

Pkav HTTP Fuzzer 使用手册 Ver 1.0

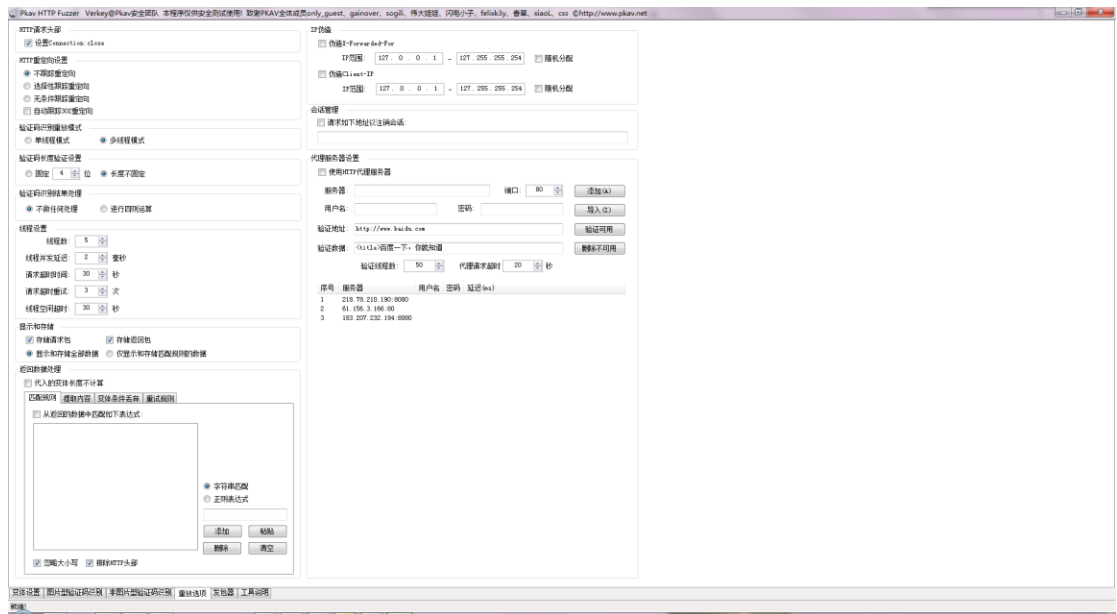
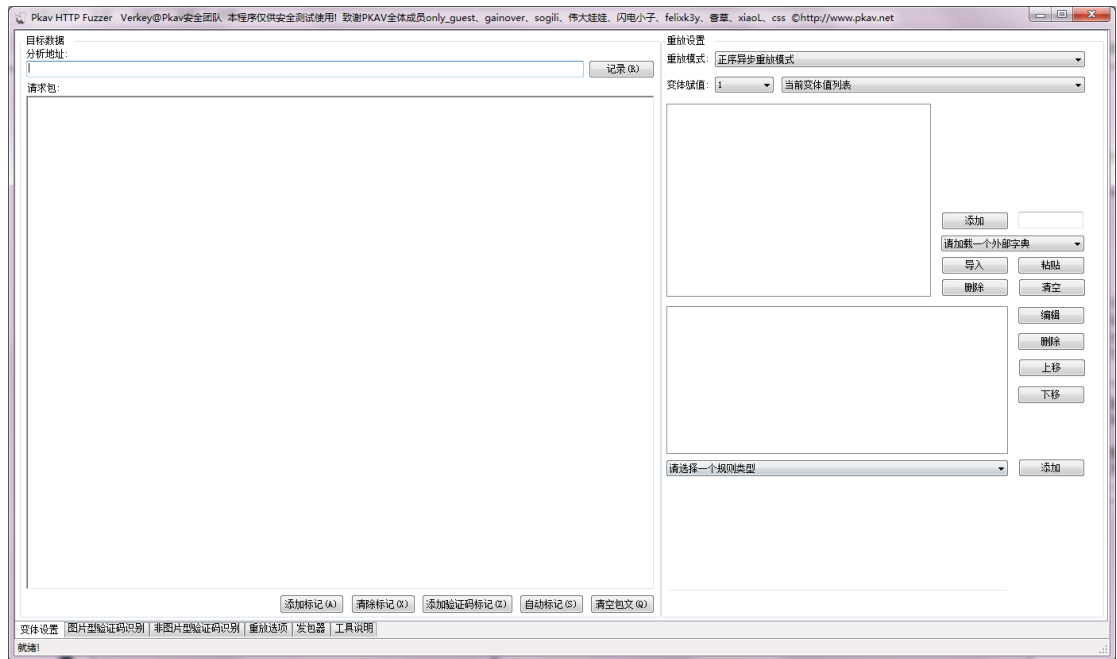
Pkav HTTP Fuzzer 是 Pkav 团队开发的一款安全测试工具，主要用于 WEB 站点安全测试。在使用它之前，请阅读并知晓如下声明：

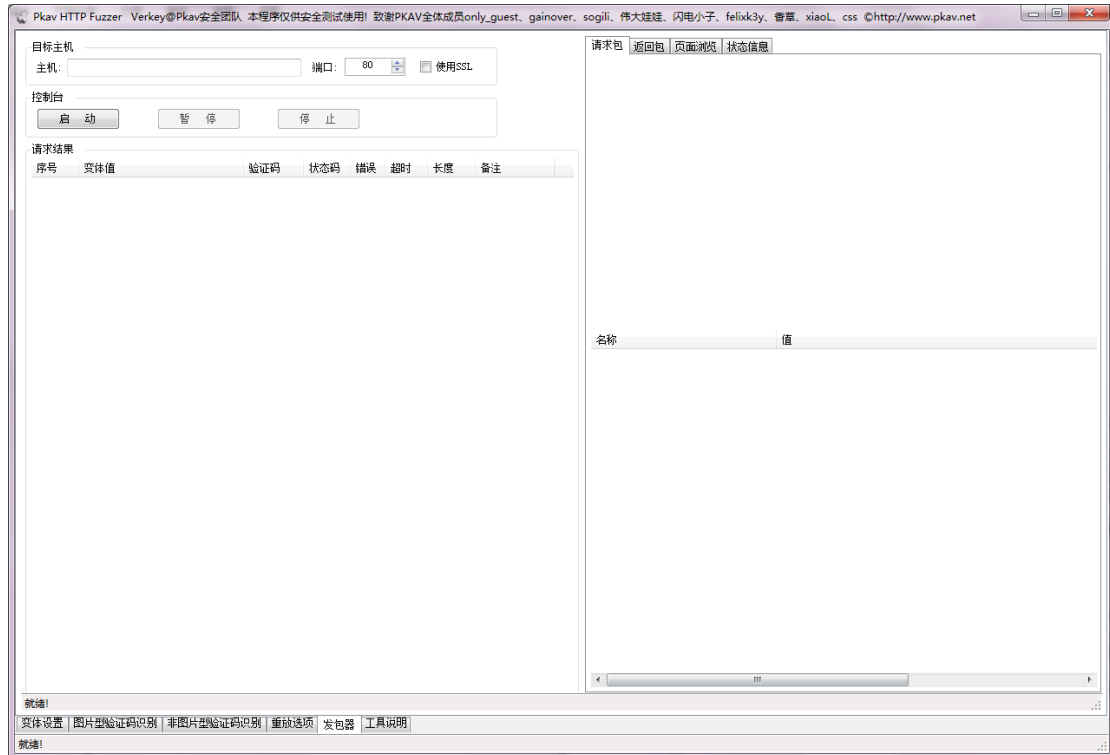
- 1、本工具为安全测试工具，用于安全测试。非正当使用本软件造成的法律纠纷，与我们无关。
- 2、本工具免费下载和使用，不存在破解版本和收费版本，不存在后门或病毒，有可能被杀毒软件误杀，请在 Pkav 官网下载。
- 3、本工具功能完整，Pkav 工具集中部分与之协同使用的工具(如 http tester)不包含在此工具内，暂不放出，敬请谅解！
- 4、如在使用中发现 bug，或您有好的意见或建议，请在 PKAV 官网留言给我们，我们会第一时间加以改进，然后发布。
- 5、Pkav 团队全体成员期待与大家一起学习和交流，共同进步！

