1. **管控端设计**
2. **页面功能**
   1. **任务基本信息查看**

进入"报文抓取"菜单后，默认可以显示已设置的所有抓包任务，当历史任务过多时可分页显示，默认每页显示25条，用户可自定义选择分页条数，可选项为：10条/页、25条/页、50条/页、100条/页，存在多页数据时，用户也可以手动输入页码来跳转到指定页查看；

任务信息栏直接显示的内容包括（顺序从左到右）：开始时间、结束时间、抓包流量口、抓包时长限制、文件大小限制（字节）、实际文件大小（字节）、文件名称、IP、端口、任务状态、操作；（其中IP、端口、网络层协议、传输层协议、应用层协议，如果用户无自定义输入或选择的全部，则显示”全部”）

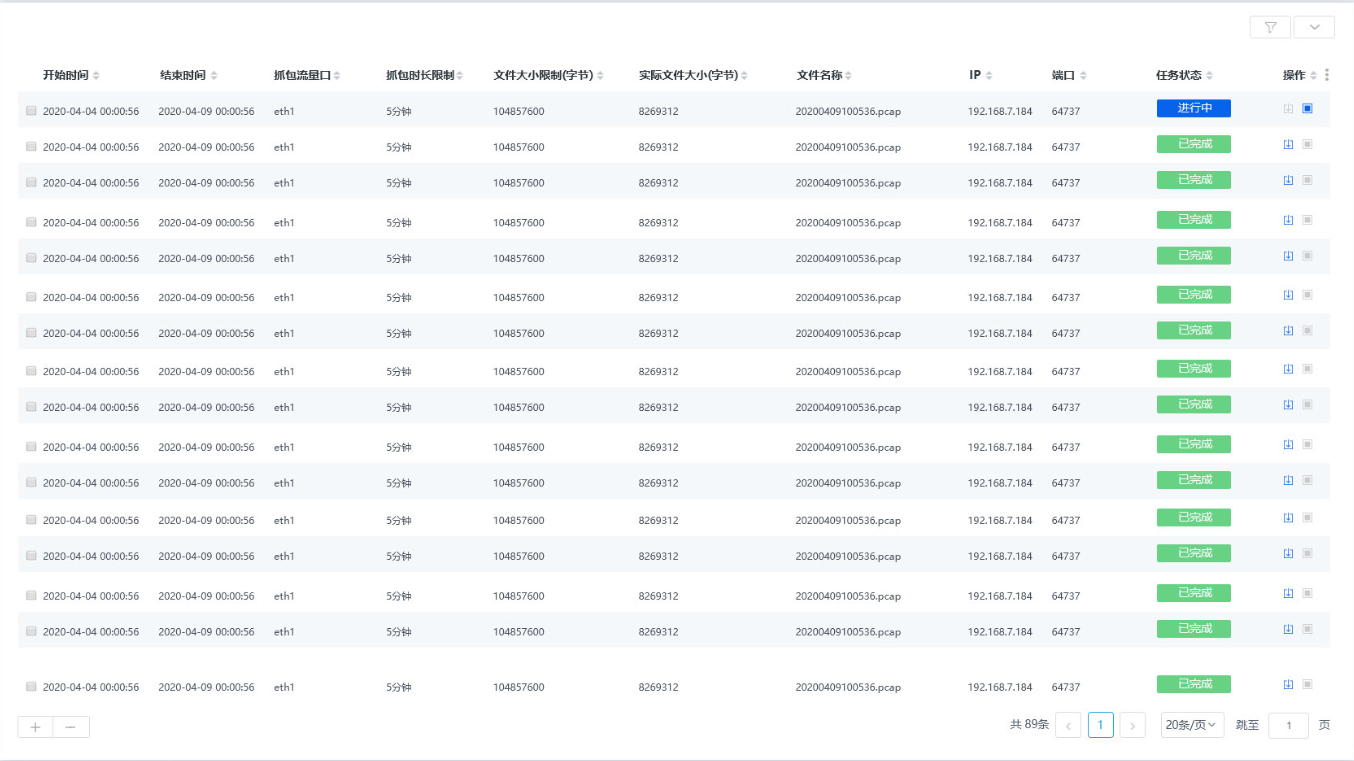
注：文件大小限制用户配置值单位为MB，之所以这里用字节，一个因为实际文件大小可能只有几KB，单位用MB会涉及计算和显示的问题，二是为了文件大小限制和实际文件大小采用同一单位，看起来统一；

