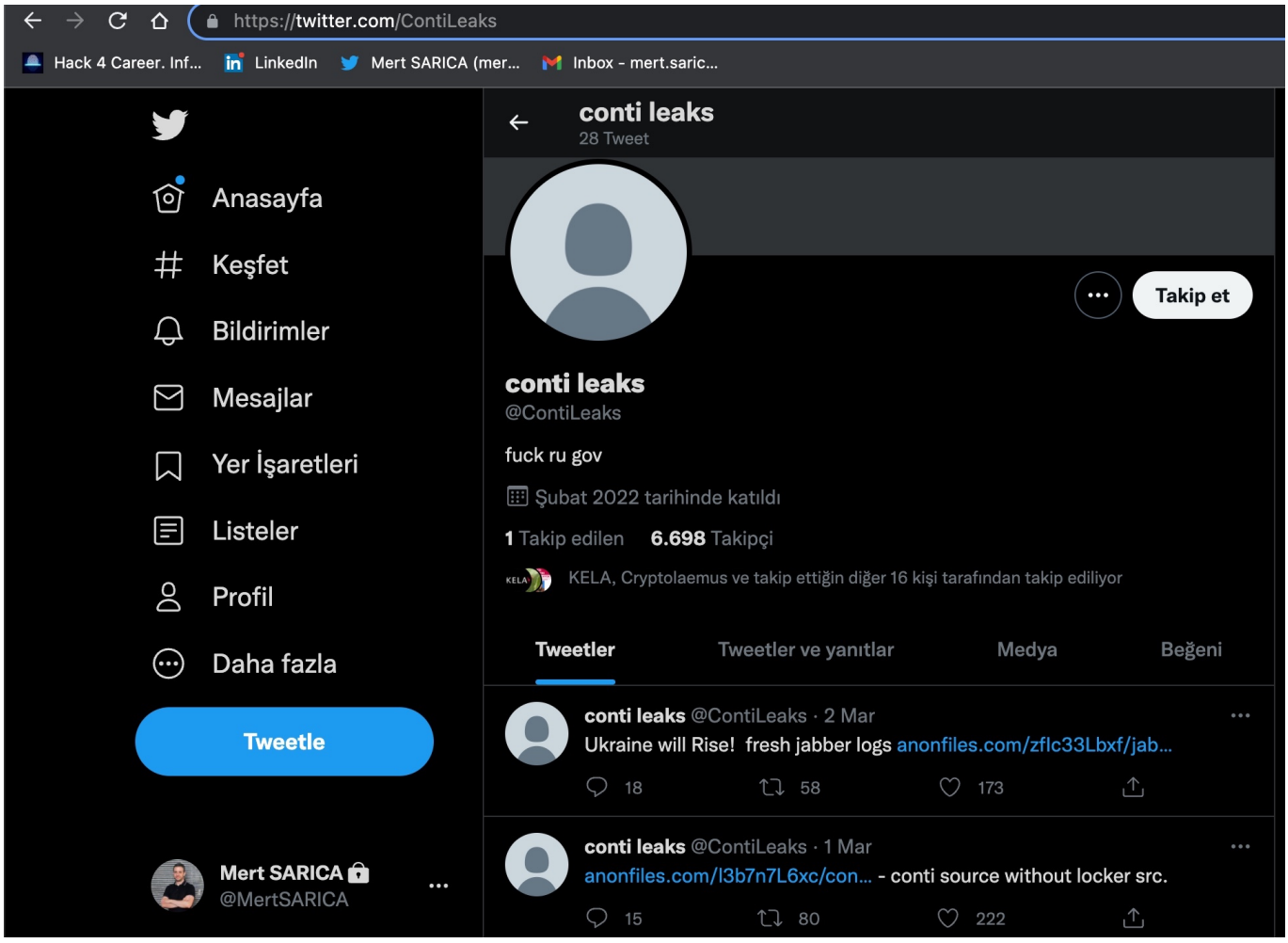
