Домашняя работа №4

Анализ записи сетевой активности.

Шараев Евгений

Задание к варианту 1.

1. Каков IP-адрес зараженного узла?
2. Каков MAC-адрес зараженного узла?
3. Каково доменное имя зараженного узла?
4. Какие сайты посетил пользователь зараженного устройства по своему желанию?
5. Посещение каких сайтов зафиксировано в сетевом трафике?
6. Каково доменное имя сайта, с которого произошла загрузка вредоносного программного обеспечения?
7. Каков IP-адрес узла, с которого произошла загрузка вредоносного программного обеспечения?
8. Загружались ли пользователем или системой без ведома пользователя файлы, не являющиеся вредоносными?
9. Какие сайты (доменные имена) задействованы в заражении пользователя вредоносным программным обеспечением (имеют следы вредоносной активности, участвуют во вредоносных действиях)?
10. Каков механизм переходов (перенаправлений) пользователя с посещенных сайтов на сайт, с которого было загружено вредоносное программное обеспечение?

Для исследования дампа сетевого трафика мной были использованы сетевые следующие инструменты:

- ***wireshark*** для поиска, фильтрации, сортировки и чтения пакетов
- ***tcpdump*** для фильтрации пакетов и получения общих статистических сведений
- ***pandas*** для табличного представления и статистической обработки данных
- ***matplotlib*** и ***plotly*** для визуализации данных

In [1]:
```
# Поиссследую дамп сетевого трафика утилитой tcpdump

# Всего в дампе 3053 перехваченных пакета
!tcpdump -r var1.pcap --count
```

```
reading from file var1.pcap, link-type EN10MB (Ethernet), snapshot length 65535
3053 packets
```

In [2]:
```
# Самый первый пакет в дампе был в 10:11:49.324203

!tcpdump -r var1.pcap -v | head
```

```
reading from file var1.pcap, link-type EN10MB (Ethernet), snapshot length 65535
10:11:49.324203 IP (tos 0x0, ttl 128, id 482, offset 0, flags [none], proto TCP (6), length 44)
    a-0001.a-msedge.net.http > 172.16.165.165.49433: Flags [S.], cksum 0x21c9 (correct), seq 541339948, ack 9227
66772, win 64240, options [mss 1460], length 0
10:11:49.324203 IP (tos 0x0, ttl 128, id 483, offset 0, flags [none], proto TCP (6), length 44)
    a-0001.a-msedge.net.http > 172.16.165.165.49432: Flags [S.], cksum 0x4136 (correct), seq 2110835290, ack 246
0277736, win 64240, options [mss 1460], length 0
10:11:49.425739 IP (tos 0x0, ttl 128, id 484, offset 0, flags [none], proto TCP (6), length 44)
    a-0001.a-msedge.net.http > 172.16.165.165.49433: Flags [S.], cksum 0x21c9 (correct), seq 541339948, ack 9227
66772, win 64240, options [mss 1460], length 0
10:11:49.425740 IP (tos 0x0, ttl 128, id 485, offset 0, flags [none], proto TCP (6), length 44)
    a-0001.a-msedge.net.http > 172.16.165.165.49432: Flags [S.], cksum 0x4136 (correct), seq 2110835290, ack 246
0277736, win 64240, options [mss 1460], length 0
10:11:49.530499 IP (tos 0x0, ttl 128, id 486, offset 0, flags [none], proto TCP (6), length 44)
    a-0001.a-msedge.net.http > 172.16.165.165.49433: Flags [S.], cksum 0x21c9 (correct), seq 541339948, ack 9227
66772, win 64240, options [mss 1460], length 0
tcpdump: Unable to write output: Broken pipe
```

In [3]:
```
# Ну а самый последний в 10:22:45.512676

!tcpdump -r var1.pcap -v | tail
```

```
reading from file var1.pcap, link-type EN10MB (Ethernet), snapshot length 65535
    172.16.165.165.netbios-ns > 172.16.165.2.netbios-ns: UDP, length 68
10:22:41.061276 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 172.16.165.162 tell 172.16.165.2, length 46
10:22:42.512632 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 172.16.165.2 tell 172.16.165.165, length 28
10:22:42.512776 ARP, Ethernet (len 6), IPv4 (len 4), Reply 172.16.165.2 is-at 00:50:56:f3:ca:52 (oui Unknown), l
ength 46
10:22:42.512791 IP (tos 0x0, ttl 128, id 5992, offset 0, flags [none], proto UDP (17), length 96)
    172.16.165.165.netbios-ns > 172.16.165.2.netbios-ns: UDP, length 68
10:22:44.012703 IP (tos 0x0, ttl 128, id 5993, offset 0, flags [DF], proto UDP (17), length 96)
    172.16.165.165.netbios-ns > 172.16.165.2.netbios-ns: UDP, length 68
10:22:45.512676 IP (tos 0x0, ttl 128, id 5994, offset 0, flags [DF], proto UDP (17), length 96)
    172.16.165.165.netbios-ns > 172.16.165.2.netbios-ns: UDP, length 68
```

In [4]:
```
# По https было перехвачено 591 пакет

!tcpdump 'tcp port 443' -r var1.pcap --count
```

```
reading from file var1.pcap, link-type EN10MB (Ethernet), snapshot length 65535
591 packets
```

In [5]:
```
# По http на порт 80 было перехвачено 2360 пакетов

!tcpdump 'tcp port 80' -r var1.pcap --count
```

```
reading from file var1.pcap, link-type EN10MB (Ethernet), snapshot length 65535
2360 packets
```

In [6]:
```
# Сколько всего пакетов по TCP протоколу

!tcpdump 'tcp' -r var1.pcap --count
```

```
reading from file var1.pcap, link-type EN10MB (Ethernet), snapshot length 65535
2951 packets
```

In [7]:
```
# TCP запросы либо на port 80 либо на 443. Картинка сложилась

bool(2360 + 591 == 2951)
```

Out[7]:
```
True
```

In [8]:
```
# Посмотрим сколько TCP пакетов с флагом SYN было перехвачено
# 336 пакетов

!tcpdump 'tcp[tcpflags] & (tcp-syn) != 0' -r var1.pcap --count
```

```
reading from file var1.pcap, link-type EN10MB (Ethernet), snapshot length 65535
336 packets
```