Pkav HTTP Fuzzer 有如下特点：

- ✚ 支持 5 种模式的数据包重放。
- ✚ 支持 9 种模式的变体值类型。
- ✚ 支持 14 种变体值处理规则。
- ✚ 支持验证码变体标记。
- ✚ 支持图片型和非图片型的验证码识别，支持四则运算验证码。
- ✚ 可对返回数据进行各项处理。
- ✚ 支持变体值条件丢弃。
- ✚ 支持随机 IP 地址伪造。
- ✚ 支持批量 HTTP 代理轮换。
- ✚ 集成大量渗透测试所需的变体。
- ✚ 支持各选项参数实时调节。

界面部分截图如下：





Pkav HTTP Fuzzer 的通常使用流程：“获取重放数据包”->“标记数据包中的变体”->“设置数据包重放模式”->“设置变体的值来源”->“设置变体值的处理规则”->“设置变体值中需编码的字符”->“配置验证码识别(无验证码可不用配置)”->“配置重放选项”->“使用发包器发包”。

下面我们根据此流程来讲解 Pkav HTTP Fuzzer 的各种模式、功能的配置：

1.1.1.1 数据包获取

在“分析地址”的文本框中输入要截获数据包的 URL，点击“记录”按钮，HTTP Tester 会打开目标 URL，然后通过 HTTP Tester 截获所需的数据包，通过 HTTP Tester 的“发送到 HTTP Fuzzer”发送到 Pkav HTTP Fuzzer 的“请求包”文本框中。

HTTP Tester 暂时没有放出来，但是，您可以使用其他的抓包工具截获目标数据包，如 Burpsuite、charles 等等。在复制和粘贴数据包时，要注意数据包的格式，如 GET 请求最后有两个换行，POST 请求最后一行一般是提交的数据，不要随意增减。

1.1.1.2 标记数据包中的变体

渗透测试过程中，我们通常需要对某个数据包中的某参数的参数值进行修改，输入不同的攻击载荷进行反复提交和测试。在 Pkav HTTP Fuzzer 中，我们可以将该参数的参数值设置成一个“变量”，这里我们称之为“变体”，Pkav HTTP Fuzzer 会使用不同的攻击载荷替换这个“变体”，然后将修改后的数据包进行发送，并处理返回结果。

如何设置一个变体呢？很简单。选中要设置变体的数据包部分，点击“添加标记”来设置一个变体；点击“清除标记”将变体设置清除；点击“添加验证码标记”来将验证码处标记为验证码变体；点击“自动标记”来标记数据包中所有参数值；点击“清空包文”以清空文本框。

使用成对的“§”符号括起来的就是一个变体。“§ YZM §”是一个特殊的变体，它表示这里是一个可变化的验证码字符串。如果数据包中定义了验证码变体，那么必须设置验证码识别选项，如果没有，则无需设置验证码识别选项。

下图是一个登录数据包变体设置示例，定义了变体“§ 123456 §”和一个验证码变体“§ YZM §”：

目标数据
分析地址:

请求包:

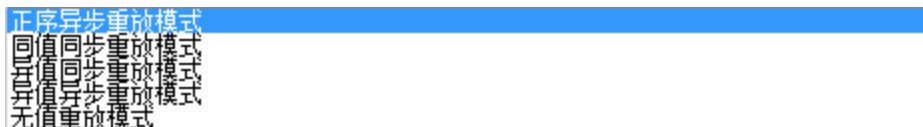
```
POST /admin/login.asp HTTP/1.1
Accept: application/x-ms-application, image/jpeg, application/xaml+xml, image/gif,
image/pjpeg, application/x-ms-xbap, */*
Referer: http://www.xxx.com.cn/admin/login.asp
Accept-Language: zh-CN
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; WOW64; Trident/5.0;
SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0;
.NET4.0C; .NET4.0E; InfoPath.3; KB974488)
Host: www.xxx.com.cn
Content-Length: 82
Connection: Keep-Alive
Cache-Control: no-cache
Cookie: ASPSESSIONIDAQCATABS=JMGIOOLAJCHIFKKMNEMDNEDM

action=login&UserName=admin&PassWord=§ 123456 §&checking=§ YZM §&Submit2=%C8%B7%C8%CF%
B5%C7%C2%BC
```

添加标记 (A) 清除标记 (X) 添加验证码标记 (Z) 自动标记 (S) 清空包文 (Q)

1.1.1.3 设置数据包重放模式

设置好了变体后，我们需要选择数据包使用哪种重放模式。Pkav HTTP Fuzzer 支持 5 种数据包重放模式，分别是“正序异步重放模式”、“同值同步重放模式”、“异值同步重放模式”、“异值异步重放模式”、“无值重放模式”。如下图所示：



各重放模式的介绍如下：

1) **正序异步同放模式：**

按照变体在数据包中的顺序，逐个进行替换，如存在变体 A=1、B=2、C=3，设置了该模式，变体为字母 X、Y，那么先重放 A=X、B=2、C=3，之后是 A=1、B=X、C=3；A=1、B=2、C=X，X 赋值完后现在赋值 Y 了，就是 A=Y、B=2、C=3；A=1、B=Y、C=3；A=1、B=2、C=Y。

由于该模式的变体值是按顺序逐个替换的，未轮到或已经轮换过的变体仍然使用数据包中的原值，故即使设置了多个变体，但只需对一个变体赋值即可。

2) **同值同步重放模式：**

所有的变体都使用同一个值，如存在变体 A=1、B=2、C=3，设置了该模式，变体为字母 X、Y、Z，那么重放结果为 A=X、B=X、C=X；A=Y、B=Y、C=Y；A=Z、B=Z、C=Z。

由于该模式的所有变体都使用的是同一个值，每次重放同时替换所有变体，故变体赋值的时候只需对一个变体赋值。

3) **异值同步重放模式：**

每个变体对应的不同的值同时重放。如存在变体 A=1、B=2，设置了变体 A 的值为 1 到 9 的数字，变体值 B 的值为 a-z 的字母，那么重放结果为：A=1、B=a；A=2、B=b；A=3、B=c... 最后为 A=9、B=i。因为最短的那个变体数只有 9 个(1-9)，因此即使变体 B 没有完，但是已经不能重放了。

4) **异值异步重放模式：**

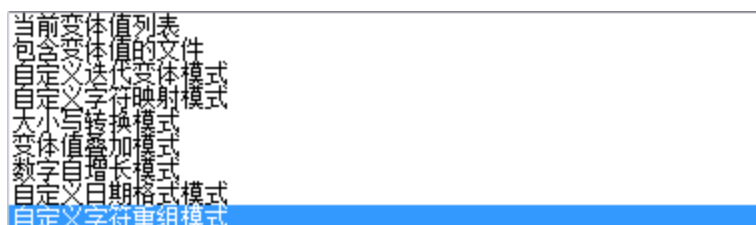
每个变体对应的不同的值然后非同步轮流重放。如存在变体 A=1、B=2，设置变体 A 的值为 1 到 9 的数字，设置变体 B 的值为 a-z 的字母，那么重放结果为：A=1、B=a；A=1、B=b、A=1、B=c... 一直到 A=1、B=z，这个时候 B 的值已经重放完了，接下来就是 A=2、B=a；A=2、B=b；A=2、B=c... 直到 A 和 B 的值都组合赋值完。

5) **无值重放模式：**

不设置变体值，只是不断重放原数据包。

1.1.1.4 设置变体的值来源

既然变体的值是可变的，那么它的值来源于哪儿呢？Pkav HTTP Fuzzer 支持 9 种变体赋值方式，如下图所示：



分别是“当前变体值列表”、“包含变体值的文件”、“自定义迭代变体模式”、“自定义字符映射模式”、“大小写转换模式”、“变体值叠加模式”、“数字自增长模式”、“自定义日期格式模式”、“自定义字符重组模式”。

