Windows网络认证之基于挑战响应认证的NTLM协议

网络认证NTLM协议简介

在平时的测试中，经常会碰到处于工作组的计算机，处于工作组的计算机之间是无法建立一个可信的信托机构的，只能是点对点进行信息的传输。举个例子就是，主机A想要访问主机B上的资源，就要向主机B发送一个存在于主机B上的一个账户，主机B接收以后会在本地进行验证，如果验证成功，才会允许主机A进行相应的访问。

NTLM 协议是一种基于`挑战（Chalenge）/响应（Response）`认证机制，仅支持Windows的网络认证协议。

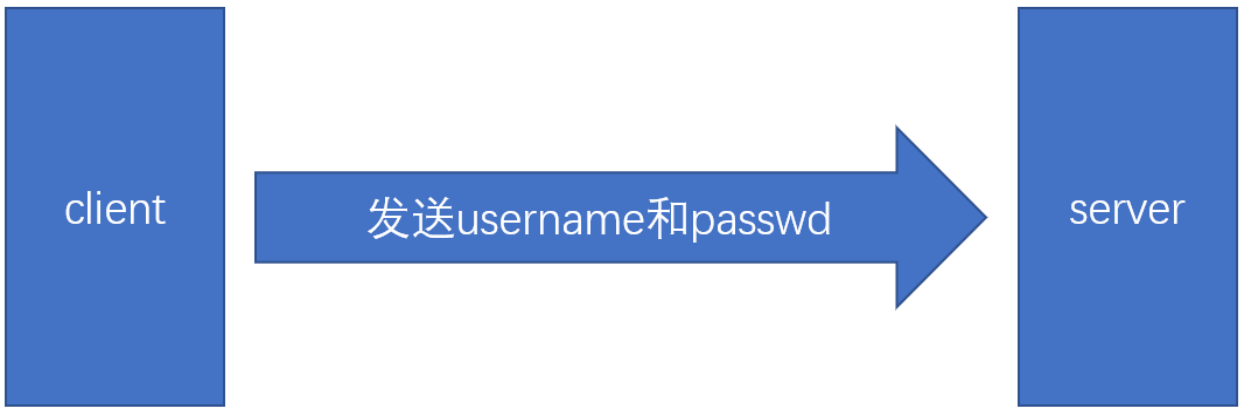
它主要分为协商、质询和验证三个步骤：

> 协商，这个是为了解决历史遗留问题，也就是为了向下兼容，双方先确定一下传输协议的版本等各种信息。
>
> 质`，这一步便是Chalenge/Response认证机制的关键之处，下面会介绍这里的步骤。
>
> 验证，对质询的最后结果进行一个验证，验证通过后，即允许访问资源

