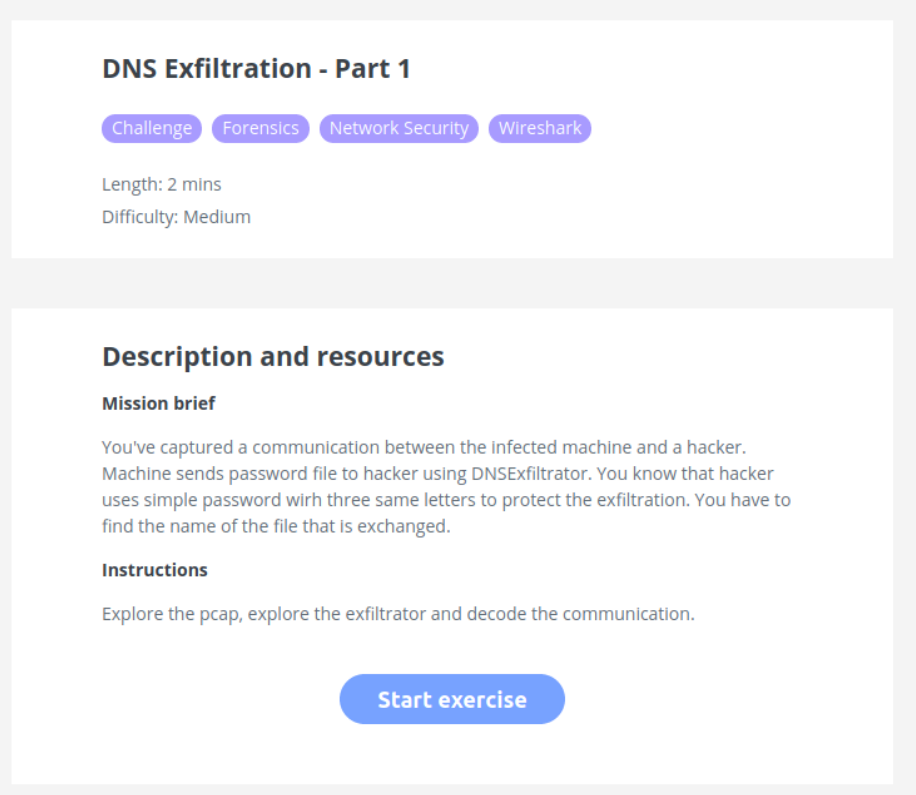
# DNSExfiltrator Challenge Writeup

Writeup for DNSExfiltrator challanges from Serbian Cybersecurity Challenge 2021.

