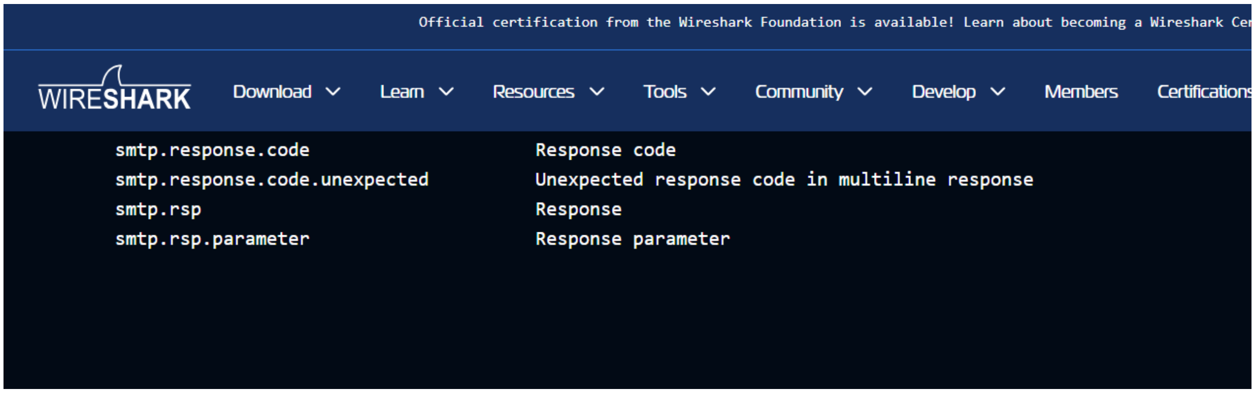
Phishing Prevention 2026/01/13

Analyzing SMTP responses

In this task, you will analyze a PCAP file with SMTP traffic. Some familiarity with traffic analysis using Wireshark will be helpful, as well as knowledge of SMTP Wireshark filters and status codes.

Go ahead and deploy the machine attached to this task. It will appear in the split-screen view when it's ready. Then, open the traffic.pcap file on the Desktop to begin your examination.

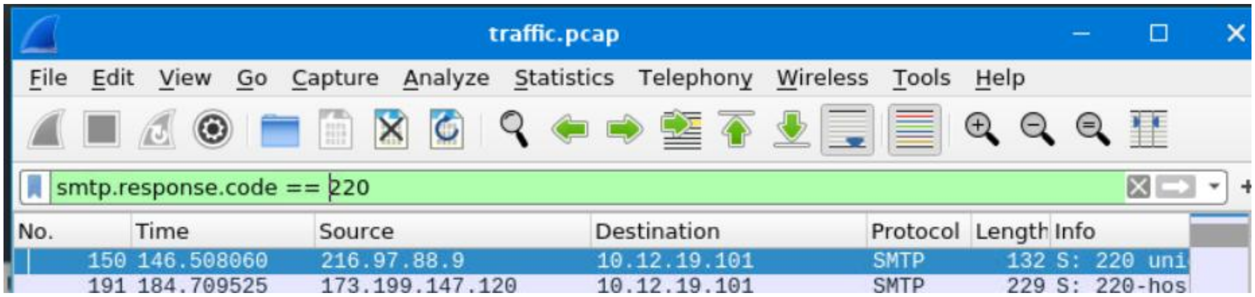
Step 1: Which Wireshark filter can you use to narrow down your results based on SMTP response codes?


Official certification from the Wireshark Foundation is available! Learn about becoming a Wireshark Cer
WIRESHARK
Download
Learn
Resources
Tools
Community
Develop
Members
Certifications

```
smtp.response.code                 Response code
smtp.response.code.unexpected      Unexpected response code in multiline response
smtp.rsp                           Response
smtp.rsp.parameter                 Response parameter
```

I opened up '**Display Filter Reference: Simple Mail Transfer Protocol**' and skimmed down the Description column to find the 'Response code' description. The field name – **smtp.response.code**.

