WAF Analysis/Research/Questions

By: Ryan Stewart

WAF Log Analysis

Web Application Firewall (WAF) is a crucial technology for securing web-based applications. Analyzing firewall or IDS/IPS logs alone is often insufficient for detecting web-based attacks, mainly due to SSL offload issues and the need to control data in the payload of web requests.

SSL Offload involves decrypting SSL-encrypted traffic, aiming to reduce the load and increase performance, as well as decrypting the traffic/request to make the content visible and controllable from a security standpoint. This process enables the detection and prevention of invisible attack vectors in encrypted traffic.

In networks equipped with WAF, end-user requests reach WAF first over the internet. The WAF inspects the request and decides whether it will be transferred to the Web Server or not. A significant advantage of WAFs is SSL Off-loading, allowing the examination of HTTPS traffic content. WAF without SSL Offloading capability cannot provide effective protection as it won't inspect the payload (data) part of HTTPS communication.

F5 Big-IP, Citrix, Imperva, and Forti WAF products are well-known examples of WAF solutions. Additionally, cloud-based solutions like Cloudflare, Akamai, and AWS WAF are used as cloud WAF solutions.

WAF systems generally handle web access requests on publicly faced systems, making them the first line of defense against web attacks. WAF logs are crucial for SOC Analysts to detect suspicious activities. Analysts need to understand their location on the network clearly when analyzing WAF logs, which provide records of all web requests and information about detected or blocked web attacks. While examining alerts, the reputation of the source IP address should be analyzed, and similar activities from the source IP in other log sources (such as IDS/IPS, Firewall) should be investigated.

Sample WAF Log:

```
<json>
date=2022-01-26 time=19:47:26 log_id=20000008 msg_id=000018341360 device_id=FVVM08
vd="root" timezone="(GMT+3:00)Istanbul" timezone_dayst="GMTg-3" type=attack
main_type="Signature Detection" sub_type="SQL Injection" severity_level=High proto=tcp
service=https/tls1.2 action=Alert policy="Alert_Policy" src=19.6.150.138 src_port=56334
dst=172.16.10.10 dst_port=443 http_method=get http_url="?v=(SELECT
(CHR(113)||CHR(120)||CHR(120)||CHR(118)||CHR(113))||(SELECT (CASE WHEN (1876=1876)
THEN 1 ELSE 0 END))::text" http_host="app.letsdefend.io" http_agent="Mozilla/5.0 (Windows;
U; Windows NT 5.1; en-US; rv:1.9b1) Gecko/2007110703 Firefox/3.0b1"
msg="Parameter(Password) triggered signature ID 030000136" signature_subclass="SQL
Injection" signature_id="030000136" srccountry="Germany" attack_type="SQL Injection"
```

Log Details:

- **date:** Date information
- **time:** Time information
- **type:** Log type
- **main_type:** Detection type
- **sub_type:** Detected activity detail
- **severity_level:** Incident severity level
- **proto:** Protocol
- **service:** Service information
- **action:** Action taken
- **policy:** Rule name
- **src:** Source IP address
- **src_port:** Source port address
- **dst:** Destination IP address
- **dst_port:** Destination port address
- **http_method:** HTTP request method
- **http_url:** URL requested
- **http_host:** Host requested
- **http_agent:** User-agent info
- **msg:** Message related to the incident
- **signature_subclass:** Signature class
- **srccountry:** Source IP country
- **attack_type:** Attack type

Log Analysis:

When analyzing the provided WAF log, check the source and target IP information, especially for a high-severity level SQL Injection attack. Verify WAF's response to this request, particularly if the reported attack is a generic (SQL injection, XSS, etc.) web attack. If the WAF did not block this request, the response returned by the application should be checked. The response code of the application (IIS, Apache, Nginx, etc.) is crucial and should be investigated. If the application responded with a code 200 for an attack that WAF could not prevent, it means the attack reached the web server and returned a successful response. In some cases, the application returns code 200 while it should return code 404 due to technical deficiencies, considering these as **false-positives** for the relevant requests.

Application Responses:

- **200 (OK):** Request received successfully, and the response was returned.
- **301 (Permanent Redirect):** Request was redirected to a different location.
- **403 (Forbidden):** Data requested is not allowed.
- **404 (Not Found):** Requested content could not be found.
- **503 (Service Unavailable):** Server cannot respond.

Response Code Categories:

- Informational responses (100–199)
- Successful responses (200–299)
- Redirection messages (300–399)
- Client error responses (400–499)
- Server error responses (500–599)

The connection request in the provided WAF log was blocked due to recognized malicious signatures, generating an alert. The request came from IP address 19.6.150.138 to port 443 of the 172.16.10.10 host behind the WAF. The applied policy for requests matching this signature on the WAF is "Alert_Policy," and the action is set to "alert" (monitoring mode). Therefore, we can say that the request reached the destination host.

If the reported attack by WAF aims to detect vulnerabilities, details of the vulnerability should be examined. For example, if the web application runs on ASP, and the vulnerability detection is a PHP application-specific scan, such a vulnerability may not be reported. However, it is good practice to take action against the IP address conducting scanning activity. The best action is to block inbound requests at the first security device at the gateway where inbound requests first interact with the network.

WAF Log Utilization:

WAF logs are instrumental in analyzing the following detections:

- Detection of known web vulnerabilities
- Detection of various web attacks like SQL Injection, XSS Attack, Code Injection, Directory Traversal
- Detection of suspicious method usage such as **PUT, DELETE**
- Top requesting IP address information
- Most requested URL information

Request Method:

Indicates the method used for 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 to create or update files
- **OPTIONS:** Specifies which methods the server accepts

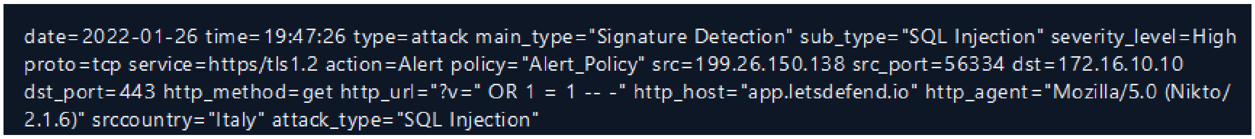
Quiz Questions:

1. Which of the following actions should be taken when examining the above WAF log?

A) Simulate whether the attack was successful or not.

B) Block requests with high source port numbers on the firewall.

C) Review the SSL certificate.



```
date=2022-01-26 time=19:47:26 type=attack main_type="Signature Detection" sub_type="SQL Injection" severity_level=High
proto=tcp service=https/tls1.2 action=Alert policy="Alert_Policy" src=199.26.150.138 src_port=56334 dst=172.16.10.10
dst_port=443 http_method=get http_url="?v=" OR 1 = 1 -- -" http_host="app.letsdefend.io" http_agent="Mozilla/5.0 (Nikto/
2.1.6)" srccountry="Italy" attack_type="SQL Injection"
```

2. Which of the following is not true according to the WAF log above?

A) The request has reached the server.

B) The server responded to the request successfully.

C) According to the log record, the request came through the automated web browsing tool.

D) The request method is GET.