请注意，为保持和tcpdump的参数生效计算方法一致，MB到KB到字节的转换，采用1000而不是1024进行换算；

任务信息展示栏数据列，可进行自定义选择，点击任务信息名称栏右侧的按钮可弹出“自定义列”操作框，对展示的信息列进行选择，比如可勾选显示默认隐藏的网络层协议、传输层协议、应用层协议（网络层协议、传输层协议、应用层协议，如果用户无自定义输入或选择的全部，则显示”全部”）；自定义列中“文件名称”为必须展示列，默认灰掉不可取消选择；

任务信息展示界面添加自动刷新机制，默认时间设为1分钟，自动刷新间隔暂时不暴露界面配置项；用户也可以点击页面右上角刷新按钮进行手动刷新；

界面效果图如下：



* 1. **任务信息排序**

任务信息展示，默认按开始时间逆序排列，最新的任务在最上面，另外，可点选开始时间、结束时间、抓包流量口、抓包时长限制、文件大小限制（字节）、实际文件大小（字节）、文件名称、IP、端口、任务状态，进行自定义排序；

（开始时间、结束时间按时间排序，抓包时长限制、文件大小限制、实际文件大小按数字排序，其余均为按字符串排序）

* 1. **任务状态和操作**

抓包任务信息展示栏，其最后为"任务状态"、"操作"两栏，"任务状态"可展示的值为"已完成"、"进行中"，"操作"栏显示下载、停止两个操作按钮，功能说明如下：

**任务状态：**

"进行中"：当前正在进行抓包的任务，仅停止按钮可点击，下载按钮置灰不可用；

"已完成"：已经结束的抓包任务，仅下载按钮可点击，停止按钮置灰不可用；（非手动点击停止触发的抓包任务结束，比如达到抓包时长或文件大小限制，需要通过界面刷新后获取新的状态值来更新，相当于界面主动获取新的状态，而不是后台主动通知界面）

效果如下：

  
**操作按钮：**

下载：下载对应任务的pcap文件，仅”已完成”的任务可以点击下载按钮；鼠标移动到

下载按钮上面，需要显示”下载”字样提示；

停止：停止进行中的抓包任务，状态变为"已完成"，同时下载按钮变为可用状态；鼠

标移动到停止按钮上面，需要显示”停止”字样提示；

（每一个抓包任务实质上为后台单独运行的一个tcpdump进程，停止抓包任务实际操作为直接杀掉进程，理论上没有延迟，前端需要等待进程杀掉之后（数据库中的状态变为“已完成”），才更新状态和允许点击下载）

* 1. **任务自定义筛选**

"报文抓取"页面右上角显示过滤按钮（显示为小漏斗图标），以及保存的过滤条件列表按钮（显示为下箭头图标），可对历史任务进行自定义筛选过滤查询；

用户通过将自定义的过滤条件保存为场景模板，下次可直接进行同条件快捷筛选；

过滤条件展开效果如下：



支持的过滤项包括：

* 开始时间：时间选择框，默认为空，代表不做开始时间过滤；
* 结束时间：时间选择框，默认为空，代表不做结束时间过滤；
* 抓包流量口：下拉框，显示已配置的流量口，下拉框默认显示水印“请选择”，代表不做任何流量口过滤，可选项为“全部”、所有流量口，选择“全部”即等同于不做任何流量口过滤；
* 文件名称：输入框，用户可自定义输入文件名称进行筛选，暂时不对用户输入的文件名做检验或特殊字符限制，输入框后面的tooltips将提示“最多允许输入256个字符”，文件名称将对所包含的英文字符大小写敏感；
* 任务状态：下拉框，默认显示水印“请选择”，代表不做任何任务状态过滤，可选项包括“全部”、“已完成”、“进行中”，选择“全部”即等同于不做任何任务状态过滤；
* IP：输入框，允许用户输入IP地址，输入框后面的tooltips将给出提示内容如下：



