Importing and Parsing Security Logs with Python

Overview

In this project, I worked with Python to import, parse, and manipulate security log files. The exercise simulated a real-world security analyst task of handling log data, appending missing entries, and creating allow-lists of trusted IP addresses.

Tools & Technologies

- **Python** (file handling, string methods, list operations)
- **Text files (.txt)** for log storage and allow-lists

Key Activities

1. Importing and Reading Security Logs

- Opened a log file (`login.txt`) using Python's `with open()` statement.
- Read the entire log into a string variable using `.read()`.
- Displayed the raw contents of the log file, which contained usernames, IP addresses, timestamps, and dates.

[1]:

```
# Assign `import_file` to the name of the text file that contains the
 security␣→log file

import_file = "data/login.txt"

# The`with` statement
# Use `open()` to import security log file and store it as a string

with open(import_file, "r") as file:

  # Use `.read()` to read the imported file and store the result in a
 variable␣→named `text`

  text = file.read()

# Display the contents of `text`

print(text)
```

```
username,ip_address,time,date
tshah,192.168.92.147,15:26:08,2022-05-10
dtanaka,192.168.98.221,9:45:18,2022-05-09
tmitchel,192.168.110.131,14:13:41,2022-05-11
```

```
daquino,192.168.168.144,7:02:35,2022-05-08
eraab,192.168.170.243,1:45:14,2022-05-11
jlansky,192.168.238.42,1:07:11,2022-05-11
acook,192.168.52.90,9:56:48,2022-05-10
asundara,192.168.58.217,23:17:52,2022-05-12
jclark,192.168.214.49,20:49:00,2022-05-10
cjackson,192.168.247.153,19:36:42,2022-05-12
jclark,192.168.197.247,14:11:04,2022-05-12
apatel,192.168.46.207,17:39:42,2022-05-10
mabadi,192.168.96.244,10:24:43,2022-05-12
iuduike,192.168.131.147,17:50:00,2022-05-11
abellmas,192.168.60.111,13:37:05,2022-05-10
gesparza,192.168.148.80,6:30:14,2022-05-11
cgriffin,192.168.4.157,23:04:05,2022-05-09
alevitsk,192.168.210.228,8:10:43,2022-05-08
eraab,192.168.24.12,11:29:27,2022-05-11
jsoto,192.168.25.60,5:09:21,2022-05-09
```

2. Parsing Logs with Python

- Applied the `.split()` method to convert the log data into a list, where each line became a separate element.
- This made it easier to analyze login activity line by line instead of working with one long string.

```
[2]: # Assign `import_file` to the name of the text file that contains the
     security␣↪log file

     import_file = "data/login.txt"

     # The`with` statement
     # Use `open()` to import security log file and store it as a string

     with open(import_file, "r") as file:

       # Use `.read()` to read the imported file and store the result in a
       variable␣↪named `text`

       text = file.read()

     # Display the contents of `text` split into separate lines

     print(text.split())
```

```
['username,ip_address,time,date', 'tshah,192.168.92.147,15:26:08,2022-05-10',
'dtanaka,192.168.98.221,9:45:18,2022-05-09',
'tmitchel,192.168.110.131,14:13:41,2022-05-11',
'daquino,192.168.168.144,7:02:35,2022-05-08',
'eraab,192.168.170.243,1:45:14,2022-05-11',
'jlansky,192.168.238.42,1:07:11,2022-05-11',
'acook,192.168.52.90,9:56:48,2022-05-10',
'asundara,192.168.58.217,23:17:52,2022-05-12',
'jclark,192.168.214.49,20:49:00,2022-05-
10','cjackson,192.168.247.153,19:36:42,2022-05-12',
'jclark,192.168.197.247,14:11:04,2022-05-12',
'apatel,192.168.46.207,17:39:42,2022-05-10',
'mabadi,192.168.96.244,10:24:43,2022-05-12',
'iuduike,192.168.131.147,17:50:00,2022-05-11',
'abellmas,192.168.60.111,13:37:05,2022-05-10',
'gesparza,192.168.148.80,6:30:14,2022-05-11',
'cgriffin,192.168.4.157,23:04:05,2022-05-09',
'alevitsk,192.168.210.228,8:10:43,2022-05-08',
'eraab,192.168.24.12,11:29:27,2022-05-11',
'jsoto,192.168.25.60,5:09:21,2022-05-09']
```

