

php ip库

qq_42030417 于2022-07-04 16:32:54

tcp/ip 网络 服务器

QQWry.dat文件是显IP版QQ的数据库文件，用于获取对方IP及位置，纯真IP数据库也采用了这个格式，并沿用至今。

纯真IP库是民间自发收集、提交、聚合而来的数据库，囊括了国内外的大量IP数据，部分记录甚至比商业付费数据库更加准确。它的官网提供了记录提交和纠错的功能，来自全国各地的网友将不同地区的ISP及位置数据上传，管理员在统一整合后每7天更新一次。从2005年到现在的16年间，这个数据库已经聚合了超过六十万条IP记录。

纯真IP库是完全免费的，它的官网上有在线查询功能，同时也可以下载离线数据库用于低延迟场景，但数据不可用于商业用途。这些记录在稍加分析处理后能达到不错的效果，基本满足定位IP所处城市及ISP信息的需求，不过它目前只支持IPv4地址，在一些应用场景下稍显不足。

获取数据文件

纯真IP数据库的文件名为 qqwry.dat，这个文件在官网上并没有公开链接可以下载，官方只提供了一个Windows工具用于查询和升级数据库。因此，如果想在服务器上实现数据库的获取与升级，必须模拟官方工具的更新机制。

手动获取

如果仅用于临时测试，可以通过下载并安装纯真IP数据库查询器来得到这个文件，它内置了 qqwry.dat 文件，同时也具备自动更新机制。

你可以在官网下载最新版的Windows安装包，将下载的 setup.zip 压缩包解压，打开里面的 setup.exe，默认安装目录为 C:\Program Files (x86)\cz88.net\ip，已解密的 qqwry.dat 文件就放置在这个文件夹下。

点击工具的解压按钮可以将数据库导出为文本文件。

可以发现，每条记录均由起始IP、终止IP和两个数据段共四部分组成，且前后两条记录的IP范围是连续相接的，覆盖了从 0.0.0.0 到 255.255.255.255 的所有IPv4地址。

自动获取

纯真官网没有提供 qqwry.dat 的下载，但其Windows查询工具内置了数据库更新功能，可以通过分析它的行为机制来获取下载和解密的算法。

抓包获取下载源

对程序抓包时，检测到它会向 update.cz88.net 发起GET请求，分别下载 /ip/copywrite.rar 与 /ip/qqwry.rar 两个文件，使用以下命令来获取它们：

获取加密的源文件

```
shell> wget http://update.cz88.net/ip/copywrite.rar
```

```
...
```

```
shell> wget http://update.cz88.net/ip/qqwry.rar
```

```
...
```



自动化脚本

将上述流程封装为一个脚本，实现自动下载并解密，输出 qqwry.dat 文件。

shell脚本

```
shell> php -v
```

...PHP版本信息...

```
shell> vim qqwryUpdate.sh
```

写入以下内容

```
#!/bin/sh
```

```
cd dirname $0
```

```
mkdir -p qqwryTemp
```

```
cd qqwryTemp
```

```
wget http://update.cz88.net/ip/copywrite.rar
```

```
wget http://update.cz88.net/ip/qqwry.rar
```

```
cat > unlock.php <<EOF
```

```
<?php  $\backslash$ copywrite = file_get_contents("copywrite.rar");  $\backslash$ qqwry = file_get_contents("qqwry.rar");  $\backslash$ key =
unpack("V6",  $\backslash$ copywrite)[6]; for ( $\backslash$ i = 0;  $\backslash$ i < 512;  $\backslash$ i++) {  $\backslash$ key = (( $\backslash$ key * 2053) + 1) & 0xFF;  $\backslash$ qqwry[ $\backslash$ i] =
chr(ord( $\backslash$ qqwry[ $\backslash$ i]) ^  $\backslash$ key); }  $\backslash$ qqwry = gzuncompress( $\backslash$ qqwry);  $\backslash$ fp = fopen("qqwry.dat", "wb"); fwrite( $\backslash$ fp,
 $\backslash$ qqwry); fclose( $\backslash$ fp); ?>
```

```
EOF
```

```
php unlock.php
```

```
cd ...
```

```
cp -f qqwryTemp/qqwry.dat qqwry.dat
```

```
rm -rf qqwryTemp/
```