In [9]:
```
# Посмотрим сколько TCP пакетов с флагом ACK было перехвачено
# 336 пакетов

!tcpdump 'tcp[tcpflags] & (tcp-ack) != 0' -r var1.pcap --count
```

```
reading from file var1.pcap, link-type EN10MB (Ethernet), snapshot length 65535
2929 packets
```

In [10]:
```
# Посмотрим более расширенную статистику по всем данным в дампе
# трафика для этого я воспользовался инструментом экспорта
# дампа pcap в формат csv для более удобной аналитики и статистики

# !sudo wireshark -r var1.pcap

# [sudo] password for evgeny:  ** (wireshark:432543) 20:00:00.182442 [GUI WARNING]
# -- QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'

# на выходе получил файл traffic.csv
!ls -l
```

```
total 1057056
drwxr-xr-x 14 evgeny evgeny       4096 Apr 11 17:06 distorm
drwxr-xr-x 14 evgeny evgeny       4096 Apr 11 17:14 distorm3
drwxr-xr-x  7 evgeny evgeny       4096 Apr  2 11:13 env
-rw-r--r--  1 evgeny evgeny     548810 Apr 12 12:42 sharaev_evgeny_hw2_2.pdf
-rw-r--r--  1 evgeny evgeny      65119 May 22 18:19 sharaev_evgeny_hw2.ipynb
-rw-r--r--  1 evgeny evgeny     509680 Apr 12 12:42 sharaev_evgeny_hw2.pdf
-rw-r--r--  1 evgeny evgeny    4212587 May 25 15:07 sharaev_evgeny_hw4.ipynb
-rw-r--r--  1 evgeny evgeny      83539 May 25 10:18 tcp_packets.png
-rw-r--r--  1 evgeny evgeny     673501 May 25 10:39 traffic.csv
-rw-r--r--  1 evgeny evgeny    2551397 May 22 17:19 var1.pcap
-rw-r--r--  1 evgeny evgeny 1073741824 May 25  2023 var-1.vmem
drwxr-xr-x  8 evgeny evgeny       4096 Apr 11 16:10 volatility
drwxr-xr-x  8 evgeny evgeny       4096 Apr  2 10:17 volatility3
```

In [11]:
```
# Вот первые несколько строк этого файла

!head traffic.csv
```

```
"No.","Time","Source","SRC_PORT","Destination","DST_PORT","Protocol","Length","Info","SYN","ACK","FIN","RST","PS
H","URG","FILE"
"1","10:11:49.324203","204.79.197.200","80","172.16.165.165","49433","TCP","60","80  >  49433 [SYN, ACK] Seq=0 A
ck=1 Win=64240 Len=0 MSS=1460","Set","Set","Not set","Not set","Not set","Not set",""
"2","10:11:49.324203","204.79.197.200","80","172.16.165.165","49432","TCP","60","80  >  49432 [SYN, ACK] Seq=0 A
ck=1 Win=64240 Len=0 MSS=1460","Set","Set","Not set","Not set","Not set","Not set",""
"3","10:11:49.425739","204.79.197.200","80","172.16.165.165","49433","TCP","60","[TCP Retransmission] 80  >  494
33 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460","Set","Set","Not set","Not set","Not set","Not set",""
"4","10:11:49.425740","204.79.197.200","80","172.16.165.165","49432","TCP","60","[TCP Retransmission] 80  >  494
32 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460","Set","Set","Not set","Not set","Not set","Not set",""
"5","10:11:49.530499","204.79.197.200","80","172.16.165.165","49433","TCP","60","[TCP Retransmission] 80  >  494
33 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460","Set","Set","Not set","Not set","Not set","Not set",""
"6","10:11:49.530500","204.79.197.200","80","172.16.165.165","49432","TCP","60","[TCP Retransmission] 80  >  494
32 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460","Set","Set","Not set","Not set","Not set","Not set",""
"7","10:11:49.624738","204.79.197.200","80","172.16.165.165","49433","TCP","60","[TCP Retransmission] 80  >  494
33 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460","Set","Set","Not set","Not set","Not set","Not set",""
"8","10:11:49.624739","204.79.197.200","80","172.16.165.165","49432","TCP","60","[TCP Retransmission] 80  >  494
32 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460","Set","Set","Not set","Not set","Not set","Not set",""
"9","10:11:49.724834","204.79.197.200","80","172.16.165.165","49433","TCP","60","[TCP Retransmission] 80  >  494
33 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460","Set","Set","Not set","Not set","Not set","Not set",""
```

In [12]:
```
import pandas as pd

# Соберу датафрейм из csv файла

df = pd.read_csv("traffic.csv", dtype={'Source': 'string'})
df.head()
```

Out[12]:

	No.	Time	Source	SRC_PORT	Destination	DST_PORT	Protocol	Length	Info	SYN	ACK	FIN	RST	PSH	URG	FIL
0	1	10:11:49.324203	204.79.197.200	80.0	172.16.165.165	49433.0	TCP	60	80 > 49433 [SYN, ACK] Seq=0 Ack=1 Win=64240 ...	Set	Set	Not set	Not set	Not set	Not set	Nal
1	2	10:11:49.324203	204.79.197.200	80.0	172.16.165.165	49432.0	TCP	60	80 > 49432 [SYN, ACK] Seq=0 Ack=1 Win=64240 ...	Set	Set	Not set	Not set	Not set	Not set	Nal
2	3	10:11:49.425739	204.79.197.200	80.0	172.16.165.165	49433.0	TCP	60	[TCP Retransmission] 80 > 49433 [SYN, ACK] S...	Set	Set	Not set	Not set	Not set	Not set	Nal
3	4	10:11:49.425740	204.79.197.200	80.0	172.16.165.165	49432.0	TCP	60	[TCP Retransmission] 80 > 49432 [SYN, ACK] S...	Set	Set	Not set	Not set	Not set	Not set	Nal
4	5	10:11:49.530499	204.79.197.200	80.0	172.16.165.165	49433.0	TCP	60	[TCP Retransmission] 80 > 49433 [SYN, ACK] S...	Set	Set	Not set	Not set	Not set	Not set	Nal