* 端口：输入框，允许用户输入端口，输入框后面的tooltips将给出提示内容“合法端口范围1~65535”；
* 网络层协议：下拉框，默认显示水印“请选择”，代表不做任何网络层协议过滤，可选值为“全部”、“IPv4”、“IPv6”，选择“全部”即等同于不做任何网络层协议过滤；
* 传输层协议：下拉框，默认显示水印“请选择”，代表不做任何传输层协议过滤，可选值为“全部”、“TCP”、“UDP”、“ICMP”、“ICMPv6”，选择“全部”即等同于不做任何传输层协议过滤；
* 应用层协议：输入框，用户可自定义输入应用层协议名称进行筛选，暂时不对用户输入的协议名称做检验或特殊字符限制，输入框后面的tooltips将提示“最多允许输入64个字符，不区分大小写”；协议名称输入不区分大小写，后台查询的时候将进行忽略大小写查询；

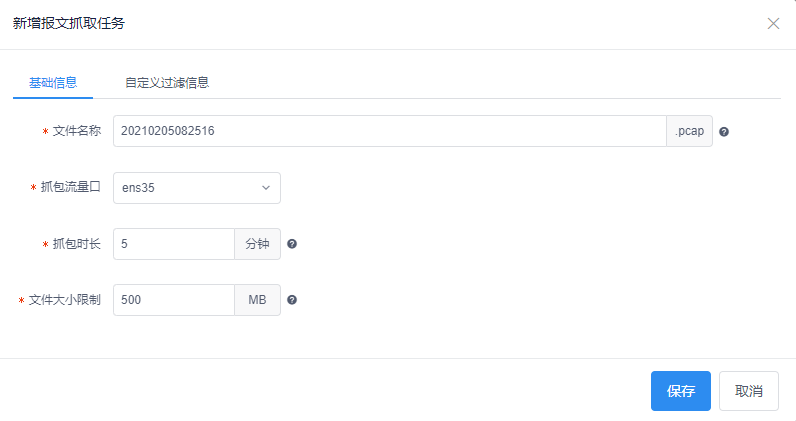
注：所有输入框内容，如文件名称、IP、端口、应用层协议，默认均为单个输入，不支持输入多个值然后分拆筛选过滤；

* 1. **任务添加和删除**
  2. **任务参数配置**

"报文抓取"页面左下角，显示添加按钮（显示为+号图标）、删除按钮（显示为-号图标），对于抓包任务，只允许添加或删除，不允许对已添加的任务进行编辑；

新增任务后抓包立即执行，操作说明如下：

点击添加按钮，弹出“新增报文抓取任务”配置页面，效果示意图如下：







用户可配置项包括：

1. **基础信息（基础信息是每个任务创建必须要有的信息，配置项通用的，但是参数值不通用）：**

* 文件名称

添加任务时，默认以当前浏览器PC端格式化时间（精确到秒）作为文件名，如20200409100536，用户可自定义进行修改；（20200409100536显示在编辑框中，固定后缀.pcap后缀显示在编辑框后面，不可修改）

暂时不对用户输入的文件名做检验或特殊字符限制，输入框后面的tooltips将提示“最多允许输入256个字符”，文件名称将对所包含的英文字符大小写敏感；

* 抓包流量口

默认读取当前用户配置的流量口，以下拉框形式展示，如果仅有一个流量口，则默认展示该流量口，且下拉框置灰不可操作；

如果配置了多个流量口，缺省显示第一个（后台接口返回的第一个流量口，也即是card.conf文件中的第一个流量口），一个任务，只能选择一个流量口进行抓包；

* 抓包时长

以输入框形式供用户输入抓包时长，输入框后面显示时间单位，默认单位为"分钟"且不可修改；（抓包时间只允许配置时长，比如5分钟、10分钟，暂不支持自定义抓包开始时间）