运行脚本即可自动获取 qqwry.dat 文件

```
shell> sh qqwryUpdate.sh
```

可以使用 crontab 等定时工具，按时运行脚本拉取更新，保持 qqwry.dat 文件一直处于最新版本

部署定时任务

```
shell> crontab -e
```

设置为每天00:00时运行更新脚本，具体参数自行更改

```
00 0 * * * /var/www/echoIP/backend/qqwryUpdate.sh
```

文件二进制结构

在得到 qqwry.dat 文件以后，使用程序自动分析结构、读取数据

一个典型的记录条目如下：

起始IP：42.83.64.0

终止IP：42.83.79.255

记录A：广东省广州市

记录B：电信天翼云计算数据中心

在文件结构上，qqwry.dat 可分为三部分，分别是文件头、记录区和索引区，文件头指出索引区的位置，索引区信息指明记录区的偏移量。

特殊记录

数据库的最后一条记录不包含IP信息，而是数据库的版本号，格式如下：

起始IP：255.255.255.0

终止IP：255.255.255.255

记录A：纯真网络

记录B: XXXX年XX月XX日IP数据

这一部分属于特殊IPv4段 240.0.0.0/4, 被标记为 SPECIAL-IPV4-FUTURE-USE-IANA-RESERVED, 即IANA特殊保留地址。对于这部分, 我们必须对其劫持并返回正确的结果, 一般标记为 IANA保留地址。同时, 也可以根据这一段信息来提取版本号, 格式为 YYYYMMDD, 用于标记当前数据库的版本信息。

代码示例

使用PHP实现, 操作被封装为 QQWry 类, 代码保存为 qqwry.php。

```
<?php

namespace classes;

// 数据来源: 纯真IP数据库 qqwry.dat
// 初始化类: new QQWry($fileName)
// 请求方式: getDetail($ip)
// 返回格式:
// {
//     "beginIP": IP段起始点
//     "endIP": IP段结束点
//     "dataA": 数据段1
//     "dataB": 数据段2
// }
//
// 请求版本: getVersion()
// 返回格式: YYYYMMDD

class QQWry
{
    private $fp; // 文件指针
    private $firstRecord; // 第一条记录的偏移地址
    private $lastRecord; // 最后一条记录的偏移地址
    private $recordNum; // 总记录条数

    public function __construct($fileName = EXTEND_PATH . 'data/qqwry.dat')
    { // 构造函数
        $this->fp = fopen($fileName, 'rb');
        $this->firstRecord = $this->read4byte();
        $this->lastRecord = $this->read4byte();
        $this->recordNum = ($this->lastRecord - $this->firstRecord) / 7; // 每条索引长度为7字
    }

    public function __destruct()
    { // 析构函数
        if ($this->fp) {
            fclose($this->fp);
        }
    }

    private function read4byte()
    { // 读取4字节并转为long
        return unpack('Vlong', fread($this->fp, 4))['long'];
    }

    private function read3byte()
    { // 读取3字节并转为long
        return unpack('Vlong', fread($this->fp, 3) . chr(0))['long'];
    }

    private function readString()
    { // 读取字符串
        $str = '';
        $char = fread($this->fp, 1);
        while (ord($char) != 0) { // 读到二进制0结束
            $str .= $char;
            $char = fread($this->fp, 1);
        }
    }
}
```

```

        return $str;
    }

    private function zipIP($ip)
    { // IP地址转为数字
        $ip_arr = explode('.', $ip);
        $tmp = (16777216 * intval($ip_arr[0])) + (65536 * intval($ip_arr[1])) + (256 * intval($ip_arr[2]));
        return pack('N', intval($tmp)); // 32位无符号大端序长整型
    }

    private function unzipIP($ip)
    { // 数字转为IP地址
        return long2ip($ip);
    }

    public function getVersion()
    { // 获取当前数据库的版本
        fseek($this->fp, $this->lastRecord + 4);
        $tmp = $this->getRecord($this->read3byte())['B'];
        return substr($tmp, 0, 4) . substr($tmp, 7, 2) . substr($tmp, 12, 2);
    }

    public function getDetail($ip)
    { // 获取IP地址区段及所在位置
        if (!filter_var($ip, FILTER_VALIDATE_IP, FILTER_FLAG_IPV4)) { // 判断是否为IPv4地址
            return null;
        }

        fseek($this->fp, $this->searchRecord($ip)); // 跳转到对应IP记录的位置
        $detail['beginIP'] = $this->unzipIP($this->read4byte()); // 目标IP所在网段的起始IP
        $offset = $this->read3byte(); // 索引后3字节为对应记录的偏移量
        fseek($this->fp, $offset);
        $detail['endIP'] = $this->unzipIP($this->read4byte()); // 目标IP所在网段的结束IP

        $tmp = $this->getRecord($offset); // 获取记录的数据A与数据B
        $detail['dataA'] = $tmp['A'];
        $detail['dataB'] = $tmp['B'];

        if ($detail['beginIP'] == '255.255.255.0') { // 去除附加信息
            $detail['dataA'] = 'IANA';
            $detail['dataB'] = '保留地址';
        }
        if ($detail['dataA'] == 'CZ88.NET' || $detail['dataA'] == '纯真网络') {
            $detail['dataA'] = '';
        }
        if ($detail['dataB'] == 'CZ88.NET') {
            $detail['dataB'] = '';
        }
        return $detail;
    }

    private function searchRecord($ip)
    { // 根据IP地址获取索引的绝对偏移量
        $ip = $this->zipIP($ip); // 转为数字以比较大小
        $down = 0;
        $up = $this->recordNum;
        while ($down <= $up) { // 二分法查找
            $mid = floor(($down + $up) / 2); // 计算二分点
            fseek($this->fp, $this->firstRecord + $mid * 7);
            $beginip = strrev(fread($this->fp, 4)); // 获取二分区域的下边界
            if ($ip < $beginip) { // 目标IP在二分区域以下
                $up = $mid - 1; // 缩小搜索的上边界
            } else {
                fseek($this->fp, $this->read3byte());
                $endip = strrev(fread($this->fp, 4)); // 获取二分区域的上边界
                if ($ip > $endip) { // 目标IP在二分区域以上
                    $down = $mid + 1; // 缩小搜索的下边界
                } else { // 目标IP在二分区域内
                    return $this->firstRecord + $mid * 7; // 返回索引的偏移量
                }
            }
        }
    }