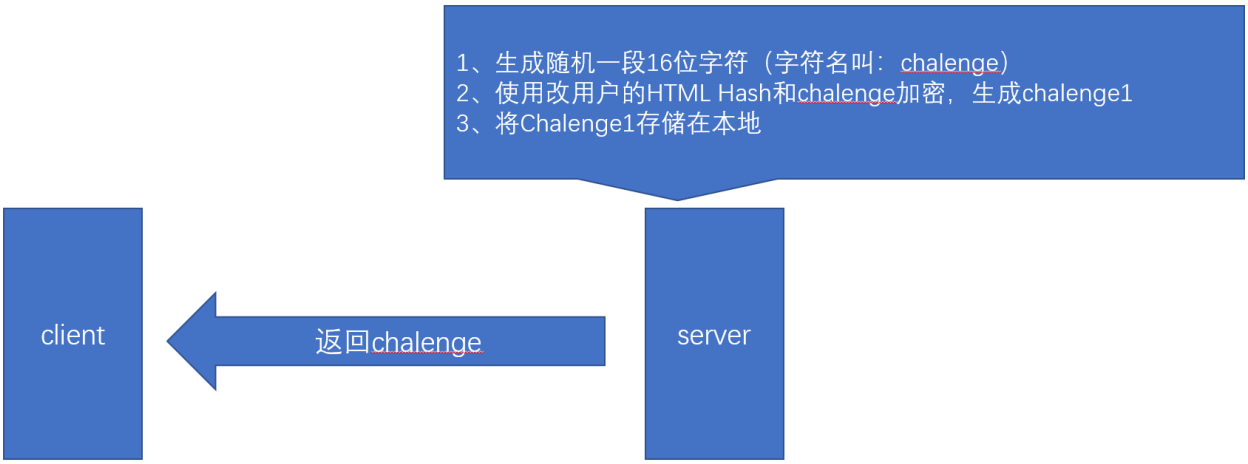
认证流程

认证成功

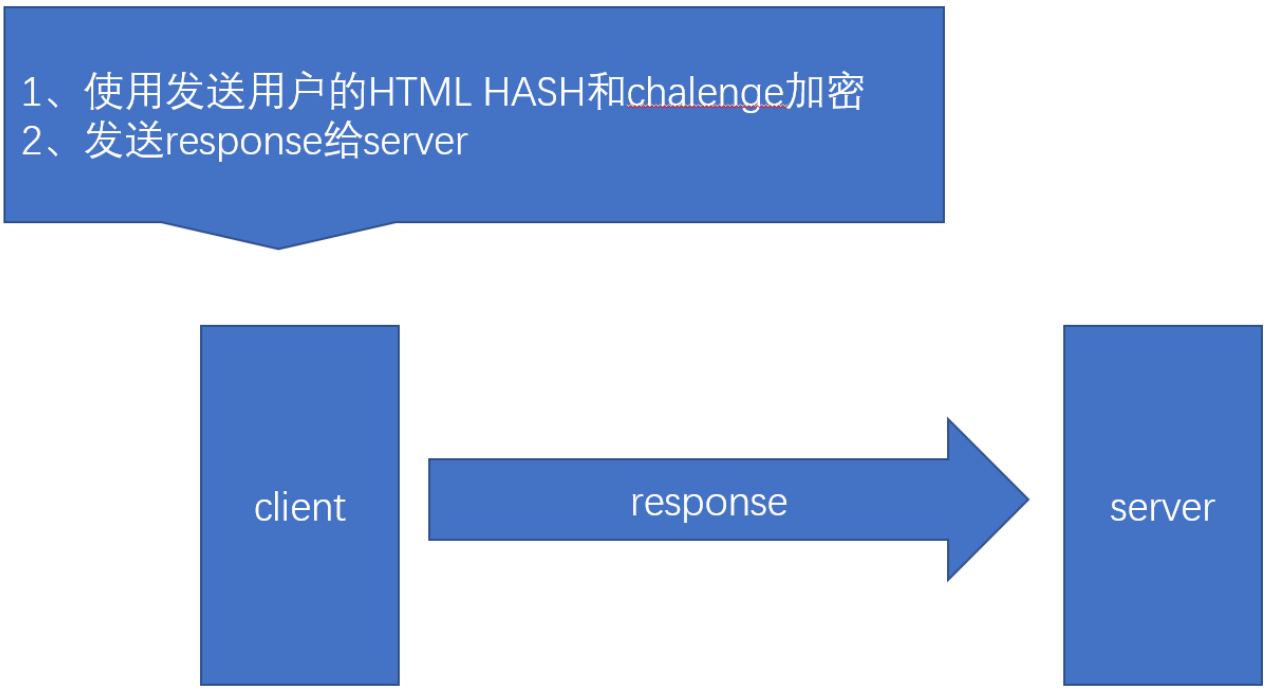
1、首先，client会向server发送一个username，这个username是存在于server上的一个用户



