Packet Capture Analysis:

I have analyzed the provided packet capture file using the free network analysis tool Wireshark. I was able to put "http" into the filter field to filter the network traffic to only see HTTP packets.

This view let me see some interesting http GET requests, which indicate that the user specifically requests information.

Sub-task 1:

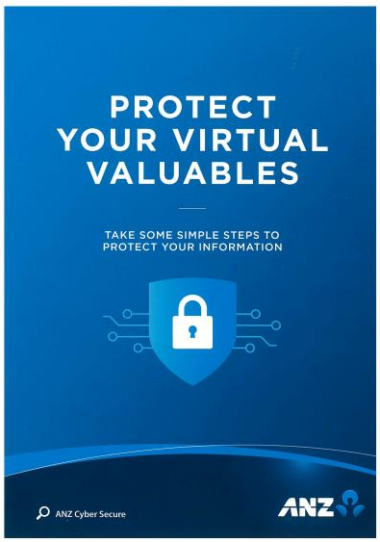
To find the images the user accessed called anz-logo.jpg and bank-card.jpg I followed the following process for both images:
First I filtered the packet capture for http traffic and looked through the remaining packets for the GET request that downloaded the image. I then right clicked the image and followed its TCP stream. In the TCP stream I saw what looked like image data.
In order to view the data in hex format, I changed the view to „raw", and then searched the hex data for a jpeg"s file signature.
After finding the file signature "FFD8" the top, and the file footer "FFD9" at the bottom, I copied everything between those two points into the hex editor HxD and saved it as a jpg image.
Resulting in the image below.

anz-logo.jpg

Bank-card.jpg

Sub-task 2:

I followed the same process to extract these images as I did in sub-task 1, which was to view the TCP stream, identify the images hex data, then copy and save that as a jpg file. The image are as follows:



ANZ-1.jpg

ANZ-2.jpg

The difference in the network traffic for these images download I discovered was a hidden message in the data after the end of the image.
The message said "You've found a hidden message in this file! Include it in your write up" and "You've found the hidden message! Images are sometimes more than they appear."

Sub-task 3:

In order to find the contents of the document, I had to view the TCP stream of the http get request for the file. The documents contents were visible in the ASCII view.

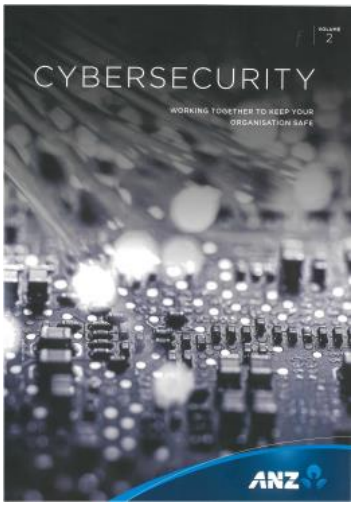
Step 1: Find target
Step 2: Hack them
This is a suspicious document

```
GET /how-to-commit-crimes.docx HTTP/1.1
Host: localhost:8000
Connection: keep-alive
Sec-Fetch-Site: same-origin
User-Agent: Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/76.0.3809.100 Safari/
537.36
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9

HTTP/1.1 200 OK
Date: Fri, 16 Aug 2019 00:48:17 GMT
Server: Apache/2.4.6 (CentOS)
Last-Modified: Mon, 05 Aug 2019 02:23:32 GMT
ETag: "46-58f5564f85059"
Accept-Ranges: bytes
Content-Length: 70
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: application/vnd.openxmlformats-officedocument.wordprocessingml.document

Step 1: Find target
Step 2: Hack them

This is a suspicious document.
```

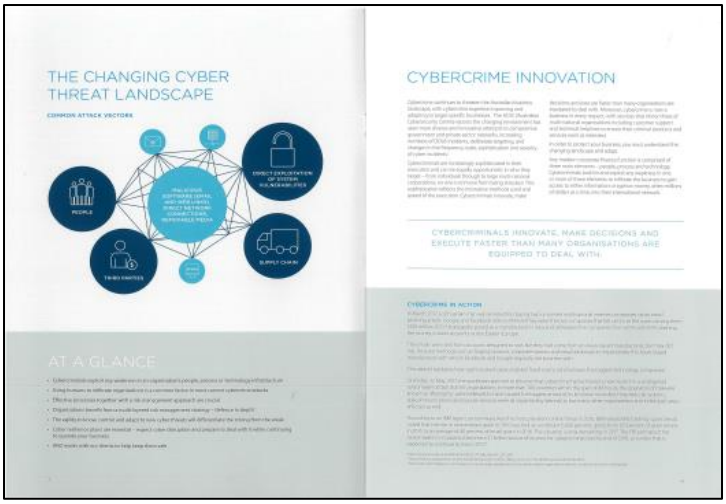
How-to-commit-crime.docx

Sub-task 4:

In order to view these PDF"s I viewed the TCP stream as usual, and found the file signature for a PDF, which was the hex data "25 50 44 46". I noticed in the ASCII view that the PDF data went until the very end of the TCP stream, so I copied all the hex date from the file signature onwards into HxD and saved it as a pdf file. The same process worked for all three files:



ANZ-Document.pdf



ANZ-Document2.pdf

Evil.pdf

Sub-task 5:

I viewed the TCP stream of this file, and noticed that instead of being plain text it was encoded data and when viewed as hex it had the same file signature as a jpg image. So I copied and saved the hex data with HxD as I have for other images, and discovered that the text file was actually this image:

hidden-message2.txt

Sub-task 6:

I viewed the TCP stream as normal when investigating this traffic, and found two sets of jpeg file signatures. In the TCP stream I saw what looked like image data. In order to view the data in hex format, I changed the view to „raw", and then searched the hex data for a jpeg"s file signature. After finding the file signature "FFD8" the top, and the file footer "FFD9" at the bottom, I copied everything between those two points into the hex editor HxD and saved it as a jpg image. I tried extracting both sets of data, and got two different images. Resulting in the image below

1. *Atm-image.jpg*

2. *Atm-image.jpg*

Sub-task 7:

To find the images the user accessed called broken.png I followed the following process for both images: First I filtered the packet capture for http traffic and looked through the remaining packets for the GET request that downloaded the image. I then right clicked the image and followed its TCP stream. In the TCP stream I saw what looked like image data. In order to view the data in hex format, I changed the view to „raw", and then searched the hex data for a jpeg"s file signature. After finding the file signature "89 50 4e 47 0d 0a 1a 0a" I copied everything after that point to end and then copy into the hex editor HxD and saved it as a png image.
The image as follow:

anz-png.png

Sub-task 8:

- *The user accessed one more document called securepdf.pdf*
- *Access this document include an image of the pdf in your report. Detail the steps to access it.*

After investigating TCP stream for securepdf.pdf I discover following thing: The data there was not for a PDF. The bottom of the file contained the hidden message: Password is “secure” It contained the file signature for a zip file, meaning that the the user downloaded was actually a zip file. So I copied the hex of the zip file into HxD and saved it as a zip file. I opened this zip file, and found it contained a pdf file called rawpdf.pdf. When opened, the pdf prompted for a password. The password „secure“ shown in the tcp stream worked and the PDF opened. It was the first two pages to a guide for internet banking.