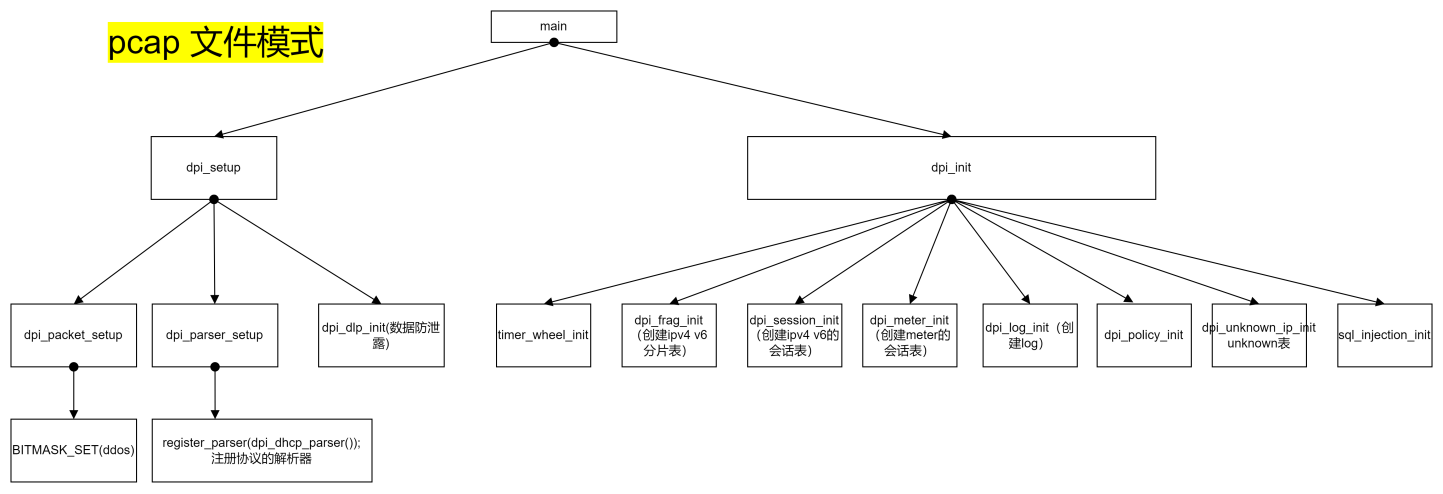
# Neovector dpi 初始化流程

本文介绍以pcap 包解析为例，

-p 选项 参数为 pcap 文件的绝对路径。



# 协议注册

## 2.1 tcp 协议注册

协议注册时：

定义初始化静态全局struct dpi\_parser 结构体，

例如：

static dpi\_parser\_t dpi\_parser\_dhcp = {

new\_session: dhcp\_new\_session,

delete\_data: NULL,

parser: dhcp\_parser,

name: "dhcp",

ip\_proto: IPPROTO\_UDP,

type: DPI\_PARSER\_DHCP,

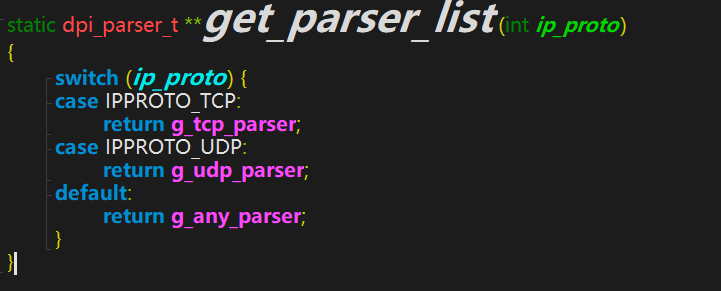
};

dpi\_dhcp\_parser 函数接口返回 注册的全局dpi\_parser 变量。

调用register\_parser接口，

调用get\_parser\_list， 根据协议自定义的dpi\_parse 变量的ip\_proto 类型， 将协议挂到TCP、UDP、其他。

List （g\_tcp\_parser、g\_udp\_parser、g\_any\_parser）是一个 指针结构体数组， 下标为协议的type(DPI\_PARSER\_DHCP), value 指向为 dpi\_parser\_dhcp。