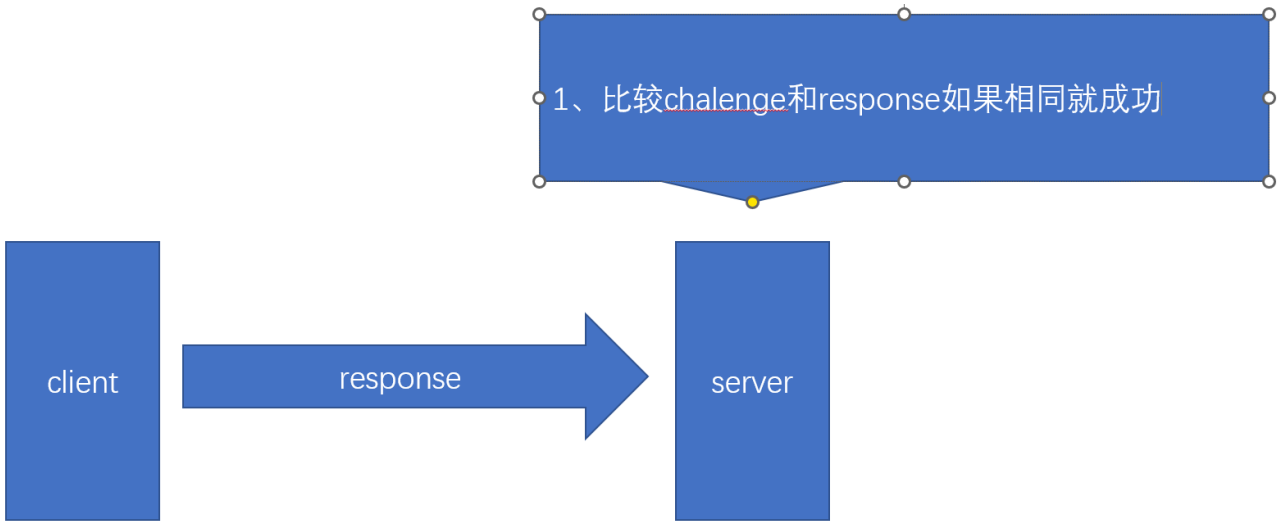
2、首先会在本地查询是否存在这样的一个用户，如果存在，将会生成一个16位的随机字符，即Chalenge，然后用查询到的这个user的NTLM hash对Chalenge进行加密，生成Chalenge1，将Chalenge1存储在本地，并将Chalenge传给client。



3、当client接收到Chalenge时，将发送的username所对应的NTLM hash对Chalenge进行加密即Response，并Response发送给server。



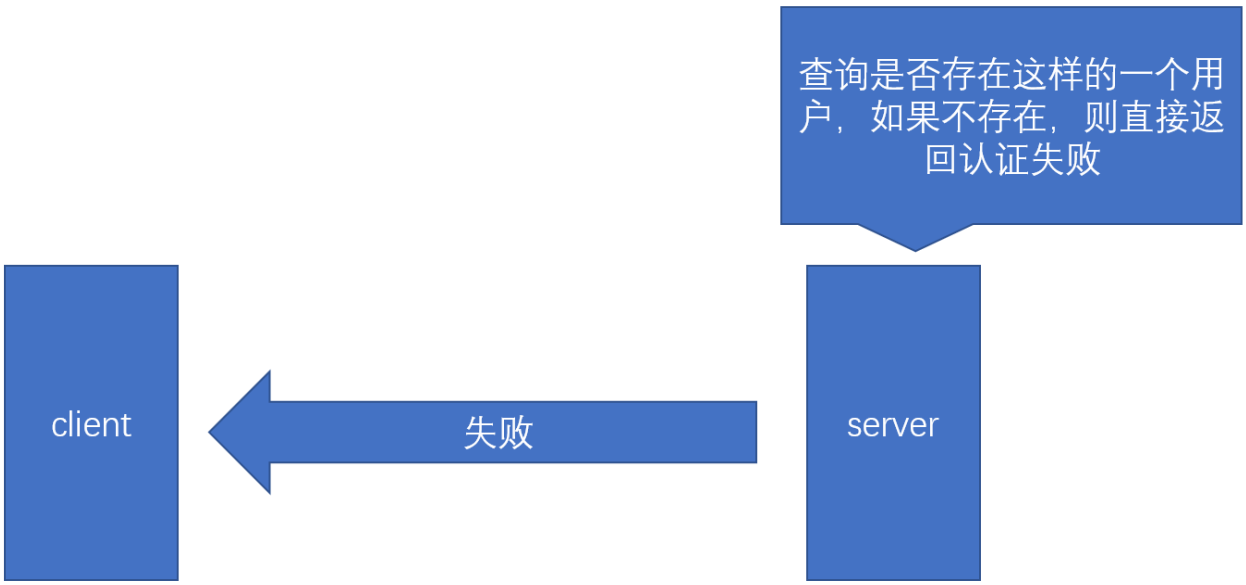
4、server在收到Response后，将其与Chalenge1进行比较，如果相同，则验证成功



