```
# install scapy and associated packages
import sys
import os
import gc
import pandas as pd
!pip install scapy
from scapy.all import hexdump
import scapy.utils
```

```
Requirement already satisfied: scapy in /usr/local/lib/python3.11/dist-packages (2.6.1)
```

```
# list pcap files available in the folder
files = [f for f in os.listdir("/content/drive/MyDrive/Colab Notebooks/Working Data") if f.endswith(".pcap")]
print(list(files))
```

```
['Copy of Malware_8.pcap']
```

```
# loop that goes through all pcap files in the folder and exports .csv files
for file in files:
  no_pcap = file.split(".")
  file_name = "/content/drive/MyDrive/Colab Notebooks/Working Data/" + file
  savefile_name = "/content/drive/MyDrive/Colab Notebooks/Working Data/" + no_pcap[0] + "_Hexdump.csv"
  # switched to pcapreader to reduce memory requirements
  a = scapy.utils.PcapReader(file_name)
  # create two empty arrays
  b = []
  c = []
  # iterate through a and pull both packet time and overall header hexdump
  for packet in a:
    b.append(hexdump(packet,dump=True))
    c.append(packet.time)
  # create empty pandas dataframe
  df = pd.DataFrame()
  # add both arrays as columns to the dataframe
  df['epoch_time'] = c
  df['hexdump'] = b
  # export the dataframe to a csv file and notify the user
  df.to_csv(savefile_name)
  print("Completed exporting " + savefile_name)
  # calculates and displays the memory usage of the largest variables
  totalmem = str((sys.getsizeof(df) + sys.getsizeof(a) + sys.getsizeof(b) + sys.getsizeof(c))/1000000)
  print("Total Memory Occupied: " + totalmem + "MB")
  # clear memory of large variables before next iteration
  del a, df, b, c
  gc.collect()
```

```
Completed exporting /content/drive/MyDrive/Colab Notebooks/Working Data/Copy of Malware_8_Hexdump.csv
Total Memory Occupied: 13.492318MB
```

Start coding or generate with AI.