Web Log Analysis/Research/Questions

By: Ryan Stewart

Web Log Analysis

In today's digital landscape, **most services** are web-based, making web services the **primary target for attackers**. SOC Analysts play a crucial role in accurately analyzing web logs to thwart potential threats. Major web servers like Microsoft IIS, Apache, and Nginx generate logs with similar content structures.

Sample Web Server Log:

```
71.16.45.142 - - [12/Dec/2021:09:24:42 +0200] "GET /?id=SELECT+*+FROM+users HTTP/1.1" 200 486 "-" "curl/7.72.0"
```

Analysis:

- **Source IP:** 71.16.45.142
- **Date:** 12/Dec/2021:09:24:42 +0200
- **Request Method:** GET
- **Requested URL:** **/?id=SELECT+*+FROM+users**
- **Version Info:** HTTP/1.1
- **Server Response:** 200
- **Data Size:** 486
- **User-Agent Info:** **curl/7.72.0**

Request Method:

Indicates the method used in the web request. Common methods include:

- **GET:** Retrieve data from the server.
- **POST:** Send data to the server (e.g., pictures, videos).
- **DELETE:** Delete data on the server.
- **PUT:** Send data to the server, creating or updating files.
- **OPTIONS:** Specifies methods the server accepts.

Note: Web servers typically do not log the content of data sent by POST or PUT requests by default.

Requested URL:

Specifies the directory/file on the server for the request. Examining URLs is crucial for detecting attacks; for instance, "SELECT+*+FROM+users" suggests a "SQL Injection" attack.

HTML Injection	/htmli_get.php?firstname=<h1>TWAFTEST<%2Fh1>&lastname=<h1>TWAFTEST<%2Fh1>&form=submit
XSS Attack	/xss_get.php?firstname=<script>alert%28test%29&lastname=<%2Fscript>&form=submit
SQL Injection	/sqli_1.php?title=%25iron%27+union+select+1%2C2%2C3%2C4%2C5%2C6%2C7--+-+%25%27&action=search
Directory Traversal Attack	/script.php?page=../../../../../../../../etc/passwd

Server Response:

The server responds to requests with numeric codes indicating success or failure:

- **200 (OK):** Successful request and response.
- **404 (Not Found):** Requested content could not be found.
- **500 (Server Error):** Server couldn't interpret the request.

User-Agent Information:

Indicates the application used for the request. Helps differentiate real users from automated tools. Common tools include "nikto," "nessus," and "nmap."

Web Log Analysis:

- Detects web attacks (SQL Injection, XSS Attack, Code Injection, Directory Traversal).
- Identify top requesting IPs and most requested URLs.
- Analyze the most received HTTP response codes.
- Detects suspicious method usage (e.g., PUT, DELETE).

Sample Web Request Analysis:

-`192.168.8.11/bwapp/sqli_1.php?title=%25iron%27+union+select+1%2Cuser%28%29%2C3%2C4%2C5%2C6%2C7--+-+%25%27&action=search`

Decoded Web Request:
- `192.168.8.11/bwapp/sqli_1.php?title=%iron' union select 1,user(),3,4,5,6,7-- -%'&action=search`

Web Log of the Attack:

192.168.8.54 - - [29/Jun/2022:07:42:48 +0300] "GET /bwapp/sqli_1.php?title=%25iron%27+union+select+1%2Cuser%28%29%2C3%2C4%2C5%2C6%2C7--+-+%25%27&action=search HTTP/1.1" **200** 13539 "http://**192.168.8.11**/bwapp/sqli_1.php?title=&action=search" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.0.0 Safari/537.36"

Log Analysis:

- The attack was a SQL injection from IP 192.168.8.54 and was successful.
- URL patterns like "union," "select," and a server response of 200 indicate a SQL injection.
- The analyst confirmed access to the database information, revealing the user "root@localhost."

Quiz Questions:

1. Which of the following is not an HTTP request method?

- A) GET
- B) OPTIONS
- C) TRACE
- D) HEAD
- E) BLOCK

2. Are there web logs with "Nmap Scripting Engine" in the user-agent information among the web requests made? - True

```
1331901000.010000       CKnDAp2ohlvN6rpiXl      192.168.202.79  50467   192.168.
229.251 80      1       HEAD    192.168.229.251 /DEASLog03.nsf  -       Mozilla/
5.0 (compatible; Nmap Scripting Engine; http://nmap.org/book/nse.html)  0
0       404     Not Found       -       -       -       (empty) -       -
```

- *cat http.log |grep "Nmap Scripting Engine" | more*

3. Are there any SQL injection attacks with a status code of 200?

```
1331901002.730000       CbQUjj4fRZ5e5OHVtl      192.168.202.79  50770   192.168.229.251 80
mpatible; Nmap Scripting Engine; http://nmap.org/book/nse.html) 0       195     200     OK
Urc     text/html
```

- *cat http.log | egrep -i 'select|union|insert|concat' |grep '200' |grep 'OK' |more*

4. Identify the highest requesting IP address.

- 192.168.203.63

- cat http.log | awk '{print $1}' | sort | uniq -c | sort -rn | head -n 1

1. **`cat http.log`: Reads the contents of the `http.log` file.**
2. **`awk '{print $1}'`: Extracts the first column, which usually contains the IP addresses.**
3. **`sort`: Sorts the IP addresses.**
4. **`uniq -c`: Counts the occurrences of each unique IP address.**
5. **`sort -rn`: Sorts the counts in reverse numerical order (highest count first).**
6. **`head -n 1`: Selects the first line (the IP address with the highest count).**

5. How many web requests are made with the "DELETE" method in total?

- 223

- cat http.log | awk '{print $8}' | grep -w DELETE | wc -l

- `cat http.log`: Concatenates and displays the contents of the `http.log` file.
- `awk '{print $8}'`: Uses `awk` to extract the 8th field (HTTP method) from each line.
- `grep -w DELETE`: Filters lines that contain the exact word "DELETE" in the 8th field.
- `wc -l`: Counts the number of lines that match the pattern.