

Winsock Packet Editor

网络封包拦截器

使用说明 (1.0.0.36)

简介

WinSockPacketEditor (网络封包拦截器), 是一款可以拦截并修改 WinSock 封包的 windows 软件, 自适应支持 32 位及 64 位的目标程序, 软件具有批量发送和高级滤镜等功能, 开发中使用了 C#的多线程和消息队列技术, 测试拦截了 1 万+的封包不会卡死或退出, 软件不定期会修复 bug 和更新功能, 每次启动的时候支持在线自动更新, 欢迎大家提出宝贵意见一起完善和改进, 谢谢!

下载和安装

本软件使用了微软的 VS2022 集成开发环境, .NET Framework 4.8 开发框架, 以及 ClickOnce 部署资源。每次版本更新后, 都会在启动程序时自动下载最新版本。如果更新服务器不可用, 也不会导致程序无法使用。当然, 如果您不希望自动更新, 也可以在启动时手动关闭自动更新, 或者直接下载离线打包版使用。

在线版安装方法:

1. 打开在线版下载网址, 点击安装按钮后下载安装程序

X-NAS 封包拦截器

名称: 封包拦截器

版本: 1.0.0.33

发行者: X-NAS

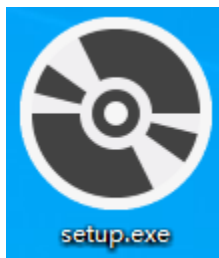
以下系统必备组件是必需的:

如果已经安装了这些组件, 您可以立即启动该应用程序, 否则, 请单击下面的按钮, 安装系统必备组件并运行该应用程序。



[ClickOnce 和 .NET Framework 资源](#)

2. 运行下载的安装程序



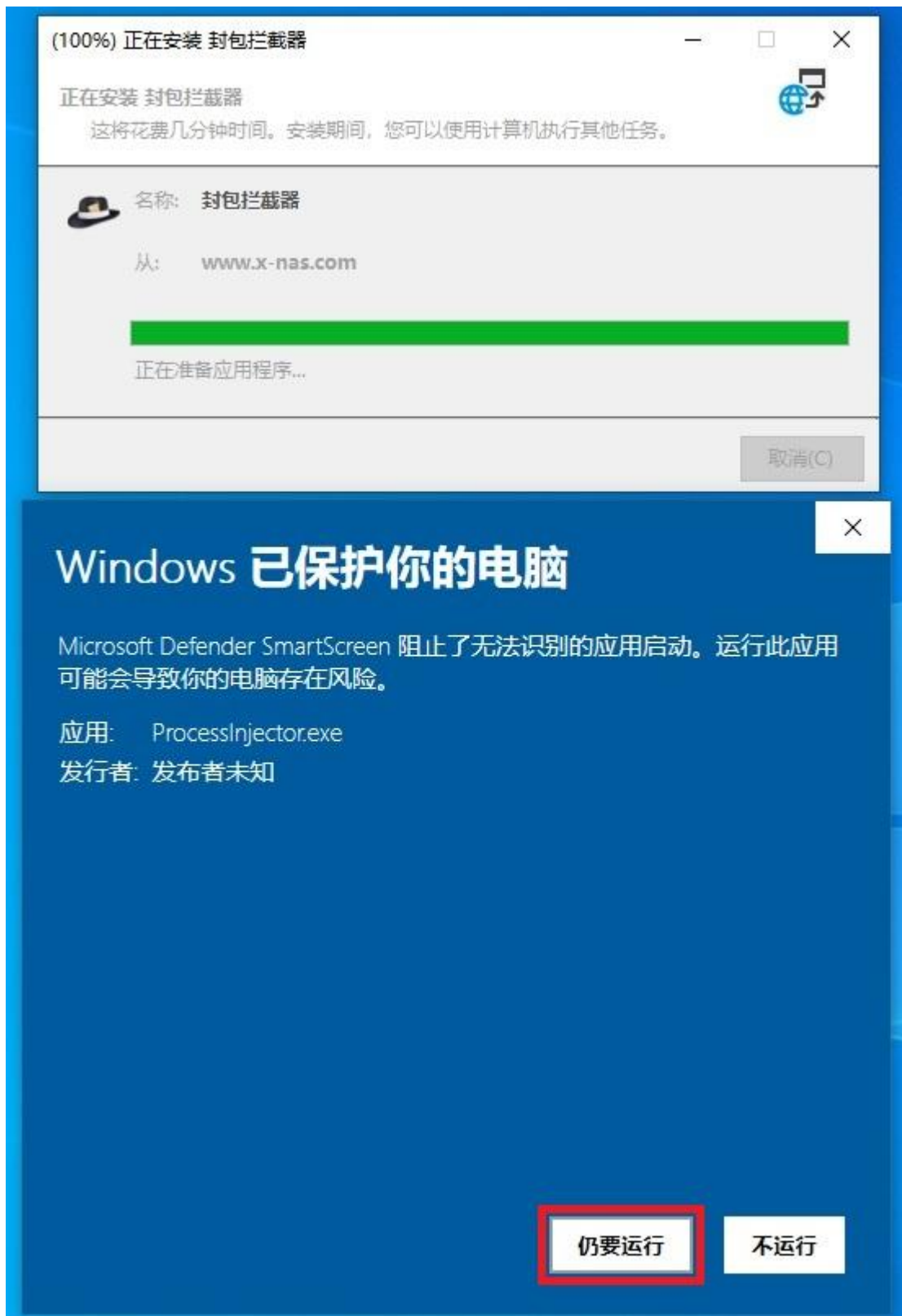
3. 如有提示请选择 “更多信息”, 并点击 “仍要运行”



4. 安装程序会自动检测操作系统的运行环境，如有需要会提示安装相关运行库（比如.net framework 4.8 等），点击下载及安装即可

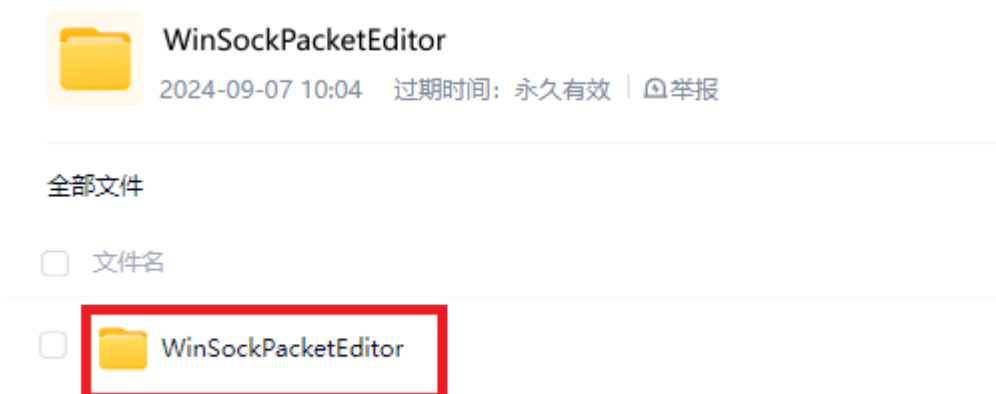


5. 安装完成后启动时如有提示，请选择“更多信息”，并点击“仍要运行”