```

```

    }
    return $this->lastRecord; // 无法找到对应索引，返回最后一条记录的偏移量
}

private function getRecord($offset)
{ // 读取IP记录的数据
    fseek($this->fp, $offset + 4);
    $flag = ord(fread($this->fp, 1));
    if ($flag == 1) { // dataA与dataB均重定向
        $offset = $this->read3byte(); // 重定向偏移
        fseek($this->fp, $offset);
        if (ord(fread($this->fp, 1)) == 2) { // dataA再次重定向
            fseek($this->fp, $this->read3byte());
            $data['A'] = $this->readString();
            fseek($this->fp, $offset + 4);
            $data['B'] = $this->getDataB();
        } else { // dataA无重定向
            fseek($this->fp, -1, SEEK_CUR); // 文件指针回退1字节
            $data['A'] = $this->readString();
            $data['B'] = $this->getDataB();
        }
    } else if ($flag == 2) { // dataA重定向
        fseek($this->fp, $this->read3byte());
        $data['A'] = $this->readString();
        fseek($this->fp, $offset + 8); // IP占4字节，重定向标志占1字节，dataA指针占3字节
        $data['B'] = $this->getDataB();
    } else { // dataA无重定向
        fseek($this->fp, -1, SEEK_CUR); // 文件指针回退1字节
        $data['A'] = $this->readString();
        $data['B'] = $this->getDataB();
    }
}
$data['A'] = iconv("GBK", "UTF-8", $data['A']); // GBK -> UTF-8
$data['B'] = iconv("GBK", "UTF-8", $data['B']);

```

```

private function getDataB()
{ // 从fp指定偏移获取dataB
    $flag = ord(fread($this->fp, 1));
    if ($flag == 0) { // dataB无信息
        return '';
    } else if ($flag == 1 || $flag == 2) { // dataB重定向
        fseek($this->fp, $this->read3byte());
        return $this->readString();
    } else { // dataB无重定向
        fseek($this->fp, -1, SEEK_CUR); // 文件指针回退1字节
        return $this->readString();
    }
}
}

```

调用示例，文件名为 demo.php，同目录下放置 qqwry.dat 数据文件。

```

<?php
include("qqwry.php"); // 引入代码
$demo = new QQWry('qqwry.dat'); // 初始化类
echo '数据库版本: ' . $demo->getVersion() . PHP_EOL;
$detail = $demo->getDetail('8.8.8.8'); // 调用查询函数
var_dump($detail); // 输出查询结果
?>

```

...

