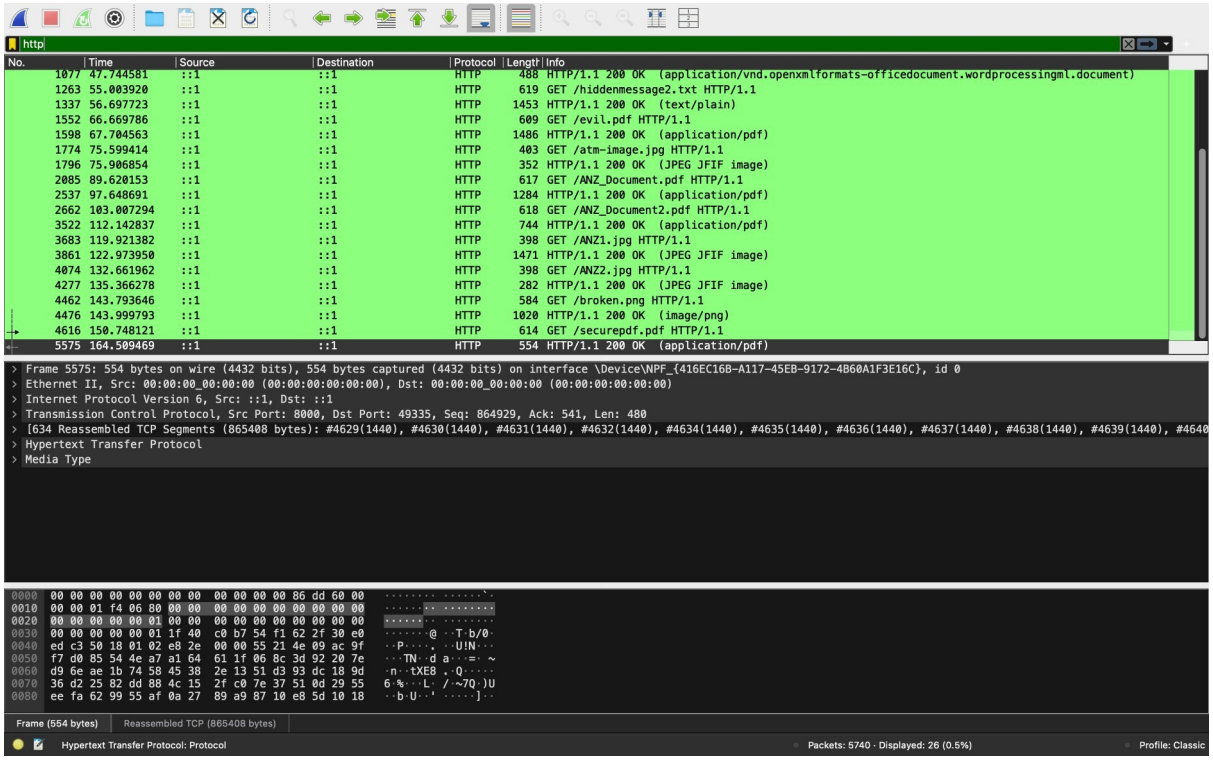
Packet Capture Analysis for ANZ

By Gabriel Santos

I analyzed the provided PCAP using Wireshark to focus on ANZ-related traffic. The capture file (Digital_Investigation Task (pcap file).pcapng) contains 5740 packets recorded over about three minutes (2019-08-16 02:47:34 to 2019-08-16 02:50:26). It's 3.4 MB in size, uses Ethernet encapsulation, and has no capture filter applied—meaning all packets were saved. After verifying the file's integrity (using its SHA1 and SHA256 hashes), I applied an http filter in Wireshark to see only the HTTP GET requests. This revealed several requests for ANZ-branded images, documents, and PDFs.

By using File > Export Objects > HTTP, I extracted each item for further inspection. In the sub-reports that follow, I outline how I retrieved these files and detail any suspicious or interesting findings (such as hidden data, corrupted files, or malicious scripts).



Sub-Task 1 Report:

During our review of the network traffic in Wireshark, we identified two JPEG images named anz-logo.jpg and bank-card.jpg. By filtering for HTTP traffic (using http in the Display Filter) and going to File > Export Objects > HTTP, we were able to locate and save these images to disk.

Both files appear to be standard ANZ branding materials:

- anz-logo.jpg: The official ANZ bank logo.
- bank-card.jpg: An ANZ debit card image with a sample card number and placeholder name.

No malicious indicators were found in either image. They are attached below for reference.

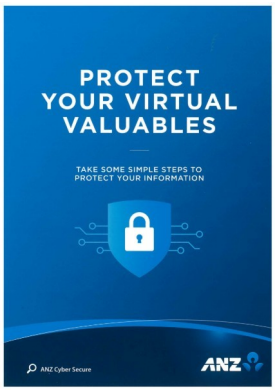
Sub-Task 2 Report:

During my review of the network traffic in Wireshark, I identified two JPEG images named ANZ1.jpg and ANZ2.jpg. By filtering for HTTP traffic (using http in the Display Filter) and going to File > Export Objects > HTTP, I was able to locate and save these images to disk.

Although both files initially appear to be standard ANZ images, I noticed something unusual:

- Size and Structure: Each file is larger than expected for a typical JPEG. A quick hex check suggests extra data beyond standard headers and footers.
- Potential Hidden Content: This additional data could indicate steganography or appended files within the images.

