第3章 Wireshark使用


视频

Wireshark是一款免费网络封包分析软件，其前身是Ethereal。在Ethereal出现之前，对网络数据报文进行分析需要购买昂贵的分析软件，Ethereal让使用者方便快捷地分析数据报文成为可能。网络封包分析软件的功能是捕获网络通信报文，并尽可能显示出详细的网络报文资料。Wireshark使用WinPCAP作为接口，直接与网卡交换数据报文。

Wireshark提供了Windows和macOS两个编译版本，在Linux平台上可以使用tcpdump实现数据报文捕获功能，并把捕获的报文用Wireshark打开，即可进行分析。

Wireshark可以在Linux/UNIX上用源码编译使用，但过程比较复杂。

3.1 BPF

使用伯克利包过滤器(Berkeley Packet Filter，BPF)过滤规则可以确定获取和检查哪些流量，以及忽略哪些流量。

1. BPF表达式

BPF表达式由一个或者多个原语(primitive)组成：

```
[not] primitive [and|or [not] primitive …]
```

2. 原语

原语由标识符(identifiers)，以及描述它的多个限定词(qualifiers)组成：

```
qualifiers Identifiers
```

例如，host www.baidu.com，这里host是限定词，www.baidu.com是标识符。

3. 标识符

标识符即名字或者数值，如www.baidu.com，80。

4. 限定词

(1) Type：设置名称或者数字所指示类型。

Type的可选值如下。

① host：IP或者域名地址。

② net：设定子网。例如，net 192.168.0.0 mask 255.255.255.0等价于net 192.168.0.0/24。

③ port：端口。

④ portrange：设置端口范围。例如，portrange 6000-8000。

(2) Dir：设置网络出入方向。Dir 的取值有 src、dst、src or dst 和 src and dst。例如，dst port 80。

(3) Proto：指定具体的协议。Proto 的可选值有 ether、fddi、tr、wlan、ip、ip6、arp、rarp、decnet、tcp、udp。

(4) 其他。

① gateway：指明网关 IP 地址，等价于 ether host ehost and not host host。

② 广播(broadcast)与多播(multicast)：如 ether broadcast 或者 ip6 multicast。

5. 运算符

运算符用于组合多个原语。其中，“非”优先级最高，“与”和“或”优先级相同，优先级的顺序为从左到右。

(1) 非：！ 或者 not。

(2) 与：&& 或者 and。

(3) 或：||或者 or。

6. 筛选

BPF 可以使用逻辑运算，格式如下：

```
expr relop expr
```

以上内容表示对数据报文内容进行过滤。其中，expr 为表达式；relop 为关系操作符，包括>、<、>=、<=、=、!=。

可以使用中括号实现截取，格式如下：

```
proto [expr : size]
```

例如，ip6[12:4]==ip6[16:4]表示捕获 IPv6 报文源地址和目的地址一致的报文。

以下是常用过滤举例：

(1) udp dst port not 53：捕获 UDP 并且目的端口不是 53 的报文。

(2) host 10.0.0.1 && host 10.0.0.2：捕获 10.0.0.1～10.0.0.2 的报文。

(3) tcp dst port 80 or 8080：捕获 TCP 中目的端口是 80 或 8080 的报文。

3.2 安装

从 Wireshark 官网(https://www.wireshark.org)找到软件下载链接，下载后直接打开，即可安装 Wireshark 软件。

3.3 捕获

打开 Wireshark 后，首先需要选择要捕获的网卡，该界面根据计算机的网卡配置会有区别，如图 3-1 所示。

在该界面中，在 Capture filter for selected interfaces 文本框中可输入捕获过滤器。若在其中输入 BPF 过滤规则表达式，则可实现针对选定网口的数据报文过滤功能。