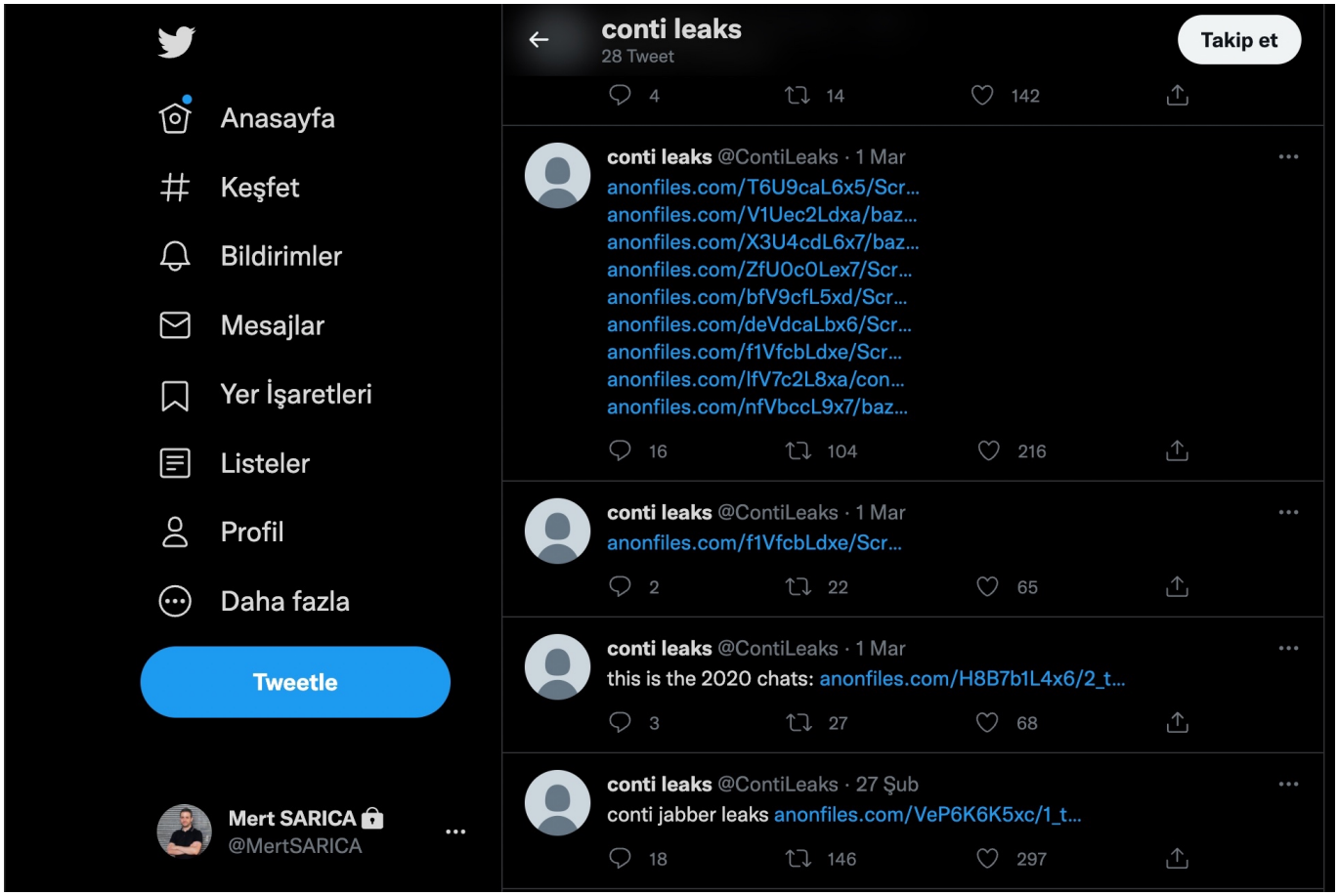
Practical Data Leakage Analysis