离线版安装方法：

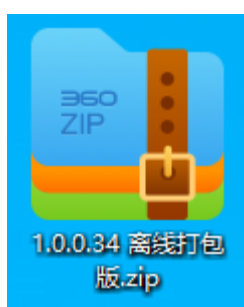
1. 打开离线版下载网页，点击 WinSockPacketEditor 文件夹



2. 选择对应要下载的离线版本

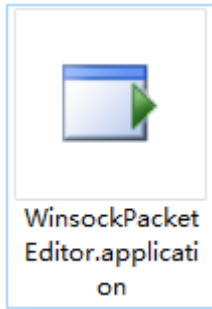


3. 下载完成后解压缩到对应的目录



离线版转在线版安装方法：

打开解压缩后的文件所在的文件夹，运行 WinsockPacketEditor.application



启动程序

在线安装版启动方法：在安装完成后，运行桌面的程序图标

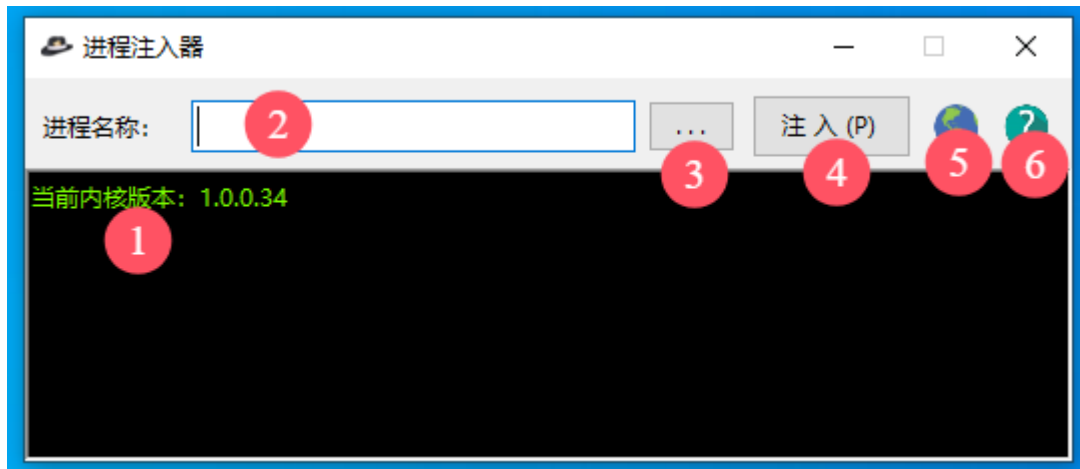


离线打包版启动方法：打开解压缩后的文件所在的文件夹，运行 WinsockPacketEditor.exe



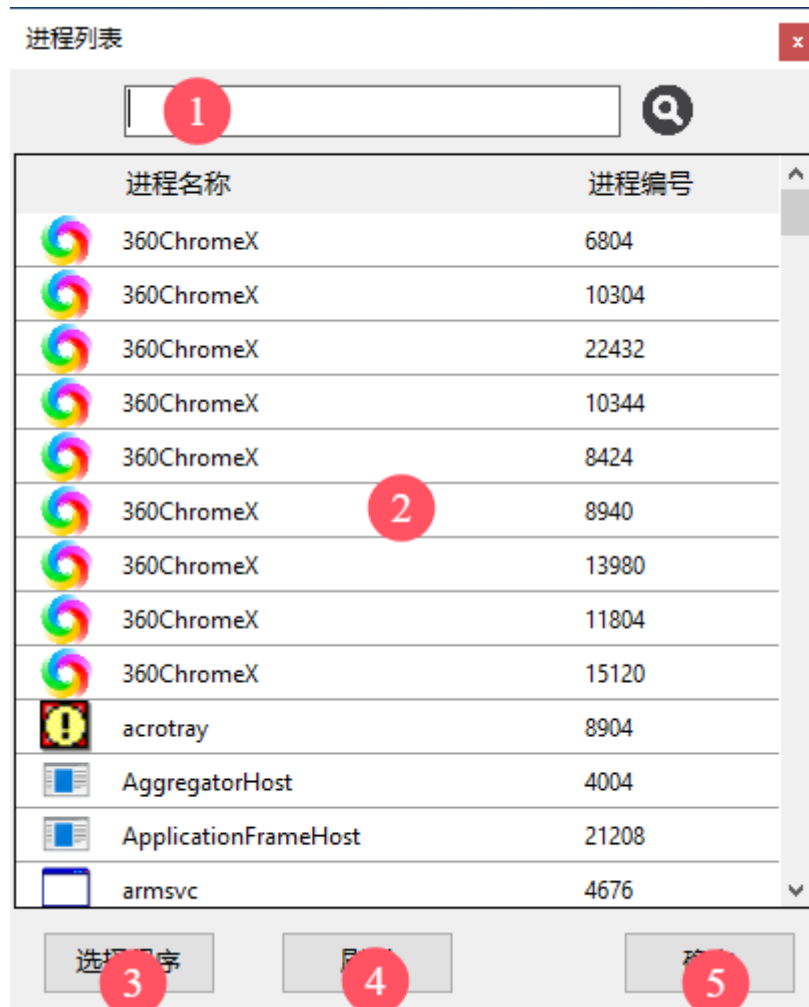
程序打开时会尝试以管理员身份启动，如有弹窗请点击“是”确认

进程注入器界面



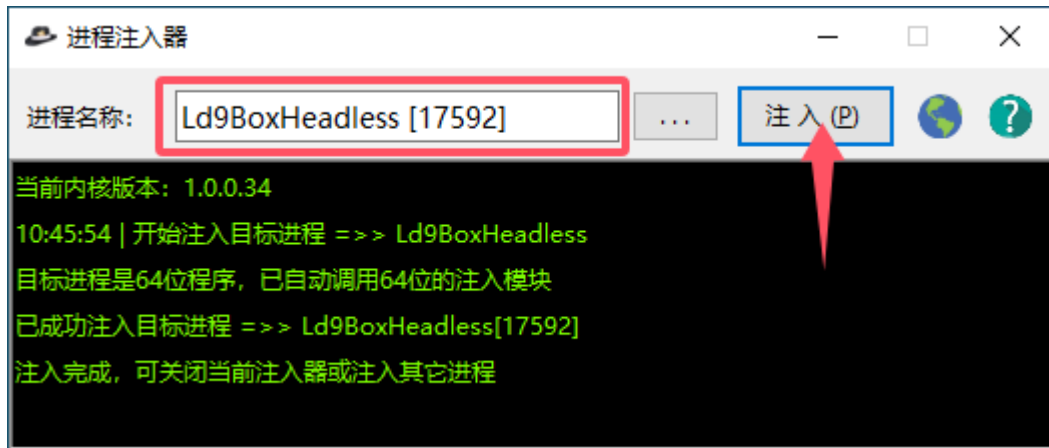
1. 显示当前版本号，以及注入进程的信息
2. 显示选择进程后的进程名称以及进程编号信息
3. 选择进程功能按钮
4. 选择好进程，点击注入按钮可注入当前选中的进程
5. 多语言选择
6. 关于信息

选择进程



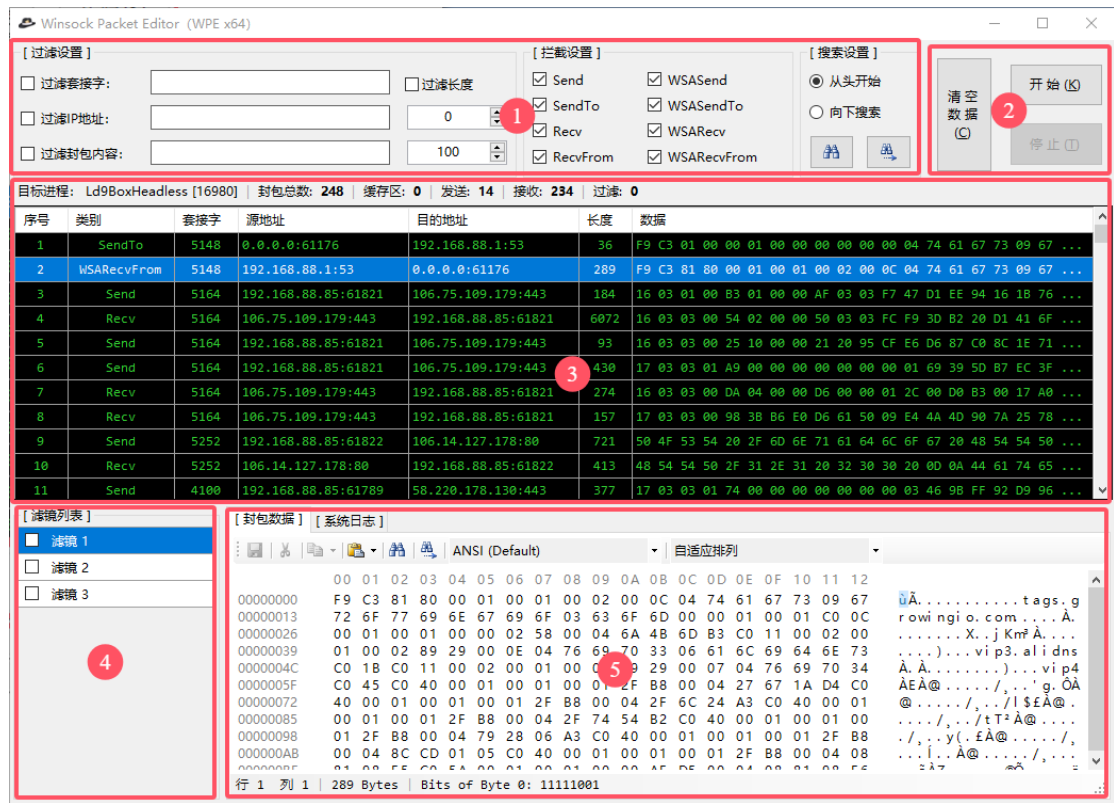
1. 快速搜索进程，输入想要搜索的进程名字的字母
2. 显示所有满足搜索条件的进程信息
3. 选择一个尚未启动的程序，通常用于需要从程序启动时就开始拦截封包的情况
4. 刷新当前进程列表
5. 选中进程后点击“确定”按钮可将进程信息带入到进程注入器

注入当前进程



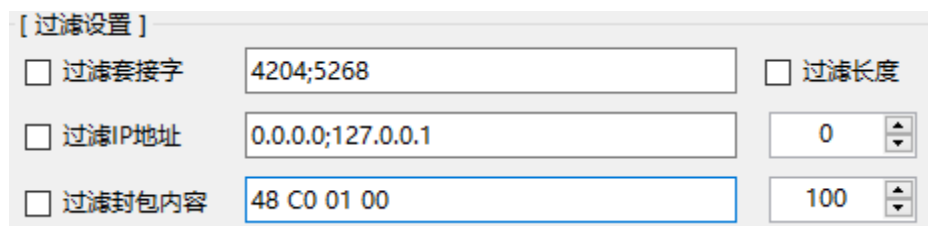
当选择好进程后，点击“注入”按钮即可注入当前进程，注入器会根据当前进程自动调用32位或者64位的注入模块，注入完成后在下方信息区会显示注入是否成功的提示信息。注入完成后会自动弹出封包拦截器主界面，如果不需要再注入其它进程的话，可以关闭当前注入器。

封包拦截器主界面



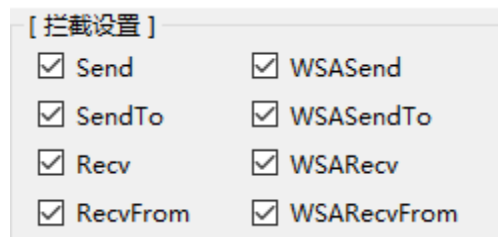
1. 过滤, 拦截和搜索参数设置区
2. 功能按钮区
3. 封包列表区, 显示拦截的封包信息
4. 滤镜列表区
5. 封包数据编辑区, 显示列表中选中封包的数据, 以及系统的运行日志

设置过滤拦截参数



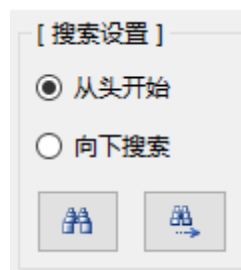
1. 过滤套接字: 选中后的套接字将被过滤掉, 不显示在下方的封包列表中



2. 过滤 IP 地址：选中后的 IP 地址将被过滤掉，不显示在下方的封包列表中
3. 过滤封包内容：选中后的封包内容将被过滤掉，不显示在下方的封包列表中
4. 过滤长度：选中后将会只显示封包长度在 0~100 之间（包含 0 和 100）的封包
5. 多个过滤条件之间使用 “;” 作为分隔符