图 3-1　Wireshark 选择网卡界面

3.4　过滤器

网络中的数据通信非常频繁，如系统升级、病毒库升级、新消息检测、邮件检测等，上层应用不断进行通信。如果要捕获全部的数据再进行分析，一方面会增加分析难度，另一方面还需要整理所有的通信数据，因此需要对报文进行过滤，仅捕获人们需要捕获的内容。BPF完成了在网络接口层数据过滤的功能，Wireshark 还提供了过滤器，用来在已经捕获的数据中过滤想要分析的内容，从而更加有效地对报文进行分析。

并不是所有的通信过程都是完整、正确的，有一部分是异常、错误的。由于不同平台、系统、软件或接口对协议的实现并不是绝对完整的，因此在实际通信过程中会出现部分协议不支持的情况。此外，由于通信故障，如断线、误码、干扰等问题，以及断电、人为阻断、数据拥塞等因素，也会造成数据错误或中断。过滤器可以帮助人们过滤错误报文，通过正确的报文掌握规则，再逐步分析错误报文，掌握处理方法。

Wireshark 过滤器的规则与 BPF 类似，允许将报文中协议的字段与指定值进行比较，仅输出满足条件的报文。过滤器除了用于过滤已经捕获报文的显示，还可用于信息统计及 Wireshark 主界面中的报文着色，如图 3-2 所示。

1. 检查某协议或字段

通过协议名或字段，过滤仅包含该协议或字段的报文。例如，http 显示 HTTP 报文记录，tcp 显示 TCP 连接报文记录。

2. 比较运算

Wireshark 支持的比较运算符如表 3-1 所示。例如，ip. addr＝＝127. 0. 0. 1，表示显示 IP 地址为 127. 0. 0. 1 的报文记录。

图 3-2 Wireshark 主界面

表 3-1 比较运算符

关 键 词	运 算 符	含 义
eq	==	Equal,等于
ne	!=	Not Equal,不等于
gt	>	Greater Than,大于
lt	<	Less Than,小于
ge	>=	Greater than or Equal to,大于或等于
le	<=	Less than or Equal to,小于或等于

3. 比较范围

使用 any 和 all 关键词限定比较范围。例如,all tcp. port > 1024,表示显示所有 TCP 端口都大于 1024 的报文记录。Wireshark 支持的其他比较运算符如表 3-2 所示。

表 3-2 其他比较运算符

关 键 词	运 算 符	含 义
eq,any_eq	==	任何字段等于
ne,all_ne	!=	所有字段都不等于
all_eq	===	所有字段都等于
any_ne	!==	任何字段都不等于

4. 查找和匹配

Wireshark 支持的查找和匹配关键词如表 3-3 所示。例如,http contains "https://www.imau.edu.cn",表示显示 HTTP 报文中包含地址 https://www.imau.edu.cn 的报文记录。

表 3-3 查找和匹配关键词

关 键 词	运 算 符	含 义
contains		协议、字段或切片是否包含
matches	~	是否匹配 perl 的正则表达式

5. 函数

Wireshark 过滤器支持的函数如表 3-4 所示。例如，upper(http. server) contains "APACHE"，表示显示 HTTP 报文中转换为大写的 server 字段内容中包含 APACHE 的报文记录。

表 3-4 函数

函 数	说 明	函 数	说 明
upper	转换字符串为大写	string	转换为字符串
lower	转换字符串为小写	max	返回一组数据中的最大值
len	字符串长度/数据字节数	min	返回一组数据中的最小值
count	返回报文中字段的出现次数	abs	取绝对值

6. 整型值

Wireshark 中的整型值可以表示为十进制数、八进制数、十六进制数、二进制数和 C 语言风格的字符串。例如，表 3-5 所示表达式的值是相等的。

表 3-5 Wireshark 整型值 10 的表示

	十 进 制	八 进 制	十六进制	二 进 制
不同进制	10	012	0xa	0b1010
C 语言风格字符	'\n'	'\x0a'	'\012'	