written by Mert SARICA | 1 June 2023

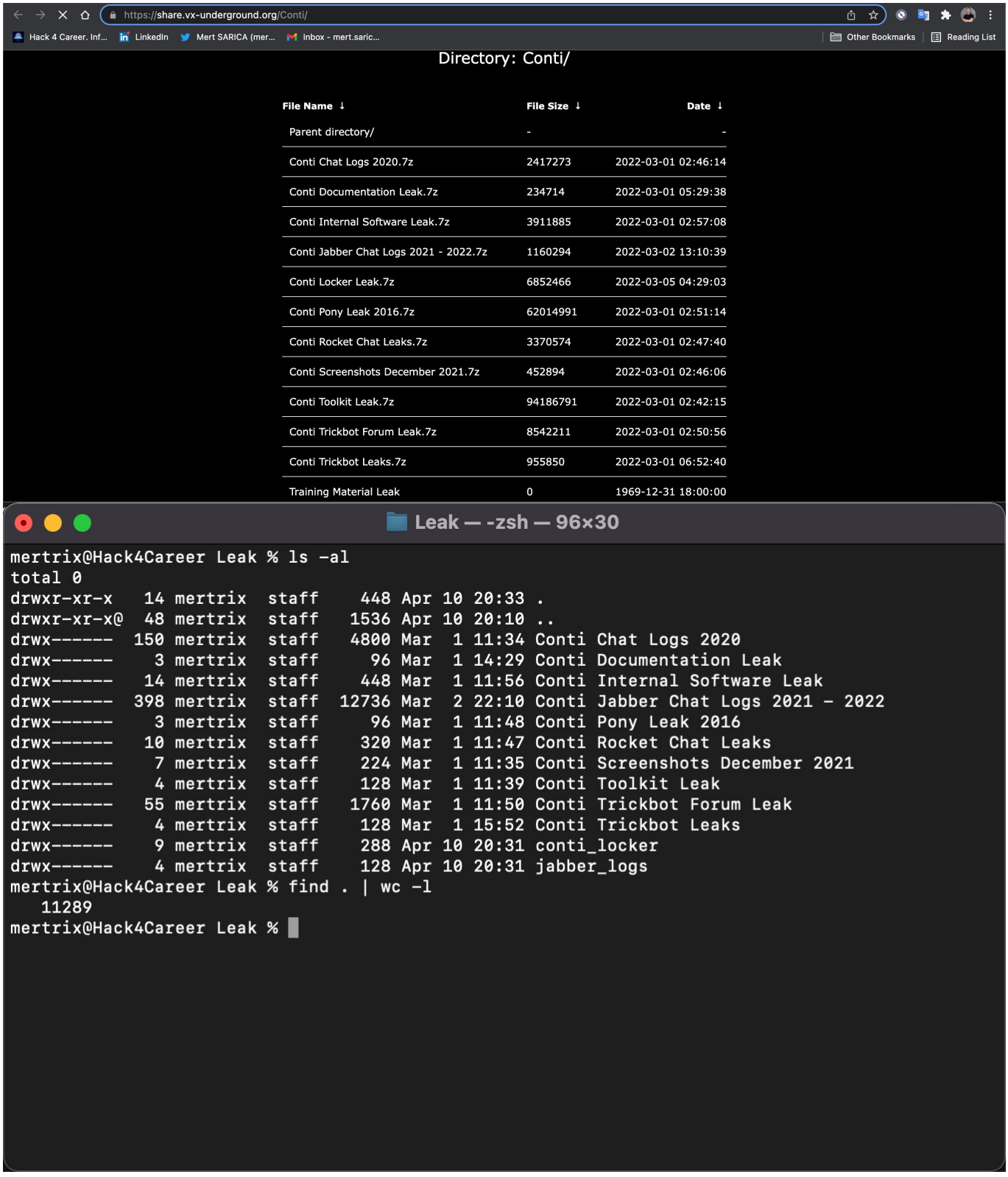
Conti, a Russian-backed cybercrime group that earned $180 million in revenue from ransomware attacks in 2021, reached a major turning point in 2022 with Russia’s invasion of Ukraine. The group publicly supported the Russian invasion, resulting in a rift among its international members. One member began leaking internal messages from 2020-2021 on a Twitter account (@ContiLeaks), including the source code for the ransomware they used in their cyberattacks. The group was considered one of the most notorious cybercrime groups in the world.





As a cybersecurity researcher, when data from such threat actors is leaked, one of the things that interests me the most is whether the data includes information about hacked organizations in Turkey, as well as non-Russian, English messages. If you ask me why, it’s because I can have the opportunity to learn how extensively Turkey is targeted by these threat actors and which nationalities are involved in such internationally organized crime groups. To find out, I decided to conduct cybersecurity research to provide insights to cybersecurity researchers who are also interested in this topic.

First, I downloaded the files that include the Conti group’s messages from the sharing area of the vx-underground website. When I extracted all the zip files, more than 11,000 files came out.

