

Objective--Network Traffic Forensics (Part 5)

Solution (at last!)

Now that we have the `super_secret_packet_capture.pcap` file, we can answer the question. Wireshark shows that the file is one TCP stream, an unencrypted SMTP connection.

upload_2a4a5ee98007cb261119b208bf9369ef.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-F>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.1.1	10.10.1.25	TCP	74	60830 → 25 [SYN] Seq=0 Win=43690 Len=0 MSS=65495 SACK_PERM=1 TSval=
2	0.000014	10.10.1.1	10.10.1.25	TCP	74	25 → 60830 [SYN, ACK] Seq=0 Ack=1 Win=43690 Len=0 MSS=65495 SACK_P
3	0.000024	10.10.1.1	10.10.1.25	TCP	66	60830 → 25 [ACK] Seq=1 Ack=1 Win=43776 Len=0 TSval=1978359160 TSecr=
4	3.146894	10.10.1.25	10.10.1.1	SMTP	116	5: 220 mail.kringlecastle.com #SMTP Postfix (Ubuntu)
5	3.146118	10.10.1.1	10.10.1.25	TCP	66	60830 → 25 [ACK] Seq=1 Ack=51 Win=43776 Len=0 TSval=1978359946 TSecr=
6	8.986508	10.10.1.1	10.10.1.25	SMTP	94	C: EHLO Mail.kringlecastle.com

Follow TCP > Stream gives us this.

[illegible]

Wireshark · Follow TCP Stream (tcp.stream eq 0) · upload_2a4a5ae98007cb26119b208bf9369ef.p... — □

```
-----=_MIME_BOUNDARY_000_11181
Content-Type: application/octet-stream
Content-Transfer-Encoding: BASE64
Content-Disposition: attachment
```

133 client pkt(s), 16 server pkt(s), 12 turn(s).

Wireshark · Follow TCP Stream (tcp.stream eq 0) · upload_2a4a5ae98007cb261119b208bf9369ef.p...

```
-----= MIME_BOUNDARY_000_11181--
```

