Packet Capture Analysis:

I have analyzed the Digital\_Investigation Task (pcap file) using the network analysis tool Wireshark.

To get the only HTTP packets, I filtered by inserting “http” in the box in left right corner.

The view let me see some fascinating 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 step-by-step process for both images:

At first, I filtered the packet capture for http traffic to get only HTTP and looked through the remaining packets for the GET request that downloaded the image. Then I have gone to TCP stream by clicking (CTRL+ALT+SHIFT+T). In the TCP stream I observed the data was in ASCII format. In order to view the data in hex format, I changed the view to “raw” and then searched the hex data for a jpegs 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. That gives the below image:



Anz-logo.jpg

And then I repeated the same step-by-step process to get the bank-card image. The image is resulted below:

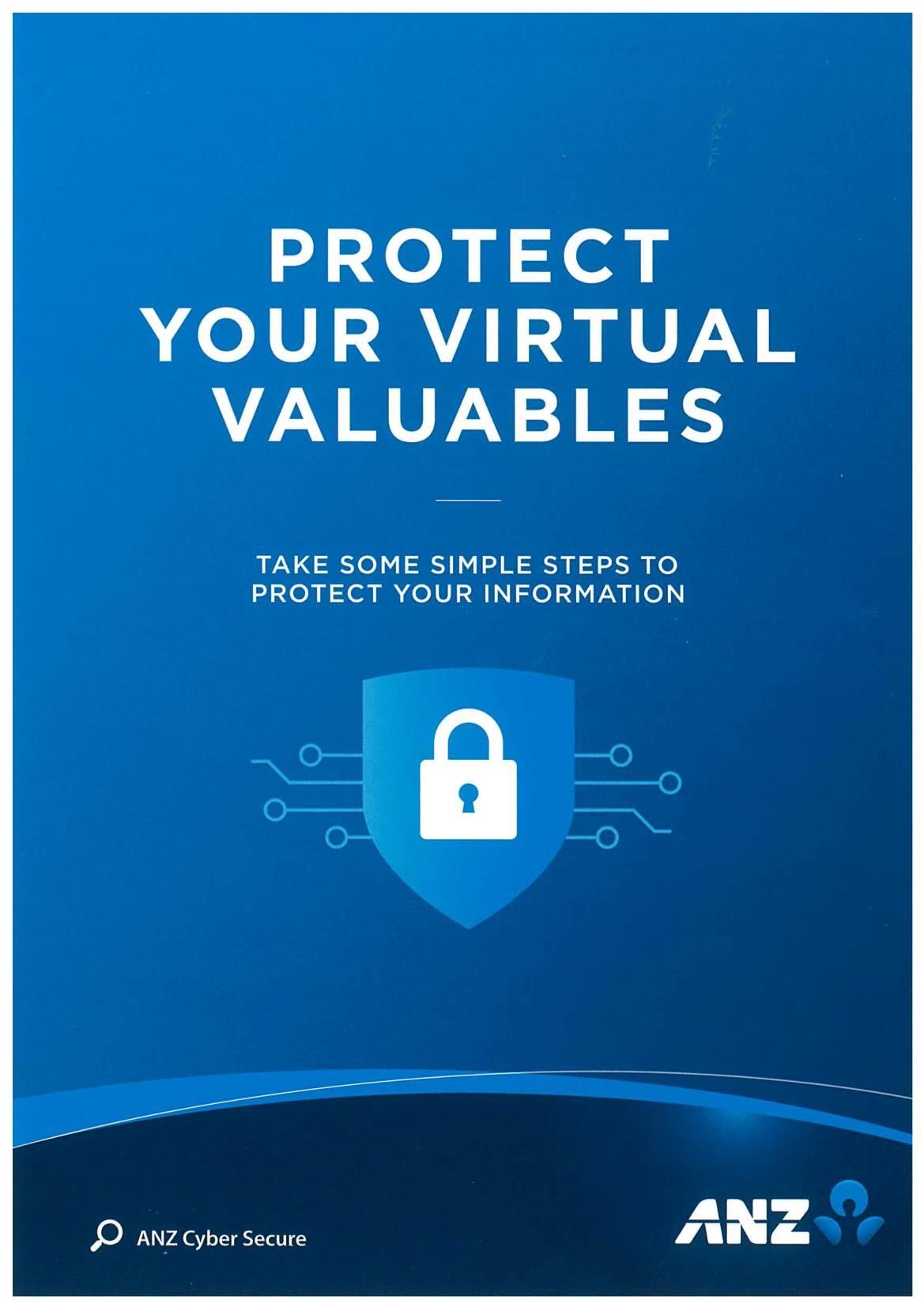


Bank-card.jpg

Sub-task 2:

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

The image I get for the anz1-image is:



The difference in the network traffic of these images that I discovered was a hidden message in the data after the end of the image.