7. 布尔型值

布尔型值的真值表示为 True 或 TRUE，或者任何非 0 值；假值表示为 False 或 FALSE，或者 0 值。

8. 字符串

Wireshark 使用双引号表示字符串，使用“\”作为转义字符。Wireshark 的转义字符如表 3-6 所示。

表 3-6 Wireshark 的转义字符

转义字符	含 义	转义字符	含 义
\'	单引号	\t	水平制表符
\"	双引号	\v	垂直制表符
\\	反斜杠	\NNN	任意八进制值
\b	退格	\xNN	任意十六进制值
\f	换页	\uNNNN	Unicode 16 码
\n	换行	\uNNNNNNNN	Unicode 32 码
\r	回车	—	—

9. 切片

Wireshark 支持对字段进行切片，通过切片可进行部分数据的比较运算。

切片规则如下。

(1) [开始偏移:长度]：从开始偏移开始，长度为指定长度的内容。

(2) [开始偏移-结束偏移]：从开始偏移到结束偏移的所有内容。

(3) [偏移]：该字段偏移量为指定值的字节。

(4) [:偏移量]：从 0 到偏移量的全部内容。

(5)[偏移量:]:从偏移量到末尾的全部内容。

(6)偏移量可以为负值,表示从反向计算。

例如,eth.src[0:3]==00:00:83,表示显示 MAC 地址的前 3 字节内容为"00:00:83"的报文记录。

切片还可以通过逗号分隔,由多个偏移范围组成。例如,ftp[1,3-5,9:]==01:03:04:05:09:0a:0b。

10. 列表取值

可以通过"in"和大括号实现判断字段值是否在其中的检查。列表可以使用".."进行范围取值。

例如,tcp.port in {80,443,8080}等价于 tcp.port==80 or tcp.port==443 or tcp.port==8080,表示 TCP 端口在 80、443 和 8080 中一个的报文记录。

再如,tcp.port in {443,4430..4434}表示 TCP 端口在 443、4430、4431、4432、4433、4434 中的一个。

11. 位操作

使用"bitwise_and"或"&"进行位与操作。例如,tcp.flags & 0x02 可以检查 tcp.flags 字段的第 2 位是否为 1,即 TCP SYN 报文。

12. 算术运算符

Wireshark 支持的算术运算符如表 3-7 所示。如果在算术运算中分组,则使用"{}"。

表 3-7 算术运算符

算术运算符	含 义
+	加
*	乘
%	取余
-	减
/	除

13. 报文引用

在 Wireshark 主界面中双击报文记录,打开报文详情界面,如图 3-3 所示。在该界面中可以单独查看该报文的详细情况,使用"${协议/字段}"的形式引用该报文,从而以该报文的内容作为条件进行过滤。例如,ip.src==${ip.src},表示选择与当前报文的源地址相同的报文记录。

