

问题描述

负载均衡设备NQA探测模板http探测是否支持识别返回值为中文?

expect { data | hex-data }命令用来配置期望测试成功的应答内容。

undo expect { data | hex-data }命令用来恢复缺省情况。

【命令】

expect { data | hex-data } string [{ offset | strict-offset } number]

undo expect { data | hex-data }

【缺省情况】

未配置期望测试成功的应答内容。

【视图】

HTTP/HTTP2/HTTPS/TCP/UDP/WAP类型的NQA模板视图

【缺省用户角色】

network-admin

mdc-admin

vsys-admin

【参数】

data: 表示使用文本格式的字符串来填充探测报文。

hex-data: 表示使用十六进制字符串来填充探测报文。该参数仅在TCP/UDP/WAP类型的NQA模板视图下支持。

string: 期望测试成功的应答内容。和data关键字配合使用时, 为1~200个字符的字符串, 区分大小写; 和hex-data关键字配合使用时, 为2~200个字符的字符串, 字符串长度必须是偶数且不区分大小写。

offset number: 所期望的内容在返回报文中的偏移量, 取值范围为0~1000, 缺省值为0。

strict-offset number: 所期望的内容在返回报文中的严格偏移量, 取值范围为0~1000, 缺省值为0。

【使用指导】

NQA客户端对于返回报文中期望成功的应答内容匹配方式如下:

- 如果未配置offset参数和strict-offset参数, 则设备直接从返回报文对应字段的第一个字节开始匹配, 若不匹配, 继续从第二个字节开始匹配, 以此类推, 直到匹配成功, 则认为测试成功, 否则认为测试失败。

- 如果配置了offset参数, 则设备从返回报文对应字段偏移量之后的第一个字节开始匹配, 若匹配失败, 则忽略该偏移量, 从返回报文的第一个字节开始匹配, 若不匹配, 继续从第二个字节开始匹配, 以此类推, 直到匹配成功, 则认为测试成功, 否则认为测试失败。

· 如果配置了strict-offset参数，则设备从返回报文对应字段偏移量之后的第一个字节开始匹配，若匹配失败，继续从第二个字节开始匹配，以此类推，直到匹配成功，则认为测试成功，否则认为测试失败。

对于HTTP/HTTP2/HTTPS类型的NQA模板，仅当回应报文的BODY类型为CONTENT或CHUNK时，进行期望应答内容的检查，否则不做检查。

对于TCP类型的NQA模板，仅当data-fill和expect { data | hex-data }命令都配置时，进行期望应答内容的检查，否则不做检查。

对于UDP类型的NQA模板：

· 仅当data-fill和expect { data | hex-data }命令都配置时，进行期望应答内容的检查，否则不做检查。

· 偏移检查和data-fill是否指定raw关键字命令有关，当指定raw关键字时，从收到报文的第一个字节开始检查；当未指定raw关键字时，UDP报文数据字段的前五个字节具有特定用途，从第六个字节开始进行偏移检查。

对于WAP类型的NQA模板，从WSP reply报文的headers字段开始进行期望应答内容的检查。当配置url命令时，WAP测试需要通过配置本命令进行期望应答内容的检查来判断测试是否成功。

测试过程中若未配置期望应答状态码，且收到的NQA应答报文中的内容匹配上本命令配置的期望应答内容，则判断本次测试为成功测试；若配置了期望应答状态码，则只有在匹配上期望的、测试成功应答状态码或者未匹配上期望的、测试失败应答状态码之后，才以本命令的匹配结果判断本次测试的成功与否。

对于HTTP/HTTP2/HTTPS类型的NQA模板，本命令不能与expect { failed-data | hex-failed-data }命令同时配置。

对于TCP/UDP类型的NQA模板，当同时配置了本命令和expect { failed-data | hex-failed-data }时，如果两条命令同时匹配上，则以expect { failed-data | hex-failed-data }的匹配结果为准，认为测试失败。

【举例】

在HTTP类型的NQA模板视图下配置期望的应答内容为welcome!。

```
<Sysname> system-view
```

```
[Sysname] nqa template http httpplt
```

```
[Sysname-nqatplt-http-httpplt] expect data welcome!
```

解决方法

支持，但是需要保持和服务器侧的编码保持一致。

一般服务器都是utf-8编码，需要在设备后台使用utf-8编码配置中文。切忌Web配置。