认证失败

1、首先，client会向server发送一个username，这个username是存在于server上的一个用户

2、当server接收到这个信息时，首先会在本地查询是否存在这样的一个用户，如果不存在，则直接返回认证失败



NTML协议v1和v2区别

NTLM V2协议，NTLMv1与NTLM v2最显著的区别就是Challenge与加密算法不同，共同点就是加密的原料都是NTLM Hash，

```
NTLM v1的Challenge有8位，NTLM v1的主要加密算法是DES
NTLM v2的Challenge为16位；NTLM v2的主要加密算法是HMAC-MD5。
```

抓包分析

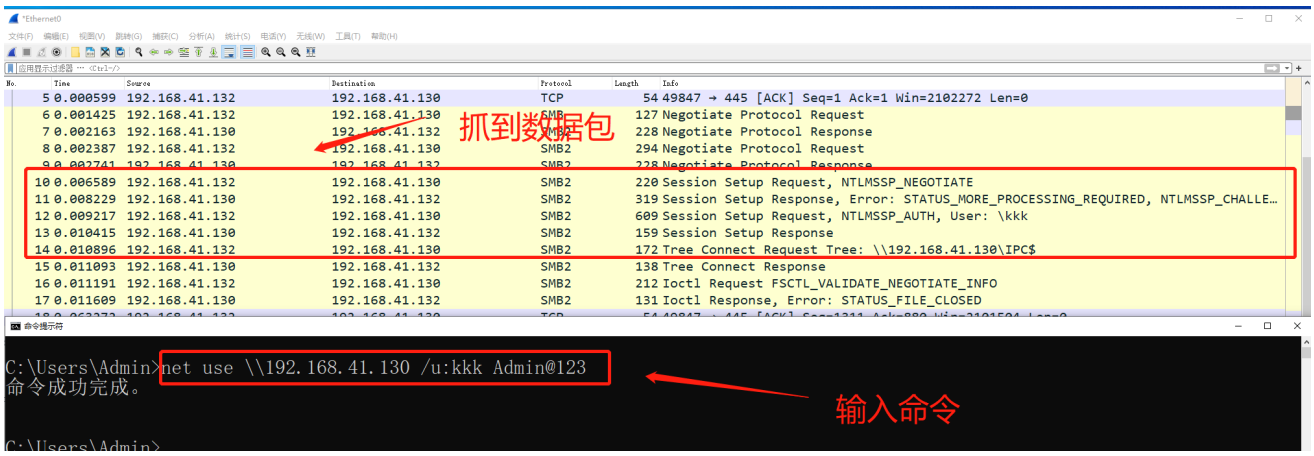
1、实验环境如下

机器名称	IP地址	账号密码
实验机器（windows 10）	192.168.41.132	自己的
靶机（windows server 2008 ）	192.168.41.130	kkk/Admin@123

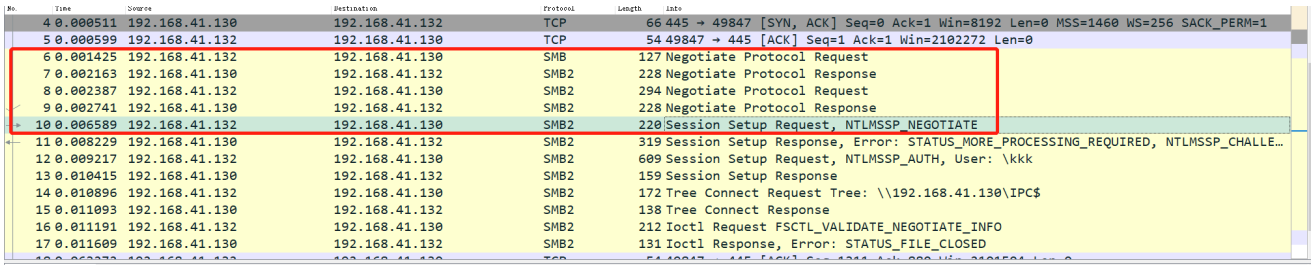
2、windows 10 上 已经安装了 wireshark

3、使用如下命令进行远程连接，并且使用wireshark 包

```
net use \\192.168.41.130 /u:kkk Admin@123
```