如果在数据包中定义了多个变体，那么怎么针对多个变体赋值呢？通过切换“变体赋值”右侧下拉框中的数字，给数据包中的变体赋值。Pkav HTTP Fuzzer 给多个变体赋值的顺序是根据变体在数据包中的顺序决定的。选择“变体赋值”下拉框中的数字，如 1，表示给第 1 个变体赋值，如 2，表示给第 2 个变体赋值。如定义了多个变体，但是下拉框中只有 1，那么说明当前的模式可能是“正序异步同放模式”或“同值同步重放模式”，这两个模式只需给一个变体赋值，而该值是应用到全部定义的变体的。

各变体赋值模式的介绍如下：

1) 当前变体值列表

使用当前列表中的值为变体赋值，可以通过手工添加、导入、粘贴等方式将字典导入列表中。默认已经内置了多个外部字典，从下拉框中选择即可导入。

变体赋值: 1 当前变体值列表

admin
admin12
admin888
admin8
admin123
sysadmin
adminxxx
adminx
6kadmin
base
feitium
admins
root
roots
test

添加

请加载一个外部字典

导入 粘贴

删除 清空

2) 包含变体值的文件

如果是给变体赋值的是外部的一个字典文件，可以点击“浏览”选择该文件，Pkav HTTP Fuzzer 会逐行读取该文件对变体进行赋值。

变体赋值: 1 包含变体值的文件

浏览..

3) 自定义迭代变体模式

提供了 1-8 个列表，每个列表中可以添加或导入数据。“列表 1”至“列表 8”中的数据是迭代生成的，数据直接可以通过“分隔符”进行分割。比如“列表 1”中的数据为“1、2、3”，“列表 2”中的数据为“a、b、c”，“列表 1”设置的分割符为“:”，那么最终给变体赋值依次为“1:a”、“1:b”、“1:c”、“2:a”、“2:b”、“2:c”、“3:a”、“3:b”、“3:c”，无不使用分隔符请留空。

应用场景：针对 HTTP 认证的暴力破解，最终应赋值给变体的是“用户名:密码”的 Base64 编码字符串。那么，我们可以将用户名导入到“列表 1”中，在“分割”文本框中填入冒号“:”，然后在“列表 2”中导入密码，最后在规则列表中添加“变体编码 -> Base64 编码”。配置完成后，Pkav HTTP Fuzzer 会使用“列表 1”中的每个用户名 + “:” + “列表 2”中的每个密码然后进行 Base64 编码，然后赋值给变体进行重放。

该模式最高支持 8 级迭代，不建议导入超大量数据进行多级迭代。

变体赋值: 1 自定义迭代变体模式

分割:

1 2 3 4 5 6 7

1
2
3

添加

请加载一个外部字典

导入 粘贴

删除 清空

4) 自定义字符映射模式

提供了一个“变体值表”和一个“映射表”，“映射表”中提供了针对于“变体值表”中数据的字符映射关系，如下图所示，“a”对应的是数字 4，“b”对应的是数字 8，“e”对应的是数字 3。当变体值中的数据为“abcd”的时候，赋给变体的值将依次被映射为 “4bcd”、“a8cd”、“48cd”、“abcd”，因为 d 没有映射关系，所以没有变，该模式会重放数据的原值，这也就是为什么最后赋值了“abcd”。

默认是不区分大小写映射的，如果勾选了“大小写敏感”选项框，那么字符直接的映射是区分大小写的，如字符“A”将不会被映射成 4。

变体赋值: 1 自定义字符映射模式

变体值表 映射表

a	>	4
b	>	8
e	>	3
g	>	6
i	>	1
o	>	0
s	>	5
t	>	7

☐ 大小写敏感

添加

请加载一个外部字典

导入 粘贴

删除 清空

5) 大小写转换模式

提供了一个变体值列表，Pkav HTTP Fuzzer 会将变体值列表中的数据根据设置转换成大写、小写等数据然后对变体赋值。如变体值列表中的数据为“aBc”，勾选了“原变体值”、“转换成小写”、“转换成大小”等选项框，那么最终变体的赋值依次为“ABC”、“abc”和“aBc”。

变体赋值：1

大小写转换模式

aBc

☒ 原变体值

☒ 转换成小写

☒ 转换成大写

添加

请加载一个外部字典

导入

粘贴

删除

清空

6) 变体值叠加模式

变体值叠加模式是指提供一个基本字符串，根据设置来生成指定的 N 个基本字符串。如基本字符串为“abc”，叠加的起始次数为 1，最大次数为 10，递增 1，那么会依次生成“abc”、“abcabc”、

“abcabcabc” …… “abcabcabcabcabcabcabcabcabcabc”。如果递增值设置为 2，那么会依次生成 1、3、5、7、9 个“abc”。

变体赋值: 变体值叠加模式

基本字符串:

起始次数:

最大次数:

递增:

7) 数字自增长模式

数字自增长模式是指：指定一个数字增长的范围和增长幅度，生成指定范围内的数字。如设置的模式为顺序模式，数据范围为 0-100，增长量为 1，数据格式为十进制，那么会依次生成 0-100 的数字作为变体值。如果设置的模式为随机模式，数据范围为 100-200，数量为 10，格式为十进制，那么会随机生成 10 个 100-200 之间的十进制数字作为变体值。

如果希望生成的变体值为 0001、0002...1234 这样的格式，请勾选“补零”选项框，工具会自动补足数字前面的 0。

变体赋值: 1 数字自增长模式

段:

从: 0

到: 100

增长: 1

数量: 1

☒ 顺序模式 ☒ 十进制 ☒ 不补零
☐ 随机模式 ☐ 十六进制 ☐ 补零

8) 自定义日期格式模式

自定义日期格式模式是指: 指定一段时间范围和增长量, 按照设置的日期格式生成指定范围内的日期。如设置的时间段为“2015 年五月 1 日”至“2015 年八月 28 日”, 日期增长量为 1 天, 日期格式为“2015-05-01”, 那么生成的日期依次为 2015-05-01、2015-05-02……2015-08-28。允许自定义日期格式, 也可以在变体规则列表中添加“Unix 时间戳转换”规则将日期转化为 unix 时间戳。

变体赋值: 1 自定义日期格式模式

从: 2015 五月 1
到: 2015 八月 28
增长: 1 天
日期格式: ☒ 2015-05-01
☐ yyyy-mm-dd

9) 自定义字符重组模式

自定义字符重组模式是指: 指定一串基础字符串, 使用该字符串的每一个字符和其他的字符进行组合, 组合成指定长度的字符串。假如基字符串为“abc”, 最小长度设置为 1, 最大设置为 2, 那么可以组合的变体值为“a”、“b”、“c”、“aa”、“ab”、“ac”、“ba”、“bb”、“bc”、“ca”、“cb”、“cc”。

变体赋值: 1 自定义字符重组模式

基字符串:

abcdefghijklmnopqrstuvwxyz0123456789

最小长度: 4

最大长度: 4

1.1.1.5 设置变体值的处理规则

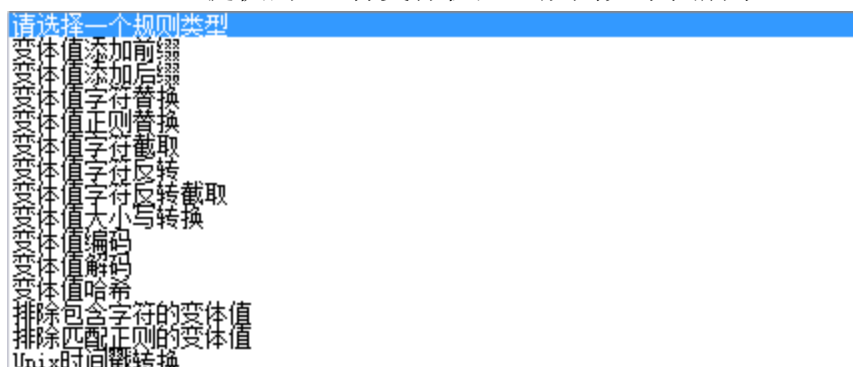
Pkav HTTP Fuzzer 提供了 14 种变体值处理规则, 可以组合使用 14 种变体值处理规则对变体值进行补充修正、格式转换、编码加解密等操作。Pkav HTTP

Fuzzer 根据“变体值处理规则列表”中的规则顺序进行解析和处理，未勾选的规则不起作用。

The screenshot shows a configuration window for Fuzzer. On the left, there is a list of rules with checkboxes:
- ☒ 添加前缀:a
- ☒ 添加后缀:b
- ☒ Base64编码
- ☒ 截取位置1起, 长度5的字符串
On the right, there are four buttons: 编辑 (Edit), 删除 (Delete), 上移 (Move Up), and 下移 (Move Down). At the bottom left, there is a dropdown menu labeled "请选择一个规则类型" (Please select a rule type). At the bottom right, there is a blue button labeled 添加 (Add).