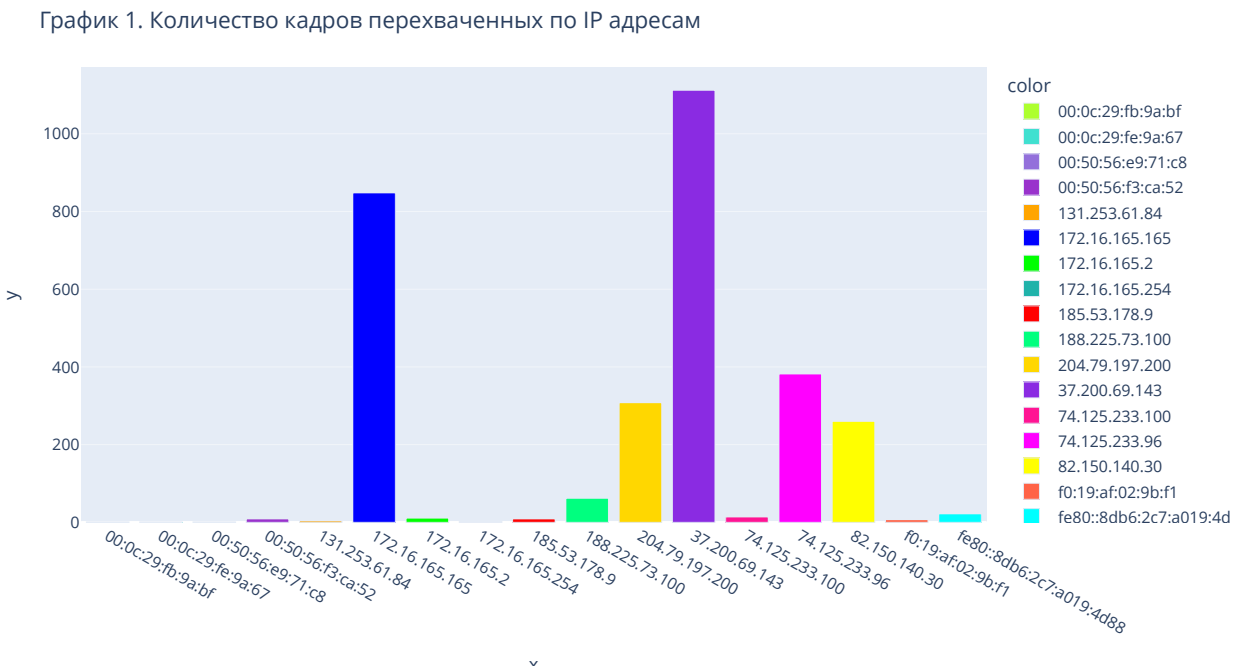
In [13]:
```
# В датафрейме по стобцам распределны следующие данные:

# No - порядковый номер пакета в дампе
# Time - абсолютное время
# Source - IP адрес источника
# SRC_PORT - порт источника
# Destination - IP адрес получателя
# DST_PORT - порт получателя
# Protocol - название протокола
# Length - длина (размер) пакета
# Info - общие сведения
# SYN, ACK, FIN, RST, PSH, URG, FILE - Наличие tcp флага в сегменте (set/ not set)
# FILE - данные файла (если имеются)


df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3053 entries, 0 to 3052
Data columns (total 16 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   No.          3053 non-null   int64
 1   Time         3053 non-null   object
 2   Source       3053 non-null   string
 3   SRC_PORT     3024 non-null   float64
 4   Destination  3053 non-null   object
 5   DST_PORT     3024 non-null   float64
 6   Protocol     3053 non-null   object
 7   Length       3053 non-null   int64
 8   Info         3053 non-null   object
 9   SYN          2951 non-null   object
 10  ACK          2951 non-null   object
 11  FIN          2951 non-null   object
 12  RST          2951 non-null   object
 13  PSH          2951 non-null   object
 14  URG          2951 non-null   object
 15  FILE         36 non-null     object
dtypes: float64(2), int64(2), object(11), string(1)
memory usage: 381.8+ KB
```

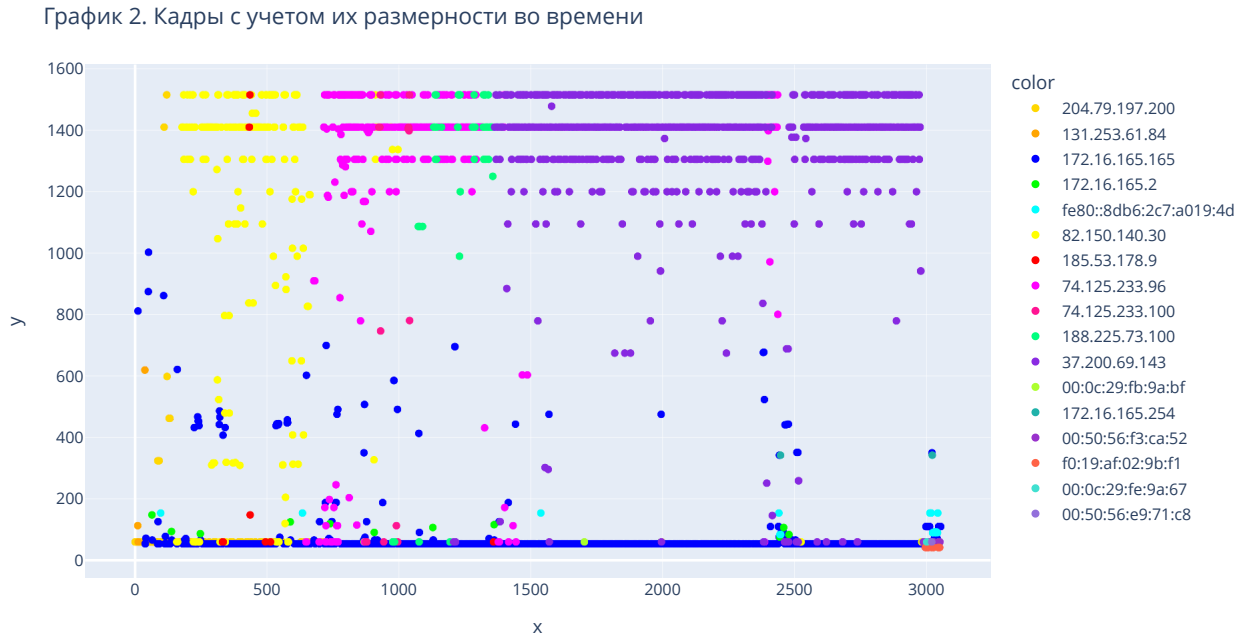
In [14]:
```
# Задам каждому Source address свой цвет
# для дальнейшей визуализации

import matplotlib


sources = df["Source"].unique()
colours = [
    "#FFD700", "#FFA500", "#0000FF", "#00FF00", "#00FFFF",
    "#FFFF00", "#FF0000", "#FF00FF", "#FF1493", "#00FF7F",
    "#8A2BE2", "#ADFF2F", "#20B2AA", "#9932CC", "#FF6347",
    "#40E0D0", "#9370DB"
]

palette = {sources[i]:colours[i] for i in range(len(sources))}
palette
```

