Data Exfiltration Techniques

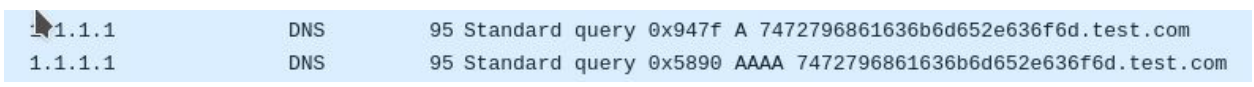
Data Exfiltration is the technique of transferring unauthorized data out of the network, this is typically why companies have Data Loss Prevention systems in place - to prevent data exfiltration. Unfortunately, Data Loss Prevention systems aren't perfect and they can allow data that is classified to leave the network. This is where the Security Operations Center comes in.

Part of the job of a SoC Tier 1 Analyst is determine if an attack had actually occurred, this can be identified by many means. An easy way to tell is to review wireshark logs from the network and search for data exfiltration techniques. It's a lot easier to spot data that you know shouldn't be leaving the network than to determine if an APT (Advanced Persistent Threat) is inside. APTs can often be hidden deep inside the network, often abusing Kerberos and it's Ticket Granting System to gain permanent access.

Some Data Exfiltration techniques include:

- DNS
- FTP/SFTP based file transfer
- SMB based file transfer
- HTTP/HTTPS
- Steganographical methods, like hiding data within images
- ICMP
- And many more. The sky's the limit for Exfiltration methods.

So how do we identify Data Exfiltration attempts? It can be incredibly simple with Wireshark as long as it is not occuring over an encrypted protocol. DNS is the single most common technique used in Data Exfiltration, mainly because it blends in with normal traffic. DNS is used on just about every single device on your network, and it can be incredibly difficult to determine malicious traffic. An attacker, for example, could have their website setup, listening for data on any subdomain other than www and record it to a file to later examine the contents



An example of using DNS to transfer Hex encoded data.

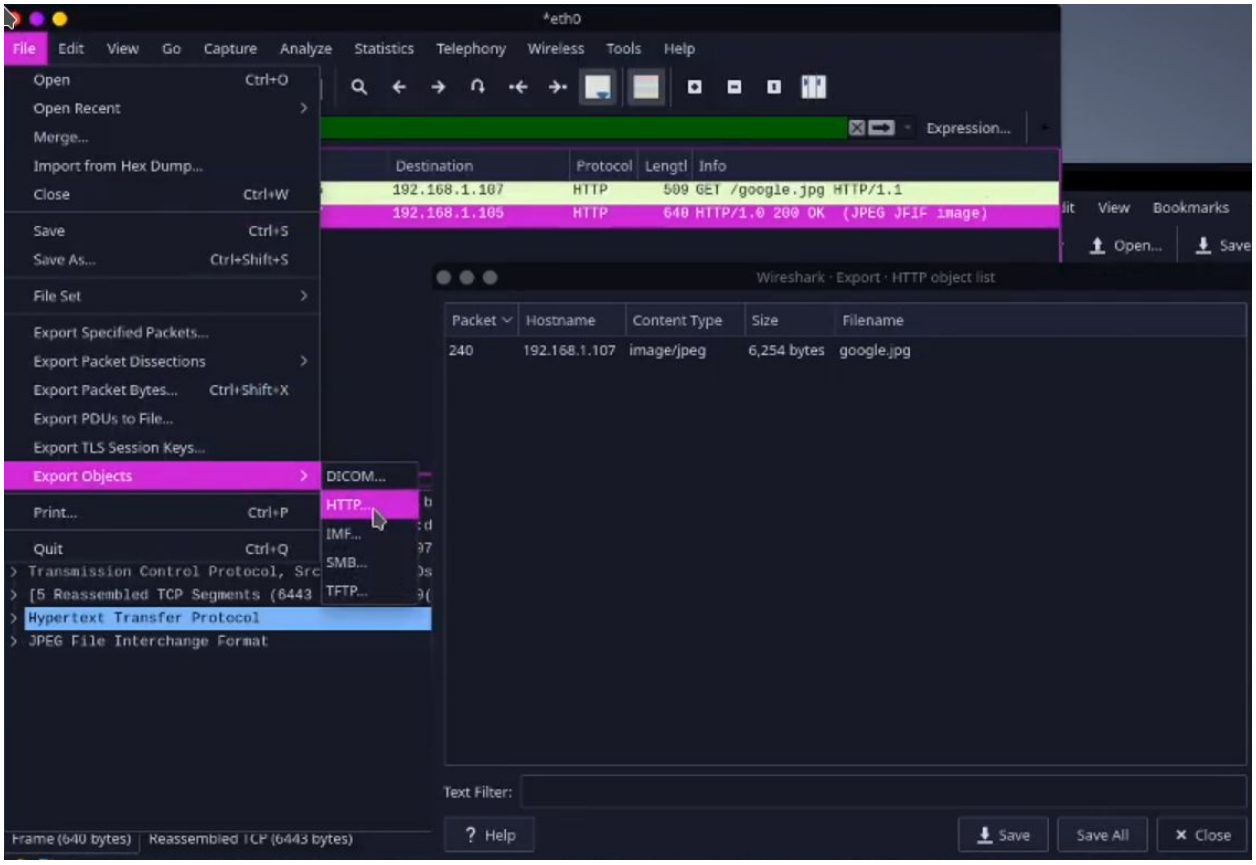
As you can see, this traffic can be easy to spot in large quantities as they are completely random and it can be semi-easy to spot.