The message is “You've found a hidden message in this file! Include it in your write up.”



This network traffic also had the message hidden in the same way.

It was “You've found the hidden message! Images are sometimes more than they appear.”

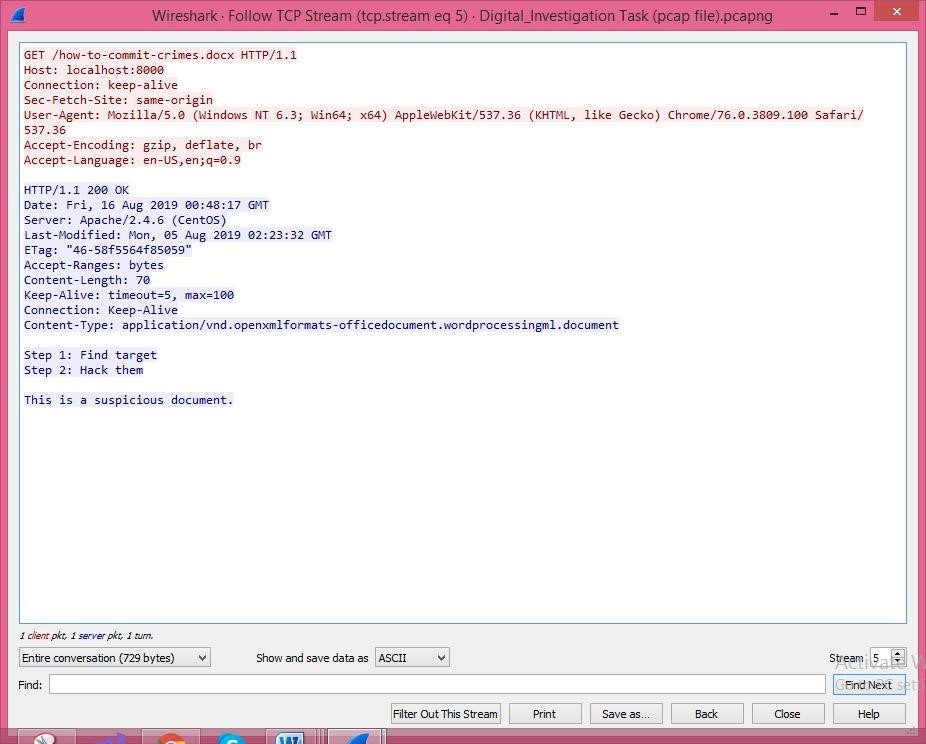
Sub task 3:

In order to view the contents of the document, I had to view the TCP stream of the http get request for the file, the same way I did for before tasks. And The documents contents were visible in the ASCII view.