Pasting files from Windows to Linux causes problems

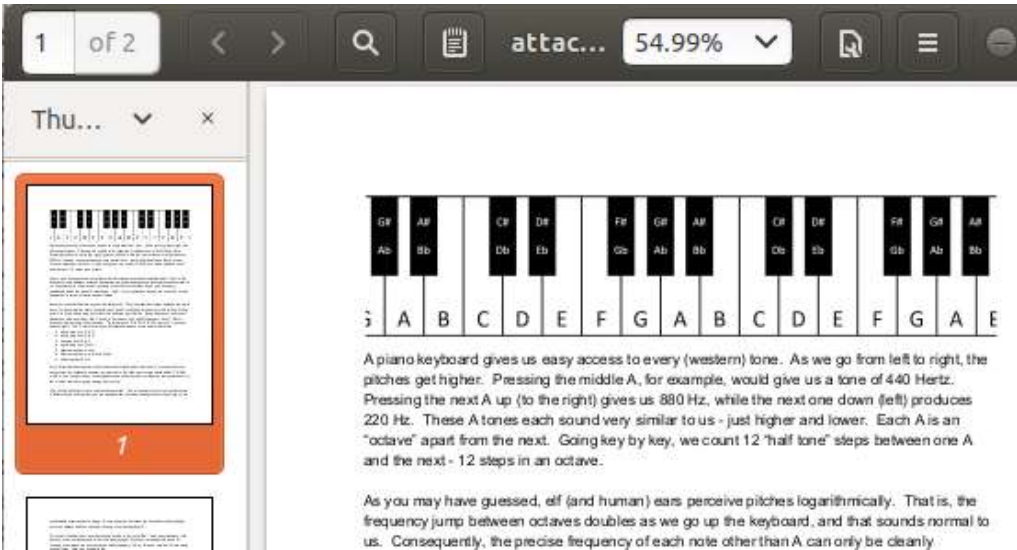
```
john@ubuntu:~$ cd smtp/
john@ubuntu:~/smtp$ file attachment.b64.txt
attachment.b64.txt: ASCII text, with CRLF line terminators
john@ubuntu:~/smtp$ cat attachment.b64.txt | base64 -d > attachment
base64: invalid input
john@ubuntu:~/smtp$ vi attachment
```

```
File Edit View Search Terminal Help
JVBERi0xLjUKJb/3ov4K0CAwIG9Iago8PCAvTGLuZWYyaXplZCAxIC9MIDk3ODMxIC9IFsgNzM4^M
IDE0MCBdIC9PIDEyIC9FIDc3MzQ0IC90IDIGL1QgOTc1MTcgPj4KZW5kb2JqCiAgICAgICAgICAg^M
ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg^M
ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg^M
ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg^M
VHlwZSAvWFJlZiAvTGVuZ3RoIDU5IC9GaWw0ZXIgL0ZsYXRlRGVjb2RlIC9EZWNVZGVQYXJtcyA8^M
PCAvQ29sdW1ucyA1IC9QcmVkaWN0b3IgMTIgPj4gLT1cgWyAxIDMgMSBdIC9JbmRleCBbIDggMjIg^M
XSAvSW5mbyAxOCAwIFRlIGL1Jvb3QgMTAgMCSIC9TaXplIDMwIC9QcmV2IDk3NTE4ICAgICAgICAg^M
ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg^M
ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg^M
YjUxNWZlZGVuZ3RoIDU5IC9GaWw0ZXIgL0ZsYXRlRGVjb2RlIC9EZWNVZGVQYXJtcyA8^M
krEHTJ4GkettQaTACSDJmLoQrHI5AXPj/2vdYPUMjJSQADt+CUSKZW5kc3RyZWZlcmVudG9Iago^M
ICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg^M
```

All those ^M characters cause problems. There is a fix. In vi, use :%s/[cntrl-V][cntrl-M]//g. This is similar to other Linux syntax for search and replace: s/searchfor/replacewith/g for global. Another way would be to use the translate command with -d for delete: tr -d '\r' > file. Both ^M and \r mean “carriage return”. There’s another character, \n or line feed. At the end of lines Windows uses \r\n (carriage return, line feed, or CRLF) and Linux just uses \n. I should have done it all in Linux. What a pain.

After fixing the ^M problems, it is easy to decode the file.

```
john@ubuntu:~/smtp$ vi attachment.b64.txt
john@ubuntu:~/smtp$ cat attachment.b64.txt | base64 -d > attachment
john@ubuntu:~/smtp$ file attachment
attachment: PDF document, version 1.5
john@ubuntu:~/smtp$ mv attachment attachment.pdf
john@ubuntu:~/smtp$
```



Decoding Base64 in Windows

PowerShell can decode Base64, although the syntax is awkward. This will decode.

```
[System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String("U2VjcmV0TWVzc2FnZQ=="))
```

This will encode.

```
[System.Convert]::ToBase64String([System.Text.Encoding]::UTF8.GetBytes("SecretMessage"))
```

Windows certutil.exe is simpler if you don't need a script.

```
PS D:\HolidayHack2018\Lessons> certutil -decode .\attachment.b64.txt attachment.pdf
Input Length = 133876
Output Length = 97831
CertUtil: -decode command completed successfully.
PS D:\HolidayHack2018\Lessons>
```

In either case, the end of the PDF file we extract has this.

And take everything down one half step for A:

C# B A B C# C# C# B B B C# E E C# B A B C# C# C# C# B B C# B A

We've just taken Mary Had a Little Lamb from Bb to A!

That's a lot of work to find "Mary Had a Little Lamb"

Up Next

We will need hints from Shinnay Upatree and his Sleigh bell lottery terminal before we tackle the last objective. You can find him on the right side of the second floor, near the stairs.