3. Appending Missing Entries

- Detected a missing log entry and appended it back into the file using `.write()` in append (`"a"`) mode.
- Verified that the new entry appeared correctly at the end of the log file.

```
[3]: # Assign `import_file` to the name of the text file that contains the
 security␣↪log file

import_file = "data/login.txt"
# Assign `missing entry` to a log that was not recorded in the log file

missing_entry = "jrafael,192.168.243.140,4:56:27,2022-05-09"

# Use `open()` to import security log file and store it as a string
# Pass in "a" as the second parameter to indicate that the file is being
 opened␣↪for appending purposes

with open(import_file, "a") as file:

    # Use `.write()` to append `missing_entry` to the log file

    file.write(missing_entry)

# Use `open()` with the parameter "r" to open the security log file for
 reading␣↪purposes

with open(import_file, "r") as file:

    # Use `.read()` to read in the contents of the log file and store
 in a␣↪variable named `text`

    text = file.read()

# Display the contents of `text`
```

```
print(text)

username,ip_address,time,date
tshah,192.168.92.147,15:26:08,2022-05-10
dtanaka,192.168.98.221,9:45:18,2022-05-09
tmitchel,192.168.110.131,14:13:41,2022-05-11
daquino,192.168.168.144,7:02:35,2022-05-08
eraab,192.168.170.243,1:45:14,2022-05-11
jlansky,192.168.238.42,1:07:11,2022-05-11
acook,192.168.52.90,9:56:48,2022-05-10
asundara,192.168.58.217,23:17:52,2022-05-12
jclark,192.168.214.49,20:49:00,2022-05-10
cjackson,192.168.247.153,19:36:42,2022-05-12
jclark,192.168.197.247,14:11:04,2022-05-12
apatel,192.168.46.207,17:39:42,2022-05-10
mabadi,192.168.96.244,10:24:43,2022-05-12
iuduike,192.168.131.147,17:50:00,2022-05-11
abellmas,192.168.60.111,13:37:05,2022-05-10
gesparza,192.168.148.80,6:30:14,2022-05-11
cgriffin,192.168.4.157,23:04:05,2022-05-
09alevitsk,192.168.210.228,8:10:43,2022-
05-08
eraab,192.168.24.12,11:29:27,2022-05-11
jsoto,192.168.25.60,5:09:21,2022-05-09
jrafael,192.168.243.140,4:56:27,2022-05-09
```

4. Creating an Allow List of IP Addresses

- Created a new file (`allow_list.txt`) to document IP addresses permitted to access restricted resources.
- Used Python's `"w"` mode with `.write()` to store the list of approved IPs.
- Reopened the file in `"r"` mode to confirm that the allow-list was correctly written and saved.

```
[4]: # Assign `import_file` to the name of the text file that you want to create

import_file = "data/allow_list.txt"

# Assign `ip_addresses` to a list of IP addresses that are allowed
 to access␣→the restricted information

ip_addresses = "192.168.218.160 192.168.97.225 192.168.145.158
 192.168.108.13␣ →192.168.60.153 192.168.96.200 192.168.247.153
 192.168.3.252 192.168.116.187␣ →192.168.15.110 192.168.39.246"

# Display `import_file`

print(import_file)

# Display `ip_addresses`
```

```
print(ip_addresses)
```

```
data/allow_list.txt
192.168.218.160 192.168.97.225 192.168.145.158 192.168.108.13 192.168.60.153
192.168.96.200 192.168.247.153 192.168.3.252 192.168.116.187 192.168.15.110
192.168.39.246
```

[5]:

```
 # Assign `import_file` to the name of the text file that you want to create

import_file = "data/allow_list.txt"

# Assign `ip_addresses` to a list of IP addresses that are allowed
 to access␣ ↪the restricted information

ip_addresses  =  "192.168.218.160  192.168.97.225  192.168.145.158
 192.168.108.13␣   ↪192.168.60.153    192.168.96.200    192.168.247.153
 192.168.3.252  192.168.116.187␣ ↪192.168.15.110  192.168.39.246"

# Create a `with` statement to write to the text file

with open(import_file, "w") as file:

  # Write `ip_addresses` to the text file



  file.write(ip_addresses)
```

[6]:

```
 # Assign `import_file` to the name of the text file that you want to create

import_file = "data/allow_list.txt"

# Assign `ip_addresses` to a list of IP addresses that are allowed
 to access␣ ↪the restricted information

ip_addresses  =  "192.168.218.160  192.168.97.225  192.168.145.158
 192.168.108.13␣   ↪192.168.60.153    192.168.96.200    192.168.247.153
 192.168.3.252  192.168.116.187␣ ↪192.168.15.110  192.168.39.246"

# Create a `with` statement to write to the text file

with open(import_file, "w") as file:

    # Write `ip_addresses` to the text file

    file.write(ip_addresses)
```

```
# Create a `with` statement to read in the text file

with open(import_file, "r") as file:

    # Read the file and store the result in a variable named `text`

    text = file.read()

# Display the contents of `text`

print(text)
```

```
192.168.218.160 192.168.97.225 192.168.145.158 192.168.108.13 192.168.60.153
192.168.96.200 192.168.247.153 192.168.3.252 192.168.116.187 192.168.15.110
192.168.39.246
```

Outcome & Skills Gained

This project gave me hands-on experience in:

- File handling with Python (`read`, `write`, `append`)
- Parsing raw log data into structured formats
- Detecting and correcting missing entries in security logs
- Creating allow-lists to support access control policies

Through this activity, I strengthened my ability to **automate security tasks with Python**, a critical skill for incident response and log analysis.