Step 1: Find target

Step 2: Hack them

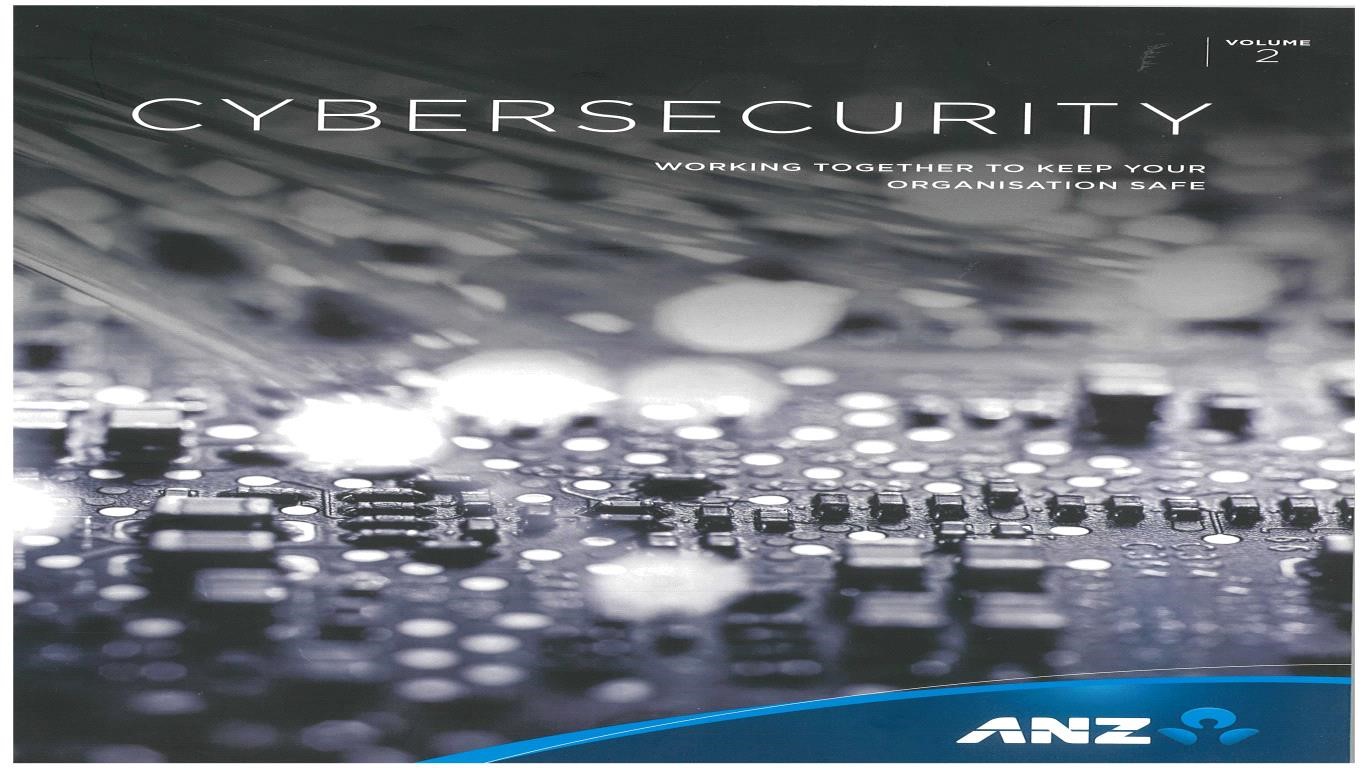
# This is a suspicious document



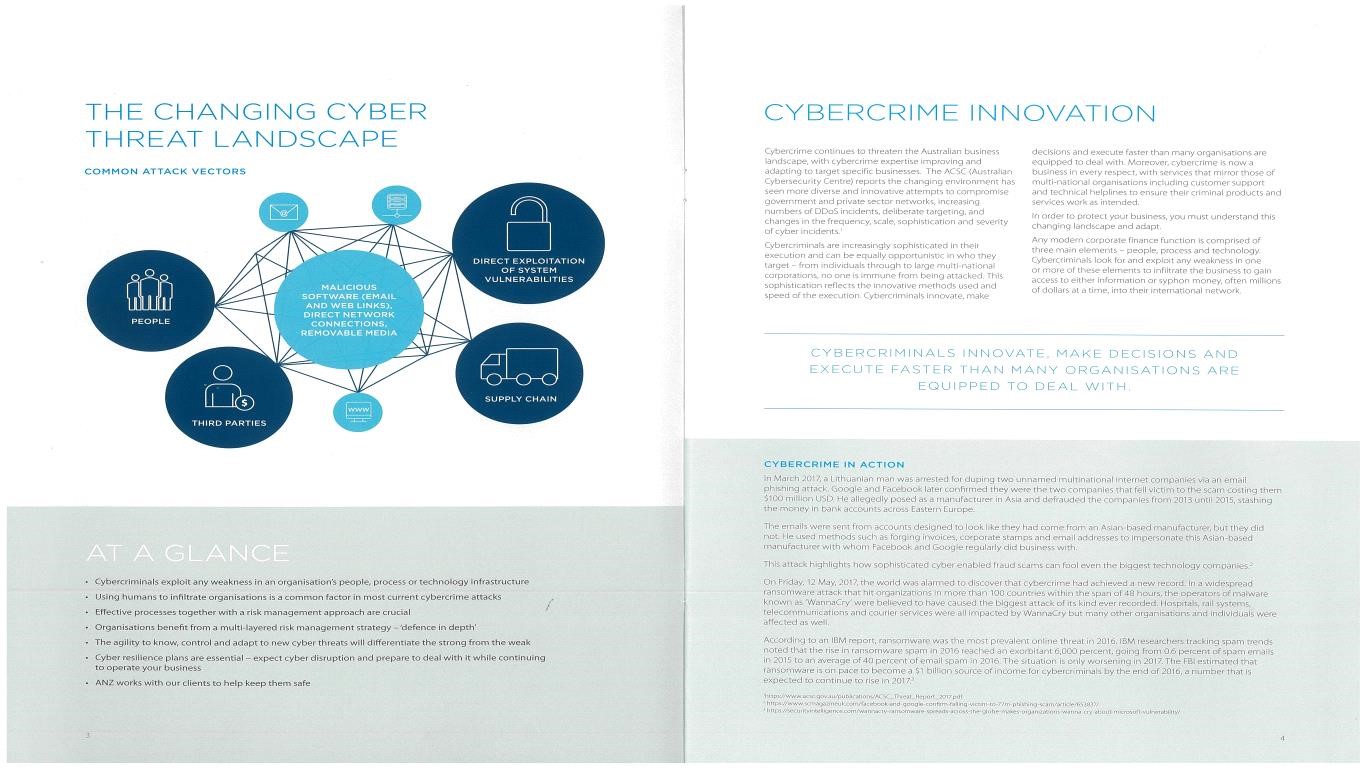
Sub-task 4:

In order to view these PDFs I viewed the TCP stream as usual, and found the file signature for a PDF, which was the hex data “25 50 44 46”. Then, I noticed in the ASCII view of that PDF data until the very end of the TCP stream, therefore, I copied all the hex date from the file signature onwards into HxD and saved it as a pdf file.

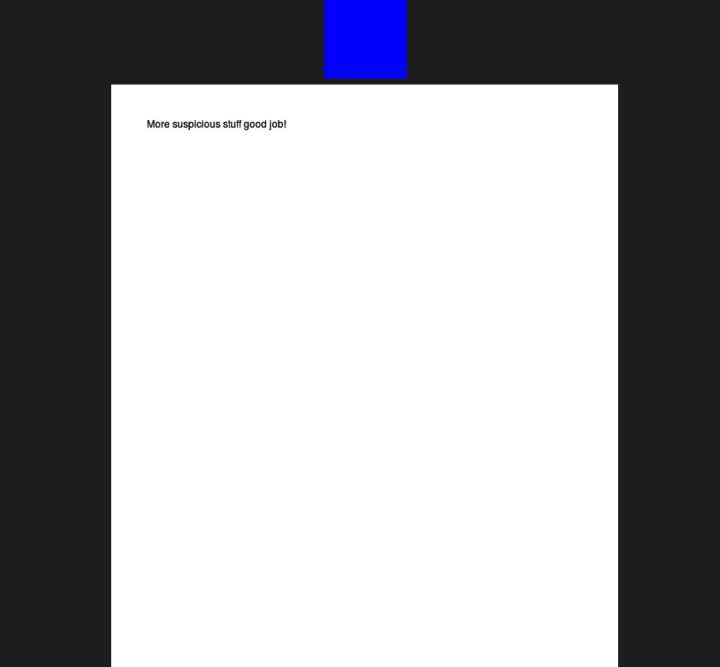
The same step-by-step process worked for all of the three files:



ANZ\_document.pdf (extracted image)



ANZ\_document2.pdf (extracted image)



*Evil.pdf(extracted image)*

Sub-task 5:

I viewed the TCP stream of this file, the same way I did before and noticed that instead of being in plain text it was in encoded data. Then I viewed as hex and it had the same file signature as .jpg image. Thus, I copied and saved the hex data with HxD, and discovered that the text file was actually the below image.



Sub-task 6:

I viewed the TCP stream for this ,while investigating this traffic, and found the two sets of jpeg file signatures present in the 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, that are resulted below:

First image:



Second image:



So the difference about this traffic is that, a single GET request performed by the user resulted in downloading two images.