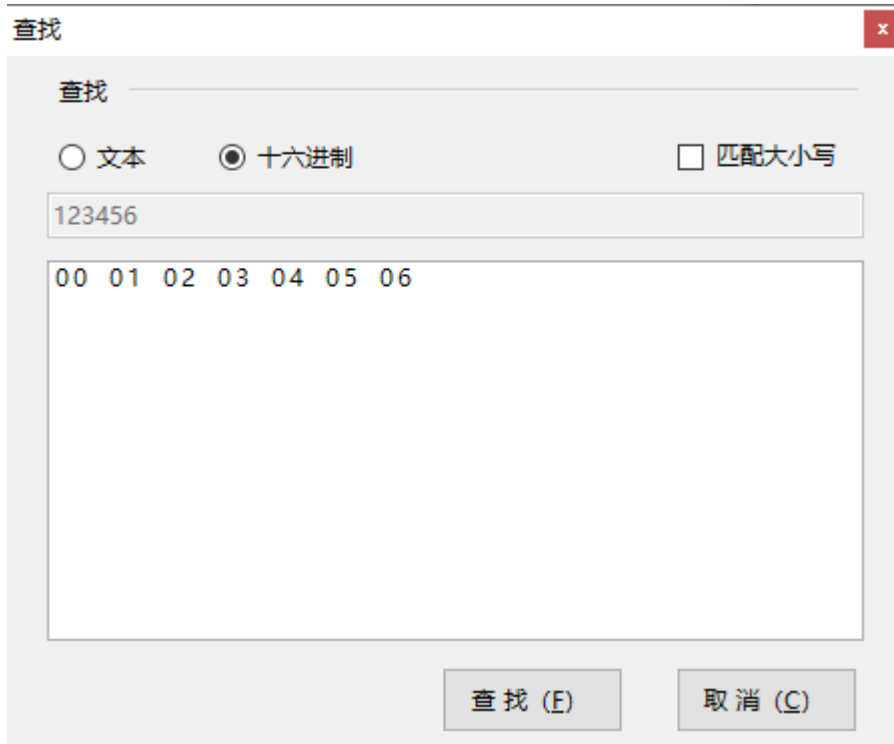


1. 软件现支持拦截 Winsock 网络接口规范下的 8 种类型的封包, 包括 TCP 协议的 Send, Recv 和 WSA Send, WSA Recv 以及 UDP 协议的 SendTo, RecvFrom 和 WSA SendTo, WSA RecvFrom
2. 系统默认拦截全部类型的封包, 如有需要可自行勾选对应要拦截的封包类型

设置搜索参数



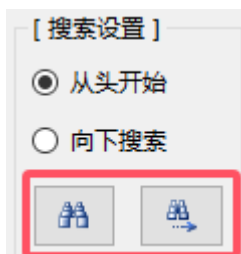
1. 从头开始：从封包列表的头部开始搜索
2. 向下搜索：从封包列表中当前选中的行开始向下搜索
3.  按钮：打开搜索参数页面
4.  按钮：按照设置的搜索参数，查找下一个满足条件的封包数据，并定位到该数据位置



1. 文本：按照文本格式 (UTF-7) 进行封包数据的全局搜索
2. 十六进制：按照 Hex 格式进行封包数据的全局搜索
3. 匹配大小写：严格匹配大小写进行封包数据的全局搜索
4. “查找”按钮：按照设置的搜索参数，进行全局搜索，并定位到匹配的数据位置
5. “取消”按钮：关闭当前窗口，不进行搜索操作

全局搜索和单个封包搜索

1. 全局搜索：



配置好搜索参数后，点击主界面顶部的搜索按钮，会对封包列表中的所有封包进行全局搜索，每点击一次“查找下一个”按钮，就会定位到下一个满足条件的封包列表位置行，

并在封包数据编辑区定位到封包内部满足条件的数据位置（高亮显示），如果没有满足条件的封包，则会弹出提示对话框

2. 单个封包搜索：



配置好搜索参数后，点击封包数据编辑区的搜索按钮，会对当前选中封包的数据进行搜索，每点击一次“查找下一个”按钮，就会定位到下一个满足条件的数据位置（高亮显示），如果没有满足条件的数据，则会弹出提示对话框

开始拦截封包



1. 在设置好过滤和拦截参数后，点击“开始”按钮即可启动拦截，拦截的封包数量和信息会在下方的封包列表区域实时刷新，由于程序使用了队列缓存模式，数据的显示会有一些的延迟间隔，待缓存区数量显示为 0 后，即可查看到所有拦截的封包信息
2. 在封包拦截期间，过滤设置和拦截设置将不可用，待拦截结束后恢复可用状态
3. 点击“清空数据”按钮，可实时清空封包列表中的所有封包数据

封包列表区

| 序号 | 类别 | 套接字 | 源地址 | 目的地址 | 长度 | 数据 |
|----|-------------|------|------------------|------------------|----|---|
| 1 | SendTo | 4204 | 0.0.0.0:50461 | 211.167.97.67:53 | 24 | 48 C0 01 00 00 01 00 00 00 00 00 00 ... |
| 2 | SendTo | 5268 | 0.0.0.0:50462 | 211.167.97.67:53 | 36 | 40 C0 01 00 00 01 00 00 00 00 00 00 ... |
| 3 | SendTo | 5220 | 0.0.0.0:50463 | 211.167.97.67:53 | 31 | 54 94 01 00 00 01 00 00 00 00 00 00 ... |
| 4 | WSARecvFrom | 5268 | 211.167.97.67:53 | 0.0.0.0:50462 | 52 | 40 C0 81 80 00 01 00 01 00 00 00 00 ... |
| 5 | WSARecvFrom | 4204 | 211.167.97.67:53 | 0.0.0.0:50461 | 99 | 48 C0 81 83 00 01 00 00 00 01 00 00 ... |
| 6 | SendTo | 5240 | 0.0.0.0:50464 | 211.167.97.67:53 | 34 | 48 25 01 00 00 01 00 00 00 00 00 00 ... |
| 7 | SendTo | 5320 | 0.0.0.0:50465 | 211.167.97.67:53 | 31 | 54 33 01 00 00 01 00 00 00 00 00 00 ... |
| 8 | SendTo | 4800 | 0.0.0.0:50466 | 211.167.97.67:53 | 33 | 66 C3 01 00 00 01 00 00 00 00 00 00 ... |
| 9 | SendTo | 5212 | 0.0.0.0:50467 | 211.167.97.67:53 | 33 | 79 ED 01 00 00 01 00 00 00 00 00 00 ... |
| 10 | SendTo | 5380 | 0.0.0.0:50468 | 211.167.97.67:53 | 38 | 05 BB 01 00 00 01 00 00 00 00 00 00 ... |
| 11 | WSARecvFrom | 5220 | 211.167.97.67:53 | 0.0.0.0:50463 | 74 | 54 94 81 80 00 01 00 02 00 00 00 00 ... |

1. 目标进程：显示当前被注入的进程信息
2. 封包总数：显示拦截的封包总数（包含被过滤的封包数）
3. 缓存区：显示暂存在封包队列中还未显示到封包列表中的封包数量
4. 发送：显示所有为发送类型封包的数量（Send, SendTo, WSASend, WSASendTo）
5. 接收：显示所有为接收类型封包的数量（Recv, RecvFrom, WSARecv, WSARecvFrom）
6. 过滤：显示所有被过滤参数拦截掉的封包的数量（不会显示在下方封包列表中）
7. 封包列表：显示被拦截的封包信息，包含拦截时的序号，协议类别，套接字，源 IP 地址（含端口号），目的 IP 地址（含端口号），封包的长度，封包的十六进制数据（显示前 80 位）

封包列表的功能菜单

| 序号 | 类别 | 套接字 | 源地址 | 目的地址 | 长度 | 数据 |
|----|-------------|------|------------------|------------------|----|---|
| 1 | SendTo | 4204 | 0.0.0.0:50461 | 211.167.97.67:53 | 24 | 48 C0 01 00 00 01 00 00 00 00 00 00 ... |
| 2 | SendTo | 5268 | 0.0.0.0:50462 | 211.167.97.67:53 | 36 | 40 C0 01 00 00 01 00 00 00 00 00 00 ... |
| 3 | SendTo | 5220 | 0.0.0.0:50463 | 211.167.97.67:53 | 31 | 54 94 01 00 00 01 00 00 00 00 00 00 ... |
| 4 | WSARecvFrom | 5268 | 211.167.97.67:53 | 0.0.0.0:50462 | 52 | 40 C0 81 80 00 01 00 01 00 00 00 00 ... |
| 5 | WSARecvFrom | 4204 | 211.167.97.67:53 | 0.0.0.0:50461 | 99 | 48 C0 81 83 00 01 00 00 00 01 00 00 ... |
| 6 | SendTo | 5240 | 0.0.0.0:50464 | 211.167.97.67:53 | 34 | 48 25 01 00 00 01 00 00 00 00 00 00 ... |
| 7 | SendTo | 5320 | 0.0.0.0:50465 | 211.167.97.67:53 | 31 | 54 33 01 00 00 01 00 00 00 00 00 00 ... |
| 8 | SendTo | 4800 | 0.0.0.0:50466 | 211.167.97.67:53 | 33 | 66 C3 01 00 00 01 00 00 00 00 00 00 ... |
| 9 | SendTo | 5212 | 0.0.0.0:50467 | 211.167.97.67:53 | 33 | 79 ED 01 00 00 01 00 00 00 00 00 00 ... |
| 10 | SendTo | 5380 | 0.0.0.0:50468 | 211.167.97.67:53 | 38 | 05 BB 01 00 00 01 00 00 00 00 00 00 ... |
| 11 | WSARecvFrom | 5220 | 211.167.97.67:53 | 0.0.0.0:50463 | 74 | 54 94 81 80 00 01 00 02 00 00 00 00 ... |

在封包列表中选中某个封包后，点击鼠标右键会弹出封包列表的功能菜单

1. 发送：将当前选中的封包带入到封包发送界面中
2. 添加到发送列表：将当前选中的封包带入到封包发送列表界面中
3. 添加到滤镜列表：将当前选中的封包带入到滤镜列表中（新增一个以当前进程名称命名的新滤镜）
4. 使用此套接字：将当前选中的封包的套接字作为发送列表中所有封包使用的全局套接字
5. 查看发送列表：显示发送列表界面
6. 导出到 Excel：将当前封包列表中的全部封包数据到处到 Excel 文件（由于程序以字符串形式导出成 excel 文件，所以在打开导出的 excel 文件时会提示格式不正确，点击确认即可正常打开，可再另存为标准的 excel 格式的文件）

发送封包界面

系统可同时打开多个发送封包界面，每个发送界面可以相对独立的发送封包



1. 发送封包界面的标题，显示当前注入的进程，以及发送的封包在封包列表的序号
2. 当前要发送的封包的数据，以十六进制格式显示

| 001 | 002 | 003 | 004 | 005 | 006 | 007 | 008 | 009 | 010 | 011 | 012 | 013 | 014 | 015 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 48 | C0 | 01 | 00 | 00 | 01 | 00 | 00 | 00 | 00 | 00 | 00 | 06 | 63 | 6F |

