DNS雷达系统设计方案

# 一、概述

DNS雷达系统（以下简称“系统”）用来对于指定的黑域名进行非递归探测，获取黑域名的活动区域，并及时对用户进行报警。

系统内核总体流程如下：



# 二、模块设计

## 1.IPFinder.jar

IPFinder.jar用来进行全国IP地址段的归属地及ISP信息收集，并将结果录入数据库。程序流程如下：



程序使用单线程执行，每次执行时会清空数据库原有记录，并对所有CIP地址段进行探测。程序可由crontab任务设定为定期执行。

程序配置文件问JSON格式，具备以下信息：

|  |  |  |
| --- | --- | --- |
| 属性 | 类型 | 说明 |
| StartIP | LONG | 计算起始IP |
| EndIP | LONG | 计算结束IP |
| Interval | Int | 计算步长 |
| sampleMethod | Enum | 抽样方法枚举(random/peak) |
| Lgnore | List<long[,]> | 排除段列表 |
| saveCountry | Enum | 要保存的国家(all/china等，其中all表示不使用该条件) |
| saveProvince | Enum | 要保存的省份(all/shangdong等，其中all表示不使用该条件) |
| saveRegion | Enum | 要保存的自治域(all/等，其中all表示不使用该条件) |
| SaveISP | Enum | 要保存的ISP(其中all表示不使用该条件) |
| Database | Enum | 数据库类型(oracle/mysql/file) |

## 2.DNSFinder.jar

DNSFinder.jar用来根据不同的ISP,使用ZAM探测IPFinder.jar计算出的IP地址，并配合抓包程序获取其中的DNS服务器。程序流程如下：



DNSFinder.jar程序使用多线程设计，每个线程对应一条虚拟网卡线路。每次执行时会清空数据库原有DNS记录并会对所有IP地址进行ZMP探测。程序会将探测到的结果保存为一份数据一份文件（文件的命名方式为：省份\_ISP.dns，格式为CSV文件）。程序可由crontab任务设定为定期执行。

程序启动时需要传入需要进行ZAMP请求的域名，格式为:

Java –jar DNSFinder.jar [domainName]

说明:DNS服务器获取方法:

1. 当ZMAP程序执行结束后，到数据库DNSLOG表中检索域名为ZMPDomain记录。
2. 将得到的地址列表与ZMAP请求的地址列表相对比，在其中的即为需要的DNS服务器
3. DNS服务器分线路方法，首先取得DNS服务器的CIP然后以CIP为条件在RADAR\_ZMAPPOOL表中获取省份，ISP信息，并根据信息在DNSQUERYSERVERLIST表中取得LISTID，最后将信息插入DNSQUERYSERVER表中。

## 3.DnsRadarService.war

DnsRadarService.war使用Restful方式设计，以API方式提供各种借口。具体服务如下：

/\*\*

\* 线路列表查询服务

\* **@param** apikey 用于身份验证

\* **@return** 线路列表JSON

\*/

@GET

@Path("/link")

@Produces(MediaType.***APPLICATION\_JSON***)

**public** RadarLink listLink(@HeaderParam("apikey") String apikey)

/\*\*

\* 新建链路列表

\* **@param** apikey

\* **@param** listName

\* **@param** active

\* **@param** describe

\* **@param** vpnip

\* **@param** country

\* **@param** province

\* **@param** isp

\* **@return** 执行结果

\*/

@POST

@Path("/link")

@Produces(MediaType.***APPLICATION\_JSON***)

**public** ServiceResponse addLink(@HeaderParam("apikey") String apikey,

@FormParam("listName") String listName,

@FormParam("active") **boolean** active,

@FormParam("describe") String describe,

@FormParam("vpnip") String vpnip,

@FormParam("country") String country,

@FormParam("province") String province, @FormParam("isp") String isp)

/\*\*

\* 启用链路

\* **@param** apikey

\* **@param** linkid

\* **@return**

\*/

@PUT

@Path("/link/enable/{linkid}")

@Produces(MediaType.***APPLICATION\_JSON***)

**public** ServiceResponse enableLink(@HeaderParam("apikey") String apikey,@PathParam("linkid")**long** linkid)

/\*\*

\* 停用链路

\* **@param** apikey

\* **@param** linkid

\* **@return**

\*/

@PUT

@Path("/link/disable/{linkid}")

@Produces(MediaType.***APPLICATION\_JSON***)

**public** ServiceResponse disableLink(@HeaderParam("apikey") String apikey,@PathParam("linkid")**long** linkid)

## 4. DNSValidator.jar

DNSValidator.jar用来验证DNS非递归探测的结果。程序流程如下:



程序提供单次执行和循环执行两种方式。在使用循环方式启动时应将程序休眠间隔以参数形式传入程序（单位：毫秒）。

# 三、数据库设计

## 1. Radar\_ZMAPPOOL

|  |  |  |  |
| --- | --- | --- | --- |
| 表名: Radar\_ZMAPPOOL | | | |
| 说明:用来存储需要进行ZAMP探测的IP地址信息 | | | |
| 字段名 | 类型 | 键 | 说明 |
| IPADDR | LONG | PK | IP地址 |
| CIPADDR | LONG |  | CIP地址 |
| COUNTRY | VARCHAR2(256) |  | 国家 |
| PROVINCE | VARCHAR2(256) |  | 省份 |
| ISP | Varchar2(256) |  | ISP |