Out[14]:
```
{'204.79.197.200': '#FFD700',
 '131.253.61.84': '#FFA500',
 '172.16.165.165': '#0000FF',
 '172.16.165.2': '#00FF00',
 'fe80::8db6:2c7:a019:4d88': '#00FFFF',
 '82.150.140.30': '#FFFF00',
 '185.53.178.9': '#FF0000',
 '74.125.233.96': '#FF00FF',
 '74.125.233.100': '#FF1493',
 '188.225.73.100': '#00FF7F',
 '37.200.69.143': '#8A2BE2',
 '00:0c:29:fb:9a:bf': '#ADFF2F',
 '172.16.165.254': '#20B2AA',
 '00:50:56:f3:ca:52': '#9932CC',
 'f0:19:af:02:9b:f1': '#FF6347',
 '00:0c:29:fe:9a:67': '#40E0D0',
 '00:50:56:e9:71:c8': '#9370DB'}
```

In [15]:
```
# Посчитаем сколько каждый IP адрес источника
# встречался в дампе. Визуализирую полученные данные

import plotly.express as px

frequency = df.groupby(by=["Source"]).size().reset_index(name="Count")
fig = px.bar(
    x=frequency["Source"],
    y=frequency["Count"],
    color=frequency["Source"],
    color_discrete_map=palette
)
fig.update_layout(title='График 1. Количество кадров перехваченных по IP адресам')
fig.show()
```

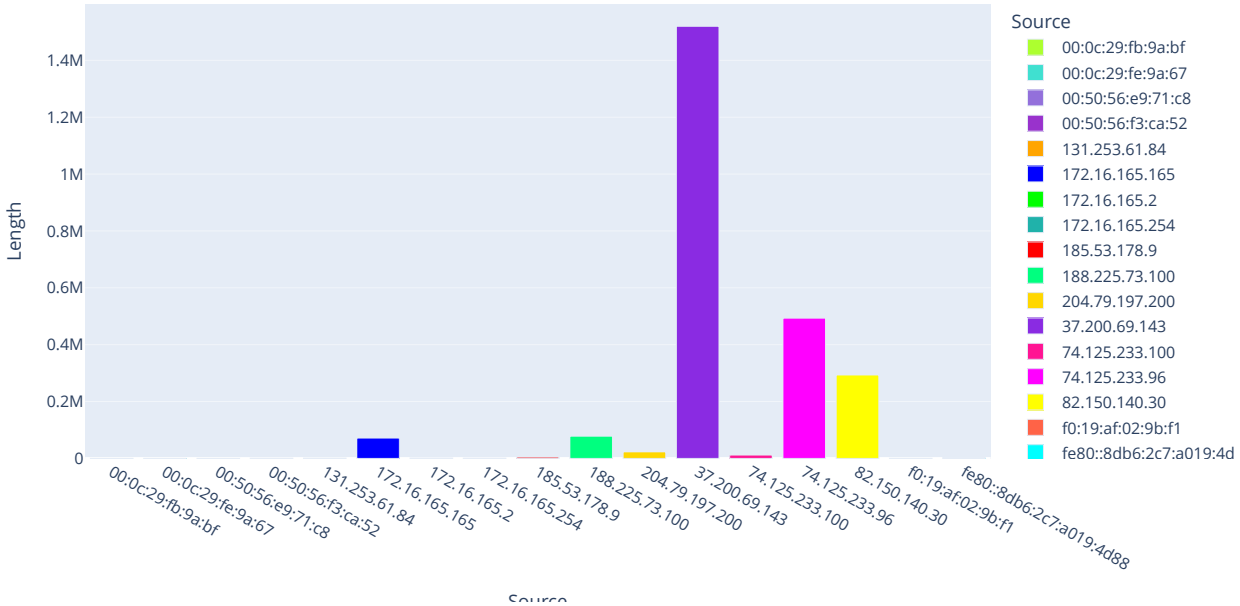


In [16]:
```
# Посмотрим визуально какого размера кадры
# проходили в момент захвата трафика

fig = px.scatter(
    x=df['No.'],
    y=df['Length'],
    color=df['Source'],
    color_discrete_map=palette
)

fig.update_layout(title='График 2. Кадры с учетом их размерности во времени')
fig.show()
```



На графике видно что среднего размера пакеты передаваоись в меньшей степени. в основном данные передавались пакетами минимального размера или максимального. Причем максимального размера пакеты ходили с 3 основных адресов выделенных на графике желтым, розовым и фиолетовым цветом. Далее я рассмотрб эти источники более подробно.

In [17]:
```
# Посмотрим аггрегированные данные объема трафика по IP адресам

sum_length_by_ip = df.groupby(['Source'])['Length'].sum().reset_index()

fig = px.bar(sum_length_by_ip, x='Source', y='Length', color="Source", color_discrete_map=palette)
fig.show()
```