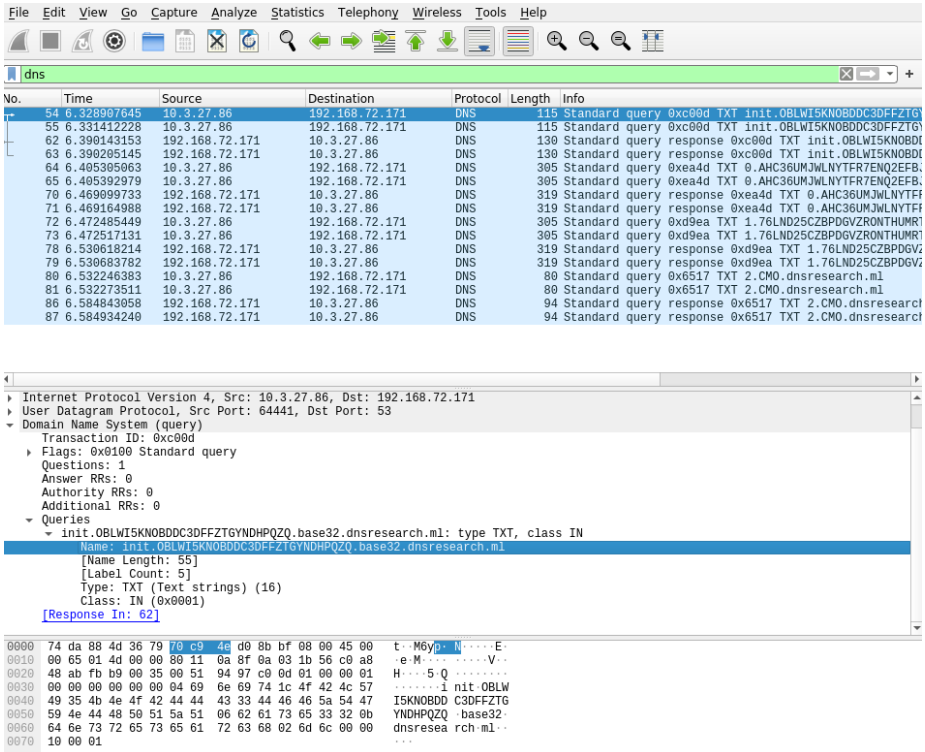
## Part 1




---

**Solution:**

We got *challengeDNS_anon.pcapng* file, and if we open it with Wireshark and apply **dns** filter we can see that first packet has name `init.OBLWI5KNOBDDC3DFFZTGYNDHPQZQ.base32.dnsresearch.ml`





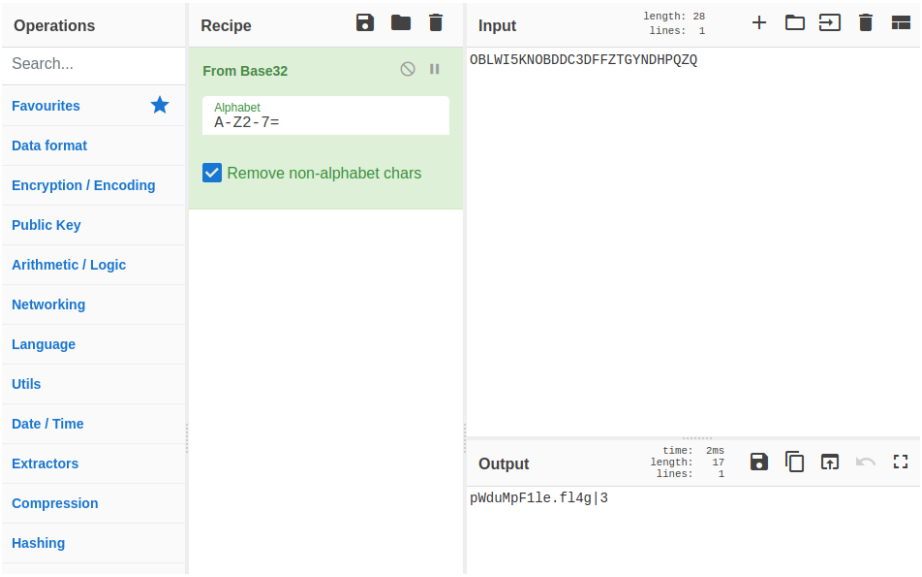


We can run next command to decode given string:

```
echo "OBLWI5KNOBDDC3DFFZTGYNDHPQZQ====" | base32 -d
```

And we got the flag for the first part, name of the file that is exchanged is **pWduMpF1le.fl4g**