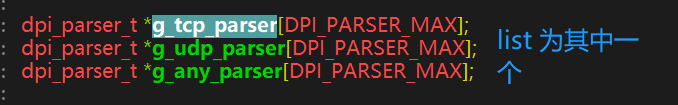


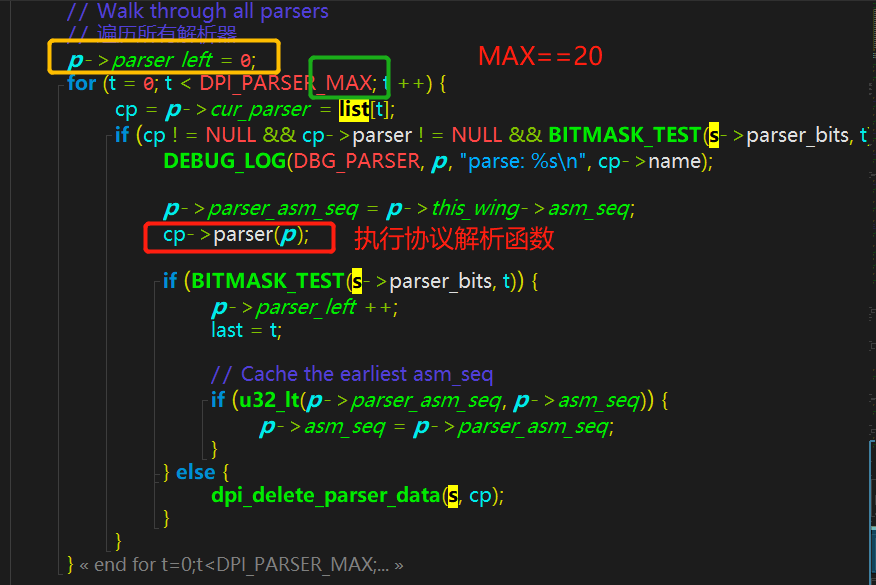
## 2.2 tcp 的应用层协议解析

dpi\_proto\_parser 接口中， 根据解析到的传输层协议，获取dpi\_parser\_t 解析器。

在循环解析的DPI\_PARSER\_MAX 是否可以考虑为，真实的协议支持能力， 避免循环空转。

这种协议解析方式， 是否会影响到引擎的解析能力。

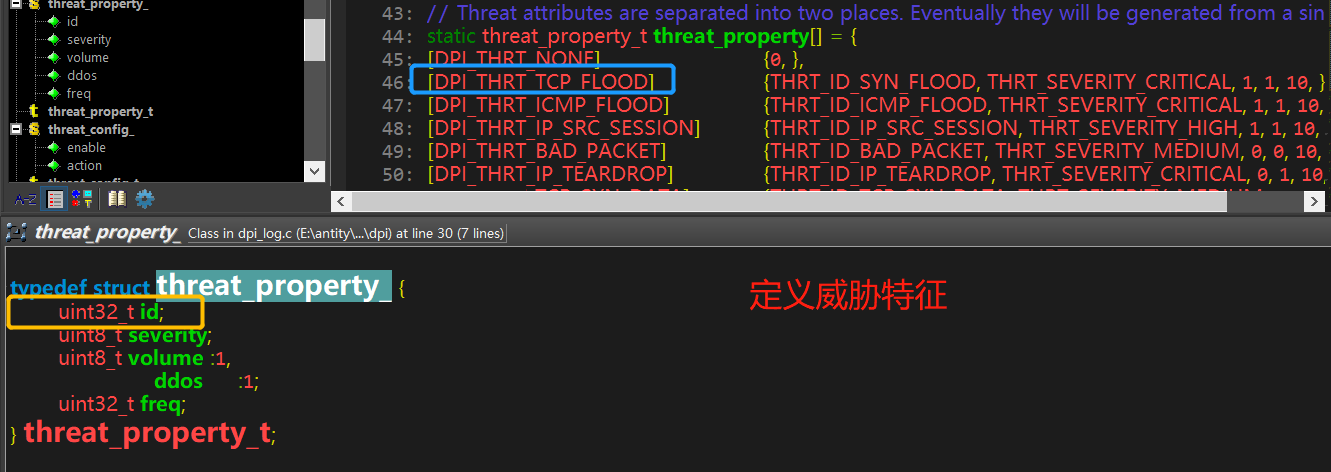




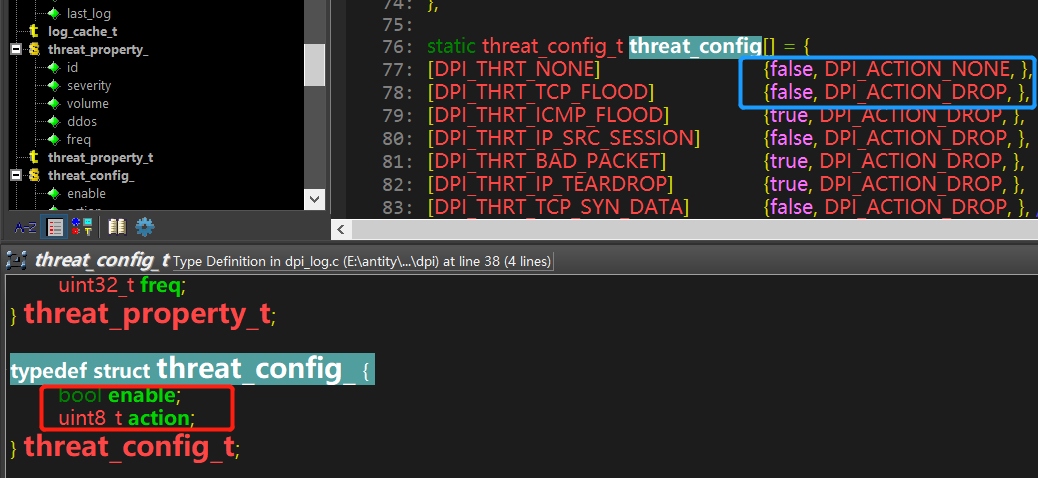
# 威胁检测

## 3.1 syn with data

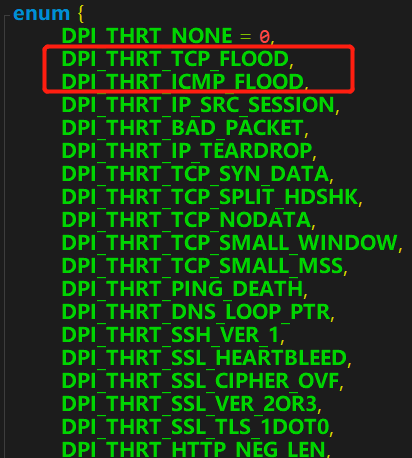