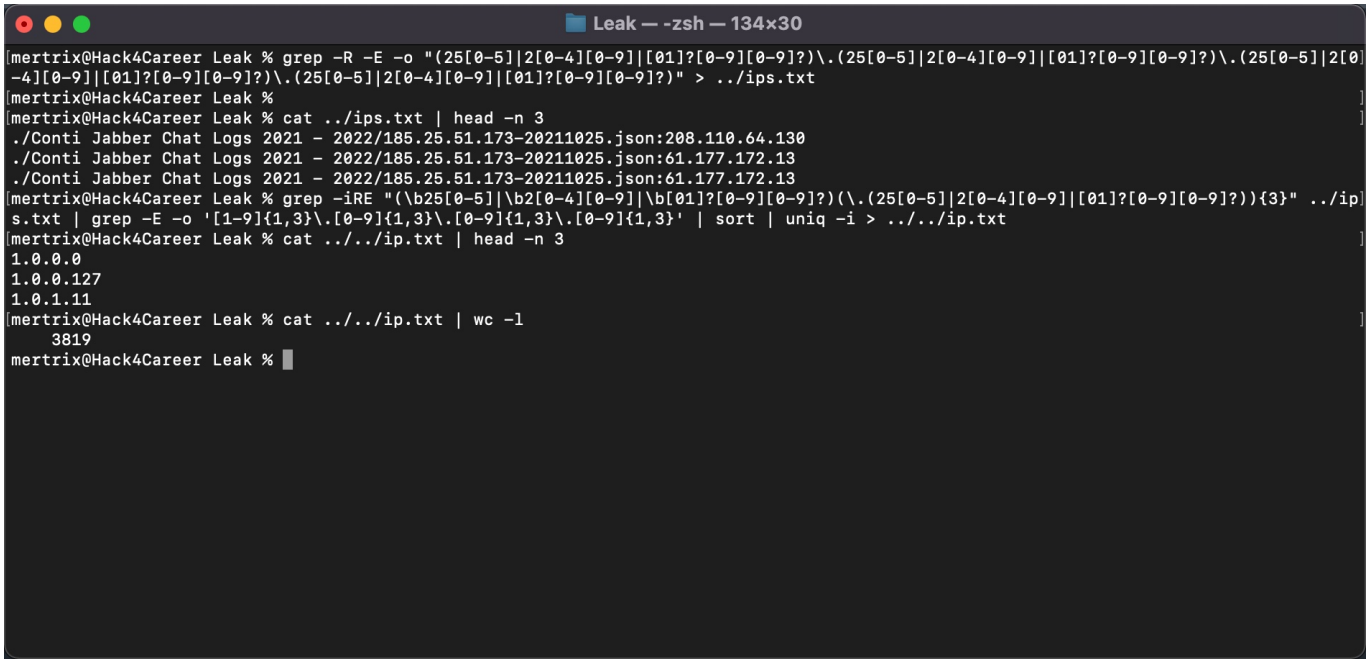


After learning that the messages are stored as readable text in JSON files (Example: 185.25.51.173-20220301.json), my first task was to use the following regex-supported GREP command to find and deduplicate all IP addresses in the files. I ended up with a total of 3819 IP addresses that match these two regex patterns, which I saved in a file named “ip.txt.”

