**Algorithm**

1. Import threading and pygeoip package
2. In main() call analyze()
3. Begin analyze()
   1. Open “attack.txt” file to write the output
   2. Define ipquery(ip)
      1. Return country name of ip
   3. Define refine()
      1. Open filter log file in read mode
         1. Define parse(line)
         2. Extract relevant information and write in csv format to a file
         3. End
         4. Read each line from log file and call parse(line)
   4. End refine()
   5. Define ips()
      1. Open ips log file in read mode
      2. Define parse(line)
      3. Extract relevant information and write in csv format to a file
      4. End
      5. Read each line from ips log file and call parse(line)
   6. End ips()
   7. Define apache()
      1. Open apache log file in read mode
      2. Define convert(line)
      3. If system file is being fetched
         1. Store source\_ip in dictionary and increase count by 1
      4. End convert()
      5. Define match(line)
      6. If status is “block”
         1. Store source\_ip in a dictionary and increase count by 1
      7. End match()
      8. Read each line from apache log file and call convert(line)
      9. Read each line from apache log file and call match(line)
      10. For every key in dictionary
          1. If value is greater than a predetermined threshold
          2. Write key to “attack.txt” for system file access violation
      11. For every key in dictionary
          1. If value is greater than a predetermined threshold
          2. Write key to “attack.txt” status code violation
   8. End apache()
   9. Define malicious()
      1. Open ips csv file in read mode
      2. Define detect(line)
         1. If “attack” or “bad” in classification then
         2. Store source\_ip in dictionary and increase count by 1
      3. End detect()
      4. Read each line from ips csv file and call detect(line)
      5. For every key in dictionary
         1. If value is greater than a predetermined threshold
         2. Write key to “attack.txt” for system file access violation
   10. End malicious()
   11. Call refine()
   12. Call ips()
   13. Call apache()
   14. Call malicious()
   15. Make a thread to call analyze() every 30 minutes
   16. Start thread
4. End analyze()