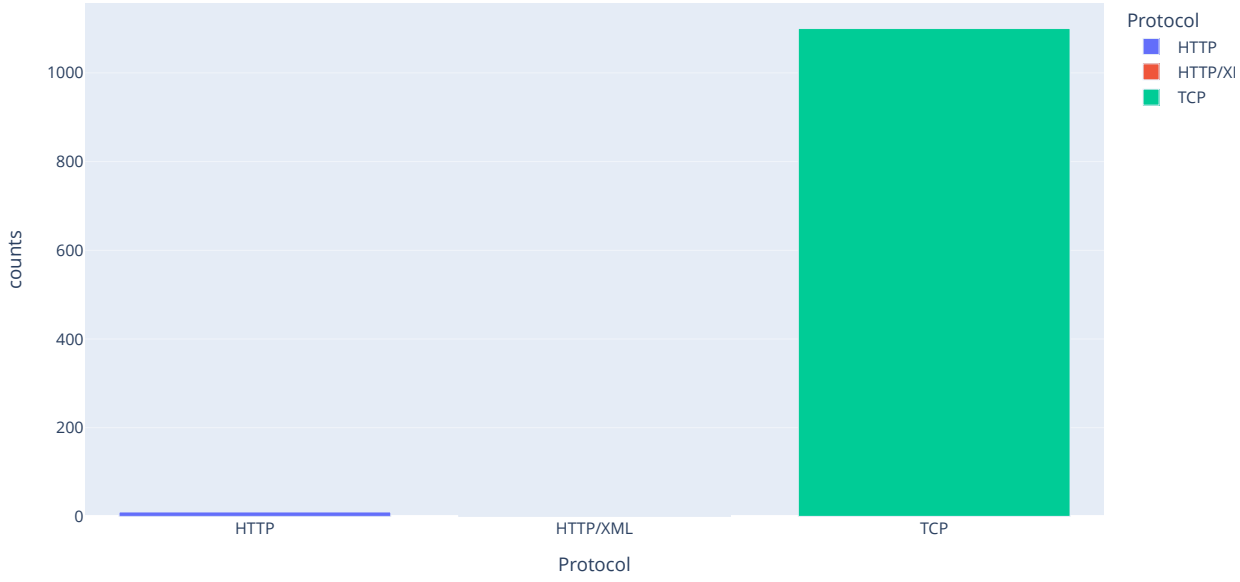


Из двух предыдущих графиков видно что больше всего данных было передано с трёх хостов (желтый, розовый и фиолетовый):

- 37.200.69.143
- 74.125.233.96
- 82.150.140.30

Расмотрим трафик с этих хостов по отдельности. Сначала отфильтрую их от остальных кадров

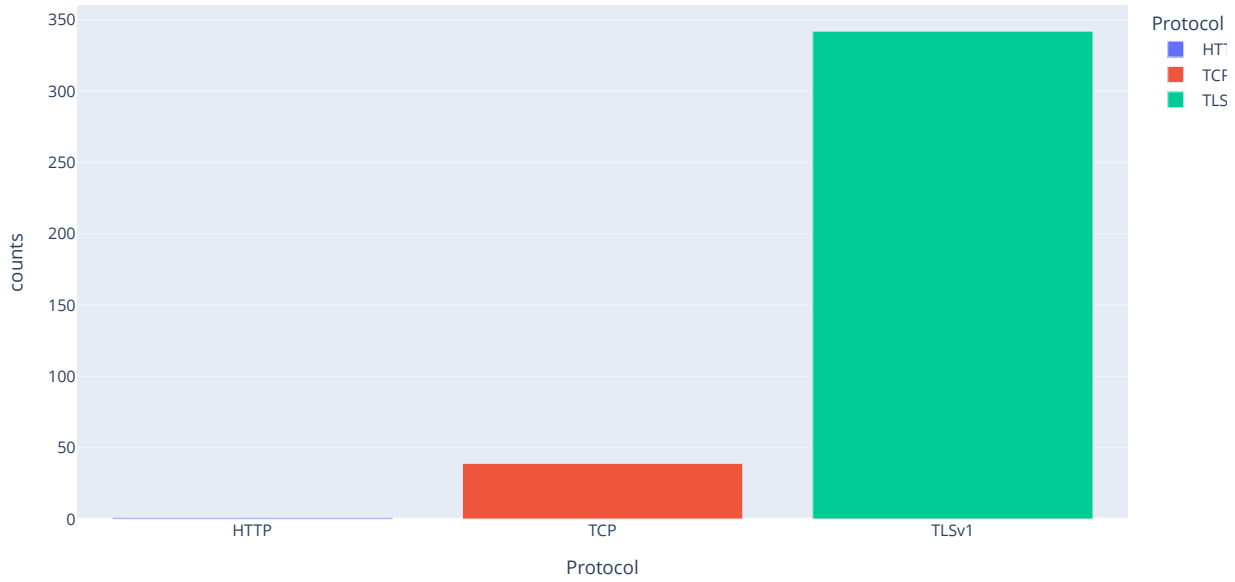
In [18]:
```
# Посмотрим с каких портов были сегменты и
# какие протоколы использовались

df_37 = df[df.Source == '37.200.69.143']
df_37_protocols = df_37.groupby(['Protocol']).size().reset_index(name='counts')
fig = px.bar(df_37_protocols, x='Protocol', y='counts', color="Protocol")
fig.show()
```



In [19]:
```
# Как видно весь трафик с хоста 37.200.69.143
# (всего 1112 кадров) шёл от исследуемого хоста
# с порта 80, то есть по нешифрованному HTTP протоколу

df_37_ports = df_37['SRC_PORT'].value_counts()
df_37_ports
```

Out[19]:
```
SRC_PORT
80.0    1112
Name: count, dtype: int64
```

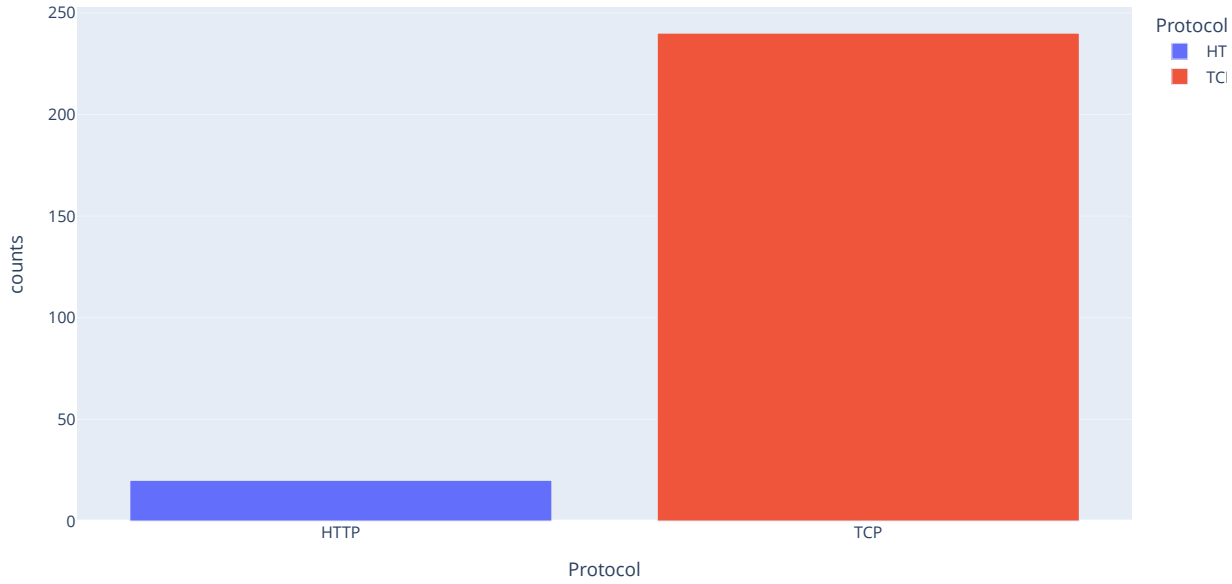
In [20]:
```
# Source 74.125.233.96

# А вот с этого источника трафик в основном шифрованый
# по протоколу TLSv1

df_74 = df[df.Source == '74.125.233.96']
df_74_protocols = df_74.groupby(['Protocol']).size().reset_index(name='counts')
fig = px.bar(df_74_protocols, x='Protocol', y='counts', color="Protocol")
fig.show()
```