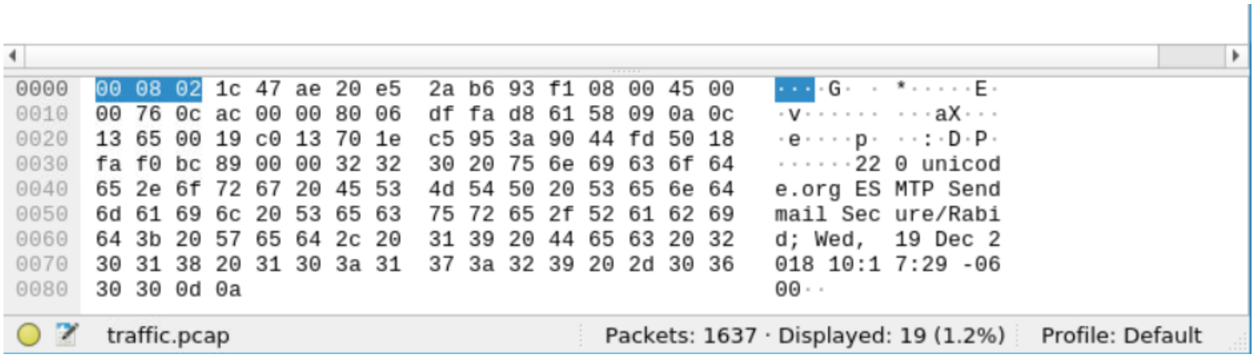
Step 2: How many packets in the capture contain the SMTP response code 220 Service ready?


traffic.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

smtp.response.code == 220

```
No.   Time         Source           Destination    Protocol Length Info
  150 146.508060   216.97.88.9      10.12.19.101   SMTP      132 S: 220 uni
  191 184.709525   173.199.147.120  10.12.19.101   SMTP      229 S: 220-hos
```

I opened up the ‘traffic.pcap’ file on Wireshark and I applied the filter - **smtp.response.code == 220**.

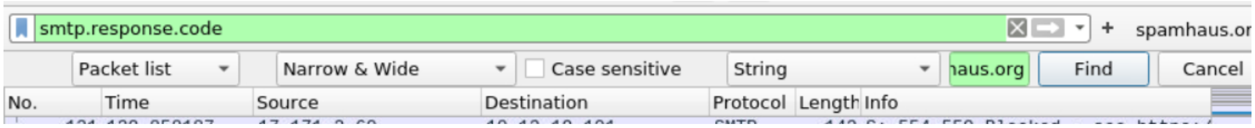


```
0000  00 08 02 1c 47 ae 20 e5  2a b6 93 f1 08 00 45 00   ····G· · *·····E·
0010  00 76 0c ac 00 00 80 06  df fa d8 61 58 09 0a 0c   ·v······ ···aX···
0020  13 65 00 19 c0 13 70 1e  c5 95 3a 90 44 fd 50 18   ·e····p· ··:·D·P·
0030  fa f0 bc 89 00 00 32 32  30 20 75 6e 69 63 6f 64   ······22 0 unicod
0040  65 2e 6f 72 67 20 45 53  4d 54 50 20 53 65 6e 64   e.org ES MTP Send
0050  6d 61 69 6c 20 53 65 63  75 72 65 2f 52 61 62 69   mail Sec ure/Rabi
0060  64 3b 20 57 65 64 2c 20  31 39 20 44 65 63 20 32   d; Wed,  19 Dec 2
0070  30 31 38 20 31 30 3a 31  37 3a 32 39 20 2d 30 36   018 10:1 7:29 -06
0080  30 30 0d 0a                                        00··
```

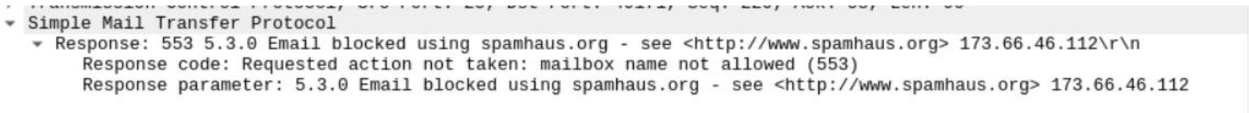
traffic.pcap   Packets: 1637 · Displayed: 19 (1.2%)   Profile: Default

The number of Displayed packets – **19**.