静态全局结构体数组 threat\_property 在代码编译时，已经定义，初始化完成。包括有特征id，验证等级，volume ，是否ddos，freq。



静态全局结构体数组，threat\_config 在代码编译时，已经定义，初始化完成。包括威胁处理方式，

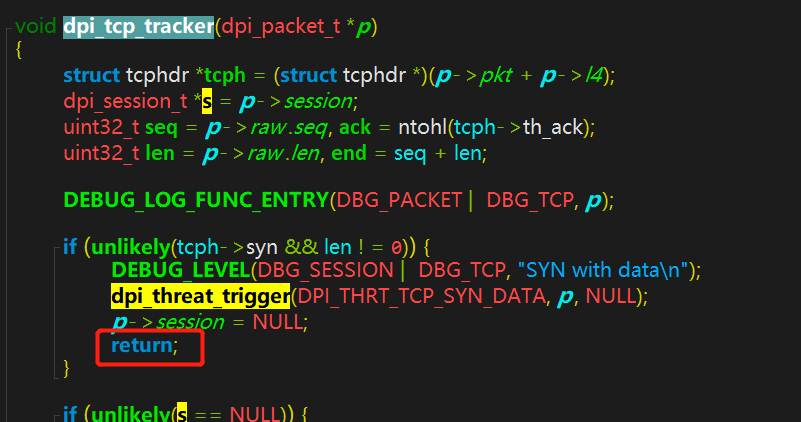


threat\_property 和threat\_config通过DPI\_THRT\_TCP\_SYN\_DATA 的枚举变量形成映射关系。



## 3.2威胁行为处理：

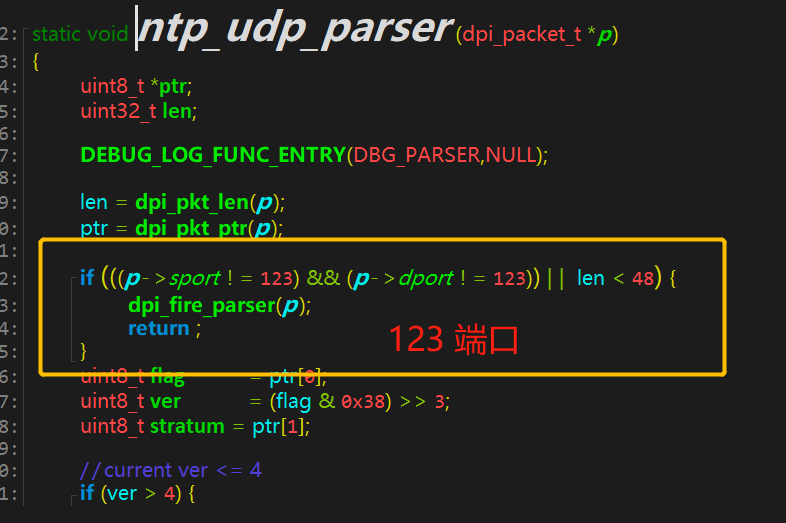
发现威胁行为，直接做丢包、阻断处理， 不走协议解析流程。 在log 中记录mac、五元组、threat\_id、severity。dump 出pcap 文件？？？