In [21]:
```
# Трафик в основном шёл с 443 порта который
# характерен для безопасного https соединения

df_74_ports = df_74['SRC_PORT'].value_counts()
df_74_ports
```

Out[21]:
```
SRC_PORT
443.0    378
80.0       4
Name: count, dtype: int64
```

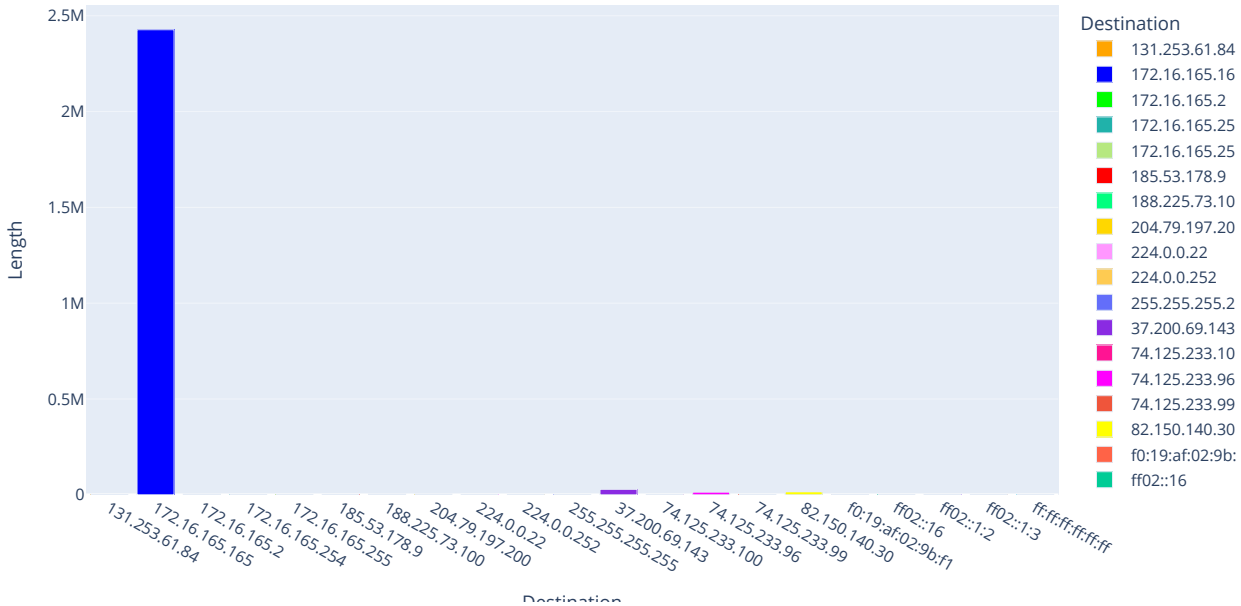
In [22]:
```
# 82.150.140.30

# Здесь аналогично первому источнику в основном
# использовался протокол HTTP

df_82 = df[df.Source == '82.150.140.30']
df_82_protocols = df_82.groupby(['Protocol']).size().reset_index(name='counts')
fig = px.bar(df_82_protocols, x='Protocol', y='counts', color="Protocol")
fig.show()
```



In [23]:
```
# И соответствующий не безопасному протоколу http
# порт номер 80

df_82_ports = df_82['SRC_PORT'].value_counts()
df_82_ports
```

Out[23]:
```
SRC_PORT
80.0    260
Name: count, dtype: int64
```

In [24]:
```
# Поисследуем распределение IP адресов получателей

sum_length_by_ip = df.groupby(['Destination'])['Length'].sum().reset_index()

fig = px.bar(sum_length_by_ip, x='Destination', y='Length', color="Destination", color_discrete_map=palette)
fig.show()
```



Ответ на вопрос 1: Каков IP-адрес зараженного узла?

Практически все пакеты были направлялись на хост ***172.16.165.165***. Делаем вывод что это адрес нашего зараженного хоста с которого собирался дамп.