Step 3: One SMTP response indicates that an email was blocked by spamhaus.org. What response code did the server return?



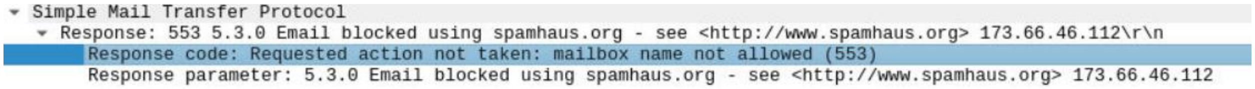
I clicked Edit > Find Packet. I changed the drop down from Display filter to string and entered – **spamhaus.org** and clicked find.



```
▾ Simple Mail Transfer Protocol
  ▾ Response: 553 5.3.0 Email blocked using spamhaus.org - see <http://www.spamhaus.org> 173.66.46.112\r\n
      Response code: Requested action not taken: mailbox name not allowed (553)
      Response parameter: 5.3.0 Email blocked using spamhaus.org - see <http://www.spamhaus.org> 173.66.46.112
```

The code here – **553**

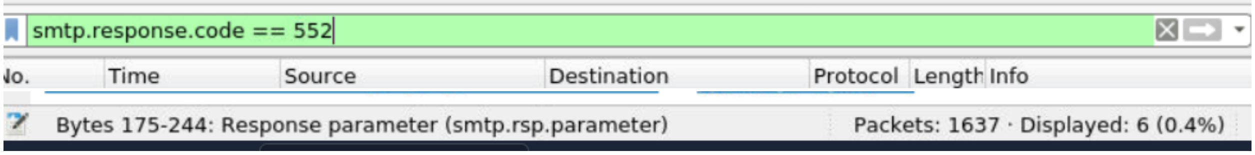
Step 4: Based on the packet from the previous question, what is the full Response code: message?



```
▾ Simple Mail Transfer Protocol
  ▾ Response: 553 5.3.0 Email blocked using spamhaus.org - see <http://www.spamhaus.org> 173.66.46.112\r\n
      Response code: Requested action not taken: mailbox name not allowed (553)
      Response parameter: 5.3.0 Email blocked using spamhaus.org - see <http://www.spamhaus.org> 173.66.46.112
```

Requested action not taken: mailbox name not allowed (553).

Step 5: Search for response code 552. How many messages were blocked for presenting potential security issues?



```
smtp.response.code == 552
```

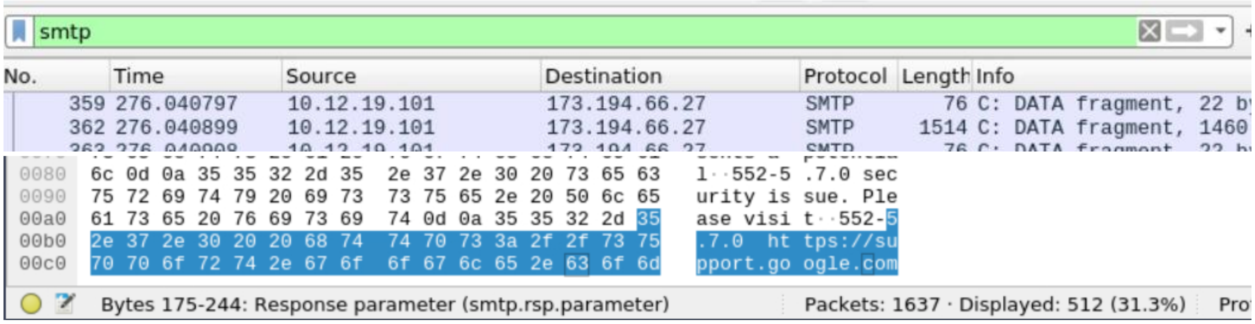
```
Bytes 175-244: Response parameter (smtp.rsp.parameter)        Packets: 1637 · Displayed: 6 (0.4%)
```

I applied the filter - **smtp.response.code == 552**. The number of blocked messages was – **6**.