上图是一个简单的变体值处理规则列表，它首先会在变体值前面添加字符“a”，在变体值末尾添加字符“b”，然后将变体值进行 Base64 编码，最后从 Base64 编码后的字符串的第 1 位开始截取 5 个长度的字符串。

Pkav HTTP Fuzzer 提供的 14 种变体值处理规则如下图所示：



各变体值处理规则的介绍如下：

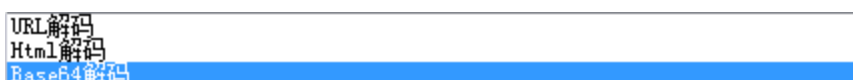
- 1) **变体值添加前缀**
在变体值前面添加指定字符串。
- 2) **变体值添加后缀**
在变体值末尾添加指定字符串。
- 3) **变体值字符替换**
将变体值中的字符串替换为指定的字符串。
- 4) **变体值正则替换**
将变体值中匹配指定正则表达式的部分替换为指定的字符串。
- 5) **变体值字符截取**
从变体值中截取指定长度的字符串。
- 6) **变体值字符反转**
将变体值进行反转。
- 7) **变体值字符反转截取**
将变体值反转后，再截取指定长度的字符串。
- 8) **变体值大小写转换**
将变体值进行大小写转换。
- 9) **变体值编码**

对变体值进行下图格式的编码：



10) 变体值解码

对变体值进行下图格式的解码：



11) 变体值哈希

对变体值进行 MD5、SHA 散列的哈希计算。

12) 排除包含字符的变体值

排除包含指定字符的变体值，该变体值将不会被使用。

13) 排除匹配正则的变体值

排除匹配指定正则表达式的变体值，该变体值将不会使用。

14) Unix 时间戳转换

将变体值转换为 unix 时间戳或将 unix 时间戳转换为日期。

1.1.1.6 设置变体值中需编码的字符

Pkav HTTP Fuzzer 默认会对变体值中的“/\=<>?+&*;:”等字符进行 URL 编码，如下图所示

☒ 对变体值中的如下字符进行编码：

/\=<>?+&*;:

如在 Fuzz 的时候不要对/进行编码，可以从上图的文本框中删除/。

1.1.1.7 设置图型验证码识别

Pkav HTTP Fuzzer 支持图形和非图形验证码的识别。在数据包中定义了验证码变体后，需要配置相关的验证码识别选项。如果是图片型验证码，那么在“图形验证码识别”选项卡中选中“图片型”，否则在“非图形验证码”选项卡中选中“非图片型”。

下图是图片型验证码的配置示例：

● 图片型

验证码地址:

● 自带识别引擎

识别模式

☒ 单个文本统一块 ☐ 单一的文本行 ☐ 一个单词 ☐ 无OSD全自动页面分割

☐ 垂直对齐文本的统一块 ☐ 可变大小文本中的一列 ☐ 无OSD或OCRG的自动页面分割

☐ 仅OSD的定位及检测 ☐ OSD模式自动页面分割 ☐ 圈内的一个单词

识别范围

☐ 不限定

☐ 清晰的数字

☒ 限定为以下字符:

● 第三方识别引擎

☒ 亦思验证码识别引擎 ☐ 次时代验证码识别引擎

识别库:

加载

识别测试:

验证码图片:


获取到的验证码为:

识别测试

● 图片型

验证码地址:

● 自带识别引擎

识别模式

☒ 单个文本统一块 ☐ 单一的文本行 ☐ 一个单词 ☐ 无OSD全自动页面分割

☐ 垂直对齐文本的统一块 ☐ 可变大小文本中的一列 ☐ 无OSD或OCRG的自动页面分割

☐ 仅OSD的定位及检测 ☐ OSD模式自动页面分割 ☐ 圈内的一个单词

识别范围

☐ 不限定

☐ 清晰的数字

☒ 限定为以下字符:

● 第三方识别引擎

☐ 亦思验证码识别引擎 ☒ 次时代验证码识别引擎

识别库:

加载

识别测试:

验证码图片:


获取到的验证码为:

识别测试

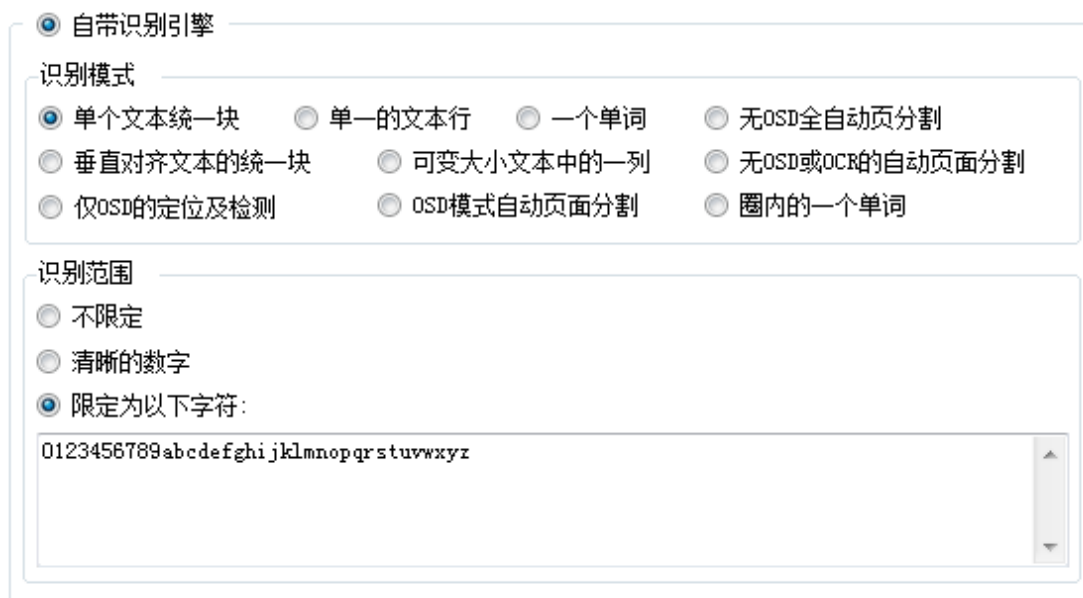
首先，在“验证码地址”处填入图片验证码的请求 URL，如“http://www.xx.com/captcha.php”，然后多次点击“识别测试”按钮，如果在“验证码图片预览框”中每次都能出现不同的新验证码，说明验证码地址设置成功。如果“验证码图片预览框”没有出现验证码图片或者图片固定不变，那么请检查验证码地址是否正确。如确定验证码地址无误，请使用抓包工具截获验证码刷新数据包，将数据包中的其他 HTTP 头部数据添加到“其他请求头部”文本框中，再多次点击“识别测试”按钮，观察是否每次都能生成新的验证码图片。