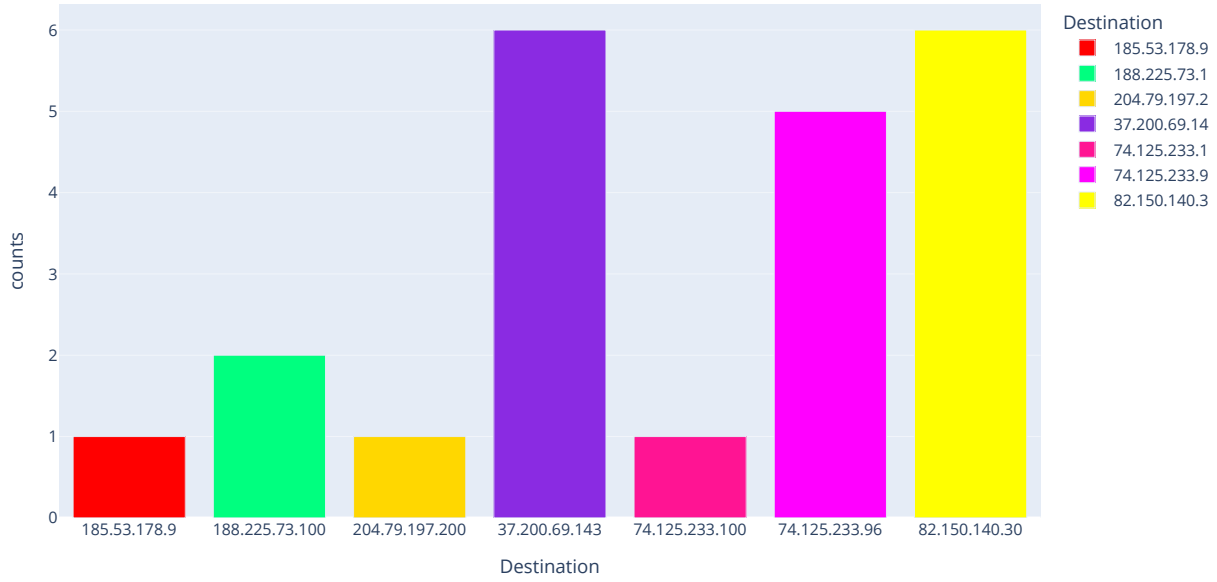
In [25]:
```
# Ключ ***-e*** утилиты tcpdump выводит заголовки ethernet.

!tcpdump 'src host 172.16.165.165' -r var1.pcap -e | head -1
```

```
reading from file var1.pcap, link-type EN10MB (Ethernet), snapshot length 65535
10:11:49.768188 f0:19:af:02:9b:f1 (oui Unknown) > 00:50:56:f3:ca:52 (oui Unknown), ethertype IPv4 (0x0800), leng
th 811: 172.16.165.165.49435 > 131.253.61.84.https: Flags [P.], seq 1385503808:1385504565, ack 17278809, win 634
81, length 757
tcpdump: Unable to write output: Broken pipe
```

Ответ на вопрос 2: Каков MAC-адрес зараженного узла?

Во втором поле видно MAK адрес источника - ***f0:19:af:02:9b:f1***, так как получатель пакета - это хост по адресу 131.253.61.84 (внешняя сеть), то второй MAK адрес 00:50:56:f3:ca:52 - это очевидно локальный сетевой интерфейс шлюза по-умолчанию (роутера)

In [26]:
```
# Поиссследуем все сайты (по IP адресам) какие посещались с зараженного хоста
# То есть те на которые с адреса 172.16.165.165 уходили TCP запросы с флагом SYN

syn_addr = df[(df["Source"] == "172.16.165.165") & (df["Protocol"] == "TCP") & (df["SYN"] == "Set")]
syn_addr_agg = syn_addr.groupby(['Destination']).size().reset_index(name='counts')
fig = px.bar(syn_addr_agg, x='Destination', y='counts', color="Destination", color_discrete_map=palette)
fig.show()
```



Большая часть этих адресов нам уже встречалась. Видим что больше всего запросов уходили примерно равновероятно на те же три адреса из предыдущих графиков. Насколько добровольно это было еще следует выяснить.

In [27]:
```
# Посмотрим запросы к ДНС и как эти адреса разрезолвились:

# ssl.bing.com                204.79.197.200
# adultbiz.in                 185.53.178.9
# www.youtube.com             74.125.233.96
# 24corp-shop.com             188.225.73.100
# www.ciniholland.nl          82.150.140.30
# stand.trustandprobaterealty.com  37.200.69.143

# Видим что с адресом 74.125.233.100 ДНС запроса
# во время захвата трафика не было. Но поскольку
# адрес 74.125.233.96 - это ютуб, то вероятно
# 74.125.233.100 тоже адрес гугловый

!tcpdump 'port 53' -r var1.pcap
```

```
reading from file var1.pcap, link-type EN10MB (Ethernet), snapshot length 65535
10:11:50.994963 IP 172.16.165.165.62720 > 172.16.165.2.domain: 51895+ A? ssl.bing.com. (30)
10:11:51.914894 IP 172.16.165.2.domain > 172.16.165.165.62720: 51895 3/0/0 CNAME ssl-bing-com.a-0001.a-msedge.net., CNAME a-0001.a-msedge.net., A 204.79.197.200 (106)
10:11:53.685362 IP 172.16.165.165.51415 > 172.16.165.2.domain: 7601+ A? www.ciniholland.nl. (36)
10:11:54.493624 IP 172.16.165.2.domain > 172.16.165.165.51415: 7601 1/0/0 A 82.150.140.30 (52)
10:11:56.195673 IP 172.16.165.165.60914 > 172.16.165.2.domain: 31133+ A? adultbiz.in. (29)
10:11:56.905440 IP 172.16.165.2.domain > 172.16.165.165.60914: 31133 1/0/0 A 185.53.178.9 (45)
10:11:59.968268 IP 172.16.165.165.52070 > 172.16.165.2.domain: 63336+ A? www.youtube.com. (33)
10:12:00.824825 IP 172.16.165.2.domain > 172.16.165.165.52070: 63336 2/0/0 CNAME youtube-ui.l.google.com., A 74.125.233.96 (83)
10:12:05.810977 IP 172.16.165.165.51871 > 172.16.165.2.domain: 42871+ A? s.ytimg.com. (29)
10:12:06.405175 IP 172.16.165.2.domain > 172.16.165.165.51871: 42871 2/0/0 CNAME ytstatic.l.google.com., A 74.125.233.96 (77)
10:12:09.734357 IP 172.16.165.165.54787 > 172.16.165.2.domain: 13278+ A? 24corp-shop.com. (33)
10:12:10.530965 IP 172.16.165.2.domain > 172.16.165.165.54787: 13278 1/0/0 A 188.225.73.100 (49)
10:12:11.958014 IP 172.16.165.165.60678 > 172.16.165.2.domain: 58844+ A? stand.trustandprobaterealty.com. (49)
10:12:12.476743 IP 172.16.165.2.domain > 172.16.165.165.60678: 58844 1/0/0 A 37.200.69.143 (65)
10:12:13.918832 IP 172.16.165.165.64324 > 172.16.165.2.domain: 18240+ A? i.ytimg.com. (29)
10:12:14.648640 IP 172.16.165.2.domain > 172.16.165.165.64324: 18240 2/0/0 CNAME ytimg.l.google.com., A 74.125.233.96 (74)
10:12:51.526013 IP 172.16.165.165.50936 > 172.16.165.2.domain: 18224+ A? wpad.localdomain. (34)
10:12:51.526505 IP 172.16.165.2.domain > 172.16.165.165.50936: 18224 NXDomain*- 0/0/0 (34)
10:12:58.627300 IP 172.16.165.165.55932 > 172.16.165.2.domain: 1803+ A? stand.trustandprobaterealty.com. (49)
10:12:59.400833 IP 172.16.165.2.domain > 172.16.165.165.55932: 1803 1/0/0 A 37.200.69.143 (65)
10:13:00.982703 IP 172.16.165.165.50173 > 172.16.165.2.domain: 19171+ A? java.com. (26)
10:13:01.859093 IP 172.16.165.2.domain > 172.16.165.165.50173: 19171 1/0/0 A 2.22.206.134 (42)
```