Another technique is using photos to transfer data via methods of Steganography. The attacker may embed a hidden file within an image using a utility like Steghide [**Note: Must be installed on Kali]** (which we can also use to extract the data). If we noticed that a specific IP has requested a specific seemingly random image (like Google's logo on a website that is not google), we may choose to investigate.

Time	Source	Destination	Protocol	Lengtl	Info
3.434522069	192.168.1.105	192.168.1.107	HTTP	509	GET /google.jpg HTTP/1.1
3.445583838	192.168.1.107	192.168.1.105	HTTP	640	HTTP/1.0 200 OK (JPEG JFIF image)

Above is a client requesting google.jg from a Private IP Address

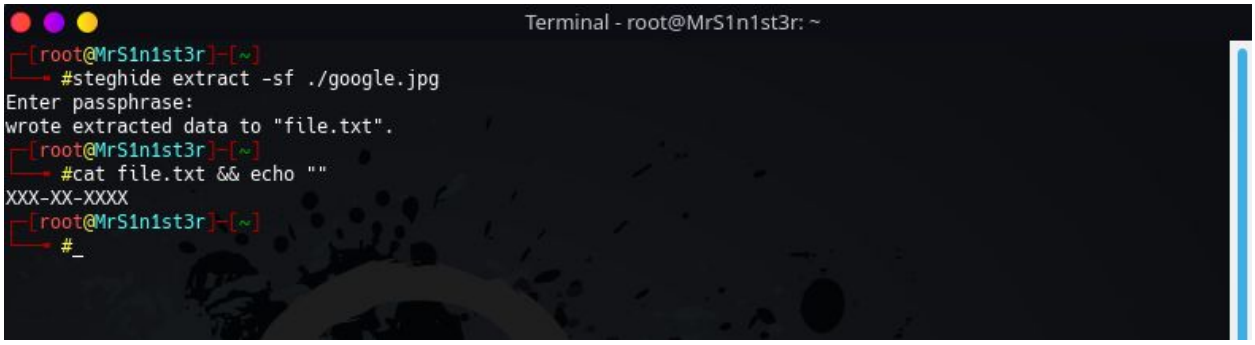
We can actually extract the contents of the file using Wireshark by going to:



File -> Export Objects -> HTTP -> google.jpg

We can then save the file for further forensics. Now we can attempt to extract any potential contents of the file with Steghide. To do so, we can execute:

steghide extract -sf ./<file to extract>.jpg

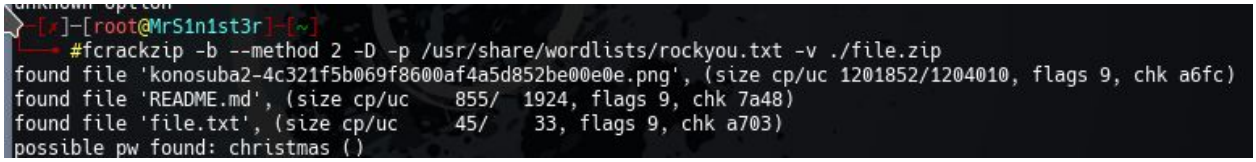


Above shows the extraction of sensitive social-security like data with Steghide.

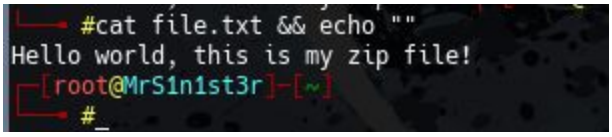
A similar process can with all sorts of file types. As long as the file over HTTP, TFTP, FTP, or SMB, the data can be extracted from the packet capture. Some attackers may attempt to obfuscate the data further by placing the data in an encrypted zip file. Luckily for us, there are tools like **fcrackzip (Note: fcrackzip is not preinstalled on Kali)** that allow us to brute force zip files. The syntax is as follows

fcrackzip -b --method 2 -D -p /usr/share/wordlists/rockyou.txt -v ./file.zip

-b specifies brute forcing, --method 2 specifies a Zip file, -D specifies a Dictionary and -V verifies the password is indeed correct.



Above depicts the successful brute forcing of a zip file



Reading the contents of the file, we can see it is just a standard text file with several items. Not everything you find will always be malicious!

Made with love for TryHackMe, By Sq00ky <3