GRE\_分析软件框架

时间：2015/12/7

背景：GRE\_分析软件框架计划；

考虑因素：

* 数据包\_IP分片；缓存，查找(找到与找不到)，老化，整合分片包（多片整合），处理合并包；转发；
* GRE\_数据包：分片，查找（找到与找不到），老化，整合分片包(单片整合)，处理合并包；转发；
* PPP\_数据包：IP数据包：压缩头(7e 21)与非压缩头(7e ff 03 00 21)； 控制链路数据包：SCIP, NCP；

程序主体框架流程图，如图1



图 1 程序主题框架\_流程图

模块流程图

ppp\_process\_packets，图2；



图 2 ppp\_process\_packets

ppp\_process\_packet,图3:



图 3 ppp\_process\_packet

ppp\_process\_packet\_normal,图4



图 4 ppp\_process\_packet\_normal

主框架处理说明

数据包进来之后，将会按照其上层协议类型判断是否是GRE(0x27)协议，不是将会按照默认处理，如果是则进入我们的处理模块中来；

数据包类型GRE，通过df来判断是否是分片包，如果df==1，则表示该数据包不是一个分片包，那么就可以使用标准GRE数据包处理模块ppp\_process\_packets进行处理，ppp\_process\_packets模块将会分析当前数据包内部包含有几个ppp报文，并重新拼接数据包使用原数据包的以太网头，IP头，GRE头，每一段ppp\_payload包含头7e和尾7e(如果没有同样还是把该段组进来，后面函数将会继续判断处理)，组成一个数据包送到单个数据包处理模块ppp\_process\_packet中；该处理模块将会判断当前单个数据包是否是完整包，如果是，将会进入ppp\_process\_packet\_normal模块中，保留MAC字段，移除外层IP字段，GRE字段，对ppp\_payload字段进行7d反转义，然后移除头部，尾部0x7e；然后将处理完成的数据包写入pcap\_dump中；如果该数据包不是一个完整包，那么就需要判断该数据包不完整状态，是1、有开头，没有结尾；2、有结尾没有开头；3、既没有开头，又没有结尾；两个都有那么就是完整的包；

1. 有开头，没有结尾；

将提取该数据包中GRE字段,key,seqnum关键字，并将该数据包进行缓存，保留pkt\_buff,和长度length；

1. 有结尾，没有开头；

提取数据包GRE字段中，key在gre\_list表中进行查找，是否有匹配通道缓存数据包，如果没有，直接丢弃；如果有，查看表中seqnum关键字是否是当前关键字seqnum-1，如果不是，则直接丢弃；如果是，则将当前数据包中部分和缓存数据包进行重组，然后送到ppp\_process\_packet\_normal模块进行处理；

1. 没有开头，也没有结尾；

提取数据包GRE字段中，key在gre\_list表中进行查找，是否有匹配通道缓存数据包，如果没有，直接丢弃；如果有，则查看表中seqnum关键字是否是当前关键字seqnum – 1，如果不是，则直接丢弃，如果是，则将该数据包中内容和缓存数据包进行merge，然后修改关键字length大小；

如果DF=0，则表述该数据包有分片包；那么查看MF标志位是否为0；如果mf==0，那么表示当前数据包应该是分片包中最后一个，然后看该数据包的offset，如果该数据包的offset=0，则表示当前数据包虽然是分片包，但是同样只有一片，这里就可以直接将该数据包当作独立包进行处理，送到ppp\_process\_packets模块；如果offset不为0，那么就需要计算当前hash\_id作为key查IP表，如果没有找到，则表明当前数据包时最有一片，但是之前几片都丢失或还没有到，那么在这里需要将该数据包丢弃；如果能够找到，则查看当前数据包中offset与IP表中oldoffset关键字是否一样，如果一样，则将数据包段和缓存中段进行重组，送到ppp\_process\_packets模块进行处理，并删除该条记录；如果不一样，那么表明IP分片包中最有一片比中间包早到了；(是否要建立一个链表将该包缓存下来？)

如果mf=1，那么表示当前数据包时分片包且之后必有数据包要过来，那么开始计算当前数据包的hash\_id并作为key查看IP表中是否有相同hash\_id，如果没有，则看当前数据包offset是否为0，如果为0，将记录该数据包新建IP表中记录，记录关键字hash\_id,oldoffset=data,并缓存当前报文，记录当前缓存部分长度length；如果offset不为0，则表示当前数据包中第一片包还没有到达，或是已经丢失；那么将当前数据包丢弃；如果hash\_id匹配成功，那么将当前数据包和上一片数据包进行merge，并修改oldoffset+=datalength;

2015/12/9

记录：关于使用8字节截断方式来降低软件缓存数据包大小；这个方式存在一定问题；

假如情况是IP分片包只有两片，其中一片包中包含着一片PPP报文，且不完整，剩余payload在下一片IP分片包中，这时我们对第一片数据包进行保留MAC,移除外层IP字段，GRE字段，格式化并反转义7d，提取内层IP信息，并将外层IP计算hash\_id，内层IP.SRC,IP.DST保存起来，将当前数据包内容按照8字节对齐方式进行打包转发，并记录当前oldoffset,newoffset，将剩余字节保存起来；当下一片IP分片包过来的时候，按照外层IP字段信息计算出hash\_id，然后查找IP表，找到相同的hash\_id，并比较当前数据包offset和IP表中记录的oldoffset是否相同，如果相同，那么将当前数据包payload进行移除PPP头字段，PPP尾7d，反转义7d，然后将当前包payload和IP表中payload合并，填充IP表中内层IP字段，并转发；那么这里出现了，如果上一层数据包中包含不止一片PPP报文，一片完整，后一片不完整；且后一片不完整数据包IP段丢失，那么导致我在提取内层IP信息的时候做不到，如果下一片包过来找到了hash\_id，融合了payload，才可以将整个IP字段都补全，可以当上一片数据包没有获取完整IP字段信息时，将IP表中设置标志位，表示:上一片包中我没有拿到完整的IP字段信息，需要下一次数据包过来时，进行从内层提取；目前我还是按照整个数据包缓存。