Sub task 1:

To locate the images, anz-logo.jpg and bank-card.jpg, that the user accessed, I followed the following steps for each image:

1. I filtered the packet capture to isolate the HTTP traffic and examined the remaining packets to find the specific GET request responsible for downloading the image.

2. Next, I right-clicked on the image and selected the option to view its TCP stream. Within the TCP stream, I observed what appeared to be image data.

3. To analyze the data in hexadecimal format, I changed the view to "raw" and proceeded to search for the file signature of a JPEG image within the hex data.

4. Upon locating the file signature, indicated by "FFD8" at the beginning, and the file footer, indicated by "FFD9" at the end, I selected and copied all the data between these two points.

5. Finally, I pasted the copied data into the hex editor software HxD and saved it as a JPG image.

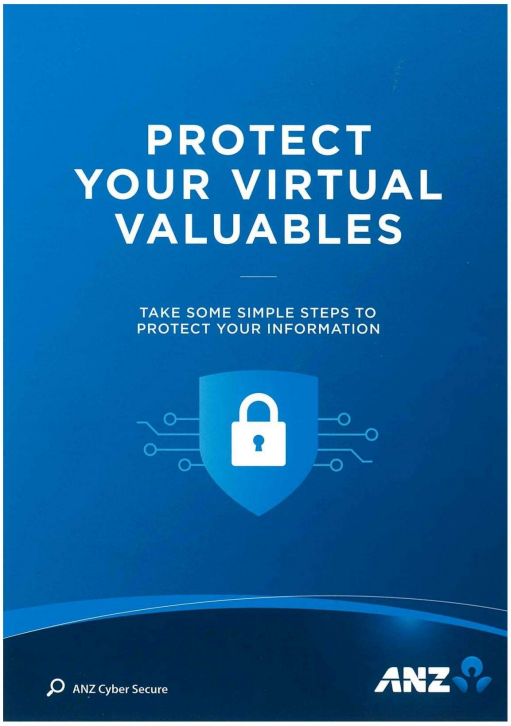
By following this process, I successfully extracted and saved the anz-logo.jpg and bank-card.jpg images from the packet capture.

Bank-logo.jpg

Card-logo.jpg

Sub Task 2:

First, I viewed the TCP stream to identify the hexadecimal data corresponding to the images. Then, I copied and saved that data as a JPG file. Same method as in sub - task 1.



During my investigation of the network traffic related to image downloads, I made an interesting discovery. Upon analyzing the data after the end of the image, I found a concealed message embedded within it.

The message contained the following text: "You've found a hidden message in this file! Include it in your write-up."

This hidden message adds an intriguing element to the investigation and should be included in the final report.

Within this network traffic, there was also a hidden message that was discovered. The message read, "Congratulations! 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.

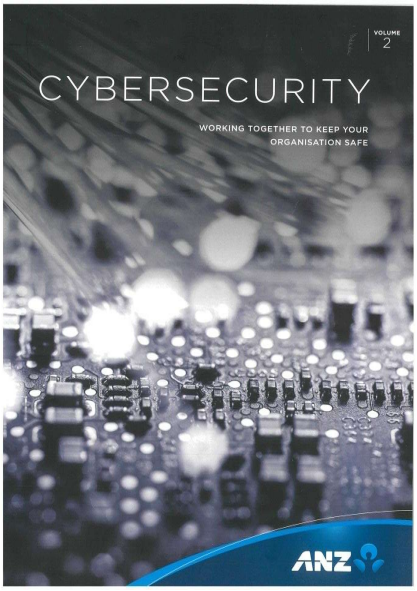
Step 1: Find target

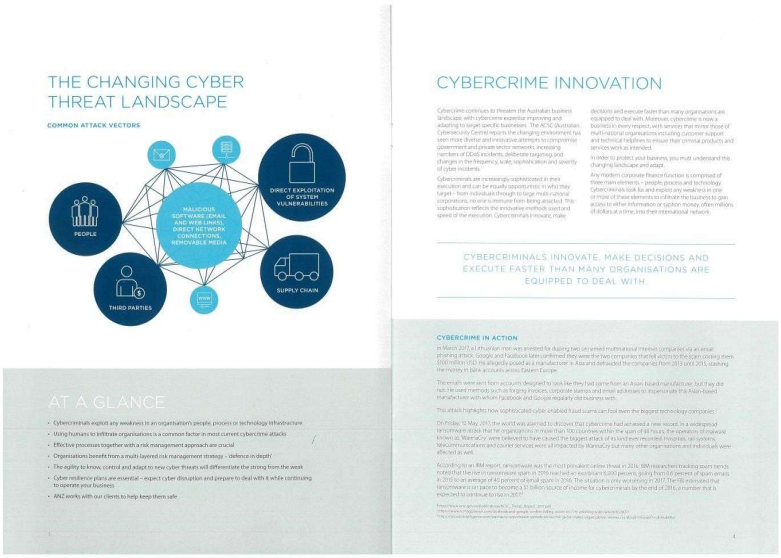
Step 2: Hack them

Sub-task 4:

To access the PDF files, I followed my usual procedure of examining the TCP stream. During this process, I identified the file signature for a PDF, which appeared as the hexadecimal data "25 50 44 46". Upon further observation in the ASCII view, I noticed that the PDF data extended to the end of the TCP stream. To extract the PDF content, I copied all the hexadecimal data from the file signature onwards and used HxD to save it as a PDF file.

These are following extracted images:





Sub-task 5:

I examined the TCP stream of the file and observed that the content was encoded rather than plain text. By viewing it in hexadecimal format, I noticed that it had the same file signature as a JPG image. To investigate further, I used HxD to copy and save the hex data, just as I have done for other images. Surprisingly, the supposed text file turned out to be an image.

Sub-task 6:

During my investigation of this network traffic, I examined the TCP stream and identified two instances of jpeg file signatures. Within the TCP stream, I observed what appeared to be image data. To analyze the data in hexadecimal format, I switched the view to "raw" and searched for the file signature of a jpeg. By locating the file signature "FFD8" at the beginning and the file footer "FFD9" at the end, I selected the data between these two points and saved it as a jpg image using the hex editor HxD.

I attempted to extract both sets of data, resulting in two distinct images. The resulting image is displayed below.


ANZ


shutterstock.com • 567329461

Sub-task 7:

I first filtered the packet capture to isolate the http traffic. Then, I examined the remaining packets to locate the specific GET request responsible for downloading the image. By right-clicking on the image, I accessed the TCP stream associated with it. Within the TCP stream, I observed what appeared to be image data.

To view the data in hexadecimal format, I changed the view to "raw" and proceeded to search the hex data for the file signature of a JPEG image, which is "89 50 4e 47 0d 0a 1a 0a." Once I identified the file signature, I copied everything after that point until the end of the data.

Next, I used the hex editor software, HxD, to paste the copied data and saved it as a PNG image.

The resulting image is folllowing:

Sub-task 8:

Upon investigating the TCP stream for the file "securepdf.pdf," I made the following discoveries:

1. The data within the file did not match the format of a PDF. Towards the end of the file, there was a hidden message indicating the password as "secure."

2. The file had the file signature of a zip file, suggesting that the user actually downloaded a zip file instead.

To proceed, I copied the hexadecimal data of the zip file into an application called HxD and saved it as a zip file. Upon opening this newly created zip file, I found a PDF file named "rawpdf.pdf." When attempting to open the PDF, it prompted me for a password. Using the password "secure" mentioned in the TCP stream, I successfully accessed the PDF. It turned out to be the first two pages of a guide related to internet banking.