定的事情：如何验证 KSK 的真实性？这就是父区域中的 DS 记录的目的。父区域将为其子区域提供 DS 记录，而此 DS 记录用于验证子区域的 KSK。

由于根服务器没有父区域，我们如何验证根服务器的真实性？根服务器的公钥是信任的根，因此它们是 DNSSEC 基础设施中最重要的密钥，被称为*信任锚*。它们必须通过安全的方式获得。

BIND 9 内置有 DNSSEC 信任锚，但可以通过/etc/bind/bind.keys 中的内容进行覆盖。在此任务中，我们将把根服务器的公钥放入此文件中。确保在此处使用 KSK，而不是 ZSK。

```
trust-anchors {
    . initial-key 257 3 8 " <根的密钥签名密钥> ";
};
```

我们还需要在本地 DNS 服务器上启用 DNSSEC，以便它将进行 DNSSEC 验证。我们只需修改本地 DNS 服务器容器的 named.conf.options 文件，将 DNSSEC 条目更改为以下内容：

```
dnssec-validation auto;
dnssec-enable yes;
```

测试。 启动本地服务器容器后，运行以下命令（10.9.0.53 是我们设置中分配给本地 DNS 服务器的 IP 地址）。

```
$ dig @10.9.0.53 www.example.edu +dnssec
;; 收到答案:
;; ->>HEADER<<- 操作: 查询, 状态: NOERROR, id: 37130
;; 标志: qr rd ra ad; 查询: 1, 答案: 2, 权威: 0, ...

;; 答案部分:
www.example.edu.        259200  IN      A       1.2.3.5
www.example.edu.        259200  IN      RRSIG   ... <签名>
```

如果一切设置正确，您应该能够获得答案，即www.example.edu的 IP 地址，如果回复是可信的。请注意 ad 标志，这意味着*真实数据*。如果设置了该标志，则回复中的所有记录都是可信的，即 DNSSEC 验证成功。如果未启用 DNSSEC，您仍会收到答案，但此标志不会被设置。

如果回复不真实，您将收到错误消息。请设计一个实验，以演示在回复假时，本地 DNS 服务器将能够检测到并显示错误消息。您无需在此实验中进行实际攻击。您可以直接修改一些现有记录而不重新生成签名（您也可以损坏签名）。

7 任务 5：添加另一个域名称服务器

假设您在 edu 域中拥有一个域名，并希望为该域建立一个启用 DNSSEC 的名称服务器。我们已经为此创建了一个备用容器（该容器中已安装 BIND 9 服务器）。您的任务是配置该名称服务器，使其支持 DNSSEC。域名应包含您的姓氏和进行本实验的年份。例如，如果您的姓氏是 Smith，年份是 2022，则域名应为 smith2022.edu。您不需要在现实世界中实际拥有此域名；我们只在实验环境的 DNS 基础设施中建立它。

8 作业提交

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。