配置好验证码地址后，我们需要选择识别该验证码的识别引擎，Pkav HTTP Fuzzer 内置了 3 个图形验证码识别引擎，分别是“自带识别引擎”、“亦思验证码识别引擎”和“次世代验证码识别引擎”。以下是这三个识别引擎的介绍：

1) 自带识别引擎：

Pkav HTTP Fuzzer 自带的识别引擎，能够识别简单的验证码图片，无需手工制作验证码识别库，只需填写验证码的 URL 地址和勾选验证码的识别模式。该引擎适用于简单的验证码图片识别。

下图是自带识别引擎配置界面：



自带识别引擎无需制作和设置验证码识别库，但是针对不同的验证码，需要设置验证码的识别模式和识别范围(也可以挨个设置，然后查看测试结果，根据结果选择最适合的。)，自带识别引擎的识别模式介绍如下：

- A. 单个文本统一块：图片上是格式一致的文本块。
- B. 单一的文本行：图片上是单一的一个文本行。
- C. 一个单词：图片上是一个单词。
- D. 无 OSD 全自动分割：无方向和脚本检测的自动页面分割。
- E. 垂直对齐文本的统一块：图片上是一个统一格式的垂直对齐文本块。
- F. 可变大小文本中的一列：图片上是大小格式不统一的一个文本列。
- G. 无 OSD 或者 OCR 的自动页面分割：无方向和脚本检测和光学字符识别的自动页面分割。
- H. 仅 OSD 的定位及检测：仅方向和脚本检测 (Orientation and script detection (OSD) only, OSD)。
- I. OSD 模式自动页面分割：方向和脚本检测的自动页面分割。
- J. 圈内的一个单词。

自带识别引擎的识别范围介绍如下：

- A. 不限定：不限定识别引擎的文字和符合范围。
- B. 清晰的数字：限定识别范围为清晰的数字。
- C. 限定为以下字符：提供一个自定义范围。

2) 亦思验证码识别引擎：

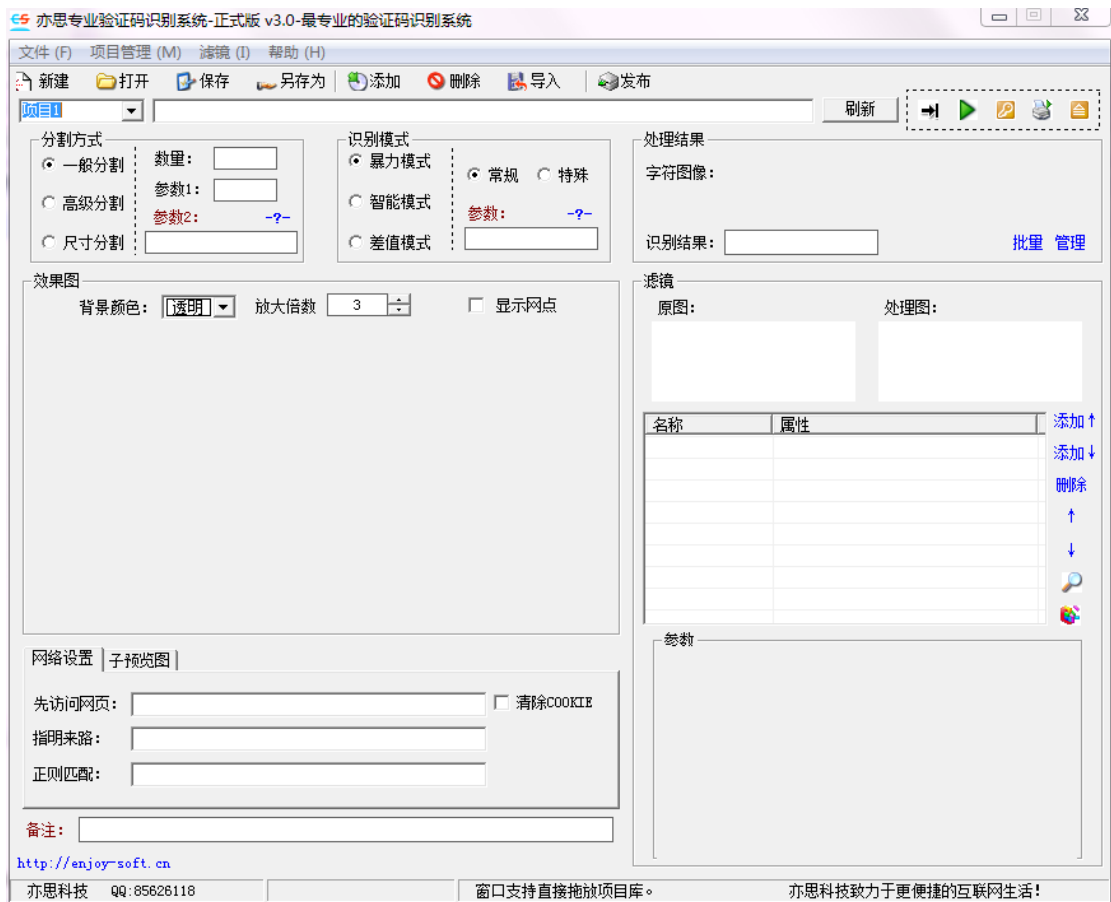
亦思验证码识别系统是一款第三方的收费的验证码识别系统。Pkav HTTP Fuzzer

只是集成了该系统的验证码识别接口。

亦思验证码识别系统是一个第三程序，不包含在工具中，需自行下载。它支持 3 大分割方式，3 大识别模式，将近 20 种处理滤镜，使用该程序进行简单的设置即可处理和识别较为复杂的验证码。(注：亦思验证码识别引擎好像已停止维护和更新，识别库制作需在 win32 位的系统中，建议在 win7 32 位的真实机或虚拟机中运行。) 关于亦思验证码识别系统的详细使用说明请自行网上搜索。

亦思验证码识别引擎使用非常简单，选择“第三方识别引擎”，然后选择“亦思验证码识别引擎”，点击“加载...”按钮，选择制作好的识别库，输入识别库的密码（无则留空），将亦思验证码识别系统制作出的验证码识别库导入即可。

下图是亦思验证码识别系统主程序界面：



3) 次世代验证码识别引擎：

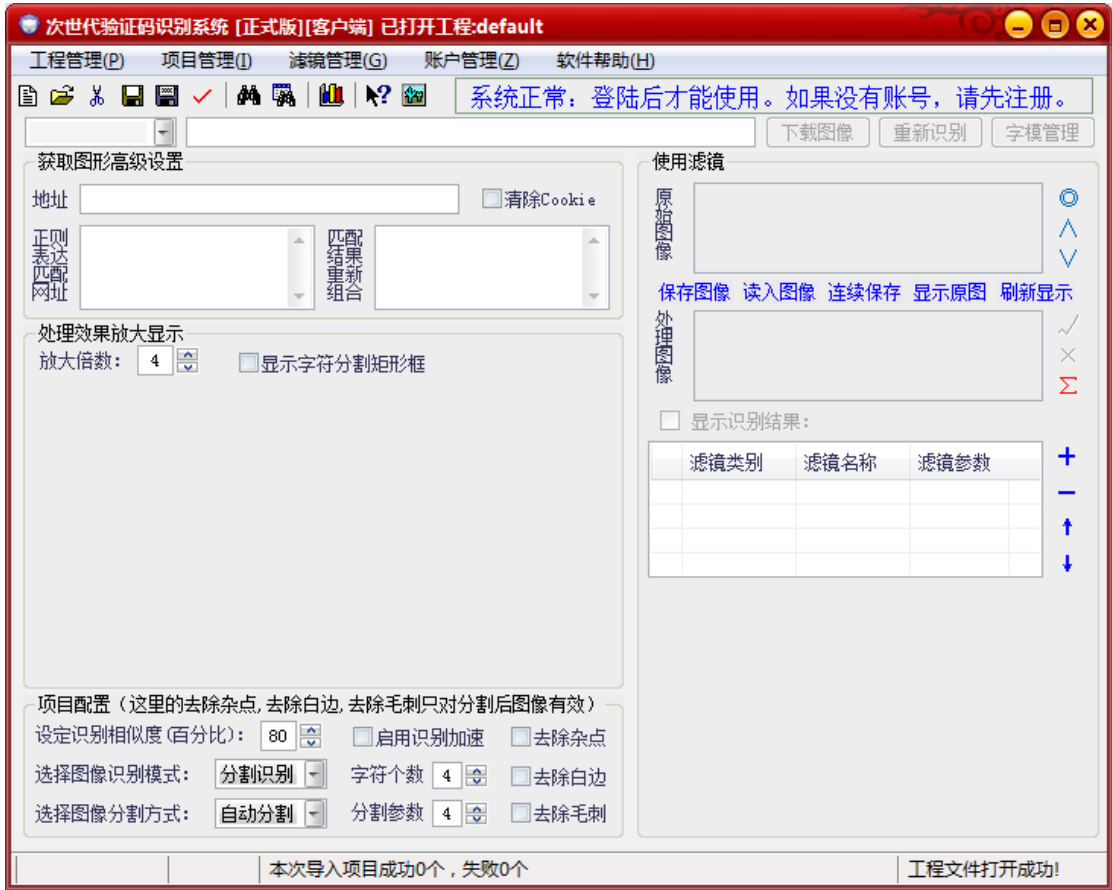
次世代验证码识别系统也是一款第三方的收费的验证码识别系统，不包含在工具中，需自行下载。Pkav HTTP Fuzzer 只是集成了该系统的验证码识别接口。