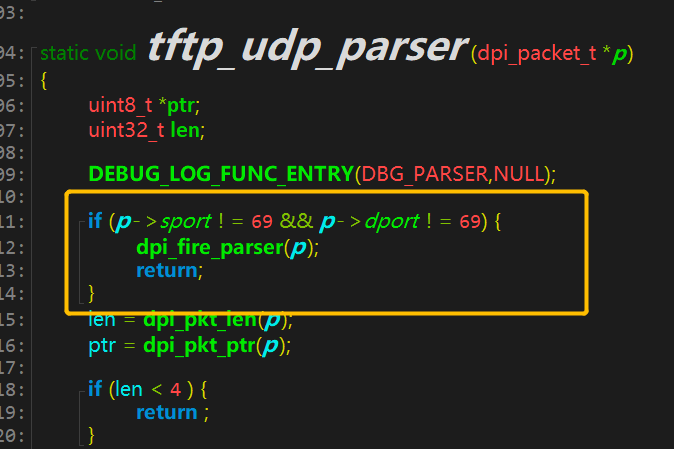
# 协议识别

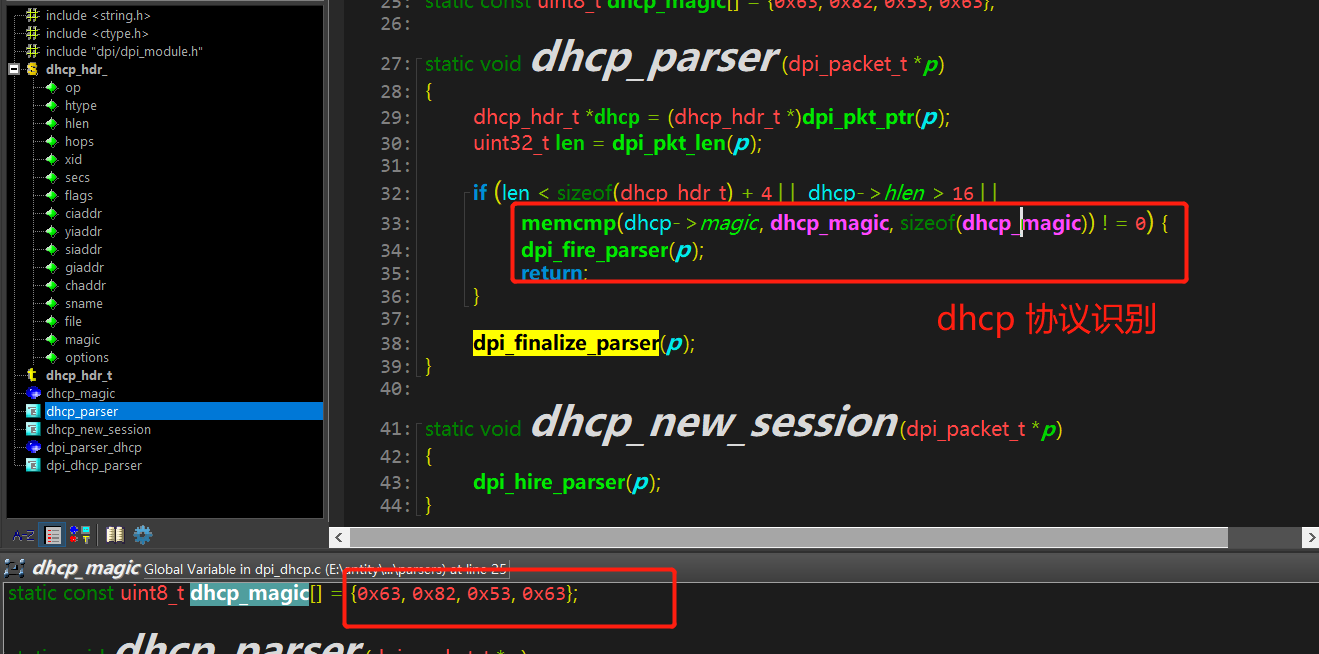
深度协议解析，可解析出协议的具体字段， http 协议，

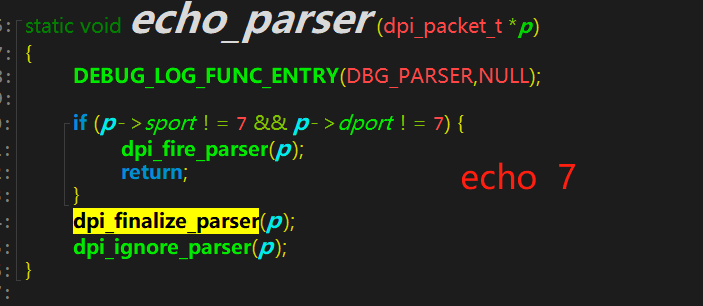
Ntp协议识别处理：