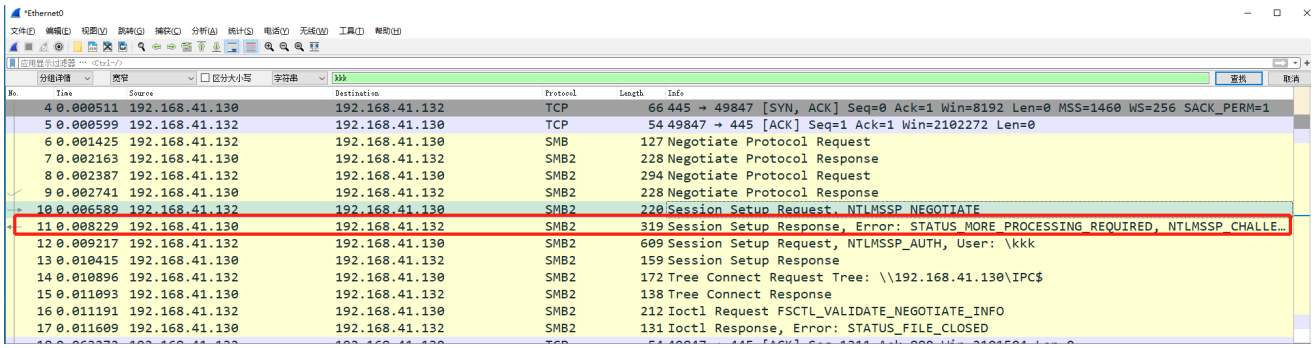


4、前5个数据包中前四条时协商，第五个是认证的第一个数据包



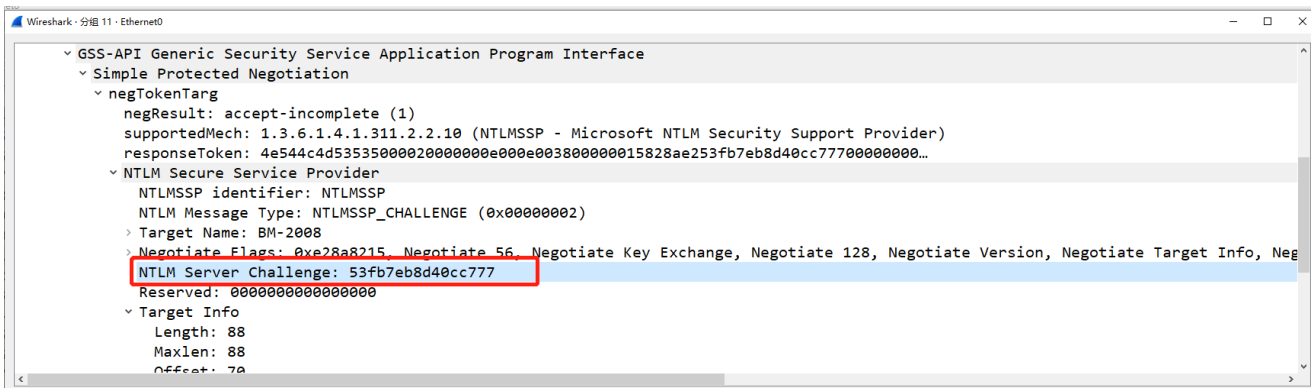
No.	Time	Source	Destination	Protocol	Length	Info
4	0.000511	192.168.41.130	192.168.41.132	TCP	66	445 → 49847 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
5	0.000599	192.168.41.132	192.168.41.130	TCP	54	49847 → 445 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
6	0.001425	192.168.41.132	192.168.41.130	SMB	127	Negotiate Protocol Request
7	0.002163	192.168.41.130	192.168.41.132	SMB2	228	Negotiate Protocol Response
8	0.002387	192.168.41.132	192.168.41.130	SMB2	294	Negotiate Protocol Request
9	0.002741	192.168.41.130	192.168.41.132	SMB2	228	Negotiate Protocol Response
10	0.006589	192.168.41.132	192.168.41.130	SMB2	220	Session Setup Request, NTLMSSP_NEGOTIATE
11	0.008229	192.168.41.130	192.168.41.132	SMB2	319	Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_CHALLE…
12	0.009217	192.168.41.132	192.168.41.130	SMB2	609	Session Setup Request, NTLMSSP_AUTH, User: \kkk
13	0.010415	192.168.41.130	192.168.41.132	SMB2	159	Session Setup Response
14	0.010896	192.168.41.132	192.168.41.130	SMB2	172	Tree Connect Request Tree: \\192.168.41.130\IPC$
15	0.011093	192.168.41.130	192.168.41.132	SMB2	138	Tree Connect Response
16	0.011191	192.168.41.132	192.168.41.130	SMB2	212	Ioctl Request FSCTL_VALIDATE_NEGOTIATE_INFO
17	0.011609	192.168.41.130	192.168.41.132	SMB2	131	Ioctl Response, Error: STATUS_FILE_CLOSED

