NetworkSecurity_Project2_Report

網工所碩一 309552005 吳偉誠

A part

In this project, I use 「**Rule-Based Method**」, I don't use machine learning because I haven't touched it. This may become another problem for me.

Simply put, it is to analyze these files and use the observed characteristics to determine whether they have reached a certain value and then to determine which attack it is.

For These five attacks, I got some inspiration after observing the tips given by the teaching assistant and checking the content of the json file.
I process the string from the json file and extract the value that I think can be used as a characteristic, and calculate what percentage of it is executed.Finally, analyze the proportion of the characteristics of each attack collected in this Case to determine which attack event this Case may be.

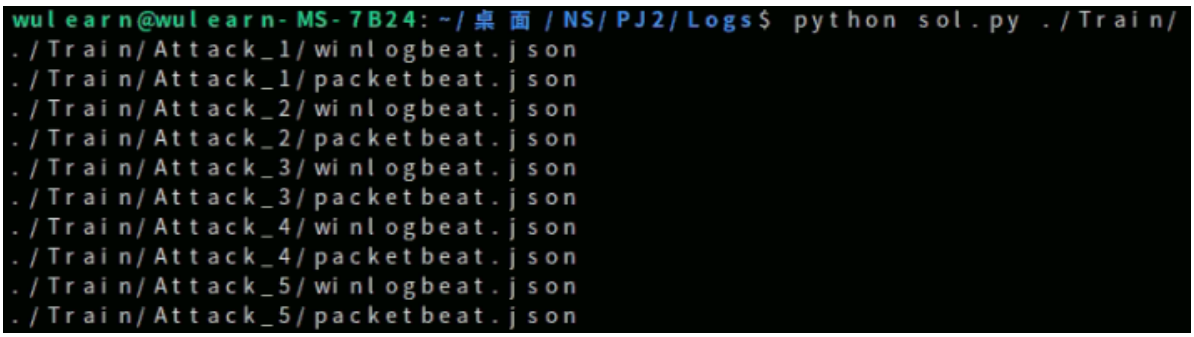
Next, I will elaborate on my entire `sol.py` architecture.

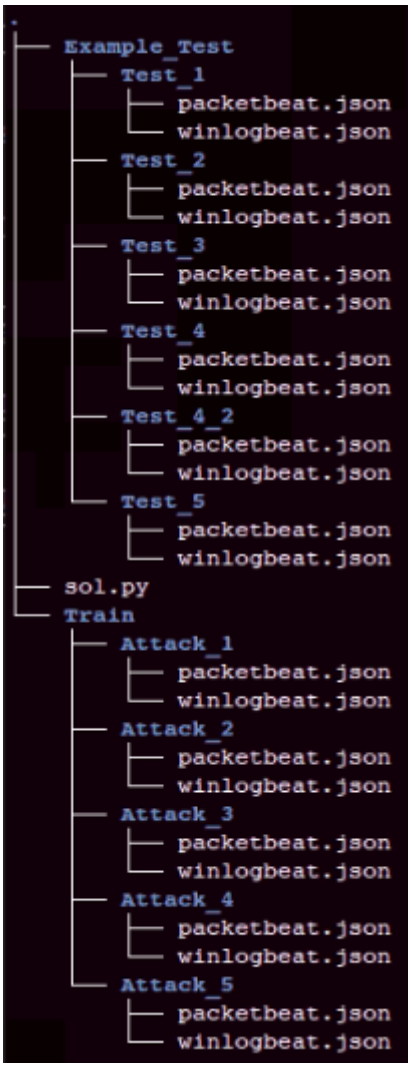
Architecture

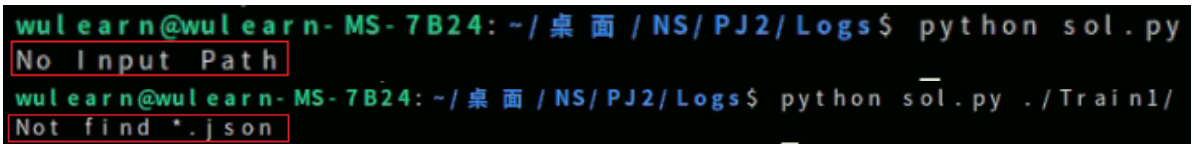
Environment: **python2.7**

(**Hint**:Some of the displays were annotated later, here is just for convenience)

- **First**, Collect files
    - After grabbing the path entered by the user, I went to visit all the folders and files under this path, and recorded the path of each JSON file

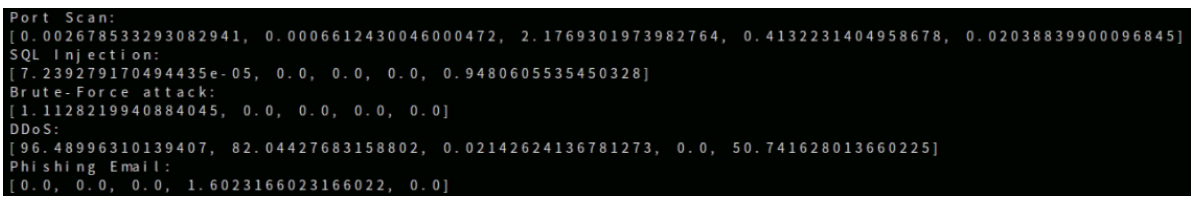


> **Note here!!**: the file to be observed must be on the same layer as `sol.py`
> ex:
>
> 
>
> Suppose there are two JSON files in each folder(winlogbeat.json,packetbeat.json)



