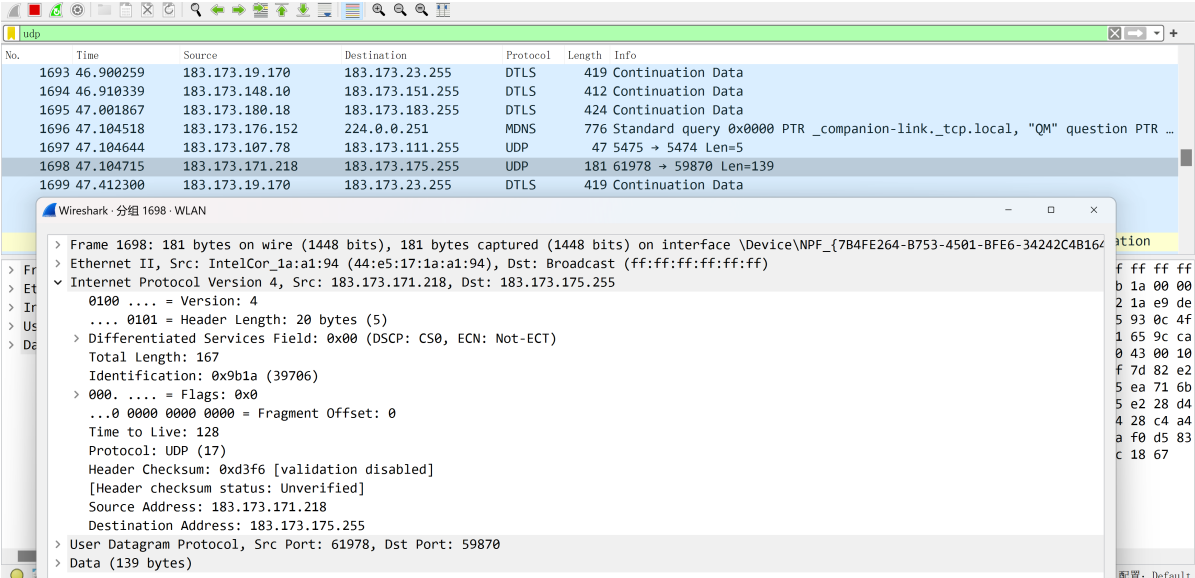


# 小实验4 Report

计11 周昉平 2021010699

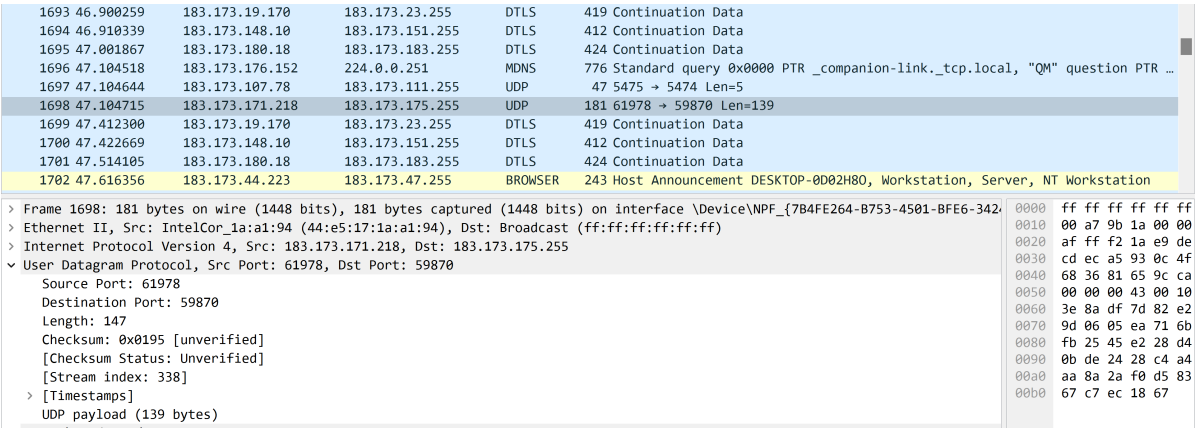
## 抓包实验 1： 观察 UDP 消息

(1) UDP 数据包在 IP 层的类型编号是？



17

(2) UDP 数据包头字段依次是？



源端口号(16bit), 目的端口号(16bit), UDP 包总长(16bit), 校验和(16bit)