次世代验证码识别系统可快速有效的识别各种复杂的验证码，它支持四大类识别模式：分割识别、混合识别、整体识别、快速识别，能有效识别字符扭曲、粘连、重叠的验证码。独有的整体识别模式，图像无需分割，甚至不需要降噪即可识别，对于干扰点、干扰线等无法清除干净的图像特别有效！它支持三大类分割模式、十一大类处理滤镜：超过 40 种的图形处理滤镜，再复杂的验证码也可以有效处理，迅速得到清爽的二值化图像。它提供了强大的字模制作和管理界面，极大节省制作字模时间，轻点鼠标即可一次性得到几十个甚至几百个字模，你唯一要做的就是输入字模对应的字符。它拥有先进的识别加速技术：双重加速机制，能有效提升验证码识别速度，达到了识别速度与准确性的平衡。次世代验证码识别系统持续更新，识

别引擎比亦思验证码识别引擎更加稳定。关于次世代验证码识别系统的详细使用说明请自行网上搜索。

同样的，次世代验证码识别使用非常简单，选择“第三方识别引擎”，然后选择“次世代验证码识别引擎”，点击“加载...”按钮，选择制作好的识别库，输入识别库的密码（无则留空），将次世代验证码识别系统制作出的验证码识别库导入即可。

下图是次世代验证码识别系统主程序界面：



选择好识别引擎并配置相应选项后，通过多次点击“识别测试”按钮，观察识别测试结果和准确率，如果能够达到较高的识别准确率，图形验证码的配置就完成了。

验证码识别率不一定非得 100%准确才行。因为在 Pkav HTTP Fuzzer 的“重放选项”设置里面，我们可以对验证码识别结果进行筛选、处理和重试。如在带验证码的后台登录暴力破解场合，我们可以根据返回的结果来设置规则，在发现验证码错误的情况下自动重试，保证不漏掉任何一次有效爆破。

1.1.1.8 设置非图型验证码识别

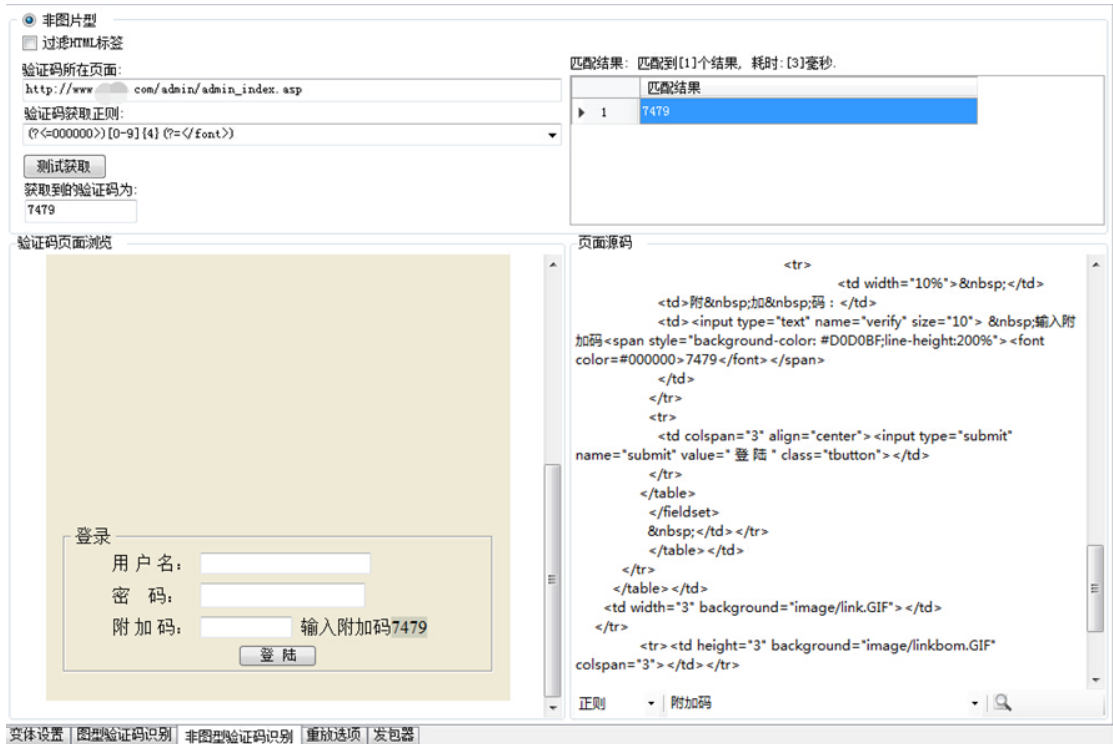
Pkav HTTP Fuzzer 支持图形和非图形验证码的识别。在数据包中定义了验证码变体后，需要配置相关的验证码识别选项。如果是图片型验证码，那么在“图形验证码识别”选项卡中选中“图片型”，否则在“非图形验证码”选项卡中选中“非图片型”。

