**Lab Report – Path Delimiter Cache Poisoning**

**Objective:  
Identify how the origin server and cache treat path delimiters differently and exploit the discrepancy to obtain another user’s API key.**

**Identify a target endpoint**

1. In Burp's browser, log in to the application using the credentials wiener:peter.

A screenshot of a computer

AI-generated content may be incorrect.

1. Notice that the response contains your API key.

A screenshot of a computer

AI-generated content may be incorrect.

**Identify path delimiters used by the origin server**

1. In **Proxy > HTTP history**, right-click the “GET /my-account” request and select “**Send to Repeater**”.

A screenshot of a computer

AI-generated content may be incorrect.

1. Go to the **Repeater** tab. Add an arbitrary segment to the path. For example, change the path to /my-account/abc.
2. Send the request. Notice the 404 Not Found response with no evidence of caching. This indicates that the origin server doesn't abstract the path to /my-account.

A screenshot of a computer

AI-generated content may be incorrect.

1. Remove the arbitrary segment and add an arbitrary string to the original path. For example, change the path to /my-accountabc.
2. Send the request. Notice the 404 Not Found response with no evidence that the response was cached. You'll use this response as a reference to help you identify characters that aren't used as delimiters.
3. Right-click the request and select **Send to Intruder**.
4. Go to the **Intruder** tab. Make sure that **Sniper attack** is selected and add a payload position after /my-account as follows: /my-account§§abc.
5. In the **Payloads** side panel, under **Payload configuration**, add a list of characters that may be used as delimiters.
6. Under **Payload encoding**, deselect **URL-encode these characters**.
7. Click “**Start attack”**. The attack runs in a new window.
8. When the attack finishes, sort the results by **Status code**. Notice that the “;” and “?” characters receive a “200” response with your API key. All other characters receive the “404” Not Found response. This indicates that the origin server uses “;” and “?” as path delimiters.

**Investigate path delimiter discrepancies**

1. Go to the **Repeater** tab that contains the “/my-accountabc” request.
2. Add the “?” character after “/my-account” and add a static file extension to the path. For example, update the path to “/my-account?abc.js”.
3. Send the request. Notice that the response doesn't contain evidence of caching. This may indicate that the cache also uses “?” as a path delimiter.

A screenshot of a computer

AI-generated content may be incorrect.

1. Repeat this test using the “;” character instead of “?”. Notice that the response contains the “X-Cache: miss” header.

A screenshot of a computer

AI-generated content may be incorrect.

1. Resend the request. Notice that the value of the “X-Cache” header changes to “hit”. This indicates that the cache doesn't use “;” as a path delimiter and has a cache rule based on the “.js” static extension. You can use this payload for an exploit.

**Craft an exploit**

1. In Burp's browser, click “**Go to exploit server”**.

A screenshot of a computer

AI-generated content may be incorrect.

1. In the “**Body”** section, craft an exploit that navigates the victim user “carlos” to the malicious URL you crafted earlier. Make sure to change the arbitrary string, so the cache creates a unique key and “carlos” caches their account details instead of receiving your previously cached response:

<script>document.location="https://YOUR-LAB-ID.web-security-academy.net/my-account;wcd.js"</script>

1. Click “**Deliver exploit to victim”**. When the victim views the exploit, the response they receive is stored in the cache.
2. Go to the URL that you delivered to “carlos”:

https://YOUR-LAB-ID.web-security-academy.net/my-account;wcd.js

1. Notice that the response includes the API key for carlos. Copy this. A screenshot of a computer

   AI-generated content may be incorrect.
2. Click “**Submit solution”**, then submit the API key for carlos to solve the lab.