抓包时长配置上设置最大限制为1440（即24小时），最小限制为1（界面需对用户配置值进行合法性校验），抓包时长和文件大小双重限制，先达到一个即停止了；

输入框后面的tooltips可提示用户“最小输入值为1，最大输入值为1440”；

* 文件大小限制

以输入框形式供用户输入抓包文件的最大尺寸，输入框后面紧跟单位，默认单位为"MB"且不可修改；

文件大小限制配置上最大限制为500，最小限制为1（界面需对用户配置值进行合法性校验），抓包时长和文件大小双重限制，先达到一个即停止了；

输入框后面的tooltips可提示用户“最小输入值为1，最大输入值为500”；

注：基础信息为用户必须输入的值，因此，基础信息名称前面都需要加上星号表示必填项；

1. **自定义过滤信息**（用户配置基础信息后，即可添加新的抓包任务，自定义过滤信息为可选项，可在用户需要时，切换到“自定义过滤信息”TAB页进行配置）：

* 协议

协议包含两个配置类别：

**协议选择：**

以单选框形式提供"全部"、"IPv4"、"IPv6"三个互斥选项，默认选择“全部”，选择 "全部" 代表不做任何协议过滤，选择 "IPv4"，代表仅抓取IPv4的报文，选择"IPv6"，代表仅抓取IPv6的报文；

当用户选择"全部"时，则默认代表选择所有协议，也无任何二级参数扩展显示；

当用户选择IPv4协议时，扩展显示"全部"、"TCP"、"UDP"、"ICMP"四个单选互斥二级参数，四个参数为单选框形式，只允许四选一，默认选择“全部”；

当用户选择IPv6协议时，扩展显示"全部"、"TCP"、"UDP"、"ICMPv6"四个单选互斥二级参数，四个参数为单选框形式，只允许四选一，默认选择“全部”；

**自定义应用协议：**

以输入框形式允许用户输入自定义应用层协议，用户输入的协议名称不允许包含以下特殊字符：”/”、”\”、”\*”、”.”、”+”（分别为正斜杠、反斜杠、星号、点、加号），其他特殊字符暂时不做校验和限制，输入框后面的tooltips将提示“最多允许输入64个字符，不区分大小写，且不能含有/\\*.+字符”，应用层协议暂不允许用户输入多个协议名称；协议名称输入不区分大小写，后台将对输入的应用层协议名称进行校验，如果输入协议不在支持列表中（/etc/services文件），则添加任务时提示错误；

自定义应用协议的配置生效逻辑，是直接配置到tcpdump的参数中，以类似 port http这样的方式下发，抓包的时候，会将应用协议名称如http，去/etc/services中查询并替换成对应的端口之后进行过滤抓包（tcpdump抓包过程中，不可能进行不依赖端口的协议识别然后进行协议过滤，只能依赖固定定义的标准端口，比如http就是80，来进行过滤）；

tcpdump去/etc/services中查询应用协议名称时，是区分大小写精确匹配的，如果界面配置自定义应用协议，也设计成区分大小写，将会导致配置如"HTTP"这样的名称出现报错，因为/etc/services中是存储的"http"；

因此，如果界面配置也区分大小写，会损失大量常用协议的用户配置体验，经过分析，/etc/services（以当前CentOS 6.9操作系统为例）中，总共生效协议为 10679个，其中，包含大写字母的，仅有61个，绝大部分，都是纯小写，因此，经过讨论，宁可损失这61个协议（几乎都不常用，列表如下），也要保证常用协议的配置体验；

