```
log_data = """192.168.1.1 - - [03/Dec/2024:10:12:34 +0000] "GET /home HTTP/1.1" 200 512
203.0.113.5 - - [03/Dec/2024:10:12:35 +0000] "POST /login HTTP/1.1" 401 128 "Invalid credentials"
10.0.0.2 - - [03/Dec/2024:10:12:36 +0000] "GET /about HTTP/1.1" 200 256
192.168.1.1 - - [03/Dec/2024:10:12:37 +0000] "GET /contact HTTP/1.1" 200 312
198.51.100.23 - - [03/Dec/2024:10:12:38 +0000] "POST /register HTTP/1.1" 200 128
203.0.113.5 - - [03/Dec/2024:10:12:39 +0000] "POST /login HTTP/1.1" 401 128 "Invalid credentials"
192.168.1.100 - - [03/Dec/2024:10:12:40 +0000] "POST /login HTTP/1.1" 401 128 "Invalid credentials"
10.0.0.2 - - [03/Dec/2024:10:12:41 +0000] "GET /dashboard HTTP/1.1" 200 1024
198.51.100.23 - - [03/Dec/2024:10:12:42 +0000] "GET /about HTTP/1.1" 200 256
192.168.1.1 - - [03/Dec/2024:10:12:43 +0000] "GET /dashboard HTTP/1.1" 200 1024
203.0.113.5 - - [03/Dec/2024:10:12:44 +0000] "POST /login HTTP/1.1" 401 128 "Invalid credentials"
203.0.113.5 - - [03/Dec/2024:10:12:45 +0000] "POST /login HTTP/1.1" 401 128 "Invalid credentials"
192.168.1.100 - - [03/Dec/2024:10:12:46 +0000] "POST /login HTTP/1.1" 401 128 "Invalid credentials"
10.0.0.2 - - [03/Dec/2024:10:12:47 +0000] "GET /profile HTTP/1.1" 200 768
192.168.1.1 - - [03/Dec/2024:10:12:48 +0000] "GET /home HTTP/1.1" 200 512
198.51.100.23 - - [03/Dec/2024:10:12:49 +0000] "POST /feedback HTTP/1.1" 200 128
203.0.113.5 - - [03/Dec/2024:10:12:50 +0000] "POST /login HTTP/1.1" 401 128 "Invalid credentials"
192.168.1.1 - - [03/Dec/2024:10:12:51 +0000] "GET /home HTTP/1.1" 200 512
198.51.100.23 - - [03/Dec/2024:10:12:52 +0000] "GET /about HTTP/1.1" 200 256
203.0.113.5 - - [03/Dec/2024:10:12:53 +0000] "POST /login HTTP/1.1" 401 128 "Invalid credentials"
192.168.1.100 - - [03/Dec/2024:10:12:54 +0000] "POST /login HTTP/1.1" 401 128 "Invalid credentials"
10.0.0.2 - - [03/Dec/2024:10:12:55 +0000] "GET /contact HTTP/1.1" 200 512
198.51.100.23 - - [03/Dec/2024:10:12:56 +0000] "GET /home HTTP/1.1" 200 512
192.168.1.100 - - [03/Dec/2024:10:12:57 +0000] "POST /login HTTP/1.1" 401 128 "Invalid credentials"
203.0.113.5 - - [03/Dec/2024:10:12:58 +0000] "POST /login HTTP/1.1" 401 128 "Invalid credentials"
10.0.0.2 - - [03/Dec/2024:10:12:59 +0000] "GET /dashboard HTTP/1.1" 200 1024
192.168.1.1 - - [03/Dec/2024:10:13:00 +0000] "GET /about HTTP/1.1" 200 256
198.51.100.23 - - [03/Dec/2024:10:13:01 +0000] "POST /register HTTP/1.1" 200 128
203.0.113.5 - - [03/Dec/2024:10:13:02 +0000] "POST /login HTTP/1.1" 401 128 "Invalid credentials"
192.168.1.100 - - [03/Dec/2024:10:13:03 +0000] "POST /login HTTP/1.1" 401 128 "Invalid credentials"
10.0.0.2 - - [03/Dec/2024:10:13:04 +0000] "GET /profile HTTP/1.1" 200 768
198.51.100.23 - - [03/Dec/2024:10:13:05 +0000] "GET /about HTTP/1.1" 200 256
192.168.1.1 - - [03/Dec/2024:10:13:06 +0000] "GET /home HTTP/1.1" 200 512
198.51.100.23 - - [03/Dec/2024:10:13:07 +0000] "POST /feedback HTTP/1.1" 200 128"""


with open("sample.log", "w") as file:
    file.write(log_data)


import re
import csv
import pandas as pd
from collections import defaultdict

FAILED_LOGIN_THRESHOLD = 10
LOG_FILE = "sample.log"
OUTPUT_CSV = "log_analysis_results.csv"


def parse_log(log_lines):
    data = []
    log_pattern = (
        r'^(?P<ip>\d+\.\d+\.\d+\.\d+) - - \[(?P<datetime>[^\]]+)\] '
        r'"(?P<method>[A-Z]+) (?P<endpoint>\S+) HTTP/[0-9.]+" (?P<status>\d+) (?P<size>\d+)(?: ".*")?$'
    )
    for line in log_lines:
        match = re.match(log_pattern, line)
        if match:
            data.append(match.groupdict())
    return data


def count_requests_per_ip(log_lines):
    ip_count = defaultdict(int)
    for line in log_lines:
        match = re.search(r'^(\d+\.\d+\.\d+\.\d+)', line)
        if match:
            ip_count[match.group(1)] += 1
    return sorted(ip_count.items(), key=lambda x: x[1], reverse=True)


def most_frequently_accessed_endpoint(log_lines):
    endpoint_count = defaultdict(int)
    for line in log_lines:
        match = re.search(r'"[A-Z]+ (\S+) HTTP/', line)
        if match:
```

```
        if match:
