# GSATE SIP 验证功能

能够验证 SIP , htpp 的头域内容。

有 Define， Match， SetTime， Before， Before等功能。

## 1.Define

Define 来定义 待验证的数据包，之后根据给定的信息过滤.

Syntax:

Pcap:Define pcaketName

Example：

Pcap:Define pcaket

定义完了之后可以访问该packet，对packet 指定过滤内容, 可以指定 destination, source, transport, protocol, requestType, responseType,header,sdp。

destination: ipv4 目标地址，ip层

source： ipv4 来源地址，ip层

transport： Tcp 或者是 Udp 协议

protocol： sip 或者 http

requestType： 请求包的 请求类型，如sip 的INVITE，OPTION，http的POST，GET

responseType： 响应包的 状态码 ， 如200 ok

header: sip 或 http中的头域内容。

sdp: sip 中的sdp 部分，如果有的话。

不给定值的属性代表不在该属性上进行过滤。 规则是 ip，protocol 这些给定上才能匹配到， 而header/sdp 头域只要 含有这个头域，并且内容是实际数据包的一部分（子串）就算匹配成功。

header 和 sdp 点出来 “.”属性 意味着， 你希望找到一个数据含有这个头域。

Syntax:

pcapName.attr = content

Example：

packet.destination = “192.168.92.200”

packet.protocol = “sip”

packet.header.X-Grandstream-PBX = "true"

此packet 在后续使用时， 会匹配目标ip是92.200的数据包， 协议是sip，带有“X-Grandstream-PBX”头域，内容含有 “true”字段。 其他内容例如其他头域，是udp 还是tcp，没有指定不会去限制。

## 2. Match

Match 功能 来对定义的好的并赋值的 数据包 进行匹配， 找到第一个数据包就算成功， 如果需要精确定位， 那么可以给定较详细的过滤信息。

可以添加 一个变量 存放匹配到的数据包， 后续可以访问。 如果没有添加这个变量， 匹配失败，测试就算失败case就会终止。 添加后无论匹配结果如何 case 都将继续运行。（后续有需要可以通过Console：Stop 来结束case） 可以通过访问 变量 empty 属性判断是否匹配成功， 匹配失败 empty 为“true”。 如果为”false” 则匹配成功， 并能通过该变量访问数据包的内容， 在case 中后续使用。

匹配时 是在所有抓到SIP http 包中进行 查找 （已经按照当前 case 所用到的device 的mac 来过滤）， 所以 建议 在测试步骤做完 来进行 匹配检测。

如果需要实时等待目标数据包， 可以通过SetTime 来设置timeout 来边等边检测。

Syntax：

Pcap:Match pcapName [Var]

Example：

Var temp

Pcap:Match packet temp

Pcap:Match Packet

## Behind & Before

Match 功能的强化版， 定义多个 数据包， 匹配 目标数据包在 其他包出现之前或者出现之后。

没有找到目标包会失败， Before 在找到目标包之前先找到了其他定义包会失败，Behind 在找到目标包之后 再找到其他包会失败， 即： Before 需要目标包在 其他包前面， Behind需要目标包在其他包后面。

同样的，最后加个变量做参数，会把匹配的数据包存入变量，并失败后继续运行脚本。

如需实时等待匹配， 可以对目标 数据包SetTime。

Syntax：

Pcap:Behind pcapName <pcapDat Vec> [var]

Example:

Pcap:Define p1

Pcap:Define p2

Pcap:Define p3

Pcap:Define p4

Vec pcapList = {p2,p3,p4}

Var temp

Pcap:Behind p1 pcapList temp

Pcap:Before p1 pcapList

## SetTime

该功能 设置目标数据包的 timeout， 没有设置的timeout 的 数据包 进行匹配时 会在当前所有包中 查找一次 （Match，Before，Behind）。 设置timeout了的数据包， 会查找当前已经抓到过了的数据包， 并且实时等待目标数据包（按照对应function的规则）， 直到满足条件（找到）， 或者超时。

单位： 秒

Syntax：

Pcap:SetTime pcapName num

Example:

Pcap:Define packet

Pcap:SetTime packet 10

packet.portocol = “http”

packet.requestType = “GET”

packet.destination = “192.168.92.72”

Pcap:Match packet

此处一直 会匹配，查找来自92.72 的 http Get 请求， 最多10秒。