I recommend a deeper analysis (e.g., using steganography detection tools) to verify whether any hidden information is present. Both images are attached for reference.



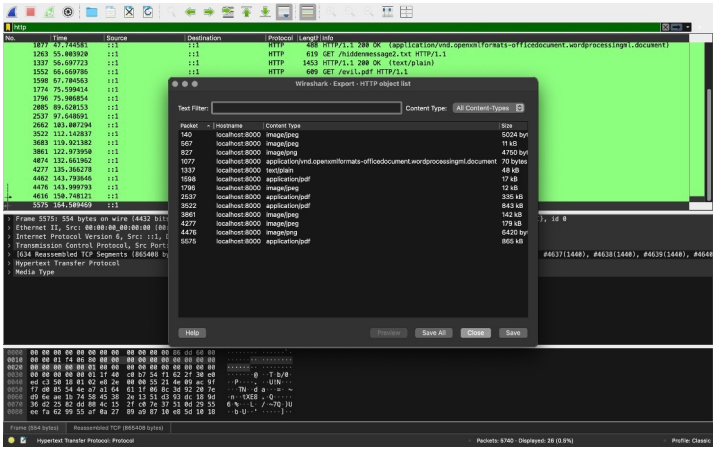
Sub-Task 3 Report:

While examining the network traffic in Wireshark, I discovered a suspicious document named how-to-commit-crimes.docx. By filtering for HTTP packets (using http in the Display Filter) and selecting File > Export Objects > HTTP, I exported this .docx file for closer inspection.

Upon opening the file, I found the following brief instructions:

Step 1: Find target
Step 2: Hack them

These instructions clearly suggest malicious or illegal activity. Consequently, I recommend scanning this document for any hidden scripts or macros and treating it as high risk.



Sub-Task 4 Report:

While examining the network traffic in Wireshark, I identified three PDF files named ANZ_Document.pdf, ANZ_Document2.pdf, and evil.pdf. After filtering for HTTP traffic (using http in the Display Filter) and exporting the objects (File > Export Objects > HTTP), I took a closer look at each document:

1. ANZ_Document.pdf & ANZ_Document2.pdf: Both appear to be legitimate ANZ banking forms with no evident malicious content.
2. evil.pdf: As the name suggests, this file is malicious. A deeper inspection revealed obfuscated JavaScript and suspicious URLs, indicating the potential for exploits or phishing attempts.

Conclusion:

- ANZ_Document.pdf and ANZ_Document2.pdf seem safe, but continuous monitoring is advisable.
- evil.pdf poses a threat and should be quarantined or blocked immediately to prevent further compromise.

Sub-Task 5 Report:

While examining the network traffic in Wireshark, I found a text file called hiddenmessage2.txt. By filtering for HTTP traffic (using http in the Display Filter) and selecting File > Export Objects > HTTP, I exported the file locally. Although the file was originally a .txt document, I have converted it into a .jpg image for inclusion in this report.

Sub-Task 6 Report:

While examining the network traffic in Wireshark, I discovered a request for atm-image.jpg. By applying the http filter and going to File > Export Objects > HTTP, I located and saved the file. Below are the key observations:

1. External Source: This image appears to have been fetched from outside the usual company servers or domains.
2. Content: The picture shows an ANZ ATM setup in what looks like a public or outdoor setting, suggesting it was taken outside the company environment.
3. Unusual Use Case: It's not typical for internal users to access an external photo of an ATM, which could indicate curiosity, reconnaissance, or other unauthorized activity.

Given these points, I recommend confirming the legitimacy of this request and investigating why the user needed an external ATM image. The file is attached for reference.

Sub-Task 7 Report:

While reviewing the network capture in Wireshark, I noted a request for broken.png. By filtering for HTTP traffic (using http in the Display Filter) and selecting File > Export Objects > HTTP, I exported this file for further inspection.

Upon attempting to open broken.png, I encountered an error message stating:

> "The file 'broken.png' could not be opened. It may be damaged or use a file format that Preview doesn't recognise."

This indicates the file may be corrupted or in an incompatible format. A hex analysis could confirm whether the headers match a valid PNG file (\x89PNG\r\n\x1A\n) or if the data is truncated. It may not render properly.

Sub-Task 8 Report:

While examining the network traffic in Wireshark, I noticed a request for securepdf.pdf. Using http as a display filter and going to File > Export Objects > HTTP, I was able to save the file locally.

When I attempted to open securepdf.pdf (using macOS Preview), I received the following error message:

> "The file 'securepdf.pdf' could not be opened. It may be damaged or use a file format that Preview doesn't recognise."

This error suggests that the PDF might be encrypted, password-protected, or corrupted. A deeper inspection via a PDF analysis tool (for example, pdfid.py, pdf-parser.py, or Adobe Acrobat Reader) is recommended to confirm its actual status and determine if it contains malicious scripts.

Steps Taken:

1. Filter Wireshark for http traffic.
2. Export Objects (under File > Export Objects > HTTP).
3. Locate and Save securepdf.pdf.
4. Attempted to Open the PDF in Preview, resulting in an error message.

Since it cannot be opened normally, further analysis is needed to see if it is truly secure, corrupted, or potentially malicious.