Cyber Security Lab 4: Forensic Analysis of Logfiles Using Python.

Due 5 June 2024.

This is a Group Lab.

Software needed

PyCharm or any other IDE of choice

Description

1. **Define Predefined Criteria:**
    - Before writing the Python code, browse through the log file and record predefined criteria that indicate suspicious activity, and store in a file. Examples of predefined criteria include:
        - Multiple failed login attempts (e.g., HTTP status code 401).
        - Access to sensitive directories or files (e.g., "/admin", "/wp-admin", "/phpmyadmin").
        - Unusual user-agent strings (e.g., non-standard or uncommon user-agents).
2. **Iterate Through Log Entries:**
    - Use a loop to iterate through each entry in the access log file.
    - For each entry, extract relevant information such as IP address, requested URL, response code, and user-agent.
3. **Apply Predefined Criteria:**
    - Implement conditional statements to check if the extracted data meets the predefined criteria for suspicious activity.
    - For example:

- Check if the response code indicates a failed login attempt (e.g., status code 401 or 403).
- Check if the requested URL matches any sensitive directories or files.
- Check if the user-agent string is unusual or uncommon.

4. **Flag Suspicious Activity:**
   - If the entry meets any of the predefined criteria for suspicious activity, flag it for further investigation.
   - Store the flagged entries in a separate data structure (e.g., a list) for easy access and analysis.
5. **Output:**
   - Print the details of flagged entries to the console for manual inspection.
   - Generate summary statistics (number of times each predefined criteria occurred, how many different unusual user agents etc) to highlight patterns of suspicious activity.
6. Conclusion:

   Analyse the output statistics with your group and state what kind of cyber attack that might have taken place during the time the log file was generated.

Deliverables

Submit a PDF of a lab report containing your analysis, list of predefined criteria, and complete python script. Include the names of all group members that participated in this lab activity.

Submit your pdf document on Lea.

What are User agents?

Take an example from the provided access log file:

`14.139.187.130 - - [01/Jan/2017:02:16:51 -0800] "GET / HTTP/1.1" 200 10267 "https://www.google.co.in/" "Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/55.0.2883.87 Safari/537.36"`

In this example, the user agent string is:

`Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/55.0.2883.87 Safari/537.36`

Let's break down this user agent string:

- `Mozilla/5.0`: This indicates that the client is compatible with Mozilla browsers, such as Firefox.
- `(Windows NT 6.1):` This part indicates that the client is running on a Windows operating system, specifically Windows 7 (Windows NT version 6.1).
- `AppleWebKit/537.36`: This indicates that the client is using the WebKit rendering engine, commonly used in browsers like Chrome and Safari.
- `(KHTML, like Gecko)`: This part indicates compatibility with Gecko-based browsers, such as Firefox.
- `Chrome/55.0.2883.87`: This indicates that the client is using Chrome version 55.0.2883.87.
- `Safari/537.36`: This indicates that the client is also compatible with Safari, and the version of Safari's rendering engine is 537.36.

So, based on this user agent string, we can infer that the client is using the Chrome browser on a Windows 7 operating system, with compatibility for both Chrome and Safari. This information can be useful for web analytics, as it provides insights into the types of browsers and operating systems used by visitors accessing the website.