## 抓包实验 2： 观察 TCP 消息

(1) TCP 数据包在 IP 层的类型编号是？

No.	Time	Source	Destination	Protocol	Length	Info
55	3.988773	183.173.106.44	202.89.233.101	TCP	66	57657 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
56	4.034339	202.89.233.101	183.173.106.44	TCP	66	443 → 57657 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1300 WS=256 SA...
57	4.034432	183.173.106.44	202.89.233.101	TCP	54	57657 → 443 [ACK] Seq=1 Ack=1 Win=131072 Len=0
58	4.034787	183.173.106.44	202.89.233.101	TLSv1.2	571	Client Hello
59	4.061953	202.89.233.101	183.173.106.44	TCP	60	443 → 57657 [ACK] Seq=1 Ack=518 Win=4194560 Len=0
60	4.061953	202.89.233.101	183.173.106.44	TCP	6554	443 → 57657 [ACK] Seq=1 Ack=518 Win=4194560 Len=6500 [TCP segment of ...
61	4.061953	202.89.233.101	183.173.106.44	TLSv1.2	590	Server Hello, Certificate, Certificate Status, Server Key Exchange, S...
62	4.062105	183.173.106.44	202.89.233.101	TCP	54	57657 → 443 [ACK] Seq=518 Ack=7037 Win=131072 Len=0
63	4.065819	183.173.106.44	202.89.233.101	TLSv1.2	212	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
64	4.065956	183.173.106.44	202.89.233.101	TLSv1.2	159	Application Data

Internet Protocol Version 4, Src: 183.173.106.44, Dst: 202.89.233.101	0000	94 29 2f 37 88 02
0100 .... = Version: 4	0010	00 34 65 22 40 0e
.... 0101 = Header Length: 20 bytes (5)	0020	e9 65 e1 39 01 b1
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)	0030	fa f0 a0 8e 00 00
Total Length: 52	0040	04 02
Identification: 0x6522 (25890)		
> 010. .... = Flags: 0x2, Don't fragment		
...0 0000 0000 0000 = Fragment Offset: 0		
Time to Live: 128		
Protocol: TCP (6)		
Header Checksum: 0xc008 [validation disabled]		
[Header checksum status: Unverified]		
Source Address: 183.173.106.44		
Destination Address: 202.89.233.101		

6

## (2) TCP 数据包头字段依次是？

No.	Time	Source	Destination	Protocol	Length	Info
55	3.988773	183.173.106.44	202.89.233.101	TCP	66	57657 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
56	4.034339	202.89.233.101	183.173.106.44	TCP	66	443 → 57657 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1300 WS=256 SA...
57	4.034432	183.173.106.44	202.89.233.101	TCP	54	57657 → 443 [ACK] Seq=1 Ack=1 Win=131072 Len=0
58	4.034787	183.173.106.44	202.89.233.101	TLSv1.2	571	Client Hello
59	4.061953	202.89.233.101	183.173.106.44	TCP	60	443 → 57657 [ACK] Seq=1 Ack=518 Win=4194560 Len=0
60	4.061953	202.89.233.101	183.173.106.44	TCP	6554	443 → 57657 [ACK] Seq=1 Ack=518 Win=4194560 Len=6500 [TCP segment of ...
61	4.061953	202.89.233.101	183.173.106.44	TLSv1.2	590	Server Hello, Certificate, Certificate Status, Server Key Exchange, S...
62	4.062105	183.173.106.44	202.89.233.101	TCP	54	57657 → 443 [ACK] Seq=518 Ack=7037 Win=131072 Len=0
63	4.065819	183.173.106.44	202.89.233.101	TLSv1.2	212	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
64	4.065956	183.173.106.44	202.89.233.101	TLSv1.2	159	Application Data