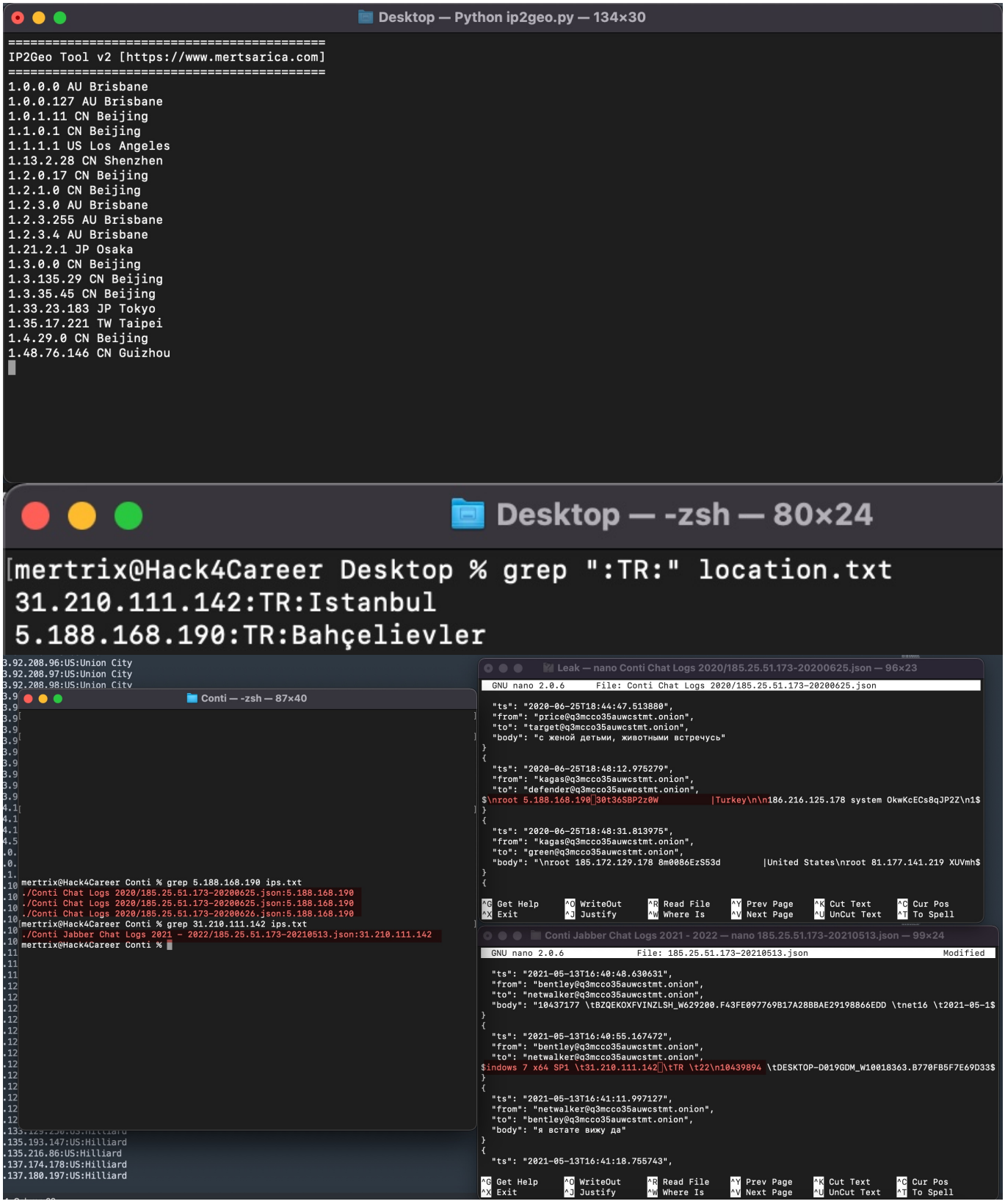
```
grep -R -E -o
```

```
"(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)" > ../ips.txt
```