5、第6个数据包就是返回chalenge挑战值



No.	Time	Source	Destination	Protocol	Length	Info
4	0.000511	192.168.41.130	192.168.41.132	TCP	66	445 → 49847 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
5	0.000599	192.168.41.132	192.168.41.130	TCP	54	49847 → 445 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
6	0.001425	192.168.41.132	192.168.41.130	SMB	127	Negotiate Protocol Request
7	0.002163	192.168.41.130	192.168.41.132	SMB2	228	Negotiate Protocol Response
8	0.002387	192.168.41.132	192.168.41.130	SMB2	294	Negotiate Protocol Request
9	0.002741	192.168.41.130	192.168.41.132	SMB2	228	Negotiate Protocol Response
10	0.006589	192.168.41.132	192.168.41.130	SMB2	220	Session Setup Request, NTLMSSP_NEGOTIATE
11	0.008229	192.168.41.130	192.168.41.132	SMB2	319	Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_CHALLE…
12	0.009217	192.168.41.132	192.168.41.130	SMB2	609	Session Setup Request, NTLMSSP_AUTH, User: \kkk
13	0.010415	192.168.41.130	192.168.41.132	SMB2	159	Session Setup Response
14	0.010896	192.168.41.132	192.168.41.130	SMB2	172	Tree Connect Request Tree: \\192.168.41.130\IPC$
15	0.011093	192.168.41.130	192.168.41.132	SMB2	138	Tree Connect Response
16	0.011191	192.168.41.132	192.168.41.130	SMB2	212	Ioctl Request FSCTL_VALIDATE_NEGOTIATE_INFO
17	0.011609	192.168.41.130	192.168.41.132	SMB2	131	Ioctl Response, Error: STATUS_FILE_CLOSED

分析该数据包得到chalenge值 `53fb7eb8d40cc777`

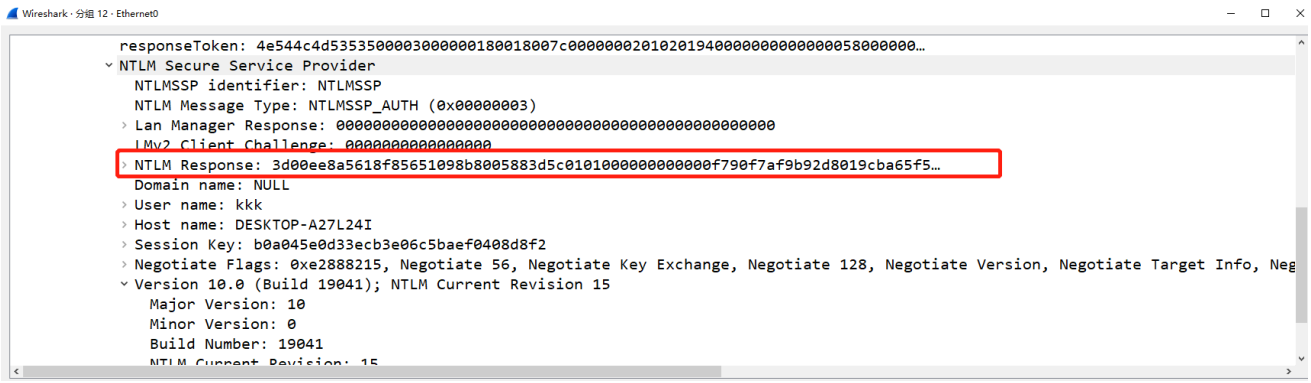


```
GSS-API Generic Security Service Application Program Interface
  Simple Protected Negotiation
    negTokenTarg
      negResult: accept-incomplete (1)
      supportedMech: 1.3.6.1.4.1.311.2.2.10 (NTLMSSP - Microsoft NTLM Security Support Provider)
      responseToken: 4e544c4d535350000200000000e000e003800000015828ae253fb7eb8d40cc77700000000…
      NTLM Secure Service Provider
        NTLMSSP identifier: NTLMSSP
        NTLM Message Type: NTLMSSP_CHALLENGE (0x00000002)
        Target Name: BM-2008
        Negotiate Flags: 0xe28a8215, Negotiate 56, Negotiate Key Exchange, Negotiate 128, Negotiate Version, Negotiate Target Info, Neg
        NTLM Server Challenge: 53fb7eb8d40cc777
        Reserved: 0000000000000000
        Target Info
          Length: 88
          Maxlen: 88
          Offset: 70
```