Ответ на вопрос 5: Посещение каких сайтов зафиксировано в сетевом трафике?

- ssl.bing.com
- adultbiz.in
- www.youtube.com (http://www.youtube.com)
- 24corp-shop.com
- www.ciniholland.nl (http://www.ciniholland.nl)
- stand.trustandprobaterealty.com

In [28]:
```
# Поиссследуем файлы данных которые были переданы
# во время перехвата трафика

df_file = df[df['FILE'].notnull()]
df_file.head(5)
```

Out[28]:

	No.	Time	Source	SRC_PORT	Destination	DST_PORT	Protocol	Length	Info	SYN	ACK	FIN	RST	PSH	URG
51	52	10:11:51.345014	172.16.165.165	49431.0	204.79.197.200	80.0	HTTP/XML	1002	POST /fd/ls/lsp.aspx HTTP/1.1	Not set	Set	Not set	Not set	Set	Set
129	130	10:11:54.208035	204.79.197.200	80.0	172.16.165.165	49429.0	HTTP	462	HTTP/1.1 200 OK (GIF89a)	Not set	Set	Not set	Not set	Set	Set
310	311	10:11:57.571273	82.150.140.30	80.0	172.16.165.165	49439.0	HTTP	1271	HTTP/1.1 200 OK (text/css)	Not set	Set	Not set	Not set	Set	Set
312	313	10:11:57.571274	82.150.140.30	80.0	172.16.165.165	49442.0	HTTP	587	HTTP/1.1 200 OK (text/javascript)	Not set	Set	Not set	Not set	Set	Set
313	314	10:11:57.571313	82.150.140.30	80.0	172.16.165.165	49441.0	HTTP	1046	HTTP/1.1 200 OK (text/css)	Not set	Set	Not set	Not set	Set	Set

In [29]:
```
# Поиисследую каждый IP источник полученный
# датасете по отдельности


#  '204.79.197.200'
#  '131.253.61.84'
#  '172.16.165.165'
#  '172.16.165.2'
#  '82.150.140.30'
#  '185.53.178.9'   # Вот тут скрытый <body>
#  '74.125.233.96'
#  '74.125.233.100'
#  '188.225.73.100'
#  '37.200.69.143'
#  '00:0c:29:fb:9a:bf'
#  '172.16.165.254'
#  '00:50:56:f3:ca:52'
#  'f0:19:af:02:9b:f1'
#  '00:0c:29:fe:9a:67'
#  '00:50:56:e9:71:c8'

# Выведу на экран содержимое файла
# переданного с источника 185.53.178.9

df_file_82 = df_file[df_file["Source"] == "185.53.178.9"]
for i in df_file_82["FILE"]:
    print(i)
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitio
nal.dtd">\\n<html data-adblockkey="MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBALquDFETXRn0Hr05fUP7EJT77xYnPmRbpMy4vk8KYiHnk
NpednjOANJcaXDXcKQJN0nXKZJL7TciJD8AoHXK158CAwEAAQ==_C846BpEU0hsFM/xTjTqlLXN2LNVJLkmeoXaJY26UuFsolp3W8Z480CT99oaw
gOGdkTjw8uGCNTSS6sv9BLfQuQ==" xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">\\n<head>\\n<meta htt
p-equiv="Content-Type" content="text/html; charset=utf-8"/>\\n<title>adultbiz.in</title>\\n<script src="http://w
ww.google.com/adsense/domains/caf.js" type="text/javascript" ></script>\\n<link href="http://d1vbm0eveofcle.clou
dfront.net/themes/saledefault.css" rel="stylesheet" type="text/css" media="screen" />\\t<link href="http://d1vbm
0eveofcle.cloudfront.net/themes/assets/style.css" rel="stylesheet" type="text/css" media="screen" />\\n<link hre
f="http://d1vbm0eveofcle.cloudfront.net/themes/cleanPeppermintBlack/style.css" rel="stylesheet" type="text/css"
media="screen" />\\n<link href='http://fonts.googleapis.com/css?family=Libre+Baskerville:400,700' rel='styleshee
t' type='text/css'>\\n</head>\\n<body id="afs" style="visibility:hidden;">\\n<script src="http://www.parkingcre
w.net/scripts/sale_form.js" type="text/javascript"></script>\\n<div id="sale_link">\\n<a href="http://www.sedo.c
om/search/details.php4?partnerid=&domain=adultbiz.in" target="_blank" onmousedown="tlink('ing', 'adultbiz.i
n');">\\nBuy this domain.\\t\\t</a>\\n</div><div id="holder" class="secondPage">\\n<div id="header">\\n<div id
="domainname">adultbiz.in</div>\\n<div id="searchHolder">\\n<div id="searchbox"></div>\\n</div>\\n</div><!--head
er-->\\n<div id="content">\\n<div id="rsHolder">\\n<div id="rs"></div>\\n</div>\\n<div id="adsHolder" class="ads
Holder">\\n<div id="ads"></div>\\n</div>\\n<div class="clear"></div>\\n</div>\\n</div>\\n<div id="copyright">\\n
<script type="text/javascript">function showImprint(){var imprintwnd = window.open('','pcrew_imprint','width=64
0,height=480,left=200,top=200,menubar=no,status=yes,toolbar=no');imprintwnd.
```

In [30]:
```
# !wireshark -r var1.pcap
```

In [ ]:
```

```