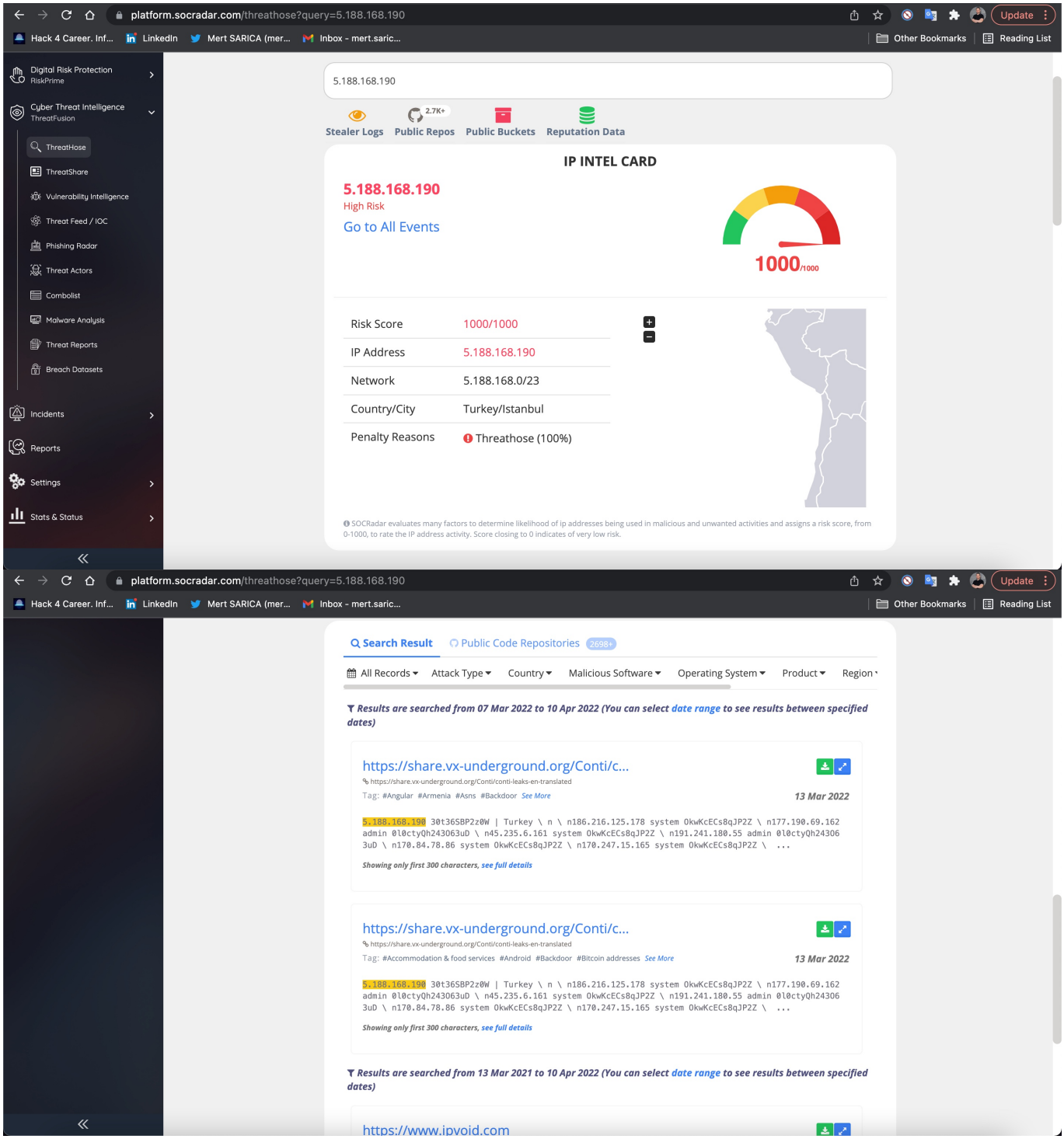
```
grep -iRE "(\b25[0-5]|\b2[0-4][0-9]|\b[01]?[0-9][0-9]?)(\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)){3}" ../ips.txt | grep -E -o '[1-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}' | sort | uniq -i > ../../ip.txt
```



When it came to finding out which of these IP addresses belong to Turkey, I found help in the IPinfo API and its Python library. By using this library with the IP2Geo Tool v2 that I developed, I queried all the IP addresses in my possession (ip.txt), and I learned that two of these IP addresses (31.210.111.142, 5.188.168.19) are located in Turkey.



To validate the results I obtained, I took a look at SOCRadar, Extended Cyber Threat Intelligence Platform that provides real-time notifications to organizations regarding data breaches. I found that the results aligned with what I discovered earlier, thus clarifying my initial curiosity. :)



When it came to my curiosity about the other topic, I decided to explore Python libraries capable of language detection from text. After a brief research, I came across several prominent libraries in this field, including fastText, langdetect and langid

While testing the libraries individually on the text from the leaked Conti data, I observed that each library made accurate language detections for some texts but produced incorrect results for others. As I pondered over which library to use, I decided to develop a tool that combines all three libraries and allows users to specify the confidence level parameter according to their

needs and preferences. This approach would provide a more reliable way to determine the language in a customizable manner.

After merging the leaked Conti data into a single file using the command find . -type f -print -exec cat {} \; > ../logs.txt, I used the Language Identification tool I developed to check each line in the “logs.txt” file for Turkish language detection using the three libraries (with the confidence level set to “High”).

To use the Language Identification tool, you need to provide the following parameters.