双击某个位置后，可编辑当前位置的数据（十六进制格式），也可支持直接复制某一段

封包数据（十六进制格式）后直接黏贴到当前封包中

3. 当前要发送的封包的发送参数

- 1) 套接字：默认显示封包的原始套接字，支持自定义修改
- 2) 长度：默认显示封包的原始长度，支持自定义修改
- 3) 目的 IP，目的端口：默认显示封包发送的目的 IP 地址和端口号，不支持自定义修改
- 4) 递进位置：显示当前选中的封包的某个位置的数据，勾选后在发送列表时，会递进修改当前位置的数据后再发送封包
- 5) 每次步长：显示按照递进值每次递进修改数据后的值，支持正值和负值递进
- 6) 循环：设置需要循环发送的次数
- 7) 间隔：设置每次循环发送的间隔（毫秒）

4. 显示发送封包的完成情况

发送封包的功能菜单



在封包数据区域点击鼠标右键会弹出发送封包的功能菜单

1. 添加到发送列表：将当前封包带入到封包发送列表中
2. 添加到滤镜列表：将当前封包带入到滤镜列表中（新增一个以当前进程名称命名的新滤镜）

发送列表界面

发送列表界面全局唯一，任何时候一个注入的进程只会有一个发送列表



1. 发送列表的标题：显示当前注入的进程信息
2. 发送列表区：显示所有发送的封包信息，只有勾选的封包才会被发送

双击当前选中封包的备注栏后，可单独添加备注信息



3. 发送封包的功能区
 - 1) 全选/取消：全选或者取消发送列表的勾选框
 - 2) 使用此套接字：勾选后发送列表中的所有封包会使用此全局套接字进行发送，否则使用每个封包默认的套接字发送
 - 3) 循环次数：循环发送列表中所有选中封包的次数
 - 4) 发送间隔：发送每个封包时的间隔时间（毫秒）
4. 显示发送封包列表的完成情况

发送列表界面的功能菜单

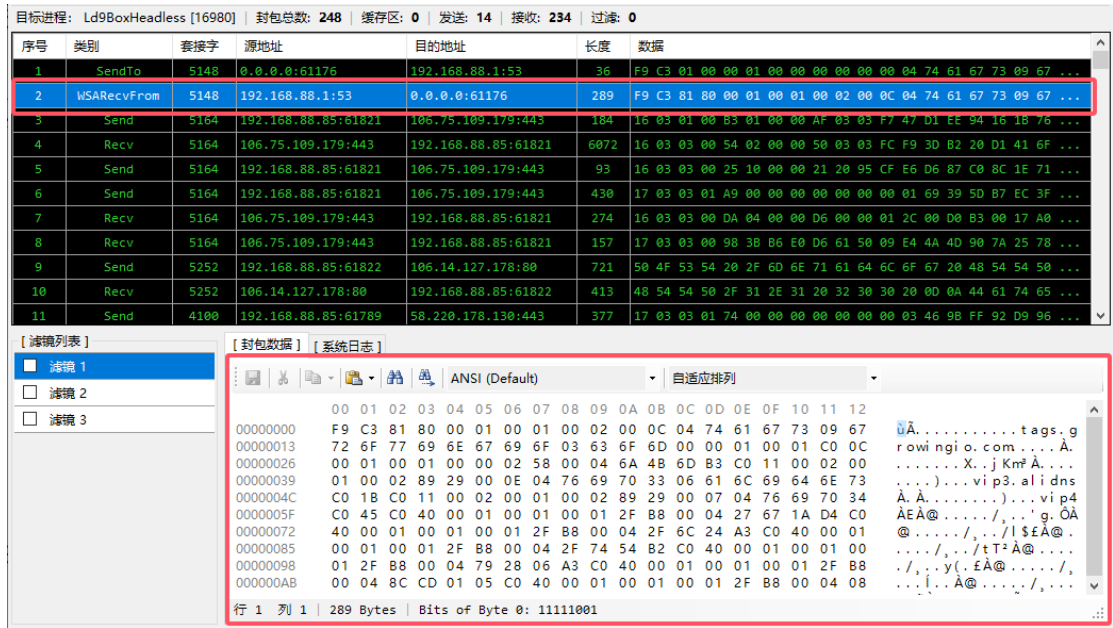


在发送列表区域点击鼠标右键可以弹出发送列表的功能菜单

1. 从列表中移除：将当前选中的封包从列表中移除
2. 清空发送列表：移除当前发送列表中的所有封包
3. 保存此列表数据：将当前发送列表中的所有封包数据保存到 .sp 文件中
4. 加载发送列表：从 .sp 文件中加载封包数据到发送列表

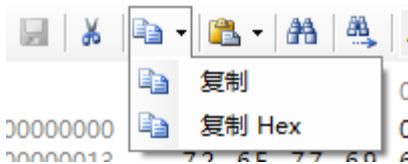
编辑封包数据

如果想要查看和编辑某个封包的完整数据，可以通过在封包列表区域选中该封包后，在下侧的封包数据编辑区进行修改，支持剪切，复制，粘贴，全选等多种编辑操作

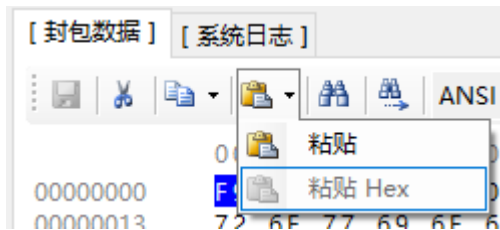


1. 当封包数据有变化后，“保存”按钮将变的可用
2. 当选中一段封包数据后，“剪切”按钮将变的可用
3. 当选中一段封包数据后，“复制”按钮将变的可用，可选择复制文本值，或者复制十六