Sub-task 7:

To find the images the user accessed called “*broken.png”.* I followed the following process for both images:

At First, I filtered the packet capture for http traffic and looked through the remaining packets for the GET request that downloaded the image. And followed to the TCP stream.

In the TCP strea, I observed it is the image data. In order to view the data in hex format, I changed the view to

“raw” and then searched the hex data for .jpeg’s file signature. After finding the file signature “89 50 4e 47 0d 0a 1a 0a” I copied everything after that point till the end and then copy into the hex editor HxD and saved it as a png file. The image is as follow:



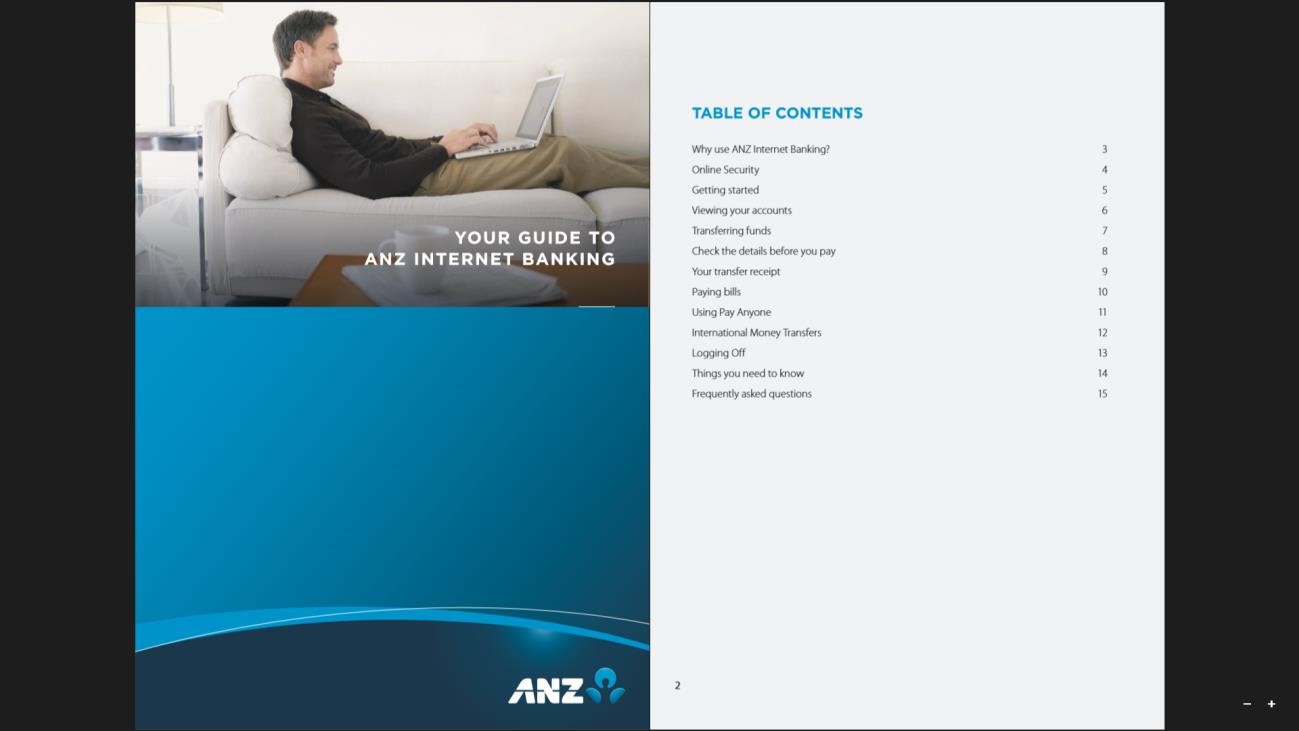
Sub-task 8:

After investigating TCP stream for “securepdf.pdf” I discover the following:

The data was not meant for a PDF, In addition to that, the bottom of the file contained the hidden message: Password is “secure” .

It contained the file signature for a zip file, which actually was a zip file.

Therefore, 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 was worked and the PDF opened. The PDF contains the first two pages to a guide for internet banking.



-K Sai Poornesh.