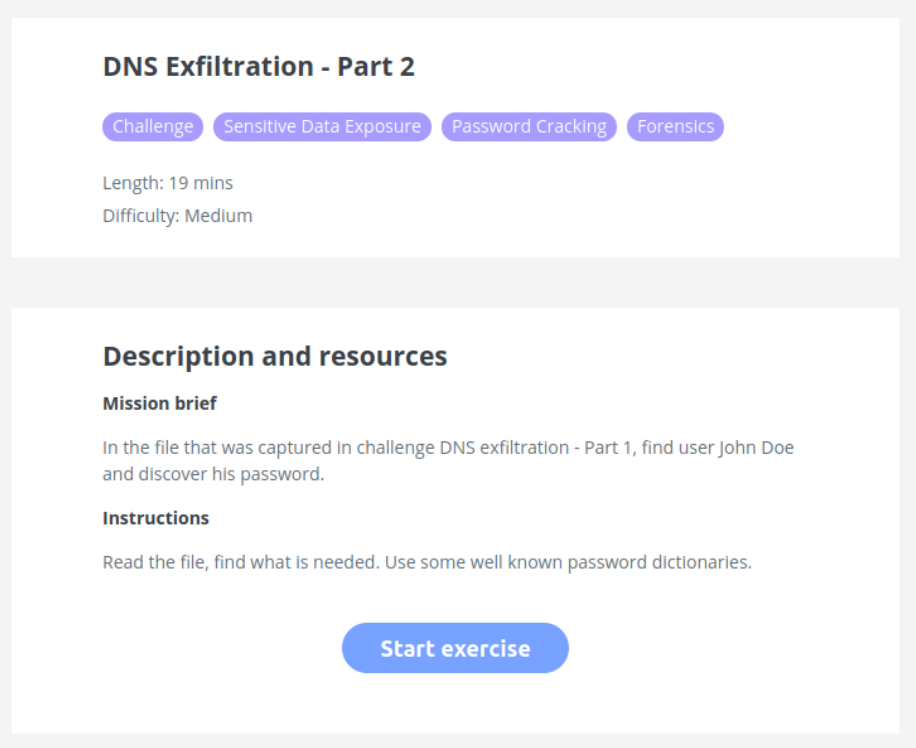
Alternatively, we can use CyberChef or any other online tool that can decode base32:




If someone would like to see how this part can be solved using Python, you can take look at file solve_part_1.py

---

## Part 2




---

**Solution:**

To solve the second part, we need to go back to description of part 1, and find out how DNS exfiltration works. We can find source code of DNSExfiltrator tool on github:

```
#---- Have we received all chunks of data ?
if chunkIndex == nbChunks:
        print '\n'
        try:
                # Create and initialize the RC4 decryptor object
                rc4Decryptor = RC4(args.password)

                # Save data to a file
                outputFileName = fileName + ".zip"
                print color("[+] Decrypting using password [{}] and saving to output file [{}]".format(args.password,outputFileName))
                with open(outputFileName, 'wb+') as fileHandle:
                        if useBase32:
                                fileHandle.write(rc4Decryptor.binaryDecrypt(bytearray(fromBase32(fileData))))
                        else:
                                fileHandle.write(rc4Decryptor.binaryDecrypt(bytearray(fromBase64URL(fileData))))
                        fileHandle.close()
                        print color("[+] Output file [{}] saved successfully".format(outputFileName))
        except IOError:
                print color("[!] Could not write file [{}]".format(outputFileName))
```

Here we can see, that on receiving side the data is firstly decoded from base32, then decrypted with RC4, and at the end we write data into file with .zip extension.

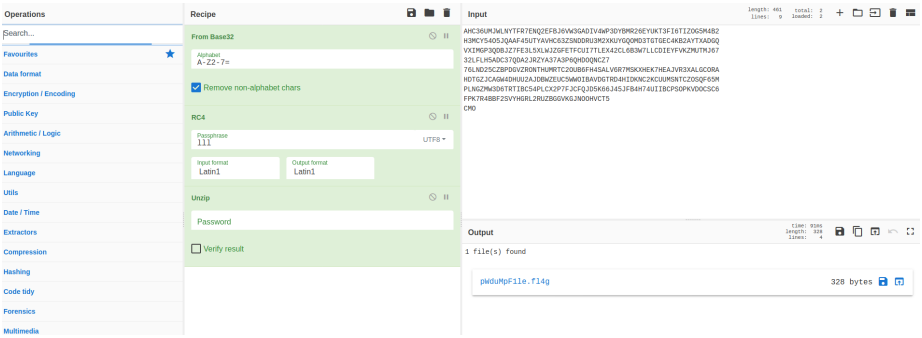
So to solve the challange we need to extract base32 encoded data from packets. Apply filter: **dns && ip.dst == 10.3.27.86**, and go to File -> Export Packet Dissections -> As Plain Text, from there we can extract base32 encoded strings, and join them into one:

```
AHC36UMJWLNYTFR7ENQ2EFBJ6VW3GADIV4WP3DYBMR26EYUKT3FI6TIZOG5M4B2
H3MCY54O5JQAAF45UTYAVHC63ZSNDDRU3M2XKUYGQOMD3TGTGEC4KB2AYTXADGQ
VXIMGP3QDBJZ7FE3L5XLWJZGFETFCUI7TLEX42CL6B3W7LLCDIEYFVKZMUTMJ67
32LFLH5ADC37QDA2JRZYA37A3P6QHDOQNCZ7
76LND25CZBPDGVZRONTHUMRTC2OUB6FH4SALV6R7MSKXHEK7HEAJVR3XALGCORA
HDTGZJCAGW4DHUU2AJDBWZEUC5WWOIBAVDGTRD4HIDKNC2KCUUMSNTCZOSQF65M
PLNGZMW3D6TRTIBC54PLCX2P7FJCFQJD5K66J45JFB4H74UIIBCPSOPKVDOCSC6
FPK7R4BBF2SVYHGRL2RUZBGGVKGJNOOHVCT5
CMO
```