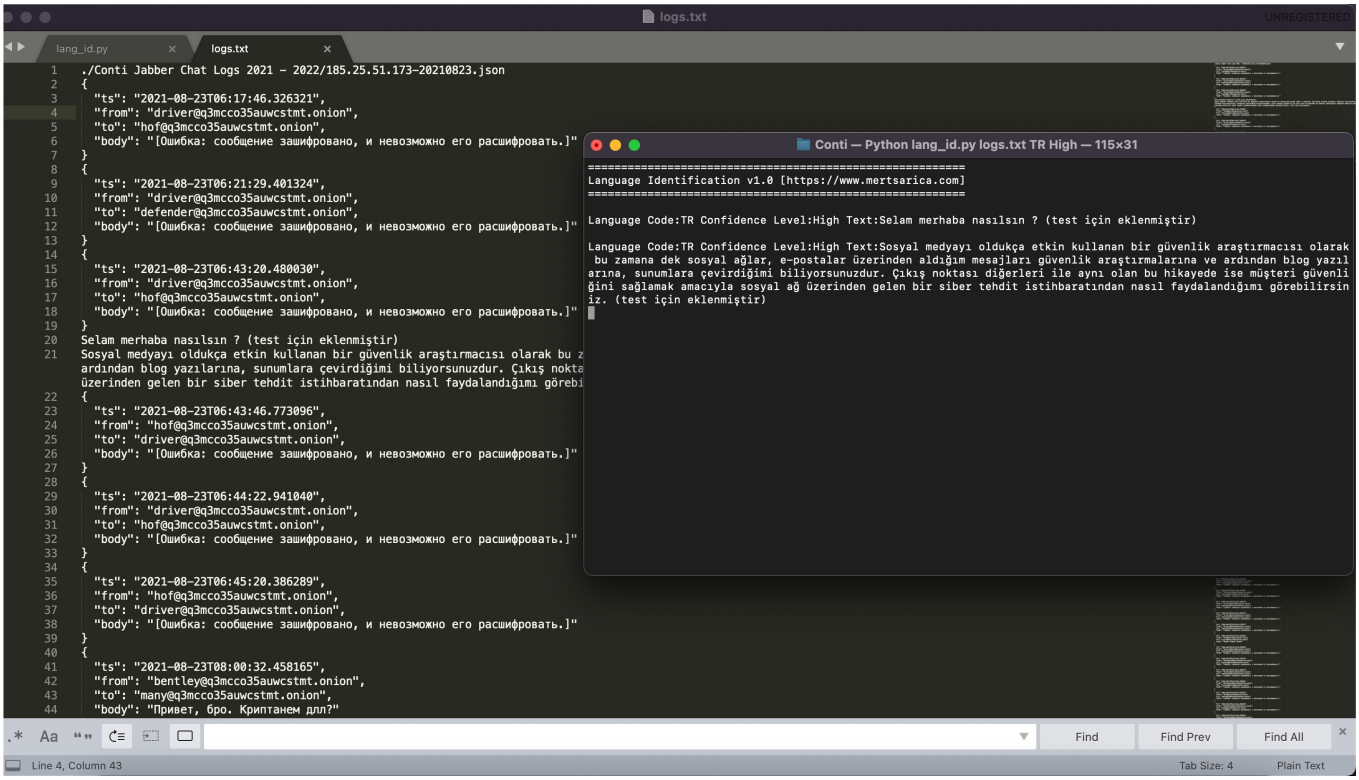
1. The first parameter is the text file you want to analyze, specifying it line by line.
2. The second parameter is the language code for the language you want to detect (e.g., “TR” for Turkish, “EN” for English).
3. The optional third parameter determines the confidence level. If you set it to “High,” when all three libraries detect the language code you specified, it will indicate it on the screen.

Here’s an example command using the tool:

python3 lang_id.py logs.txt TR High

This command will analyze each line in the “logs.txt” file for Turkish language detection with a high confidence level.

Since there were no Turkish words or sentences used in the text files, there was no language detection indicating the usage of Turkish language by any of the three libraries. However, to test the tool’s functionality, I added three fake Turkish texts to the “logs.txt” file. As a result, I successfully observed that the program detected them correctly. Through this analysis, I learned from the leaked Conti data that there was no Turkish conversation among the group members, thereby clarifying my final curiosity.



I hope this method I have followed and the two tools I have developed will be beneficial for security researchers and experts in data leakage analysis.

Hope to see you in the following articles.