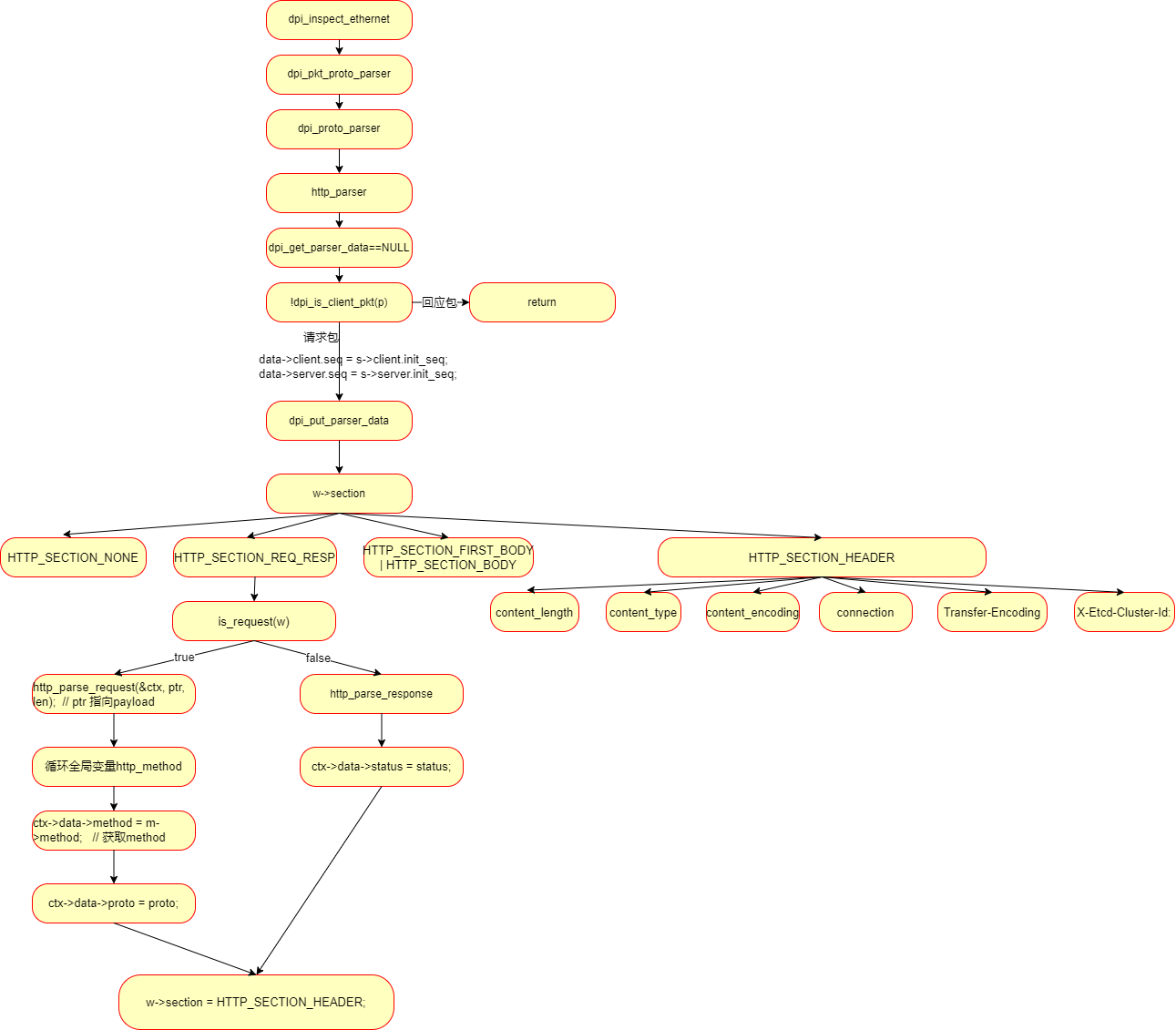
Tftp 协议







## 4.1 http 协议 深度解析流程



## 4.2 协议流程疑惑

1. 协议的解析结果，最终出口？

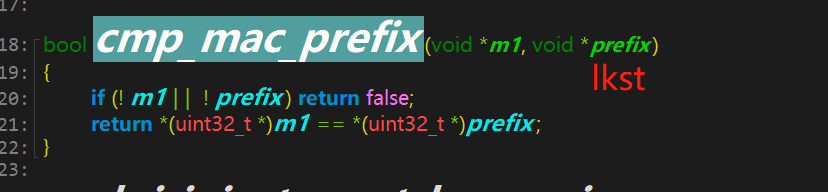
2. 查看http 协议后，发现是单包处理的， 类似 一个会话的所有 请求 回应数据如何关联在一起？