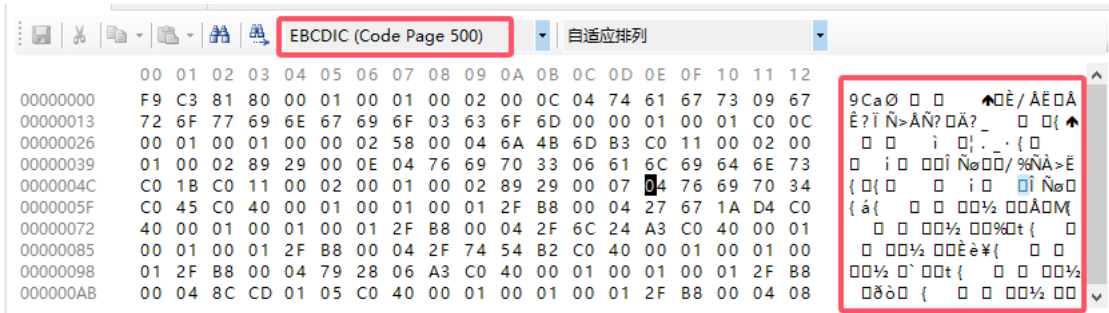
进制数据值



4. 当复制了一段封包数据后，“粘贴”按钮将变的可用，可选择粘贴文本值，或者粘贴十六进制数据值

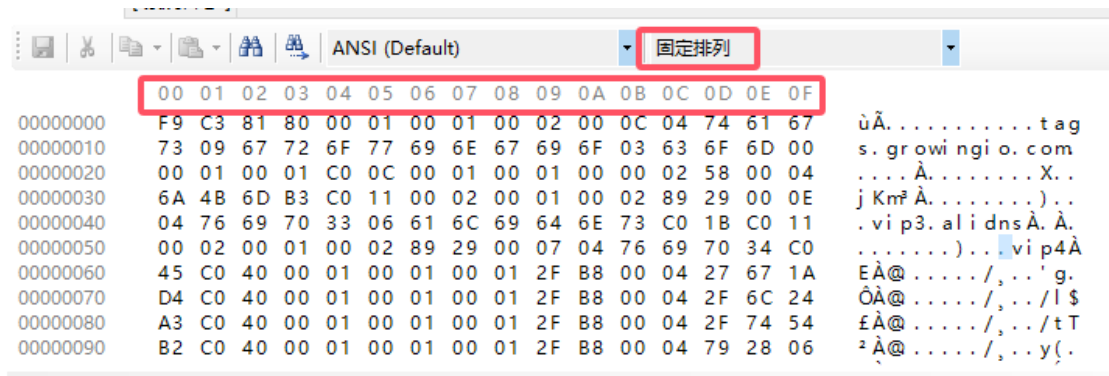


5. 封包数据编辑区的“查找”和“查找下一个”按钮的搜索范围为当前选中的封包
6. 编码选择下拉框：显示可用于显示的 ASCII 编码格式

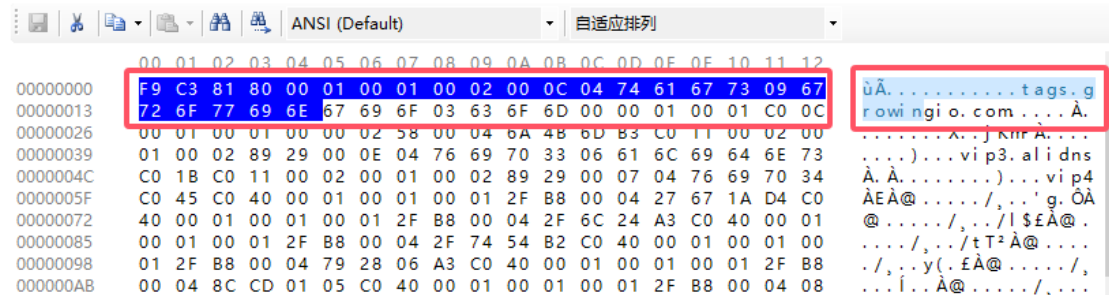


7. 排列模式下拉框：显示编辑区显示的数据的排列格式，自适应窗体界面宽度，或者按照

固定的 16 位宽度排列



8. 对任意位置封包数据的修改，都将同时作用于十六进制和文本两种格式



9. 封包数据修改后，必须点击“保存”按钮才会生效，如果没有保存就切换显示其它封包，

则将丢失所有未保存的修改

10. 状态区会显示当前选中的封包数据的行列位置，封包修改后的大小，以及当前选中数据

在整个封包数据中的位置和对应的二进制编码值



滤镜列表及其功能菜单

首次使用本程序的时候，系统会默认新建 3 个空滤镜，待用户自行添加或者编辑完滤镜后，

系统会将滤镜列表进行保存，待下次启动时自动加载之前保存的所有滤镜

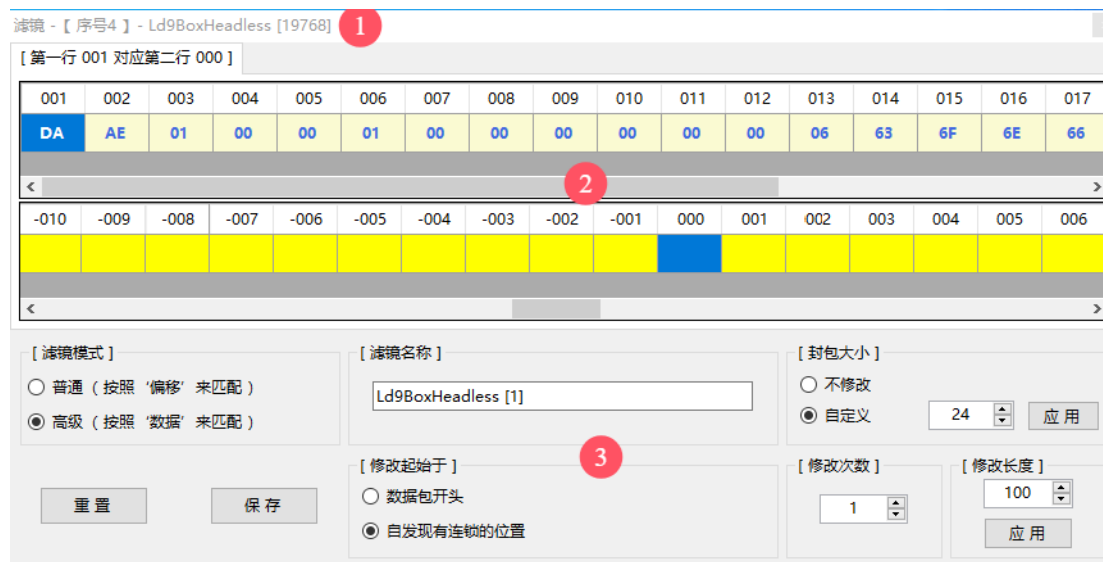
在滤镜列表区域点击鼠标右键，即可弹出滤镜列表的功能菜单



1. 查看选中滤镜：在滤镜界面中打开当前选中的滤镜
2. 添加新滤镜：自动新建一个空的新滤镜
3. 删除选中滤镜：删除在滤镜列表中选中的滤镜
4. 保存此列表数据：将滤镜列表中的所有滤镜保存到一个 .fp 文件中
5. 清除所有滤镜：自动清除滤镜列表中的所有滤镜
6. 加载滤镜列表：从一个 .fp 文件读取所有滤镜到滤镜列表中

滤镜主界面

每一个滤镜都可以打开一个配置界面，各个滤镜之间相对独立，可单独编辑和配置



1. 滤镜界面标题，显示当前注入的进程信息，当前打开的滤镜名称及编号
2. 滤镜的匹配数据及修改数据的编辑区域
3. 滤镜参数配置区域

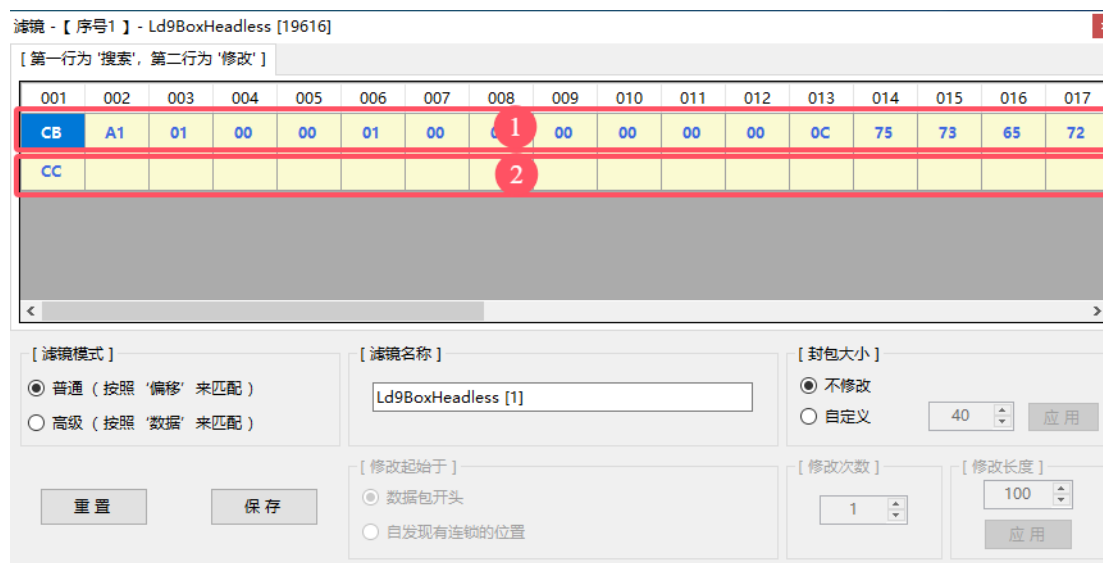
滤镜模式

滤镜可以分为“普通滤镜”和“高级滤镜”两种模式，通过滤镜配置区域的【滤镜模式】来进行配置，两种模式的区别如下：



1. 普通滤镜：按照每个“偏移”位置上所对应的数据来进行匹配，修改数据也是按照“偏移”上的数据来进行修改
2. 高级滤镜：按照数据内容来进行匹配，匹配成功后按照所选的修改参数进行数据修改

如何使用普通滤镜



1. 当滤镜模式选择为“普通”模式的时候，滤镜数据区域一共显示两行，第一行的数据用来进行匹配，第二行的数据用来进行修改
2. 在需要填写数据的单元格上双击鼠标左键可编辑当前位置的数据
3. 可以在复制一串数据（十六进制）后，选中某个单元格并按键 Ctrl+V 直接从选中的单元格开始黏贴所选的数据串
4. 普通滤镜严格按照“偏移”和对应位置上的“数据”进行一对一的匹配，当全部“偏移”位置上的非空数据，都跟封包数据匹配一致后，即认为该滤镜可生效
5. 在普通模式下，仅可编辑滤镜的名称，以及封包大小，其它参数均不可修改
6. 如有多个滤镜，在匹配到第一个满足条件的滤镜后，不再匹配后续的其他滤镜
7. 自定义封包大小后，需要点击“应用”按钮，才可生效
8. 修改完滤镜数据后，需要点击“保存”按钮，此滤镜才会正式可用，此时滤镜界面会自动关闭，当前滤镜列表的数据也会被自动保存

举例：假设我们接收到一个游戏封包数据 “CB A1 01 00 00 01 00 00 00 00 00 00 0C 75

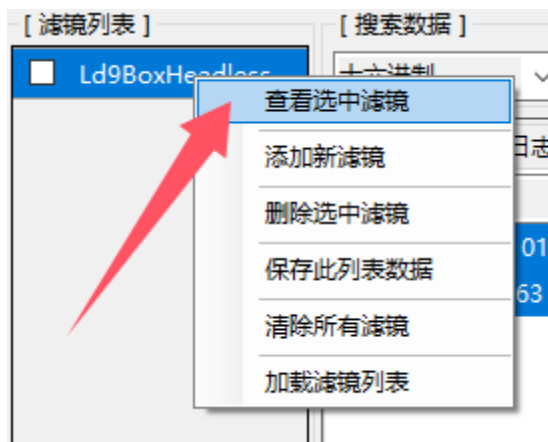
73 65 72 2D 73 65 72 76 69 63 65 05 6C 64 6D 6E 71 03 63 6F 6D 00 00 01 00 01”

经过分析，该数据封包的前6位为“有价值”的数据字段（CB A1 01 00 00 01），并且每次收到的封包都是以固定的前5位数据开头（CB A1 01 00 00），第6位数据为我们可以修改的有效数据位（01）那我们就可以按照如下步骤来配置一个滤镜：

1. 将接收到的包含有可修改数据的封包添加到滤镜列表

| 序号 | 类别 | 套接字 | 源地址 | 目的地址 | 长度 | 数据 |
|----|-------------|------|----------------------|----------------------|-----|---|
| 1 | SendTo | 5104 | 0.0.0.0:60480 | 211.167.97.67:53 | 40 | CB A1 01 00 00 01 00 00 00 00 00 00 ... |
| 2 | WSARecvFrom | 5104 | 211.167.97.67:53 | 0.0.0.0:0 | 40 | CB A1 81 80 00 01 00 03 00 00 00 00 ... |
| 3 | Send | 5128 | 10.172.179.202:54796 | 47.100.224.108:443 | 16 | 16 03 01 00 B6 01 00 00 B2 03 03 6C ... |
| 4 | SendTo | 5896 | 0.0.0.0:60481 | 211.167.97.67:53 | 40 | A0 15 01 00 00 01 00 00 00 00 00 00 ... |
| 5 | SendTo | 5812 | 0.0.0.0:60482 | 0.0.0.0:0 | 52 | 84 7C 01 00 00 01 00 00 00 00 00 00 ... |
| 6 | SendTo | 5544 | 0.0.0.0:60483 | 211.167.97.67:53 | 159 | DD 5C 01 00 00 01 00 00 00 00 00 00 ... |
| 7 | Recv | 5128 | 47.100.224.108:443 | 10.172.179.202:54796 | 16 | 16 03 03 00 4A 02 00 00 46 03 03 7F ... |
| 8 | WSARecvFrom | 5544 | 211.167.97.67:53 | 0.0.0.0:0 | 40 | DD 5C 81 80 00 01 00 04 00 00 00 00 ... |
| 9 | WSARecvFrom | 5812 | 211.167.97.67:53 | 0.0.0.0:60482 | 52 | 84 7C 81 80 00 01 00 01 00 00 00 00 ... |
| 10 | WSARecvFrom | 5896 | 211.167.97.67:53 | 0.0.0.0:60481 | 159 | A0 15 81 80 00 01 00 05 00 00 00 00 ... |
| 11 | Send | 5128 | 0.0.0.0:0 | 0.0.0.0:0 | 93 | 16 03 03 00 25 10 00 00 21 20 62 62 ... |