## 2. DNSQUERYSERVERLIST

|  |  |  |  |
| --- | --- | --- | --- |
| 表名: DNSQUERYSERVERLIST | | | |
| 说明:用来存储线路列表信息 | | | |
| 字段名 | 类型 | 键 | 说明 |
| LISTID | LONG | PK,序列 | 线路列表ID |
| LISTNAME | VARCHAR2(256) |  | 线路名称 |
| ACTIVE | BOOLEAN |  | 是否使用（1使用，0不使用） |
| DESCRIBE | VARCHAR2(256) |  | 描述 |
| STARTTIME | DATE |  | 开始时间 |
| VPNIP | VARCHAR2(256) |  | VPNIP地址 |
| COUNTRY | VARCHAR2(256) |  | 国家 |
| PROVINCE | VARCHAR2(256) |  | 省份 |
| ISP | Varchar2(256) |  | ISP |

## 3.DNSQUERYSERVER

|  |  |  |  |
| --- | --- | --- | --- |
| 表名: DNSQUERYSERVERLIST | | | |
| 说明:用来存储需要进行探测的DNS服务器信息 | | | |
| 字段名 | 类型 | 键 | 说明 |
| IPADDR | VARCHAR2(256) | PK | DNS服务器地址 |
| ADDRESS | VARCHAR2(256) |  | 物理地址 |
| ACTIVE | BOOLEAN |  | 是否使用（1使用，0不使用） |
| DESCRIBE | VARCHAR2(256) |  | 描述 |
| STARTTIME | DATE |  | 开始时间 |
| TYPE | VARCHAR2(256) |  | 类型 |
| LISTID | LONG | FK | 线路列表ID |

## 4.DNSLOG

|  |  |  |  |
| --- | --- | --- | --- |
| 表名: DNSLOG | | | |
| 说明:抓包程序结果表，用来进行DNS服务器获取及验证使用 | | | |
| 字段名 | 类型 | 键 | 说明 |
| LOGID | LONG | PK | 日志ID |
| IPADDR | VARCHAR2(256) |  | 请求IP |
| DNSNAME | VARCHAR2(256) |  | 请求域名 |
| TYPE | VARCHAR2(256) |  | 域名类型 |
| CNAME | VARCHAR2(256) |  | 域名别名 |
| TTL | Int |  | 域名请求TTL值 |
| DNSIPADDR | VARCHAR2(256) |  | 域名解析地址 |
| COUNTRY | VARCHAR2(256) |  | 国家 |
| ADDRESS | VARCHAR2(256) |  | 物理地址 |
| REQUESTTIME | DATE |  | 请求时间 |
| DNSTYPE |  |  |  |
| ISANALYSIS |  |  |  |

## 5.DNSQUERYHOST

|  |  |  |  |
| --- | --- | --- | --- |
| 表名: DNSQUERYHOST | | | |
| 说明:非递归探测域名表 | | | |
| 字段名 | 类型 | 键 | 说明 |
| NAME | VARCHAR2(256) | PK | 要探测的黑域名 |
| QUERYTYPE | SHORT |  | 请求类型 0非递归1递归 |
| ACTIVE | BOOLEAN |  | 是否开启0不开启1开启 |
| STARTTIME | DATE |  | 添加时间 |
| MEMO | VARCHAR2(256) |  | 备注 |
| INTERVAL | LONG |  | 时间间隔 |
| ENDTIME | DATE |  | 结束时间 |
| OPERATOR | VARCHAR2(256) |  | 操作员 |

## 6.DNSQUERYLOG

|  |  |  |  |
| --- | --- | --- | --- |
| 表名: DNSQUERYLOG | | | |
| 说明:非递归探测结果表 | | | |
| 字段名 | 类型 | 键 | 说明 |
| TRANID | LONG | PK，序列 |  |
| NAME | VARCHAR2(256) |  | 探测的域名 |
| DNSSERVER | VARCHAR2(256) |  | 探测到的DNS服务器 |
| RTT | LONG |  | 探测到的TTL值 |
| STARTTIME | DATE |  | 探测到的时间 |
| EFFECTIVE | BIT |  | 是否有效0无效，1有效，默认1 |

## 7.DNSANSWERLOG

|  |  |  |  |
| --- | --- | --- | --- |
| 表名: DNSANSWERLOG | | | |
| 说明:非递归探测结果详细信息表 | | | |
| 字段名 | 类型 | 键 | 说明 |
| TRANID | LONG | PK,FK,序列 |  |
| RNAME | VARCHAR2(256) |  | 探测的域名 |
| CNAME | VARCHAR2(256) |  | 域名对应的别名 |
| IPADDR | VARCHAR2(256) |  | 域名解析IP |
| CIPADDR | VARCHAR2(256) |  | 域名解析IP的CIP |
| BIPADDR | VARCHAR2(256) |  | 域名解析IP的BIP |
| TTL | LONG |  | 探测到的TTL值 |
| COUNTRY | VARCHAR2(256) |  | 国家 |
| ADDRESS | VARCHAR2(256) |  | 物理地址 |
| ASN | VARCHAR2(256) |  | 自治域 |

疑问：

1. ZAMP程序探测结束后，抓包程序是否可以完整的得到探测结果？如果不能怎么办？
2. 按照目前DNSQUERYHOST表结构如何实现某些域名只探测某些线路？