当验证码是文字型的字符串时，我们需要从指定的页面将验证码提取出来。

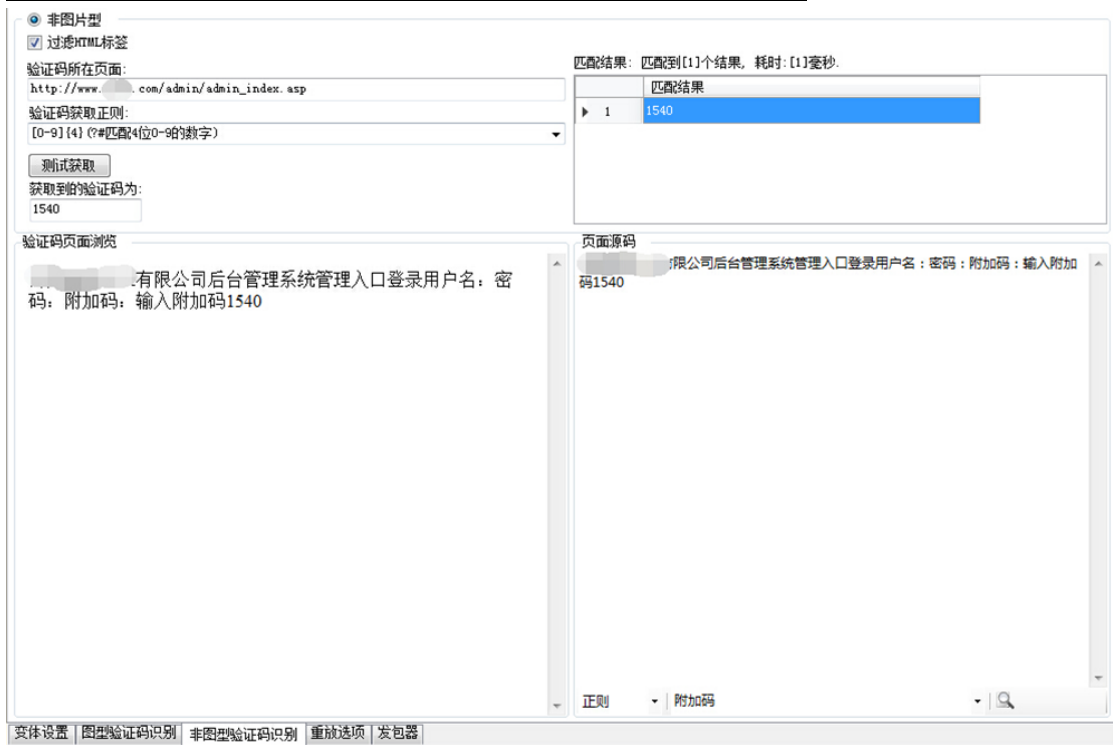
在非图型验证识别选项卡中选择“非图片型”，在“验证码所在页面”文本框上填写验证码的所在网页 URL，在“验证码获取正则”下拉框中输入提取验证码的正则表达式，点击“测试获取”按钮测试是否能获取到目标验证码。

“验证码获取正则”中内置了许多简易的验证码提取正则表达式，并且附带了使用的注释说明。

下图是非图型验证码的配置示例：



勾选“过滤 HTML”标签选项框，Pkav HTTP Fuzzer 会将网页源代码过滤成纯文本内容，降低正则表达式的复杂度，增加匹配的效率，如下图所示：



1.1.1.9 设置重放选项

在对 Pkav HTTP Fuzzer 的变体、重放模式、变体赋值方式、验证码识别引擎等进行配置后，我们还需要根据情况来设置对应的重放选项。

下面对各重放选项的配置进行详细说明：

1) HTTP 请求头部：

HTTP请求头部

☒ 设置Connection:close

在 HTTP 头部中添加“Connection: close”，数据包在发生完后立即断开与服务端的连接。

2) HTTP 重定向设置：

HTTP重定向设置

☒ 不跟踪重定向

☐ 选择性跟踪重定向

☐ 无条件跟踪重定向

☐ 自动跟踪302重定向

不跟踪重定向:不对重定向地址进行跟踪。无条件跟踪重定向:只要是重定向就进行跟踪。选择性跟踪重定向:仅跟踪隶属于同一服务器的资源重定向。自动跟踪 302 重定向:跟踪 302 重定向，仅仅在开启“无条件跟踪重定向”时，跟踪 302 重定向有效。

3) 验证码识别重放模式：

验证码识别重放模式

☐ 单线程模式 ☒ 多线程模式

在有验证码识别的数据包重放情况下，如果目标站点是一个 IP 绑定一个验证码的情况下，建议使用单线程模式。如果目标站点是一个会话绑定一个验证码的情况下，建议使用多线程模式。默认使用多线程模式。

4) 验证码长度验证码设置：

验证码长度验证设置

☐ 固定 位 ☒ 长度不固定

如果目标站点的验证码是长度固定的，建议勾选“固定 N 位”，如固定了 4 位长度的验证码，但是识别引擎识别出超过 4 位长度的验证码，那么识别引擎将重新获取验证码进行识别。如果目标站点的验证码长度是不固定的，那么建议选择“长度不固定”。如果是四则运算类的验证码，不允许选择固定验证码选项。

5) 验证码识别结果处理：

验证码识别结果处理

☒ 不做任何处理 ☐ 进行四则运算

如果目标站点的验证码为四则运算类型的验证码，请选择“进行四则运算”，否则

请选择“不做任何处理”。

6) 线程设置:

线程设置

线程数:

5

线程并发延迟:

2

毫秒

请求超时时间:

30

秒

请求超时重试:

3

次

线程空闲超时:

30

秒

A. 线程数:

允许开启的最大线程数量, 最小 1, 最大 300, 在设置了验证码识别的情况下, 建议小一点, 不要超过 20。

B. 线程并发延迟:

线程并发请求的延迟时间(单位:毫秒), 延迟越低, 扫描速度越快; 延迟越高, CPU 占用率越低, 扫描速度越慢。通过调整线程并发延迟时间可以调整目标站点的扫描速度, 防止被目标站点禁止访问。

C. 请求超时时间:

线程的超时时间(单位:秒), 线程在发出请求后超过指定时间没有收到响应则回收。

D. 空闲线程超时:

空闲线程超时时间(单位:秒), 线程处于空闲状态超过指定时间后将被销毁。

E. 请求超时重试:

请求超时或连接失败时, 自动重试的次数。

7) 显示和存储:

显示和存储

☒ 存储请求包

☒ 存储返回包

☒ 显示和存储全部数据

☐ 仅显示和存储匹配规则的数据

Pkav HTTP Fuzzer 默认会在发包器的“已发送数据包列表”中显示发送的全部数据包, 也会在本地的磁盘存储发送的请求包和请求返回的数据包, 会降低发送的效率和消耗一定的内存和磁盘空间。

如果在某些情况下, 要对显示和存储的数据进行筛选、过滤, 可以在此处进行配置。其中, “仅显示和存储匹配规则的数据”中的相关匹配规则, 请参考下面的“返回数据处理->匹配规则”的设置。

8) IP 伪造:

IP 伪造

☐ 使用 HTTP 代理服务器

☐ 伪造 X-Forwarded-For

IP 范围:

127 . 0 . 0 . 1

-

127 . 255 . 255 . 254

☐ 随机分配

☐ 伪造 Client-IP

IP 范围:

127 . 0 . 0 . 1

-

127 . 255 . 255 . 254

☐ 随机分配

某些站点对请求 IP 的访问频率和次数进行了严格的限定，某些站点对请求 IP 进行了采集和记录。我们可以设置 IP 伪造功能，伪造 HTTP 的 X-Forwarded-For 和 Client-IP 头部，每次请求都给目标站点假的 IP 地址，突破目标站点的访问限制，实现自身的简单隐蔽。

我们也可以勾选“使用 HTTP 代理服务器”，使用“设置选项->认证代理->HTTP 代理”的代理列表中的 HTTP 代理服务器来帮助我们代理请求，绕过访问限制和实现对自身的隐蔽效果。

9) 返回数据处理：

Pkav HTTP Fuzzer 提供了多种 HTTP 请求返回数据包的处理方法和功能，包括“代入的变体值长度不计算”、“匹配规则”、“提取内容”、“变体值条件丢弃”、“重试规则”等。下面我们将对各功能模块进行详细讲解：

A. 代入的变体值长度不计算：

大多数情况下，我们会根据请求返回的数据长度来辨别不同的 Fuzz 结果，对于一般的场景，同一种情况下返回的数据包长度是一致的，但是在某些场景，页面会将变体值也输出出来，因为变体值的长度是不同的，所以返回的数据包长度也不同。勾选此选项可以在计算返回数据长度时，不计算代入的变体值长度，保持同类型的返回数据的长度一致。

B. 匹配规则：

返回数据处理

☐ 代入的变体值长度不计算

匹配规则 提取内容 变体值条件丢弃 重试规则

☐ 从返回的数据中匹配如下表达式：

☒ 字符串匹配
☐ 正则表达式

添加 粘贴
删除 清空

☒ 忽略大小写 ☒ 排除HTTP头部

对 Fuzz 的结果进行辨别和筛选除了数据包长度排序、状态码排序外，更精确的方法是设置一个匹配规则。如暴力破解登录帐号密码时，密码错误会返回“密码错误！”，我们可以将“密码错误”添加到匹配规则列表中，在“发包器”的已发送数据包列表中，会在“匹配”列显示是否匹配该规则，如果匹配，会显示“是”，我们只需对该规则进行排序，筛选出“匹配”列为“否”的记录即可。