6、第7个数据包就是返回response的数据包

rsponse数据如下：

```
3d00ee8a5618f85651098b8005883d5c0101000000000000f790f7af9b92d8019cba65f5e39a1ea90000000002000e00
42004d002d003200300030003800001000e0042004d002d003200300030003800004000e0042004d002d00320030003000
380003000e0042004d002d003200300030003800070008 00f790f7af9b92d80106000400020000000800300030000000
0000000001000000002000009906b326309f0ba76eb46b2271795e5d12df73e87035391df48f0fad1ce073380a001000
0000000000000000000000000000000000090026006300690066007300 2f00310039003200 2e00310036003800 2e003400
31002e00310033003000000000000000000000
```

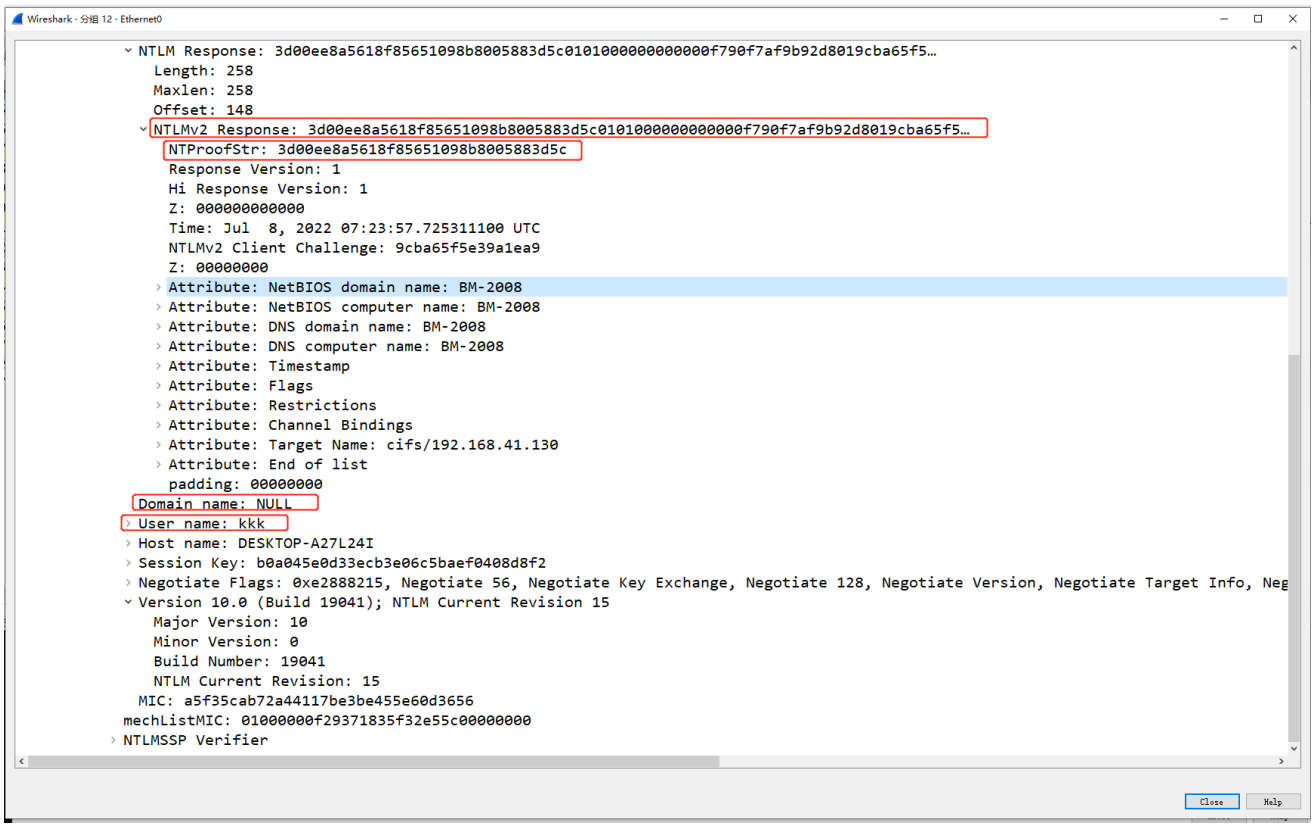


7、接下来得到NTLMv2 数据，NTLMv2格式如下：

```
username::domain:challenge:HMAC-MD5:blob
```

介绍如下：

```
username：对应数据包中 user name
domain:对应数据包中的 Domain name
HMAC-MD5：对应数据包中的NTProofStr
blob：数据库包中rsponse去掉HMAC-MD5的值
```



8、最终的到HTLNv2如下：

```
kkk:::53fb7eb8d40cc777:3d00ee8a5618f85651098b8005883d5c:0101000000000000f790f7af9b92d8019cba65f5
e39a1ea90000000002000e0042004d002d003200300030003800001000e0042004d002d003200300030003800040000e00
42004d002d003200300030003800003000e0042004d002d003200300030003800070008000f790f7af9b92d80106000400
0200000008003000300000000000000001000000002000009906b326309f0ba76eb46b2271795e5d12df73e87035391d
f48f0fad1ce073380a001000000000000000000000000000000000000090026006300690066007300 2f00310039003200
2e003100360038002e00340031002e003100330030000000000000000000
```

9、使用hashcat 破解密码

```
hashcat -m 5600
kkk:::53fb7eb8d40cc777:3d00ee8a5618f85651098b8005883d5c:0101000000000000f790f7af9b92d8019cba65f5
e39a1ea90000000002000e0042004d002d003200300030003800001000e0042004d002d003200300030003800040000e00
42004d002d003200300030003800003000e0042004d002d003200300030003800070008000f790f7af9b92d80106000400
0200000008003000300000000000000001000000002000009906b326309f0ba76eb46b2271795e5d12df73e87035391d