2. 查看刚才添加的滤镜



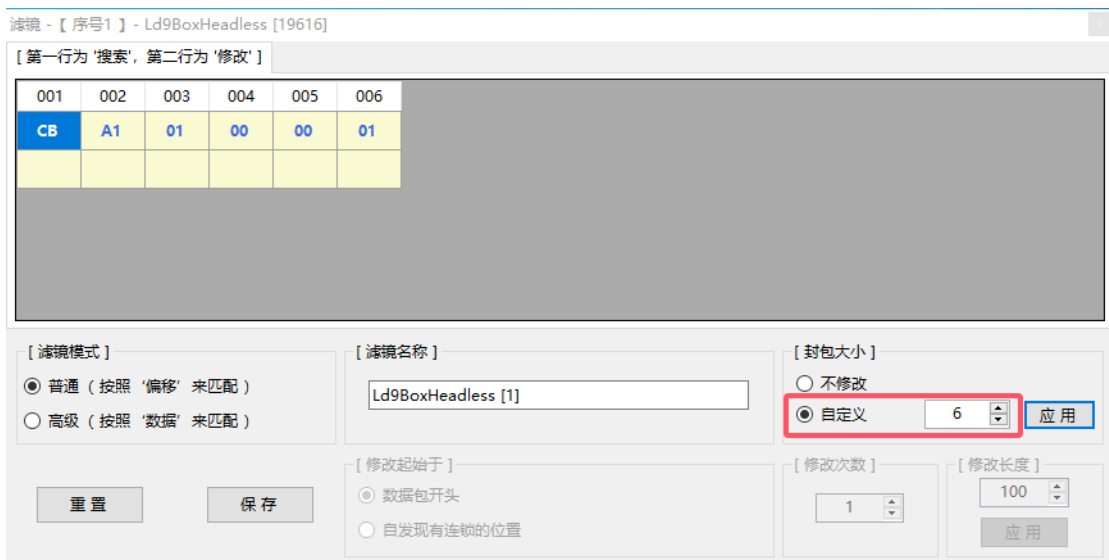


3. 调整滤镜模式

由于有效的数据位置是固定的，所以我们选择“普通”模式即可

4. 调整封包大小

前面我们已经分析过了，因为有价值的数据就在前6位，前5位固定不变，第6位是我们需要的数据位置，其余位置都是无效的数据，不能用于滤镜匹配，所以这里封包大小需要自定义为6



5. 调整匹配行数据

因为封包的前5位是固定的，第6位会变化，所以我们只能以前5位的数据 (CB A1 01

00 00) 来匹配封包，低位数据做为修改位来使用，所以我们需要在第一行匹配数据行上，把第 6 位数据置空，空位置的数据滤镜会跳过不做匹配

[第一行为 '搜索', 第二行为 '修改']

| 001 | 002 | 003 | 004 | 005 | 006 |
|-----|-----|-----|-----|-----|-----|
| CB | A1 | 01 | 00 | 00 | |
| | | | | | |

6. 调整修改行数据

前面我们知道，只有第 6 位的数据才是用来修改的，所以我们在滤镜的第二行修改行上，在第 6 位的位置上填上我们想要修改的数据值，比如 06

[第一行为 '搜索', 第二行为 '修改']

| 001 | 002 | 003 | 004 | 005 | 006 |
|-----|-----|-----|-----|-----|-----|
| CB | A1 | 01 | 00 | 00 | |
| | | | | | 06 |

7. 修改滤镜名称

给滤镜起一个直观的名字吧，方便下次直接使用，全部修改完毕后，切记点“保存”！

如果发现改错了改乱了，也可以点击“重置”按钮，恢复到上次保存时候的样子

滤镜 - 【序号1】 - Ld9BoxHeadless [19616]

[第一行为 '搜索', 第二行为 '修改']

| 001 | 002 | 003 | 004 | 005 | 006 |
|-----|-----|-----|-----|-----|-----|
| CB | A1 | 01 | 00 | 00 | |
| | | | | | 06 |

[滤镜模式]

普通 (按照 '偏移' 来匹配)

高级 (按照 '数据' 来匹配)

[滤镜名称]

某某游戏修改某某值

[封包大小]

不修改

自定义 应用

[修改起始于]

数据包开头

自发现有连锁的位置

[修改次数]

应用

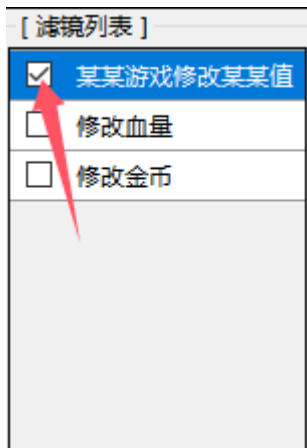
[修改长度]

应用

重置 保存

8. 启用当前滤镜

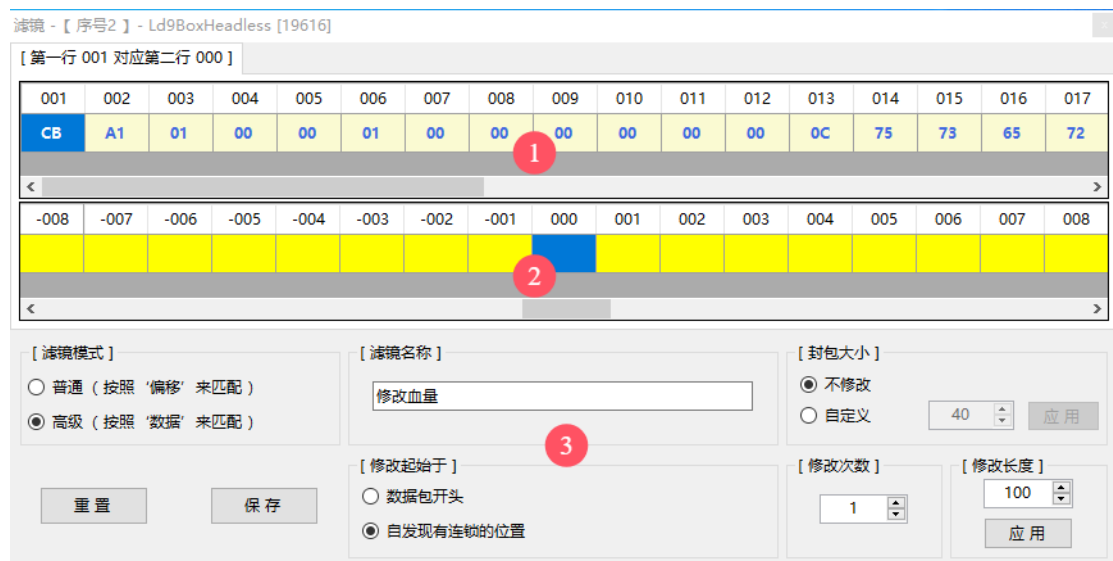
在滤镜列表中，勾选上刚才保存的滤镜后，这个滤镜才会被启用，不勾选则不会启用



经过上述配置后，当我们接收到一个以“CB A1 01 00 00”数据开头的封包后，滤镜就会自动把这个封包的第 6 位数据改成“06”以此来达到修改封包数据的目的

如何使用高级滤镜

我们前面说过，高级滤镜跟普通滤镜不同的地方是，高级滤镜是按照“数据”来进行匹配的，不管这个数据出现在封包的开头，中间，还是在结尾。而普通滤镜是按照“偏移”也就是位置来进行匹配的，必须严格按照数据出现的位置来进行匹配，所以两种滤镜的工作方法不完全相同，高级滤镜的匹配更灵活，配置也更复杂



1. 高级滤镜的第一行数据用来进行匹配

- 1) 匹配行数据跟普通滤镜类似，有数据的会进行匹配，没有数据留空的会跳过不进行匹配
- 2) 高级滤镜是基于数据来进行匹配的，与填写数据的位置无关，但是与整个用于匹配的数据串相对应某个数据所在的位置有关，在填写时需要注意数据和数据之间的相对位置不要填错

2. 高级滤镜的第二行数据用来进行修改

- 1) 第二行填写的数据用于在滤镜匹配成功后执行修改时所用到的数据
- 2) 依据 [修改起始于] 选项的不同选择，第二行修改数据行的填写方式也不一样，在参数配置说明里会详细讲到
- 3) 当选择“自发现有连锁的位置”选项后，匹配行的“001”位置，对应修改行的“000”位置

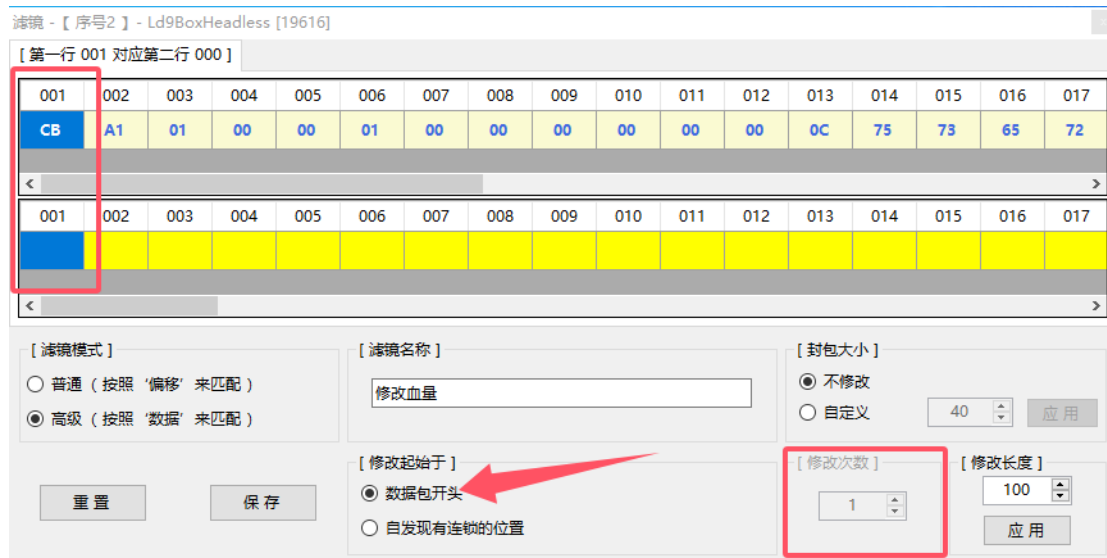
位置

[第一行 001 对应第二行 000]