Tshark is a terminal oriented version of Wireshark, so we also can use this tool to extract data from file:

```
tshark -r challengeDNS_anon.pcapng -Tfields -e dns.qry.name ip.addr==10.3.27.86 \
 | sed -n '0~4p'
```

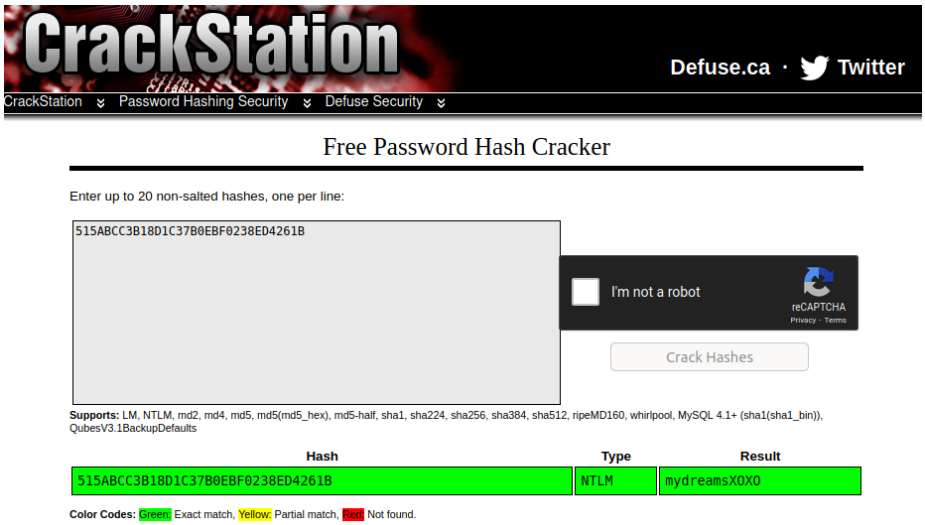
From the first part description, we know that the hacker uses simple password with three same letters to protect the exfiltration. So when we setup our cyberchef receip we can manually change letters (aaa, bbb, ccc...) for RC4 password, until we come to **lll** which will give us a valid zip file.




Inside the downloaded file we have this:

```
Administrator:500:NO PASSWORD**********************:31D4CFE1D1DAE531B73C59D7E0C089C0:::
Guest:501:NO PASSWORD**********************:NO PASSWORD**********************:::
admin:1000:NO PASSWORD**********************:0857632E42732DCC64201A80A05DD8D9:::
John Doe:1001:NO PASSWORD**********************:515ABCC3B18D1C37B0EBF0238ED4261B:::
```

On the crackstation.net we can supply our hash for John Doe password:




Our flag is: **mydreamsXOXO**

If someone would like to see how this part can be solved using Python, you can take look at file solve_part_2.py