f48f0fad1ce073380a001000000000000000000000000000000000000090026006300690066007300 2f00310039003200
2e003100360038002e00340031002e003100330030000000000000000000 1.txt --force
```

上面的1.txt是我自己的密码字典，你们用自己的

10、使用hashcat破解得到密码

```
┌──(root㉿kali)-[~/Desktop]
└─# hashcat -m 5600 kkk:::53fb7eb8d40cc777:3d00ee8a5618f85651098b8005883d5c:0101000000000000f7
90f7af9b92d8019cba65f5e39a1ea90000000002000e0042004d002d003200300030003800001000e0042004d002d00
3200300030003800040000e0042004d002d003200300030003800030000e0042004d002d003200300030003800070008
00f790f7af9b92d80106000400020000000800300030000000000000000100000000200000990 6b326309f0ba76eb4
6b2271795e5d12df73e87035391df48f0fad1ce073380a00100000000000000000000000000000000000000900260063
006900660073002f00310039003200 2e003100360038002e00340031002e003100330030000000000000000000 1.l
ist --force
```

```
KKK:::53fb7eb8d40cc777:3d00ee8a5618f85651098b8005883d5c:0101000000000000f790f7af9b92d8019cba65
f5e39a1ea90000000002000e0042004d002d003200300030003800001000e0042004d002d003200300030003800040000
0e0042004d002d003200300030003800003000e0042004d002d003200300030003800070008000f790f7af9b92d80106
000400020000000800300030000000000000000100000000200000990 6b326309f0ba76eb46b2271795e5d12df73e8
7035391df48f0fad1ce073380a00100000000000000000000000000000000000000900260063006900660073002f0031
00390032002e003100360038002e00340031002e003100330030000000000000000000:Admin@123

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 5600 (NetNTLMv2)
Hash.Target......: KKK:::53fb7eb8d40cc777:3d00ee8a5618f85651098b800588 ... 000000
Time.Started.....: Fri Jul  8 03:50:36 2022, (0 secs)
Time.Estimated ...: Fri Jul  8 03:50:36 2022, (0 secs)
Kernel.Feature ...: Pure Kernel
Guess.Base.......: File (1.list)
Guess.Queue......: 1/1 (100.00%)
```