> Frame 55: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{7B4FE264-B753-4501-BFE6-34242C41...}	0000	94 29 2f 37 88 02
> Ethernet II, Src: IntelCor_9c:a0:31 (b0:a4:60:9c:a0:31), Dst: NewH3CTe_37:88:02 (94:29:2f:37:88:02)	0010	00 34 65 22 40 0e
> Internet Protocol Version 4, Src: 183.173.106.44, Dst: 202.89.233.101	0020	e9 65 e1 39 01 b1
> Transmission Control Protocol, Src Port: 57657, Dst Port: 443, Seq: 0, Len: 0	0030	fa f0 a0 8e 00 00
Source Port: 57657	0040	04 02
Destination Port: 443		
[Stream index: 0]		
[Conversation completeness: Incomplete, DATA (15)]		
[TCP Segment Len: 0]		
Sequence Number: 0 (relative sequence number)		
Sequence Number (raw): 2898619965		
[Next Sequence Number: 1 (relative sequence number)]		
Acknowledgment Number: 0		
Acknowledgment number (raw): 0		
1000 .... = Header Length: 32 bytes (8)		
1000 .... = Header Length: 32 bytes (8)		
> Flags: 0x002 (SYN)		
Window: 64240		
[Calculated window size: 64240]		
Checksum: 0xa08e [unverified]		
[Checksum Status: Unverified]		
Urgent Pointer: 0		

源端口号(16bit)，目的端口号(16bit)，报文序列号(32bit)，报文确认序列号(32bit)，包头长度(4bit)，保留位和标记位(12bit)，窗口大小(16bit)，校验和(16bit)，紧急指针 (16bit)，选项

## (3) TCP 三次握手过程使用三个数据包，他们的标记位，序列号，确认序列号有什么特点？ TCP 握手时使用选项协商链接参数，举出一个例子？

[TCP Segment Len: 0]	0000	94 29 2f 37 88 02
Sequence Number: 0 (relative sequence number)	0010	00 34 65 22 40 0e
Sequence Number (raw): 2898619965	0020	e9 65 e1 39 01 b1
[Next Sequence Number: 1 (relative sequence number)]	0030	fa f0 a0 8e 00 00
Acknowledgment Number: 0	0040	04 02
Acknowledgment number (raw): 0		
1000 .... = Header Length: 32 bytes (8)		
> Flags: 0x002 (SYN)		
Window: 64240		

[TCP Segment Len: 0]	0000	b0 a4 60 9c a0 31
Sequence Number: 0 (relative sequence number)	0010	00 34 56 b3 40 00
Sequence Number (raw): 356335323	0020	6a 2c 01 bb e1 39
[Next Sequence Number: 1 (relative sequence number)]	0030	ff ff 47 f6 00 00
Acknowledgment Number: 1 (relative ack number)	0040	04 02
Acknowledgment number (raw): 2898619966		
1000 .... = Header Length: 32 bytes (8)		
> Flags: 0x012 (SYN, ACK)		
Window: 65535		

[TCP Segment Len: 0]	0000	94 29 2f 37 88 02
Sequence Number: 1 (relative sequence number)	0010	00 28 65 23 40 00
Sequence Number (raw): 2898619966	0020	e9 65 e1 39 01 b1
[Next Sequence Number: 1 (relative sequence number)]	0030	02 00 86 29 00 00
Acknowledgment Number: 1 (relative ack number)		
Acknowledgment number (raw): 356335324		
0101 .... = Header Length: 20 bytes (5)		
> Flags: 0x010 (ACK)		
Window: 512		

(流向, 标记位, 序列号, 确认序列号)依次为:

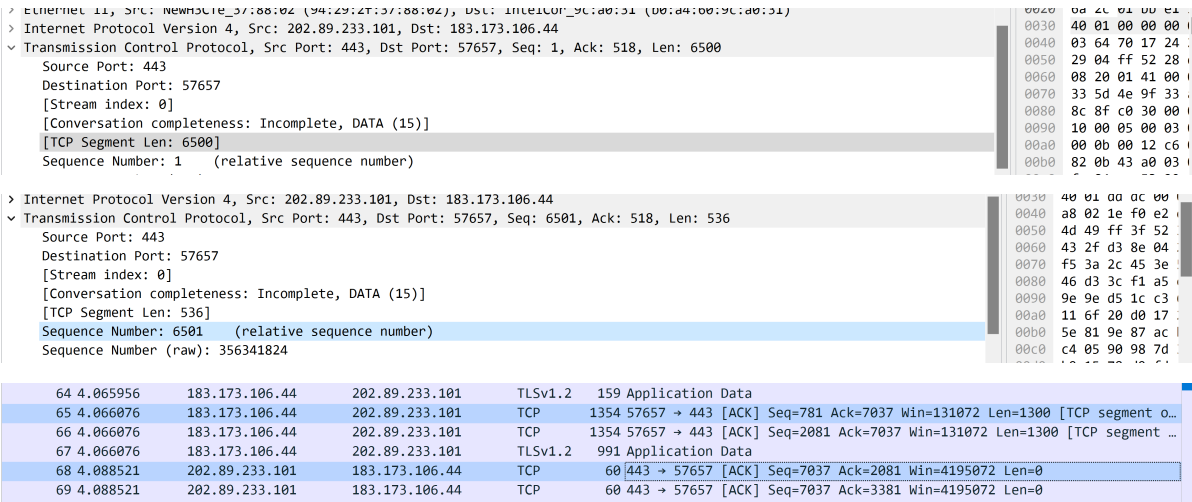
(A->B,SYN,SEQ=X,ACK=0)

(B->A,SYN| |ACK,SEQ=Y,ACK=X+1)

(A->B,ACK,SEQ=X+1,ACK=Y+1)

通过选项协商的例子: MTU 大小, 对 SACK 的支持与否等等

**(4) TCP 传输过程中利用序列号和确认序列号实现数据的可靠传输。序列号增长和包长关系是什么? 确认包确认序列号和原包序列号的关系是什么?**



TCP 序列号的增长差值与前一个包的 TCP 段长度(TCP 数据字段)相等; 数据包对应确认包中, 确认序列号与(原包序列号+段长)相等

**简述题**

**(1) TCP 建立连接时使用选项协商 MTU 信息。上网查资料, TCP 选项还支持什么特殊的功能?**

- 1. 窗口扩大选项: TCP连接的发送方和接收方可以通过这个选项来扩大窗口大小, 以便支持更高的吞吐量和更大的网络延迟。
- 2. 时间戳选项: 该选项允许发送方在TCP报文段中包含时间戳信息, 用于测量往返时间 (RTT) 和计算报文段的往返时间偏差, 从而用于拥塞控制和性能优化。
- 3. 选择确认选项: SACK选项允许接收方向发送方提供更详细的确认信息, 指示接收方已成功接收到哪些连续的数据段, 以便发送方可以仅重传丢失的数据段, 而不是整个窗口的数据。
- 4. 最大报文段长度选项: MSS选项用于在连接建立时协商两端所能接收的最大TCP报文段长度, 以便进行分段和重组。

**(2) 反射 DoS 攻击中, 攻击者将数据包源地址改为受害者 IP 向公共服务(DNS, NTP 等等)发送请求, 公共服务回复数据包至受害者IP, 使受害者带宽耗尽。为什么此类攻击大多使用基于 UDP 的公共服务, 而不是基于 TCP 呢?**

- 1. 反射放大效应: UDP协议中的公共服务 (如DNS、NTP) 通常会产生比请求更大的响应数据包。而TCP协议的连接建立过程相对复杂, 需要进行三次握手, 没有明显的放大效应。
- 2. UDP的无连接性: UDP协议是面向无连接的, 没有像TCP那样的连接状态和会话跟踪机制。这使得攻击者可以轻松地伪造源IP地址, 将受害者IP作为源地址发送UDP请求, 而无需与目标服务器建立连接。目标服务器收到请求后, 将响应发送给伪造的源地址 (即受害者IP), 从而使受害者面临大量响应数据包的洪泛攻击。

3. TCP连接资源消耗：TCP协议在建立连接时需要进行三次握手，这对于攻击者而言会耗费大量的资源。同时，TCP连接具有状态，目标服务器会为每个连接维护一定的状态信息，这使得TCP连接资源更容易耗尽。相比之下，UDP无连接性使得反射型攻击更容易实施，攻击者可以发送大量伪造的UDP请求，而无需维护连接状态，从而降低了资源消耗。