3. 部分协议识别是直接按端口方式， 是否考虑再添加识别规则？

4. 查看 http 等部分协议， 未做协议识别， 直接进如协议解析流程。是否考虑添加部分协议识别规则。

5. 如何根据mac 判断是否走代理？

#define PROXYMESH\_MAC\_PREFIX "lkst"//0x6c6b7374



# standalone 模式下抓包流程

## 5.1 抓包模式

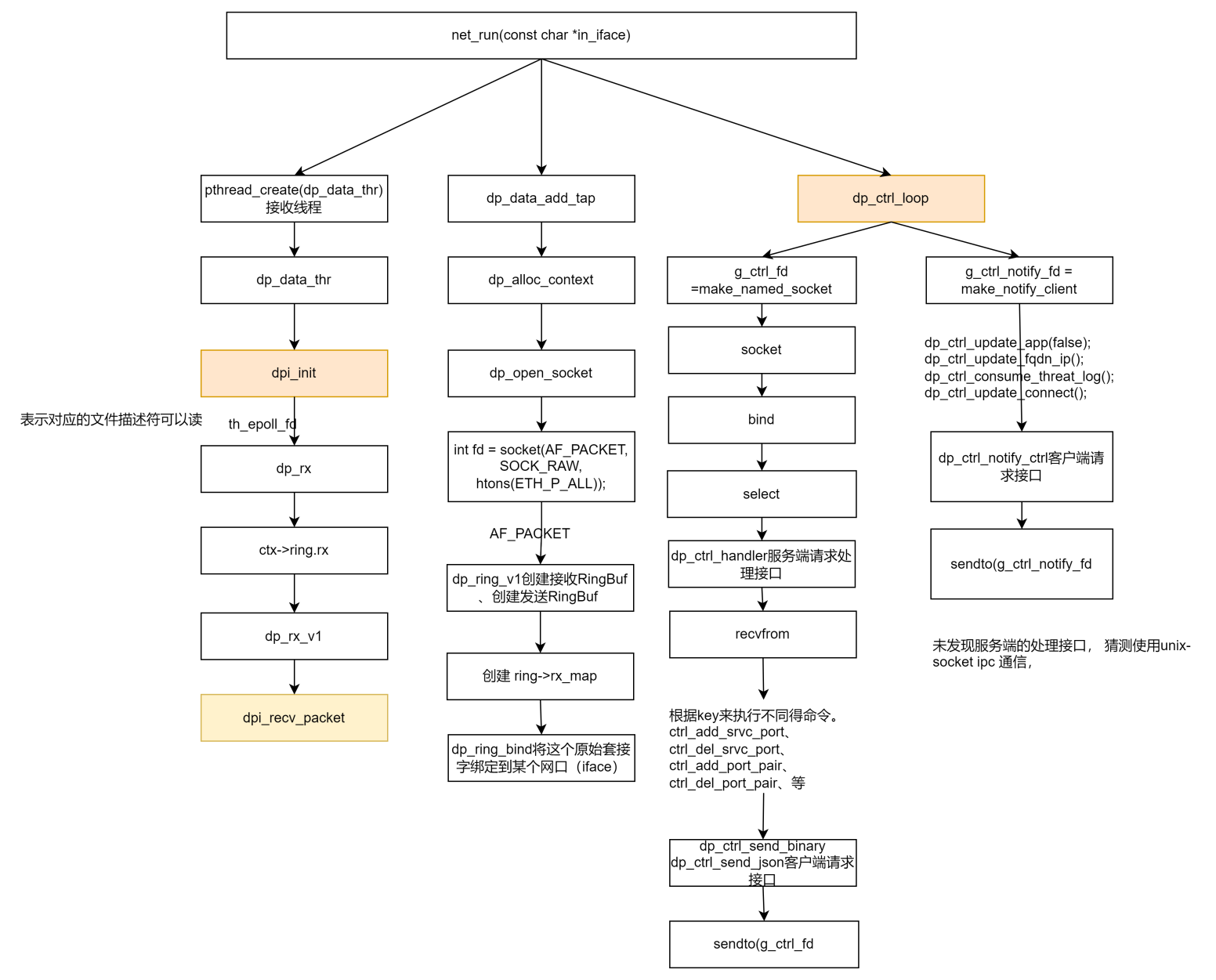
Standalone 模式下使用AF\_PACKET方式进行抓包，linux提供了原始套接字RAW\_SOCKET，可以抓取数据链路层的报文。这样可以对报文进行深入分析。今天介绍一下AF\_PACKET的用法，分为两种方式。

第一种方法是通过套接字，打开指定的网卡，然后使用recvmsg读取，实际过程需要需要将报文从内核区拷贝到用户区。

第二种方法是使用packet\_mmap，使用共享内存方式，在内核空间中分配一块内核缓冲区，然后用户空间程序调用mmap映射到用户空间。将接收到的skb拷贝到那块内核缓冲区中，这样用户空间的程序就可以直接读到捕获的数据包了。PACKET\_MMAP减少了系统调用，

不用recvmsg就可以读取到捕获的报文，相比原始套接字+recvfrom的方式，减少了一次拷贝和一次系统调用。Neuvector 的Standalone 模式 就是采用第二种方式。