            endpoint_count[match.group(1)] += 1
    most_accessed = max(endpoint_count.items(), key=lambda x: x[1])
    return most_accessed


def detect_suspicious_activity(log_lines):
    failed_logins = defaultdict(int)
    for line in log_lines:
        if '401' in line or 'Invalid credentials' in line:
            match = re.search(r'^(\d+\.\d+\.\d+\.\d+)', line)
            if match:
                failed_logins[match.group(1)] += 1
    suspicious_ips = {ip: count for ip, count in failed_logins.items() if count > FAILED_LOGIN_THRESHOLD}
    return suspicious_ips


def save_to_csv(requests_per_ip, most_accessed, suspicious_ips):
    with open(OUTPUT_CSV, mode='w', newline='') as file:
        writer = csv.writer(file)

        writer.writerow(["IP Address", "Request Count"])
        writer.writerows(requests_per_ip)


        writer.writerow([])
        writer.writerow(["Endpoint", "Access Count"])
        writer.writerow(most_accessed)

        writer.writerow([])
        writer.writerow(["IP Address", "Failed Login Count"])
        writer.writerows(suspicious_ips.items())

def main():
    with open(LOG_FILE, 'r') as file:
        log_lines = file.readlines()


    parsed_data = parse_log(log_lines)
    if not parsed_data:
        print("No data parsed from the log file. Check the log format.")
        return



    df = pd.DataFrame(parsed_data)


    df['status'] = df['status'].astype(int)
    df['size'] = df['size'].astype(int)


    print('HEAD AND INFO\n')

    print("DataFrame Head:")
    print(df.head())
    print("_________________________________________________________________\n")

    print("\nDataFrame Info:")
    print(df.info())

    print("_________________________________________________________________\n")

    print('Count Requests per IP address')
    requests_per_ip = count_requests_per_ip(log_lines)
    most_accessed = most_frequently_accessed_endpoint(log_lines)
    suspicious_ips = detect_suspicious_activity(log_lines)

    print("\nRequests per IP:")
    for ip, count in requests_per_ip:
        print(f"{ip}: {count}")

    print("_________________________________________________________________\n")

    print("\nMost Frequently Accessed Endpoint:")
    print(f"{most_accessed[0]} accessed {most_accessed[1]} times")

    print("_________________________________________________________________\n")
    print("\nSuspicious Activity Detected:")
    for ip, count in suspicious_ips.items():
        print(f"{ip}: {count} failed login attempts")


    save_to_csv(requests_per_ip, most_accessed, suspicious_ips)

if __name__ == "__main__":
    main()
```

```
HEAD AND INFO

DataFrame Head:
              ip                    datetime method   endpoint  status  size
0    192.168.1.1  03/Dec/2024:10:12:34 +0000    GET      /home     200   512
1    203.0.113.5  03/Dec/2024:10:12:35 +0000   POST     /login     401   128
2       10.0.0.2  03/Dec/2024:10:12:36 +0000    GET     /about     200   256
3    192.168.1.1  03/Dec/2024:10:12:37 +0000    GET   /contact     200   312
4  198.51.100.23  03/Dec/2024:10:12:38 +0000   POST  /register     200   128
─────────────────────────────────────────────────────────────────


DataFrame Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 34 entries, 0 to 33
Data columns (total 6 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   ip        34 non-null     object
 1   datetime  34 non-null     object
 2   method    34 non-null     object
 3   endpoint  34 non-null     object
 4   status    34 non-null     int64 
 5   size      34 non-null     int64 
dtypes: int64(2), object(4)
memory usage: 1.7+ KB
None
─────────────────────────────────────────────────────────────────

Count Requests per IP address

Requests per IP:
203.0.113.5: 8
198.51.100.23: 8
192.168.1.1: 7
10.0.0.2: 6
192.168.1.100: 5
─────────────────────────────────────────────────────────────────


Most Frequently Accessed Endpoint:
/login accessed 13 times
─────────────────────────────────────────────────────────────────


Suspicious Activity Detected:
```