Inspecting Emails and Attachments

In this task, you'll move beyond SMTP status codes and responses and begin analyzing SMTP traffic using the same traffic capture from the task above. You will utilize the **Internet Message Format** (IMF) to examine the inner details of emails, such as sender and recipient fields, content type, and attachments.

Step 1: How many SMTP packets are available for analysis?



smtp

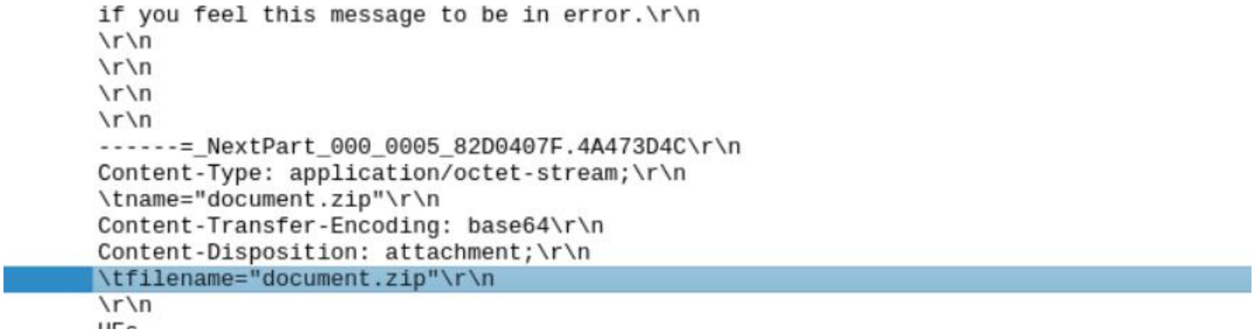
```
No.   Time        Source        Destination     Protocol Length Info
  359 276.040797  10.12.19.101  173.194.66.27   SMTP         76 C: DATA fragment, 22 by
  362 276.040899  10.12.19.101  173.194.66.27   SMTP       1514 C: DATA fragment, 1460
  363 276.040908  10.12.19.101  173.194.66.27   SMTP         76 C: DATA fragment, 22 by
```

```
0080  6c 0d 0a 35 35 32 2d 35  2e 37 2e 30 20 73 65 63   l··552-5 .7.0 sec
0090  75 72 69 74 79 20 69 73  73 75 65 2e 20 50 6c 65   urity is sue. Ple
00a0  61 73 65 20 76 69 73 69  74 0d 0a 35 35 32 2d 35   ase visi t··552-5
00b0  2e 37 2e 30 20 20 68 74  74 70 73 3a 2f 2f 73 75   .7.0  ht tps://su
00c0  70 70 6f 72 74 2e 67 6f  6f 67 6c 65 2e 63 6f 6d   pport.go ogle.com
```

Bytes 175-244: Response parameter (smtp.rsp.parameter)   Packets: 1637 · Displayed: 512 (31.3%)   Pro

I applied the Display filter – **smpt**. I then discovered the number of packets available – **512**.

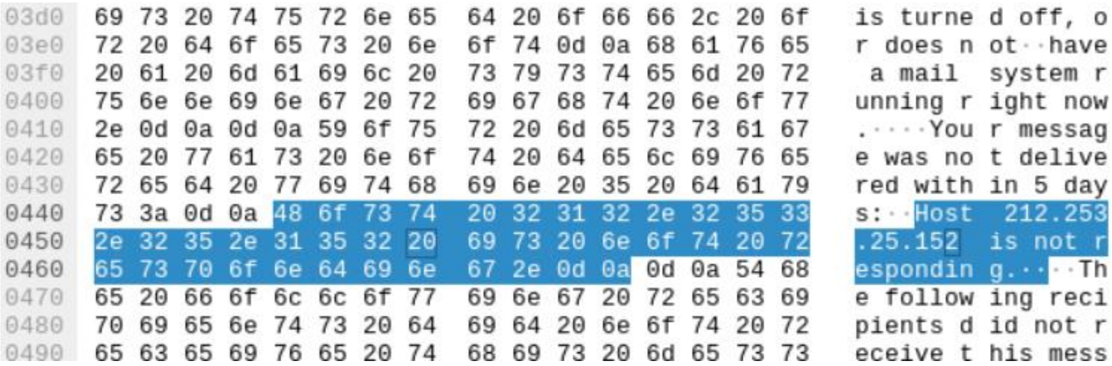
Step 2: What is the name of the attachment in packet 270?



```
if you feel this message to be in error.\r\n
\r\n
\r\n
\r\n
\r\n
------=_NextPart_000_0005_82D0407F.4A473D4C\r\n
Content-Type: application/octet-stream;\r\n
\tname="document.zip"\r\n
Content-Transfer-Encoding: base64\r\n
Content-Disposition: attachment;\r\n
\tfilename="document.zip"\r\n
\r\n
```