libpcap就是采用第二种方式。suricata默认方式也是使用packet mmap抓包。



重要接口梳理：

1. dp\_data\_thr

创建epoll\_create

创建eventfd， epoll\_ctl添加req\_event\_fd

dpi\_init

epoll\_wait

查看是否 是req\_event

True dpi\_handle\_ctrl\_req

False dp\_rx

Dpi\_recv\_packet

1. dp\_data\_add\_tap

创建dp\_context\_t ctx

dp\_alloc\_context

dp\_open\_socket

创建AF\_PACKET得socket

dp\_ring\_v1

创建mmap ringbuff

dp\_ring\_bind 将这个原始套接字绑定到某个网口（iface）

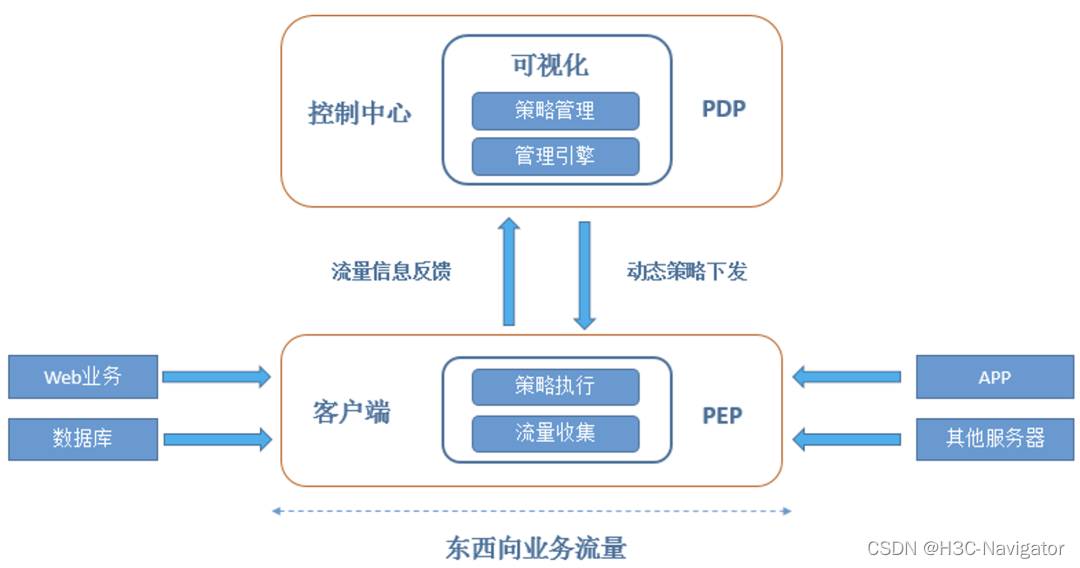
dp\_epoll\_add\_ctx 添加open\_sock 返回得sock 到epool 中。

# 微隔离

## 6.1 微隔离原理

微隔离的实现方式是将数据中心内部所有的业务按照特定的原则划分为数个微小的网络节点，根据动态策略分析对这些节点执行访问控制，在逻辑上将这些节点隔离开，限制用户横向移动，这就是微隔离。在微隔离的架构中，不再存在内、外网的概念，而是将数据中心网络隔离成了很多微小的计算单元，这里我们简称节点。每个节点要访问其他节点的资源，都需要经过微隔离客户端的认证，如果节点身份认证不通过，或不具备访问权限，会被客户端拦截。

节点可以是门户网站，可以是数据库、审计设备，甚至一个文件服务器，只要具备一定的数据处理能力的单元，都可以成为一个节点。他们不再因处于内网而被认为是“可信的”，所有节点都被逻辑隔离，节点之间的访问都是受控的。节点划分越细致，控制中心对整个数据中心网络的流量可视化就越清晰。



## 6.2 微隔离主要元素说明

■ 微隔离客户端：通过代理或虚墙实现，主要包括流量信息收集和策略执行两个部分。向控制中心反馈当前网络中的业务流量信息，实时上报业务动态，接收控制中心下发的策略控制指令，执行安全策略动作，

■ 微隔离控制中心：主要包括管理引擎和策略管理两个控制块。管理引擎接收客户端发送的流量数据，并根据这些信息建立业务模型，交由策略管理模块分析当前网络形势，进行多维度策略运算，动态生成安全策略，并下发给客户端执行，通过流量自学习实现策略自适应。

## 6.3 微隔离系统运行流程

1）确定业务主体：我们首先要明确微隔离系统需要管理的对象是什么，如何划分节点。节点的划分原则主要有两个，一个是要符合数据中心的业务体系，一个是要尽可能降低策略计算的复杂度。例如按当前网络中的业务类型划分、按照业务或部门职权划分，按照重要性优先级划分等；