包含大写字母的应用协议列表如下，这些协议，用户基本很难用到，也很难配置正确（必须大小写都准确），自定义应用协议支持选择性牺牲掉这些协议：

wnn4\_Kr  
wnn4\_Cn  
wnn4\_Tw  
CAIlic  
CAIlic  
tnETOS  
tnETOS  
dsETOS  
dsETOS  
warmspotMgmt  
warmspotMgmt  
XSIP-network  
XSIP-network  
3Com-nsd  
3Com-nsd  
EtherNet/IP-1  
EtherNet/IP-1  
LiebDevMgmt\_C  
LiebDevMgmt\_C  
LiebDevMgmt\_DM  
LiebDevMgmt\_DM  
LiebDevMgmt\_A  
LiebDevMgmt\_A  
BESApi  
BESApi  
iRAPP  
iRAPP  
PowerAlert-nsa  
PowerAlert-nsa  
MaxumSP  
MaxumSP  
prRegister  
prRegister  
SunVTS-RMI  
SunVTS-RMI  
joaJewelSuite  
joaJewelSuite  
MobilitySrv  
MobilitySrv  
XmlIpcRegSvc  
XmlIpcRegSvc  
Mon  
Mon  
MOS-lower  
MOS-lower  
MOS-upper  
MOS-upper  
MOS-aux  
MOS-aux  
MOS-soap  
MOS-soap  
MOS-soap-opt  
MOS-soap-opt  
WibuKey  
WibuKey  
CodeMeter  
CodeMeter  
DMExpress  
DMExpress  
EtherNet/IP-2  
EtherNet/IP-2

关于自定义应用协议的特殊字符限制逻辑，/etc/services中包含特殊字符有以下几种：

/、.、\*、+、-

其中：

" - "为常见字符，下发和生效都没有任何问题；

" / "验证有问题，需要禁止输入；

" \* "、" + "，通过实测，发现tcpdump自己都不支持，如下：  
[root@iSec ~]# tcpdump "port whois++" -i eth1 -s0 -C 100 -Z root -w /home/bds/capture/3.pcap   
[PF\_RING] mmap() failed: try with a smaller snaplen  
tcpdump: WARNING: eth1: no IPv4 address assigned  
tcpdump: illegal token: ++  
[root@iSec ~]# tcpdump "port sql\*net" -i eth1 -s0 -C 100 -Z root -w /home/bds/capture/3.pcap   
[PF\_RING] mmap() failed: try with a smaller snaplen  
tcpdump: WARNING: eth1: no IPv4 address assigned  
tcpdump: unknown port 'sql'  
  
而剩下的"."，虽然tcpdump可以支持，但是下发过程中的合法性判断（对/etc/services进行grep），会把其当作grep正则的通配符来处理，也有隐患，而包含" . "的，只有2种协议，如下：

[root@iSec ~]# awk -F' ' '{print $1}' /etc/services | grep -v "^#" | grep -F "."  
z39.50  
z39.50  
novell-lu6.2  
novell-lu6.2  
并不属于常见协议；

综上，最简单也最稳妥的方案，就是直接通过界面，禁止输入”/”、”\”、”\*”、”.”、”+”（分别为正斜杠、反斜杠、星号、点、加号）这几个特殊字符，避免产生干扰；

如果用户选择了ICMP或ICMPv6协议，则“自定义应用协议”输入框隐藏不可输入；

* IP

以输入框形式允许用户输入自定义IP，支持格式包括单个IP（如192.168.1.2）、带掩码的IP段（如192.168.1.0/24），界面需对用户配置值进行合法性校验，IP或IP段只允许用户配置一个；

IP在输入框后面需采用tooltips的方式给予用户提示，提示内容如下：



（A-B格式的地址范围，因为BPF语法原生并不支持，需要涉及复杂中间语法转，因此暂不支持A-B格式）

* 端口

以输入框形式允许用户输入自定义端口，端口合法值为 1 ~ 65535，界面需对用户配置值进行合法性校验，端口允许用户添加一个或多个；

端口输入框后面的tooltips给出提示“合法端口值 1 ~ 65535，可输入多个端口，多个端口之间用英文字符逗号分隔”

