SLOWLORIS抓包解析

`Slowloris Attack`又被称为`Headers Attack`。

- **攻击方式：** 以低速向服务器发送`HTTP GET`请求，长时间保持本该释放的连接不释放。耗尽服务器资源
- **原理：** `HTTP`报文结构，`Header + CRLF + Data（可以没有）`。如果不发送`CRLF`则服务器认为`Header`未接收完毕，继续等待接收，直到超时。`Header`由多行构成，每行以`CRLF`结尾（即以`\r\n`结尾）。
- **验证：** 正常HTTP请求`Header`结尾为`0D0A`。`Slowloris`攻击时`Header`结尾则没有`0D0A`。（并非行尾的`0D0A`，而是指用来间隔`Header`和`Data`的`CRLF`。）

> - `\r`为回车符，`Carriage Return`，缩写为`CR`。`ASCII`十六进制为`0D`。意为将光标调整为本行行首。
> - `\n`为换行符，`Line Feed`，缩写为`LF`。`ASCII`十六进制为`0A`。意为将光标调整为下一行（不一定是行首）。

一、监听网卡

此次以网页`khdxs7.test`为例。抓取我们和服务器`80`端口的之间的通信数据。

设置`wireshark`抓包过滤器规则：

```
host khdxs7.test and port 80
```

二、正常HTTP GET 请求

按`Ctrl + F5`强制刷新网页。

- 查看`HTTP`请求包。每行以`\r\n`结尾，在`Header`结尾存在起间隔作用的`0D0A`。

三、Slowloris Attack 时的 HTTP GET 请求

- 重新开始捕获数据包。
- 使用`slowloris`发送一个`HTTP`请求。

```
slowloris -v khdxs7.test -p 80 -s 1 --sleeptime 10000
```

- 数据包4为`Header`的第一个包，数据包6为`Header`的第二个包。两个包合并在一起才是一个完整的`HTTP`请求，因此均不能被识别为`HTTP`协议。查看`Header`信息，在`Header`结尾不存在起间隔作用的`0D0A`。
- 验证了正常`HTTP`请求与`Slowloris`的区别就在于`Header`是否正常结束。