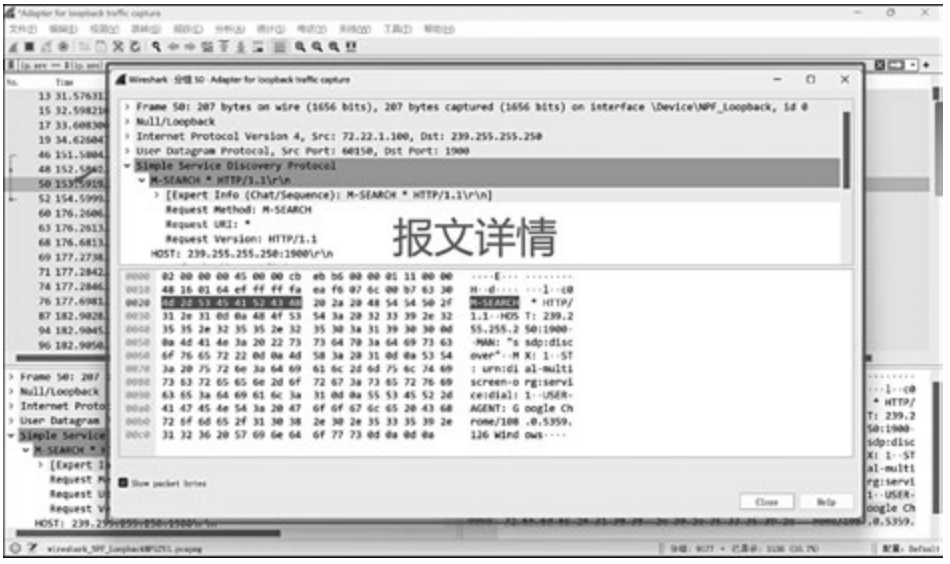

图 3-3 Wireshark 报文详情界面

3.5 HTTP 报文捕获示例

表 3-8 所示是一段 Wireshark 捕获的 HTTP/1.1 请求报文的前 160 字节。

表 3-8 Wireshark 捕获的 HTTP/1.1 请求报文的前 160 字节

索　引	数　据	ASCII
0000	02 00 00 00 45 00 02 ee a2 68 40 00 80 06 00 00	....E....h@.....
0010	7f 00 00 01 7f 00 00 01 e3 5e 00 50 c1 5c df 35	.........^.P.\.5
0020	3c 71 a0 08 50 18 04 ff 84 53 00 00 47 45 54 20	<q..P....S..GET
0030	2f 68 65 6c 6c 6f 77 6f 72 6c 64 2e 68 74 6d 6c	/helloworld.html
0040	20 48 54 54 50 2f 31 2e 31 0d 0a 48 6f 73 74 3a	HTTP/1.1..Host:
0050	20 31 32 37 2e 30 2e 30 2e 31 0d 0a 43 6f 6e 6e	127.0.0.1..Conn
0060	65 63 74 69 6f 6e 3a 20 6b 65 65 70 2d 61 6c 69	ection: keep-ali
0070	76 65 0d 0a 50 72 61 67 6d 61 3a 20 6e 6f 2d 63	ve..Pragma: no-c
0080	61 63 68 65 0d 0a 43 61 63 68 65 2d 43 6f 6e 74	ache..Cache-Cont
0090	72 6f 6c 3a 20 6e 6f 2d 63 61 63 68 65 0d 0a 73	rol: no-cache..s

该报文是通过 Loopback(直接访问本机)方式请求的,“02 00 00 00”表示本机的 IPv4 MAC 地址。

注意:以下字节从 1 开始排序,而表 3-8 中从 0 开始排序。

1. IP 部分

第 5 字节:45 用二进制表示为 0b01000101。使用 IPv4,IP 报文头长度为 20 字节(5×4)。

第 6 字节:00,即 0b00000000。服务类型为默认。

第 7、8 字节:02 ee,即 0b00000010 0b11101110。总长度=2×256+238=750。

第 9、10 字节:a2 68,IP 报文 ID 为 41576。

第 11、12 字节:40 00,即 0b01000000 0b00000000。不允许分片,片偏移量为 0。

第 13 字节:80,即 0b10000000。TTL=128。

第 14 字节:06。协议号为 6,即 TCP。

第 15、16 字节:00 00。IP 头部校验和,环回访问无效。

第 17～20 字节:7f 00 00 01,即源地址 127.0.0.1。

第 21～24 字节:7f 00 00 01,即目的地址 127.0.0.1。

2. TCP 部分

第 25、26 字节:e3 5e,即 58206,源端口号。

第 27、28 字节:00 50,即 80,目的端口号。

第 29～32 字节:c1 5c df 35,即序列号 3244089141。

第 33～36 字节:3c 71 a0 08,即确认号 1014079496。

第 37、38 字节:50 18,即 0b01010000 0b00011000,TCP 头部长度为 20 字节(5×4),确认号有效,PSH 有效。

第 39、40 字节:04 ff,TCP 窗口大小为 1279。

第 41、42 字节:84 53,TCP 校验和。

第 43、44 字节:00 00,紧急指针为 0。

3. HTTP 部分

从 HTTP 部分开始，其后全部为 ASCII 码。

第 45～75 字节：47 45 54 20 2f 68 65 6c 6c 6f 77 6f 72 6c 64 2e 68 74 6d 6c 20 48 54 54 50 2f 31 2e 31 0d 0a。将以上十六进制数值转换为 ASCII 码字符为

```
GET /helloworld.html HTTP/1.1\r\n
```