I scrolled to find packet **270**. I clicked on it. I navigated to the 'Simple Mail Transfer Protocol' > 'Line based Text Data' section in the lower section and clicked on it. I scrolled further down to find the file – **document.zip**.

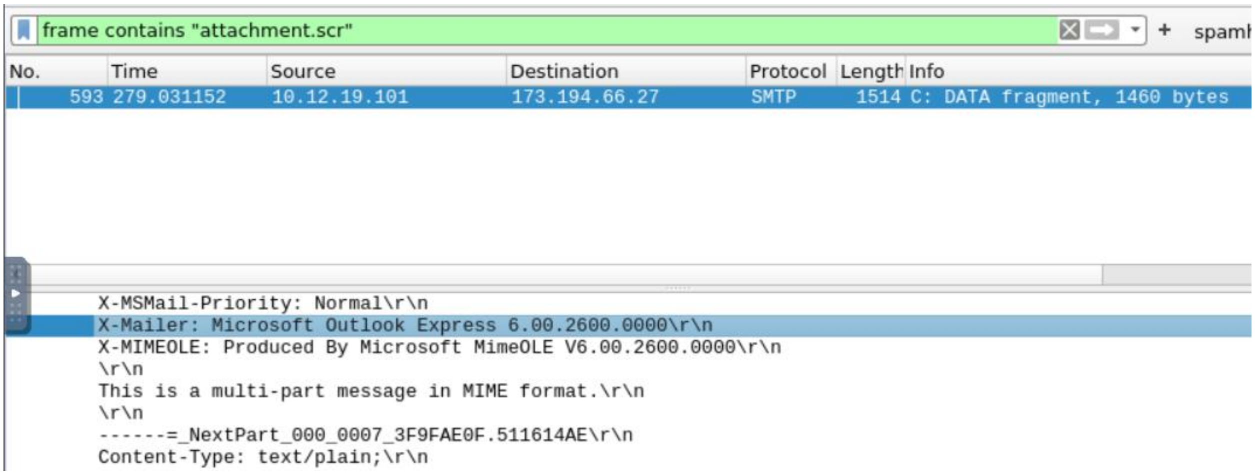
Step 3: According to the message in packet 270, which Host IP address is not responding, making the message undeliverable?



```
03d0  69 73 20 74 75 72 6e 65  64 20 6f 66 66 2c 20 6f   is turne d off, o
03e0  72 20 64 6f 65 73 20 6e  6f 74 0d 0a 68 61 76 65   r does n ot··have
03f0  20 61 20 6d 61 69 6c 20  73 79 73 74 65 6d 20 72    a mail  system r
0400  75 6e 6e 69 6e 67 20 72  69 67 68 74 20 6e 6f 77   unning r ight now
0410  2e 0d 0a 0d 0a 59 6f 75  72 20 6d 65 73 73 61 67   .····You r messag
0420  65 20 77 61 73 20 6e 6f  74 20 64 65 6c 69 76 65   e was no t delive
0430  72 65 64 20 77 69 74 68  69 6e 20 35 20 64 61 79   red with in 5 day
0440  73 3a 0d 0a 48 6f 73 74  20 32 31 32 2e 32 35 33   s:··Host  212.253
0450  2e 32 35 2e 31 35 32 20  69 73 20 6e 6f 74 20 72   .25.152  is not r
0460  65 73 70 6f 6e 64 69 6e  67 2e 0d 0a 0d 0a 54 68   espondin g.····Th
0470  65 20 66 6f 6c 6c 6f 77  69 6e 67 20 72 65 63 69   e follow ing reci
0480  70 69 65 6e 74 73 20 64  69 64 20 6e 6f 74 20 72   pients d id not r
0490  65 63 65 69 76 65 20 74  68 69 73 20 6d 65 73 73   eceive t his mess
```

After investigating contents of the packet, I was able to identify the Host IP address – **212.253.25.152**.