“匹配规则”可以使用字符串匹配和正则表达式匹配，“匹配规则”列表中的规则之间是“或”的关系，即其中任何一个规则能够匹配就属于匹配成功。同时，可以根据情况选择“忽略大小写”匹配和“排除 HTTP 头部”。

C. 提取内容：

☐ 代入的变体值长度不计算

匹配规则 提取内容 变体值条件丢弃 重试规则

☐ 从返回的数据中提取匹配如下正则表达式的内容

☒ 忽略大小写
☐ 匹配多个结果
☒ 排除HTTP头部

添加 粘贴
删除 清空

☐ 直接输出到文件 ☐ 输出对应变体值 ☒ 不输出匹配为空的数据

浏览

如果需从 Fuzz 的结果中提取所需的数据，我们可以使用“提取内容”功能。Pkav HTTP Fuzzer 可以通过设置内容提取正则表达式从返回的数据中提出所需的数据，可以将提取的数据显示在“发包器”的列表中，也可以直接提取保存到指定文件中。

使用方法很简单，勾选“从返回的数据中提取匹配如下正则的表达式的内容”，然后设置匹配是否区分大小写、是否匹配多个结果、是否从匹配的数据中排除 HTTP 头部，添加内容提取正则表达式到正则表达式列表中，最后选择是否输出对应变体值，是否直接输出到文件，是否不输出匹配为空的数据。

如果勾选了“直接输出到文件”，那么请先点击“浏览”设置输出的文件位置。勾选此选项后，提取的内容不会在“发包器”中显示。如果未勾选此选项，提取的内容会显示在“发包器”的“提取内容”列中（建议先不勾选查看是否能够提取成功，再勾选此选项直接输出到文件）。

勾选了“输出对应变体值”后，提取的内容会以“变体值 提取的内容”这样对应的格式输出。建议勾选“不输出匹配为空”的数据。

D. 变体值条件丢弃：

想象一下，如果要对一个目标站点进行暴力破解，但是站点限制了每个用户名最多只能登录 10 次，超过 10 次会提示“用户 xx 在 30 分钟内禁止登录！”，你该怎么办？这个时候，我们就可以使用“变体条件丢弃”功能了。“变体值条件丢弃”是指，当确认设定的变体值被设置的匹配规则匹配成功时，将该变体值丢弃，不再尝试与其相关的其他操作。

如上面的例子，我们设置了用户名变体和密码变体，在规则列表中添加“在 30 分钟内禁止登录”的字符串匹配规则，设置该规则应用于变体 1（依照变体在数据包中的顺序，即用户名变体）。当使用用户名“test”进行暴力破解时 11 次，返回了“用户 test 在 30 分钟内禁止登录”，与变体条件丢弃规则列表中的规则相匹配，那么会跳过“test”剩下的密码的暴力破解尝试，直接跳到下一个用户名进行暴力破解。

☐ 代入的变体值长度不计算

匹配规则 提取内容 变体值条件丢弃 重试规则

☐ 返回的数据匹配如下表达式时丢弃变体:

☒ 字符串匹配
☐ 正则表达式

应用于变体:

添加 粘贴
删除 清空

☒ 忽略大小写 ☒ 排除HTTP头部

E. 重试规则:

如果请求发生异常, 如果识别的验证码不准确, 如果验证码的识别成功率只有 50%, 如果请求需要重新发送, 该怎么办? 这时候需要使用 Pkav HTTP Fuzzer 的“重试规则”功能。勾选了“返回的数据中匹配如下表达式时重试”选项后, 如果返回的数据匹配“重试规则”列表中的规则, 那么该请求将重新发送, 直到不匹配“重试规则”列表中的规则为止(如果点了“发包器”中的“启动按钮”后一直没有结果, 那么检查检查是否“重试规则”设置错误, 导致在后台一直重试。)。

当验证码的识别成功率达不到 100%时, 我们需要设置“重试规则”, 在规则列表中添加如“验证码错误”等提示, 当请求返回的数据中能够匹配到“验证码错误”时, Pkav HTTP Fuzzer 会重新获取和识别验证码, 使用相同的变体值进行重试发送, 直到获取和识别到正确的验证码为止。

☐ 代入的变体值长度不计算

匹配规则 提取内容 变体值条件丢弃 重试规则

☐ 返回的数据中匹配如下表达式时重试:

☒ 字符串匹配
☐ 正则表达式

添加 粘贴
删除 清空

☒ 忽略大小写 ☒ 排除HTTP头部

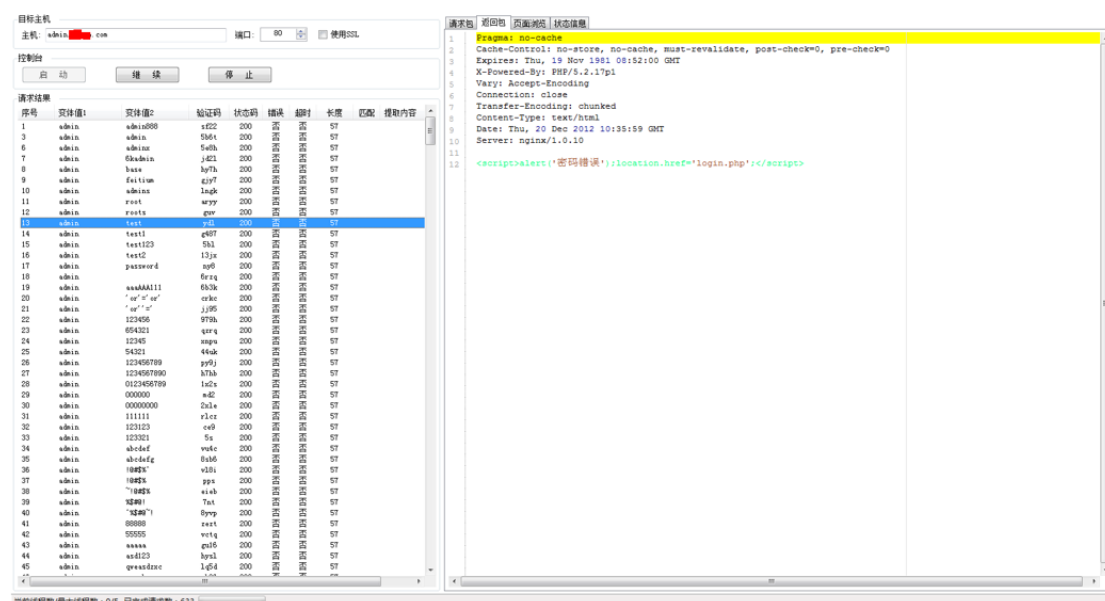
1.1.1.10使用发包器发包

所有的配置都完成了，接下来就是使用“发包器”进行数据包发送了。“发包器”可以控制数据包的发送、暂停和停止，查看发送和接收的数据包，对数据包进行排序，复制变体值和提取的内容等等。

在“启动”发包器前，需要检查目标主机、端口及 HTTP 协议。如目标是 SSL 的需要勾选“使用 SSL”。点击“启动”按钮后，Pkav HTTP Fuzzer 会对上一次 Fuzz 的缓存数据进行清理，如果缓存数据较大，可能需要几秒钟的等待时间。

点击每个列的列头即可对列表中的数据进行排序，可以根据“长度”列、“状态码”列来简单区分不同的返回数据，如果配置了“匹配规则”，还可以根据匹配规则的结果进行排序。

如下是识别图片型验证码进行后台暴力破解演示截图：



如果您发现本手册有错误需要修正或有其他意见和建议，请发邮件给我们，邮件地址：vk@pker.in。