0d 0a 即\r\n，也就是回车换行。

第 76～92 字节：48 6f 73 74 3a 20 31 32 37 2e 30 2e 30 2e 31 0d 0a，即 Host：127.0.0.1\r\n。

第 93～115 字节：43 6f 6e 6e 65 63 74 69 6f 6e 3a 20 6b 65 65 70 2d 61 6c 69 76 65 0d 0a，即 Connection：keep-alive\r\n。

后面的情况与前面类似，这里不再赘述。将全部报文进行分析后，即可得到在 HTTP/1.1 报文捕获中的请求报文。

注意：在实验中，IP 和 TCP 报文中有部分内容，如 ID、源端口号等变化内容可能与示例不同。

类似地，可以对 Wireshark 捕获的回复报文用同样的方法进行分析，从而从原始报文中得到这次 HTTP 请求的回复报文全文。

习题 3

1. BPF 的完整名称是什么？
2. BPF 表达式由什么组成？
3. BPF 的原语由标识符如何组成？
4. BPF 的限定词有哪些？
5. BPF 中 Dir 限定词有哪些取值？
6. BPF“tcp dst port not 21”的含义是什么？
7. BPF“tcp port 80 and 443”的含义是什么？
8. Wireshark 过滤器“http”的含义是什么？
9. Wireshark 过滤器“http contains "https://www.imau.edu.cn""的含义是什么？
10. Wireshark 过滤器“ip.addr!＝127.0.0.1”的含义是什么？
11. Wireshark 过滤器“any tcp.port ne 80”的含义是什么？
12. Wireshark 过滤器“udp.port in {20,4430..4434}”的含义是什么？
13. Wireshark 过滤器“http contains "login""的含义是什么？
14. 试写出 TCP 端口范围在 8000～9000 的 Wireshark 过滤器表达式。
15. 如何使用 Wireshark 捕获指定网口？
16. BPF 运算符中的“&&”和“and”是否等价？
17. 试使用 Wireshark 捕获一段 HTTP 报文。
18. 使用 BPF 和使用 Wireshark 过滤器有什么区别？
19. 为什么 BPF“ip6[12:4]＝＝ip6[16:4]”可以捕获 IPv6 报文源地址和目的地址一致的报文？

20. Wireshark 过滤器“tcp. port!=0b01010000”的含义是什么?

21. Wireshark 过滤器“http[0:20]==\"GET /helloworld. html\"”的含义是什么?

22. 试写出访问“/index. html”页面的 Wireshark 过滤器表达式。

23. Wireshark 过滤器“ip[9] bitwise_and 6”的含义是什么?

24. Wireshark 过滤器“tcp. port == {0b1001111%80+1}”的含义是什么?

25. Wireshark 过滤器“tcp. port!= ${ip. proto} * 10/2”的含义是什么?

26. 为什么 Wireshark 不能直接捕获 HTTPS 的报文?

27. 试写出捕获端口为 1883 的 BPF 表达式。

28. 试写出 Wireshark 过滤器只显示 HTTP 报文的表达式。

29. 试写出访问“http://localhost/index. html”页面的 Wireshark 过滤器表达式。

30. 试使用 Apache 搭建本地 Web 服务器,端口为 80,并建立“helloworld. html”文件,文件内容如下:

```
<! DOCTYPE html >
< html lang = "en">
< head >
  < meta charset = "utf - 8">
  < title > A simple webpage </title >
</head >
< body >
  < div >
    < h1 > Simple HTML webpage </h1 >
    < p > Hello, world! </p >
  </div >
</body >
</html >
```

31. 先使用 Wireshark 捕获本地 80 端口,再使用浏览器访问本机 Web 服务器的“helloworld. html”文件,对捕获到的数据逐条进行说明,对 HTTP 报文中的数据逐条进行说明。