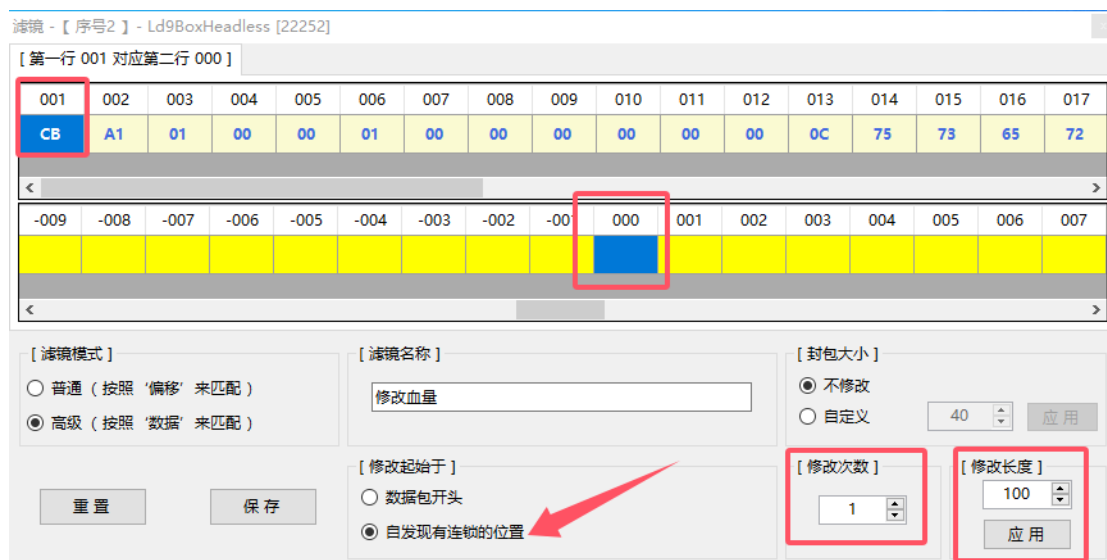
| | | | | | | | | | | | | | | | | |
|------|------|------|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 001 | 002 | 003 | 004 | 005 | 006 | 007 | 008 | 009 | 010 | 011 | 012 | 013 | 014 | 015 | 016 | 017 |
| CB | A1 | 01 | 00 | 00 | 01 | 00 | 00 | 00 | 00 | 00 | 00 | 0C | 75 | 73 | 65 | 72 |
| < > | | | | | | | | | | | | | | | | |
| -004 | -003 | -002 | -001 | 000 | 001 | 002 | 003 | 004 | 005 | 006 | 007 | 008 | 009 | 010 | 011 | 012 |
| | | | | | | | | | | | | | | | | |
| < > | | | | | | | | | | | | | | | | |

3. 高级滤镜参数配置

- 1) 滤镜模式，滤镜名称和封包大小功能与普通滤镜类似，请参考前面的说明
- 2) 修改起始于选项
 - a) 数据包开头：即滤镜匹配成功后，从封包的头部起始位置开始进行数据的修改



- i. 当选择“数据包开头”选项后，第二行修改行的起始位置会跟第一行匹配行的起始位置一样，从 001 开始
 - ii. 当选择“数据包开头”选项后，“修改次数”会变的不可用
- b) 自发现有连锁的位置：即滤镜匹配成功后，从发现匹配数据的起始位置开始进行数据的修改，如果“修改次数”参数大于 1 且有多多个匹配成功的起始位置，则会进行多次修改



- i. 当选择“自发现有连锁的位置”选项后，第二行修改行的位置将变成从负数位到正数位的两倍“修改长度”值所示的长度，调整“修改长度”参数

的值，可以调整第二行修改行的长度

- ii. 第一行匹配行的“001”位置，此时对应的是第二行修改行的“000”位置
- iii. 第二行修改行的负数位置，代表“自发现有连锁的位置”往前递进的坐标
- iv. 第二行修改行的正数位置，代表“自发现有连锁的位置”往后递进的坐标
- v. 当选择“自发现有连锁的位置”选项后，“修改次数”会变得可用

3) 修改次数参数配置

- i. 当选择“自发现有连锁的位置”选项后，“修改次数”会变得可用
- ii. 调整此参数，可调整高级滤镜在一个封包中进行匹配和修改的次数

4) 修改长度参数配置

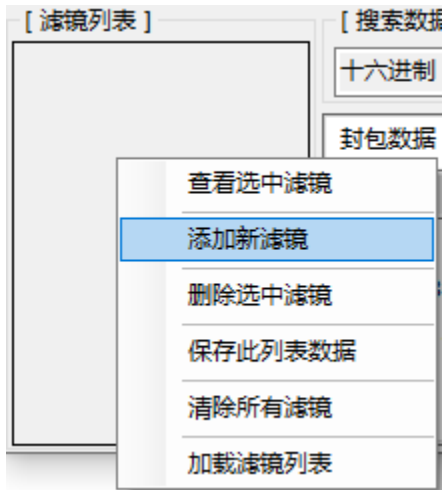
- i. 当选择“高级”滤镜模式后，“修改长度”会变得可用
- ii. 调整此参数，可改变第二行修改行的长度
- iii. 参数调整后，需要点击“应用”按钮才可生效

举例：假设我们接收到一个游戏封包数据“CB A1 01 00 00 01 00 00 00 00 00 00 0C 75

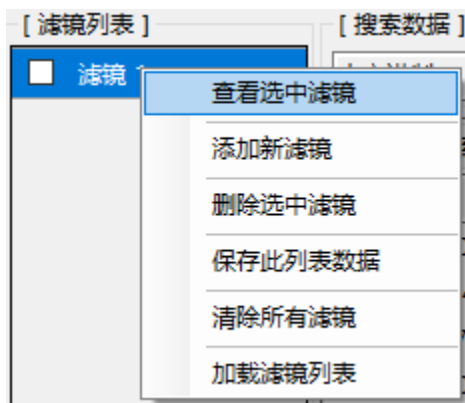
73 65 72 2D 73 65 72 76 69 63 65 05 6C 64 6D 6E 71 03 63 6F 6D 00 00 01 00 01”

经过分析，该数据封包中的“73 65 72”为有价值的位，其中“73 65”为固定不变的数据位，“72”为可修改的数据位，但是数据出现的位置不是固定在封包头部或者其它某个固定的位置，而且整个封包的长度又是动态变化的，无法使用“普通滤镜”的固定位置来匹配封包，那么这种情况下我们就需要通过“高级滤镜”来达到修改的目的了