2）客户端收集业务流量信息，上报管理中心，并实时监测业务流量走向；

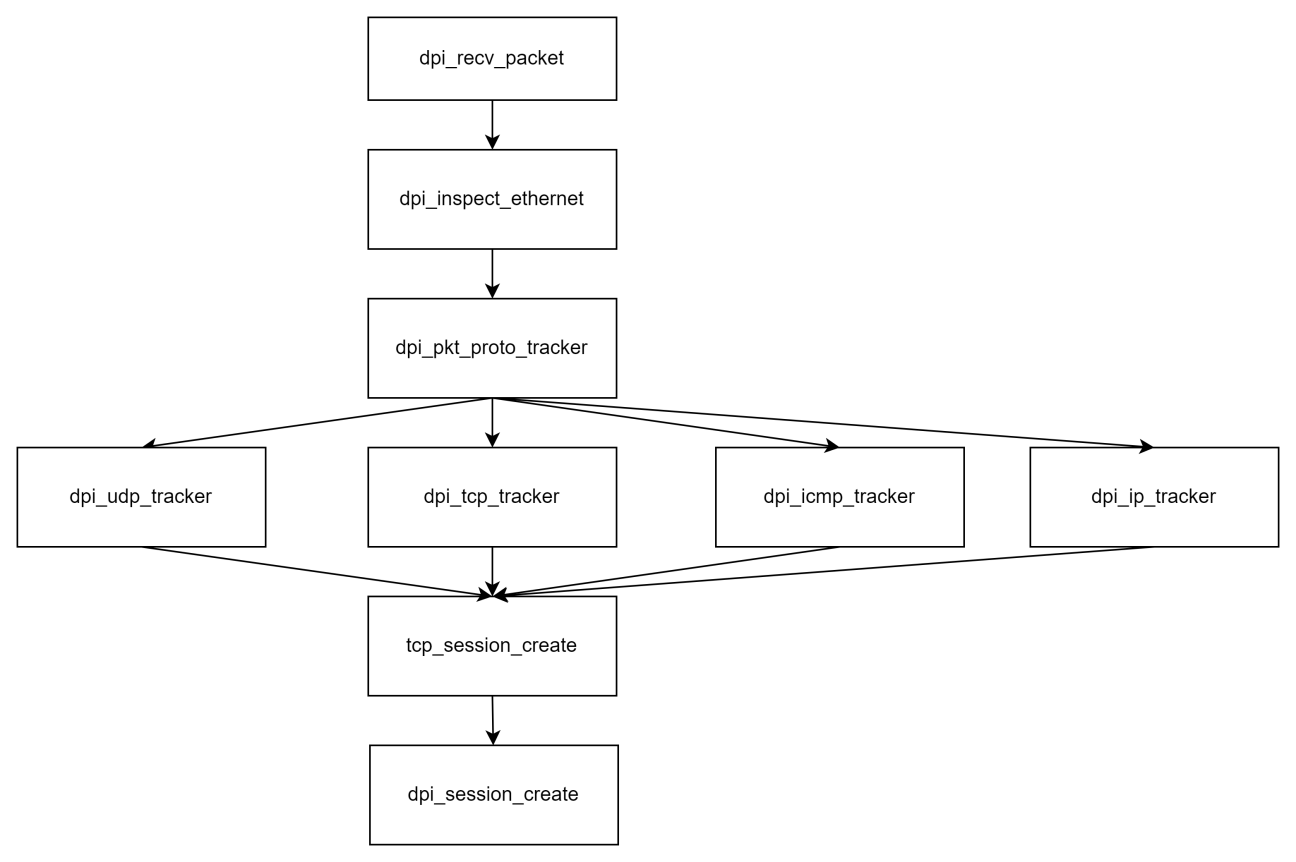
3）管理引擎绘制业务流量拓扑，根据流量数据判断网络结构变化，更新拓扑信息；

4）业务节点发送访问申请，客户端接收到申请后上报管理中心；

5）策略管理模块分析业务流走向，启动策略运算，下发安全策略至微隔离客户端；

6）客户端策略执行，业务认证通过且具有访问权限，流量放行；认证不通过或认证权限不够，执行相应的阻断动作；输出策略匹配日志信息。

# 7. dpi\_session\_create



# Dlp 数据防泄漏

数据泄密（泄露）防护（Data leakage prevention, DLP），又称为“数据丢失防护”(Data Loss prevention, DLP)，有时也称为“信息泄漏防护”(Information leakage prevention, ILP)。数据泄密防护(DLP)是通过一定的技术[手段](https://baike.baidu.com/item/%E6%89%8B%E6%AE%B5/11053605)，防止企业的指定数据或信息资产以违反安全策略规定的形式流出企业的一种策略。DLP这一概念来源于国外，是国际上最主流的信息安全和数据防护手段。

## 7.1 泄漏途径

数据泄漏的途径可归类为三种：在使用状态下的泄密、在存储状态下的泄密和在传输状态下的泄密。一般企业可通过安装防火墙、杀毒软件等方法来阻挡外部的入侵，但是事实上97%的信息泄密事件源于企业内部，所以就以上三种泄密途径分析，信息外泄的根源在于：

1、使用泄漏；1）操作失误导致技术数据泄漏或损坏；2）通过打印、剪切、复制、粘贴、另存为、重命名等操作泄漏数据。

2、存储泄漏：1） 数据中心、服务器、数据库的数据被随意下载、共享泄漏；2）离职人员通过U盘、CD/DVD、[移动硬盘](https://baike.baidu.com/item/%E7%A7%BB%E5%8A%A8%E7%A1%AC%E7%9B%98/407959)随意拷走机密资料；3）移动笔记本被盗、丢失或维修造成数据泄漏。

3、传输泄漏：1）通过email、QQ、MSN等轻易传输机密资料；2）通过[网络监听](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C%E7%9B%91%E5%90%AC)、拦截等方式篡改、伪造传输数据。

## 7.2 neovector dlp 模块

dp\_bld\_dlp\_thr 处理线程