> Of course also do some wrong case

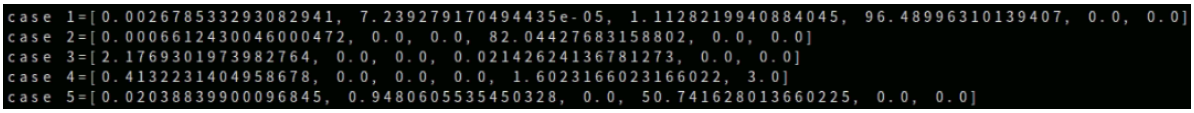
- **Second**, Analyze the JSON file
  - Open the content of each json file(in order) and grab the information I want, and calculate what percentage of the file it accounts for
  - Save them individually
    - Ex: `attack1_case_list=[99.9,80.0,0.0,30.8,...]` ([testcase1,testcase2,...], the unit is %)
  - Later will explain in detail which attack events are extracted which characteristics



> Five element,because has because has five test cases

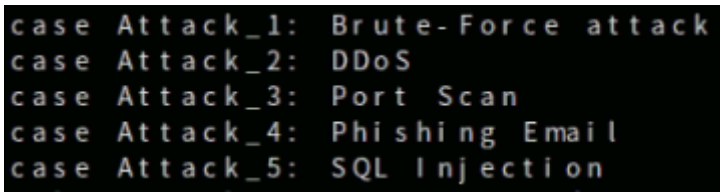
- **Third**, Integrate and collect the results of each testcase

- The data mentioned above is stored separately for the attack events, and then changed to store the number of each attack by each testcase.
  ex: `testcase1=[attack_1_%,attack_2_%,...]`



> It will be clearer to compare with the previous picture

- **Final**, To determine what the possible attack case, and display it
  - Here is to make a series of judgments by just getting the value of each of the five attack events (but be careful of the order, which will be mentioned in B part)



> The Attack_X is the name of the folder itself

Analysis of each attack event:

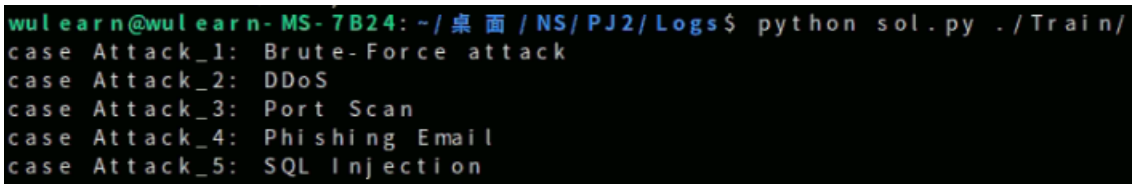
(**Hint**:Priority refers to the order of judgment with other attack events)

- **Brute-Force attack**:
  - Here I am grabbing whether the query of the URL in packetlog.json has the string "Login=Login"
  - If more than 0.5% then I conclude that it is a Brute-Force attack (**Priority is second only to SQL Injection, the premise is that the percentage of port: 80 is the highest**)
- **DDoS**:
  - Here, with TA's prompt, I am going to grab the "port" of "destination" in packetlog.json to see if it is equal to 80 (when the IP is equal to HOST)
  - If it is the highest proportion, then I conclude that it is a DDoS (**Priority is lower than Brute-Force attack and SQL Injection**)
- **Port Scan**:
  - It is very similar to DDOS, except to calculate the total number of ports scanned (same as the IP of HOST)
  - If it is the highest proportion,or the port total is more than 300, then I conclude that it is a Port Scan(**Priority is lower than Brute-Force attack and SQL Injection and DDoS**)
- **Phishing Email**:
  - To be honest here, I didn't find any characteristics, so I just counted the number of "cmd.exe" in "ProcessName" of the "event_data" of the "winlog" in winlogbeat.json, and the number of "tls" in packetbeat.json
  - If it is the highest proportion, or the percentage of "TLS" more than 1%, then I conclude that it is a Port Scan(**Lowest priority**)
- **SQL Injection**
  - The judgment here is more casual, I only calculate the percentage of the "UNION" string (not case sensitive) in the query of the URL in packetlog.json
  - If more than 0.1% then I conclude that it is a SQL Injection (**Highest priority, the premise is that the percentage of port: 80 is the highest**)
- **Unknow**

- If none of the above matches, it will display **Unknow**

Result:

- **Train**:



> Because now I know what kind of attack these Train represent, I use these **Train names** to replace these **attack names** (follow spec)

So:

- **Example_Test**:

> **Hint**:TA added Test_4_2 after

The accuracy may not be so precise(For example: can increase the observation feature value and the identification method and then improve it), but it can be distinguished for these cases! **But I have tried my best!**

B part

There are some interesting things and some problems solved in this porject:

Interesting things

1. Python does not read the files in order, must add `sorted()` before they can be read in the order of file names

2. Python's dict cannot directly use `dict [" key"]` when searching for keys. You must first use `dict.get ("key")` to determine whether there is this value.

3. Originally I wanted to use "process of elimination" to handle Phishing Email, but when the TA said that the test data may not only have one case for each attack when the demo was used, I dispelled this idea.

Problems

1. The file is too large, causing the computer to crash when the program is reading the file
sol: It was originally read all the content in the *.json once , but later changed to a line after reading and judging (calculation feature) and then read the next line...

2. Some attacks will be related, so priority issues must be addressed
sol: For example, Brute-Force attack will also access port80 (a feature of DDoS), so it was discovered that the order issue should also be dealt with. Later, some judgments were added to solve it.