输出查询结果

```

array(4) {
    ["beginIP"]=>

```

```
string(7) "8.8.8.8"
["endIP"]=>
string(7) "8.8.8.8"
["dataA"]=>
string(6) "美国"
["dataB"]=>
string(66) "加利福尼亚州圣克拉拉县山景市谷歌公司DNS服务器"
}
```

本内容为PUDN经合法授权发布，文章内容为作者独立观点，不代表PUDN立场，未经允许不得转载。

相关推荐

php168分类信息的IP库

php168分类信息的IP库，赶紧下载吧，

PHP使用纯真数据IP库转换物理IP

PHP 使用纯真数据IP库转换物理IP，QQIP地址

php ip地址查询

php ip地址查询，ip地址查询接口，附带实例和ip地址库文件

ip库+php类_ip归属地查询

ip库+php类_ip归属地查询

QQip库查询插件

替换原IP库 可手工自己改代码，也可能安装程序进行安装 1、自动安装： 上传hkh整个目录到论坛根目录下，运行install.php进行安装 ...

java IP 库 ip

java获取ip,离线文件数据库，代码已写好

php忆库相册

php忆库相册

php面试题题库

这篇文章介绍的内容是关于最全最详细的PHP面试题，有着一定的参考价值，现在分享给大家，有需要的朋友可以参考一下

IpLocation:纯真IP库 - Composer PHP包

PHP实现 基于纯真IP库 根据IP地址查找对应的地理位置信息 #####更新的版本 本库附带的qqwry.dat文件是2014.11.20更新的，如你需...

php-ip:适用于PHP的IPv4IPv6操作库

PHP IP库 受Python 启发PHP的IPv4 / IPv6操作库。要求 PHP > = 7.0 启用了IPv6支持 GMP扩展名 () 安装 推荐的方法是安装lib。只...

易语言I利用PHP利用纯真IP库显示地址连网版

易语言I利用PHP利用纯真IP库显示地址连网版源码,I利用PHP利用纯真IP库显示地址连网版

ip-lib:PHP库可处理IPv4，IPv6和IP范围

IPLib是一个现代的，符合PSR的，受测试驱动的IP地址和子网操作库。它支持CIDR格式（如::1/128或127.0.0.1/32）和模式格式（如:::*...

QQ官网最新IP资料库

QQ官网最新IP资料库 PHP版QQ官网最新IP资料库

高效路由的PHP库.zip

php namespace FastRoute; use PHPUnit\Framework\TestCase; class RouteCollectorTest extends TestCase { public function testShortcu...

PHP实现IP判断所在城市 内带QQ地址库

PHP实现IP判断所在城市 内带QQ地址库

部分ip地址库

IPv4 地址归属地数据库(IPIP.NET 版)

PHP根据IP获取所在城市

PHP根据IP获取所在城市， 通过两种方式获取IP的城市 第一种能过SINA的接口 第二种使用纯真IP库存！（IP库已包含在压缩包内）

php 获取ip所在区域实例

php中获取ip所在区域的代码，获取信息包括城市和具体区域

caozha-ip(IP地址库调用系统)-PHP

caozha-ip是基于原生PHP写的一套完整的IP转地址模块，支持自动获取IP，也支持查询指定IP，同时支持输出json、jsonp、text、xml...

PHP在线IP查询（QQWry.Dat库）

内容索引:PHP源码,查询搜索,IP,纯真IP库 PHP在线IP查询，无MYSQL支持，使用的是最新的QQWry.Dat纯真IP数据库，简单测试查询...

kefu@pudn.com

407128440

客服 09:00-12:00 13:30-18:00（周一至周五）

ICP备案号：湘ICP备2021020032号 经营性网站备案信息 网络110报警服务 中国互联网举报中心 ©长沙创客万联科技有限公司 版权与免责声明 版权申诉