Step 3: By filtering for imf, which email client was used to send the message containing the attachment attachment.scr?



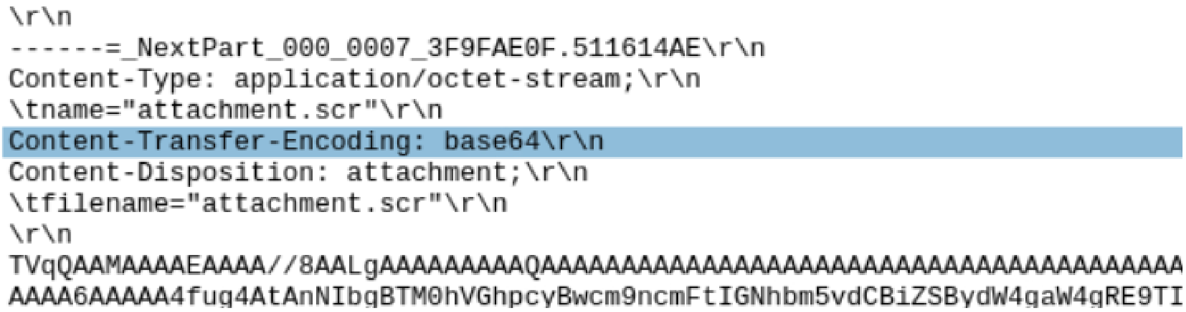
```
frame contains "attachment.scr"

No.   Time        Source        Destination    Protocol Length Info
  593 279.031152  10.12.19.101  173.194.66.27  SMTP      1514 C: DATA fragment, 1460 bytes

X-MSMail-Priority: Normal\r\n
X-Mailer: Microsoft Outlook Express 6.00.2600.0000\r\n
X-MIMEOLE: Produced By Microsoft MimeOLE V6.00.2600.0000\r\n
\r\n
This is a multi-part message in MIME format.\r\n
\r\n
------=_NextPart_000_0007_3F9FAE0F.511614AE\r\n
Content-Type: text/plain;\r\n
```

I applied the filter – 'filter contains "attachment.scr"', clicked on the packet and opened smtp in the lower section and identified the email client - **Microsoft Outlook Express 6.00.2600.0000**.

Step 4: Which type of encoding is used for this potentially malicious attachment?



```
\r\n
------=_NextPart_000_0007_3F9FAE0F.511614AE\r\n
Content-Type: application/octet-stream;\r\n
\tname="attachment.scr"\r\n
Content-Transfer-Encoding: base64\r\n
Content-Disposition: attachment;\r\n
\tfilename="attachment.scr"\r\n
\r\n
TVqQAAMAAAAEAAAA//8AALgAAAAAAAAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAA6AAAAA4fug4AtAnNIbgBTM0hVGhpcyBwcm9ncmFtIGNhbm5vdCBiZSBydW4gaW4gRE9TI
```

I scrolled further down and found that it was encoded using – **base64**.

Conclusion Summary

This exercise strengthened my understanding of phishing prevention by analysing SMTP traffic and email content at the protocol level using Wireshark. By applying targeted display filters, I was able to identify and interpret SMTP response codes, revealing how mail servers accept, reject, or block messages based on security controls such as spam reputation services. Analysing responses related to Spamhaus blocks and security-related rejections demonstrated how preventative controls operate before malicious emails reach end users.

Further inspection of SMTP packets using the Internet Message Format (IMF) allowed me to examine email headers, attachments, encoding methods, and client information. By identifying suspicious attachments, undeliverable hosts, email clients used to send messages, and Base64-encoded payloads, I gained practical insight into how potentially malicious emails are constructed and transmitted.

Overall, this task reinforced the importance of network-level and message-level analysis in phishing prevention, highlighting how SOC analysts can leverage SMTP responses and packet inspection to detect, validate, and stop malicious emails before they are delivered.