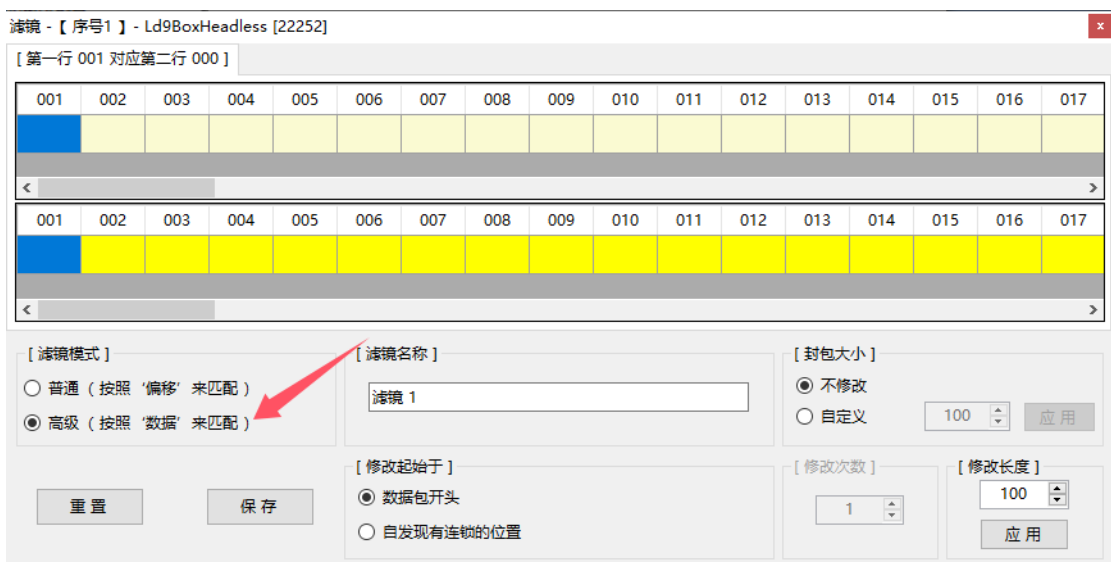
1. 首先，我们新建一个滤镜



2. 查看新建的滤镜



3. 选择“高级”滤镜模式

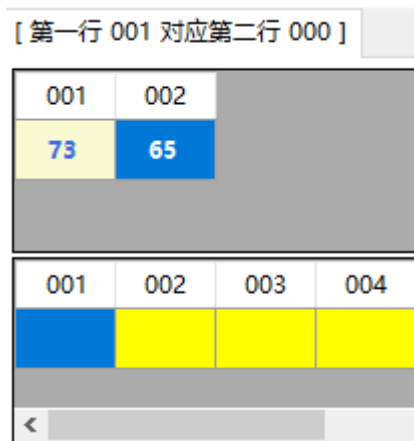


4. 我们要匹配的有价值数据是“73 65” 所以在这里把“封包大小” 自定义为 2，点击应

用生效



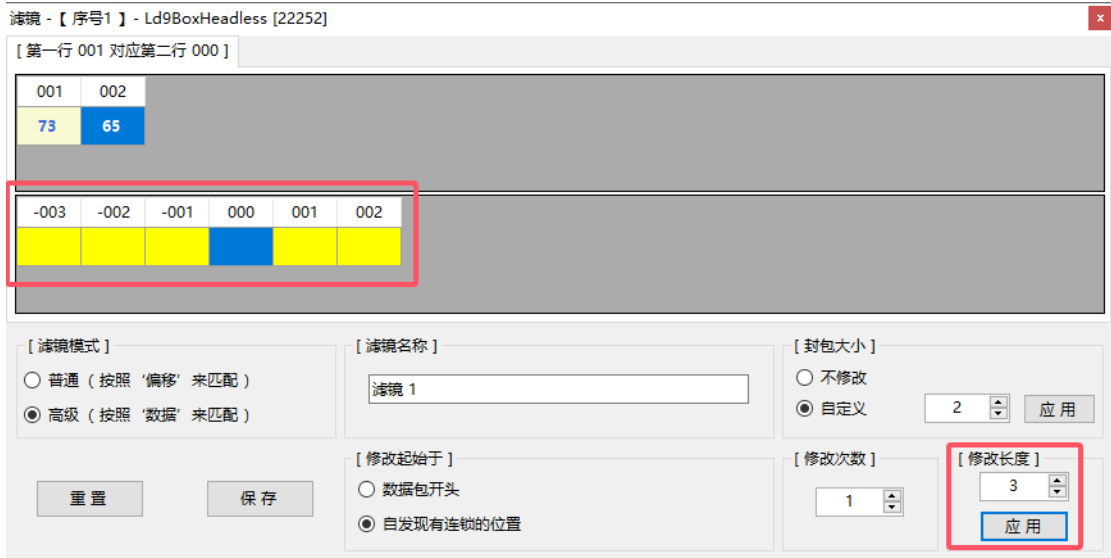
5. 将“73 65”填入第一行匹配行



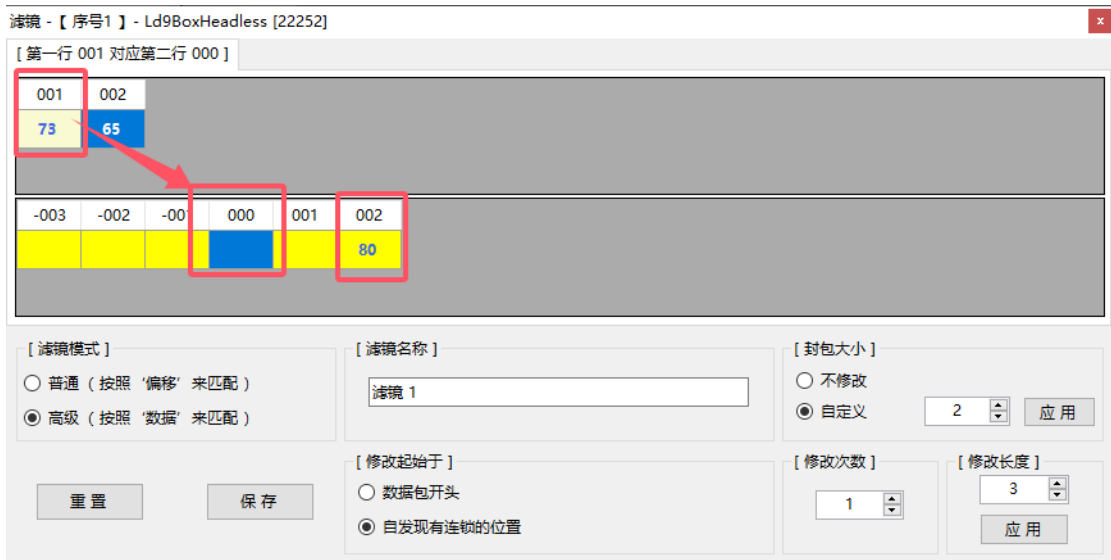
6. 因为有价值的数据位出现的位置不是固定在封包头部位置的, 需要修改的数据位也是跟在后面位置的, 所以这里需要选择“自发现有连锁的位置”



7. 前面我们分析得到“73 65”是固定的数据,需要修改的是相对位置为第三个的数据“72”,
所以这里我们把“修改长度”参数设置为 3 点击应用



8. 在“002”位置,我们填上需要修改的数据,比如“80”

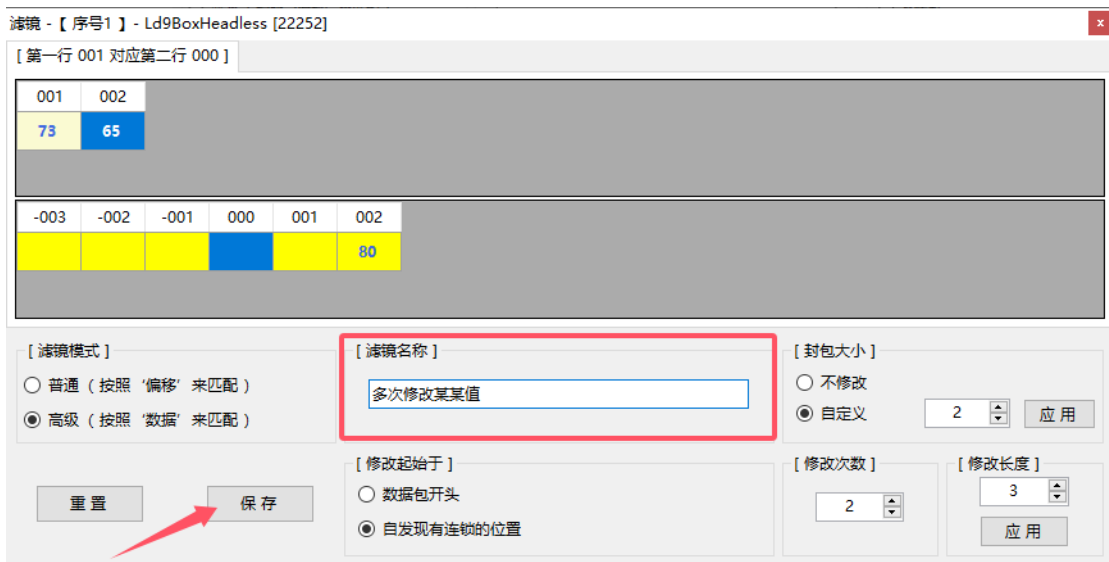


这里需要注意下,第一行匹配行的“001”位置对应的是第二行修改行的“000”位置,
而不是“001”位置,所以需要修改的“72”数据对应的位置就是“002”了

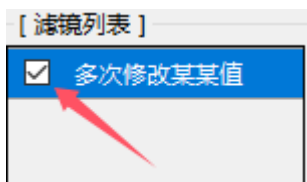
9. 由于“73 65 72”在封包中出现了 2 次,我们也需要修改 2 次,所以这里把“修改次数”参数设置成 2,当然,如果需要修改多次,并且不确定会出现多少次的情况下,可以尽量把这个参数设置的大一点,滤镜会自动忽略掉多余的次数



10. 给滤镜起一个名字后点击“保存”按钮



11. 最后在滤镜列表中勾选当前滤镜



12. 至此，一个可以按照数据来进行匹配的高级滤镜就完成了，这样配置的滤镜会根据设置的匹配数据在封包中按照设置的“修改次数”进行动态的匹配，匹配成功后根据数据出现的位置来动态修改封包

系统日志及其功能菜单

软件会在系统日志区域记录系统的运行日志，包括部分主要的功能模块运行情况，已经所有的报错信息

| 记录时间 | 模块 | 日志内容 |
|----------|------------------|------------------------------|
| 14:59:58 | InitSocketDGV | 初始化数据表完成 |
| 14:59:58 | InitSocketForm | 目标进程: Ld9BoxHeadless [19616] |
| 15:00:01 | SetSocketParam | 设置拦截参数完成 |
| 15:00:01 | bStartHook_Click | 开始拦截! |
| 15:00:08 | bStopHook_Click | 结束拦截! |

在系统日志区域点击鼠标右键可以弹出系统日志的功能菜单

| 日志内容 |
|---------------|
| 初始化数据表完成 |
| 目标进程: Ld9BoxH |
| 设置拦截参数完成 |
| 开始拦截! |
| 结束拦截! |

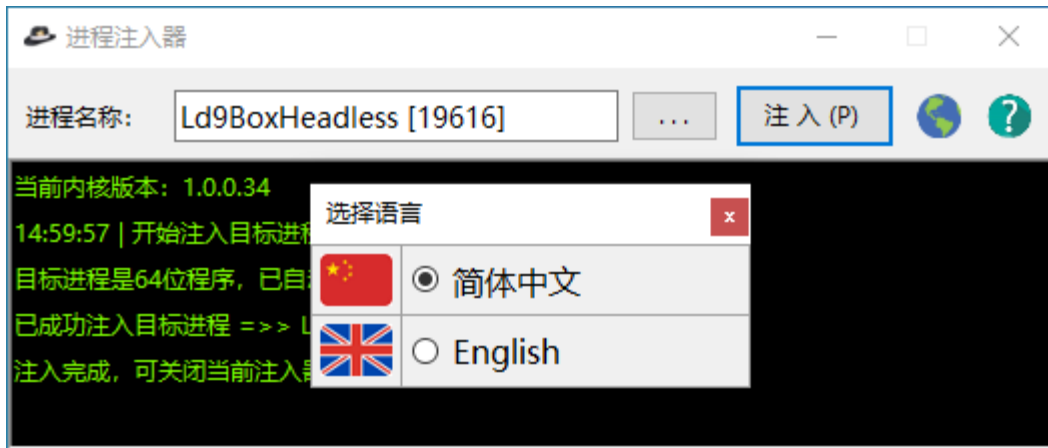
清空日志列表

导出到Excel

1. 清空日志列表: 清除日志列表区域内的所有日志信息
2. 导出到 Excel: 将日志列表区域内的所有日志信息导出到一个 Excel 文件中 (由于程序以字符串形式导出成 excel 文件, 所以在打开导出的 excel 文件时会提示格式不正确, 点击确认即可正常打开, 可再另存为标准的 excel 格式的文件)

多语言支持

本软件目前支持简体中文和英语两种语言, 后期会视使用情况再行调整



选择好语言，点击关闭窗口后会提示“软件语言已更改，需要重新启动程序”，待程序自动重新启动后即会切换到选择的对应语言，下次再启动程序会直接显示选择的语言界面

关于本软件

本软件已开源并将源代码上传到了 Github 和 Gitee 上，欢迎大家一起来添砖加瓦，如在使用过程中碰到问题或者反馈 bug 的，可以截图发到 Github 上，或者也可以直接电邮联系我处理，我会尽快给您回复，感谢大家的支持！

程序的卸载

- 在线安装版

“控制面板” - “卸载软件” - “封包拦截器”

- 离线打包版

直接删除下载的文件夹即可