Host-only

In this lesson, we'll study host-only cookies.

When a server does not include a `Domain` directive the cookie is to be considered `host-only`, meaning that its validity is restricted to the current domain only.

This is a sort of default behavior from browsers when they receive a cookie that does not have a `Domain` set. You can find a small example I wrote at github.com/odino/wasec/tree/master/cookies. It's a simple web app that sets cookies based on URL parameters, and prints cookies on the page, through some JavaScript code.

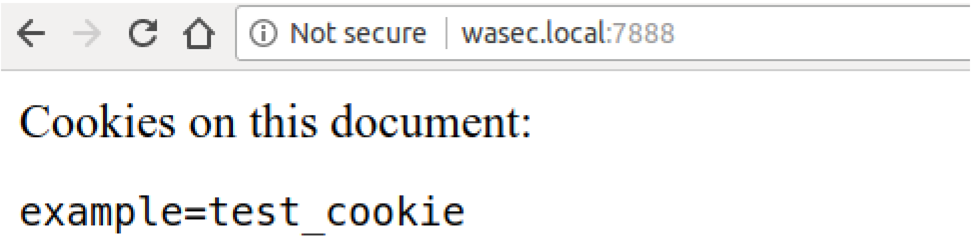
```
<html>
  <div id="output"/ >
  <script>
    let content = "none";

    if (document.cookie) {
      let cookies = document.cookie.split(';')
      content = ''

      cookies.forEach(c => {
        content += "<p><code>" + c + "</code></p>"
      })
    }

    document.getElementById('output').innerHTML = "Cookies on this document: <div>" + content + "</div>"
  </script>
<html>
```

If you follow the instructions in the `README` you will be able to access a webserver at wasec.local:7888, which illustrates how `host-only` cookies work.



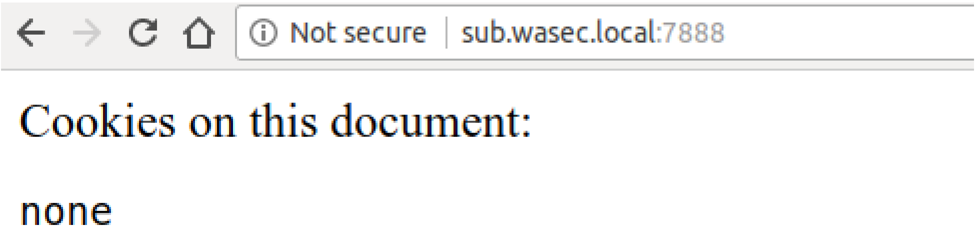
Not secure | wasec.local:7888

Cookies on this document:

example=test_cookie

Cookies on main domain

If we try to visit a subdomain, the cookies we set on the main domain are not going to be visible. Try navigating to sub.wasec.local:7888.



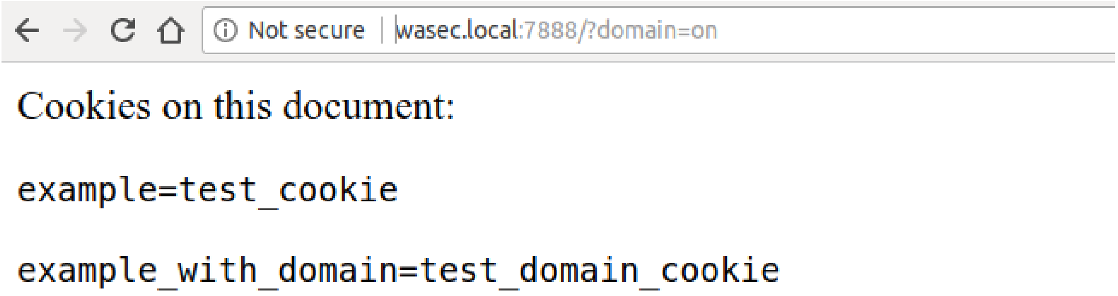
Not secure | sub.wasec.local:7888

Cookies on this document:

none

Main domain cookies do not exist on subdomain

A way to circumvent this limitation is to specify the `Domain` directive of the cookie, something we can do by visiting wasec.local:7888/?domain=on.



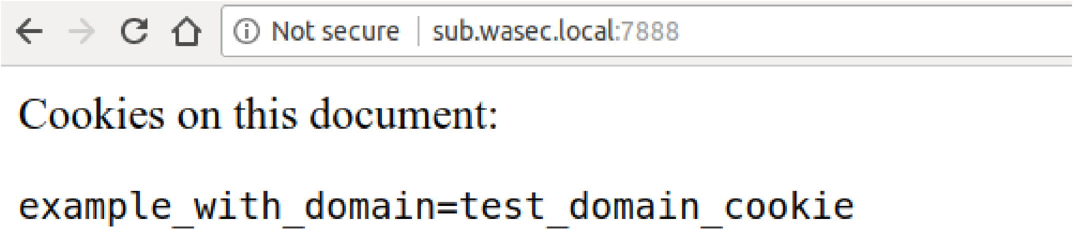
Not secure | wasec.local:7888/?domain=on

Cookies on this document:

example=test_cookie

example_with_domain=test_domain_cookie

If we have a look at the application running on the subdomain, we will now be able to see cookies set on the parent domain, as they use `Domain=wasec.local`, which allows any domain under `wasec.local` to access the cookies.



Cookies on this document:

example_with_domain=test_domain_cookie

Cookies on subdomain

In HTTP terms, this is what the responses sent from the server look like.

```
$ curl -I http://wasec.local:7888
HTTP/1.1 200 OK
Set-Cookie: example=test_cookie
Date: Fri, 24 Aug 2018 09:34:08 GMT
Connection: keep-alive
$ curl -I "http://wasec.local:7888/?domain=on"
HTTP/1.1 200 OK
Set-Cookie: example=test_cookie
Set-Cookie: example_with_domain=test_domain_cookie;Domain=wasec.local
Date: Fri, 24 Aug 2018 09:34:11 GMT
Connection: keep-alive
```

In the next lesson, we'll study Supercookies.