如果用户选择了ICMP或ICMPv6协议，则“自定义应用协议”输入框隐藏不可输入；

自定义过滤信息的组织方式内部固定，如下：

协议、IP、端口三个过滤条件之间，采用“与”的方式组织；

协议内部的一级参数，如IPv4和自定义应用协议之间，或者一级参数与二级参数之间，如IPv4和TCP，采用“与”的方式组织；

多个端口之间，采用“或”的方式组织；

* 1. **任务预检查**

任务信息及过滤参数配置完成后，点击"确定"按钮添加任务，此时流程如下：

1. 界面先检查文件名称、抓包时长、文件大小限制、IP、端口、自定义应用协议用户输入值的合法性，如不合法给出错误提示让用户重新输入；
2. 基本参数校验通过后，界面将所有任务参数以json格式组装成字符串后，传递给后台接口；
3. 后台接口直接通过数据库查询，检查当前”进行中”的任务个数，如果当前”进行中”任务已经达到了16个，则返回错误信息，界面提示用户“当前进行中的报文抓取任务已达上限（16个），无法添加新任务”，并关闭添加任务配置窗口返回；
4. 后台接口直接通过数据库查询，检查当前所有历史任务的文件大小限制和实际文件大小，通过所有已完成任务的实际文件大小，加上进行中任务的文件大小限制，计算剩余磁盘分区容量，判断新加任务配置的文件大小限制是否超过剩余容量，超过则返回错误信息，界面给出错误提示并让用户修改配置值；（界面提示的错误信息将根据后台接口返回值给出修改值建议）
   1. **任务删除**

任务信息列表单选或多选任务，点击左下角删除按钮，将弹出提示框，提示信息为”共选择1个任务，系统将删除所选任务的任务信息及其抓包文件，是否确定？”，实际数字为用户选择的任务数量，提示框允许用户点击”确定”按钮下发删除申请，或者点击”取消”按钮取消；

用户确认下发删除请求后，界面将所选任务的文件名称（数据库主键，有这个参数就够了）传递给后端接口，后端接口将会对所选任务的mysql数据库数据，及其关联pcap文件、日志文件，一并删除；

“进行中”任务不允许删除，如果用户选择了“进行中”任务，界面需要给出错误提示，提示内容如下：

“进行中”任务不允许直接删除，请取消相关任务选择，或者先停止“进行中”任务，再删除；

1. **后台设计**
2. **抓包文件存储设计**
3. **存储分区设计**

后台创建抓包文件专用存储分区，暂定名称为 /home/lany/capture，分区大小为30G；（实际可用空间限制为25G，剩余5G作为磁盘冗余，单个任务最大文件为500M，那么最少可以存储50个历史任务，当前应用场景视为足够）

1. **文件存储设计**

报文抓取任务保持的pcap文件，存储在 /home/lany/capture目录下，完整文件路径样例如下：

/home/lany/capture/20200409100536.pcap

1. **接口设计**

后台提供任务配置接口，接收界面下发的新增任务参数，进行任务检查、任务信息入库、以及任务启动；

前后端接口通信规范如下：



1. **任务数判断**

通过数据库查询，检查当前”进行中”的任务个数，如果当前”进行中”任务已经达到了16个，则返回错误信息，否则进入下一步检查；

返回错误信息为“当前进行中的报文抓取任务已达上限（16个），无法添加新任务”；

1. **任务唯一性判断**

后台接收任务参数后，首先检查文件名称的数据库主键相同信息是否存在，如果存在则返回错误信息，界面提示用户修改文件名后重新下发；

返回错误信息为“当前已存在相同文件名称的任务，请修改文件名称后重新添加”；

1. **磁盘分区使用率判断**

当前端下发添加任务，根据任务参数进行抓包文件大小和磁盘分区使用的预检查和判断，逻辑如下：

* 1. 通过数据库查询，检查当前所有历史任务的文件大小限制和实际文件大小，通过所有已完成任务的实际文件大小，加上进行中任务的文件大小限制，计算剩余磁盘分区容量，判断新加任务配置的文件大小限制是否超过剩余容量（总可使用空间目前默认固定为25G），超过则返回错误信息，否则返回成功；

返回的错误信息格式示例如下：

当前抓包文件存储分区剩余空间300M，请修改文件大小限制小于剩余空间，或手动删除不再需要的历史任务腾出空间，然后重新添加；

* 1. 由于临时流量采集是用户的主动行为、临时行为，因此后台暂不设置自动的老化清理机制；

1. **任务执行**

后台提供脚本 /home/lany/bin/utils/manual\_capture.sh ，负责实际抓包任务的执行、数据库记录的插入和更新，启动时需传入参数包括：文件名称、抓包流量口、抓包时长、文件大小限制、网络层协议、传输层协议、应用层协议、IP、端口，示例如下：

/home/lany/bin/utils/manual\_capture.sh 20200409100536.pcap eth1 5 500 ipv4 tcp,udp all 192.168.1.2 80,53

（完整启动命令 nohup /home/lany/bin/utils/manual\_capture.sh 20200409100536.pcap eth1 5 500 ipv4 tcp,udp all 192.168.1.2 80,53 >/dev/null 2>&1 &）

manual\_capture.sh脚本除启动抓包任务外，还将负责插入数据库记录、定时进行当前文件大小更新，以及任务结束时进行当前文件大小、任务状态、结束时间的更新；

单个报文抓取任务执行过程中，系统中将运行着三个可见进程：manual\_capture.sh（带任务参数）、manual\_capture.sh脚本启动的对应tcpdump进程（带任务参数，因为涉及外层运行时间控制，即达到指定时间后停止的机制，所以tcpdump进程信息会有两个）；

抓包任务执行过程中，将记录运行日志，日志目录为 /home/lany/log/manual\_capture，以文件名称命名，如/home/lany/log/manual\_capture/20200409100536.pcap.log，任务删除时，对应的运行日志需要一并删除；

1. **任务停止**

后台接口提供任务停止的操作，允许前端下发任务停止动作，此时将强行终止正在执行的抓包任务，并更新其最新文件大小、任务状态、结束时间到数据库；

1. **系统异常重启后的任务检查**

如果报文抓取任务执行过程中出现软件系统重启，则“进行中”任务无法正常结束，将会一直存在于数据库中，成为“僵尸”任务；

为了避免这种情况，在系统启动程序 /home/lany/bin/startup.py流程中添加判断，在通用工具模块启动流程开始时（/etc/lany/modules/ta-tools/module-scripts/utils/startup.sh），先启动mysql，然后强制停止所有进行中的抓包任务，并更新其最新文件大小、任务状态、结束时间到数据库；

1. **恢复出厂设置**

当用户手动进行恢复出厂设置操作时，在恢复出厂设置程序 /home/lany/bin/reinit.py流程中，通用工具模块进行恢复出厂设置时（/etc/lany/modules/ta-tools/module-scripts/utils/reinit.sh）添加对所有报文抓取任务的强制清除；（包括数据库记录和对应抓包文件）

1. **数据库表设计**

后台新创建一个mysql数据库表，用于存储抓包任务信息，字段信息包括开始时间、结束时间、抓包流量口、抓包时长限制、文件大小限制、当前文件大小、文件名称、任务状态、网络层协议、传输层协议、应用层协议、 IP、端口，文件名称为表主键；

数据库表信息描述文件如下：



1. **数据库表操作**

综上，对于数据库表数据的操作，分工如下：

1. 数据库的数据记录删除及其相关文件删除（包括日志文件），由后台接口经过任务参数检查后执行；
2. 新任务数据插入、数据库的数据更新，比如实际文件大小、任务状态、结束时间，由单个抓包任务脚本负责执行；
3. 数据库数据的查询读取，由前端界面直接执行；