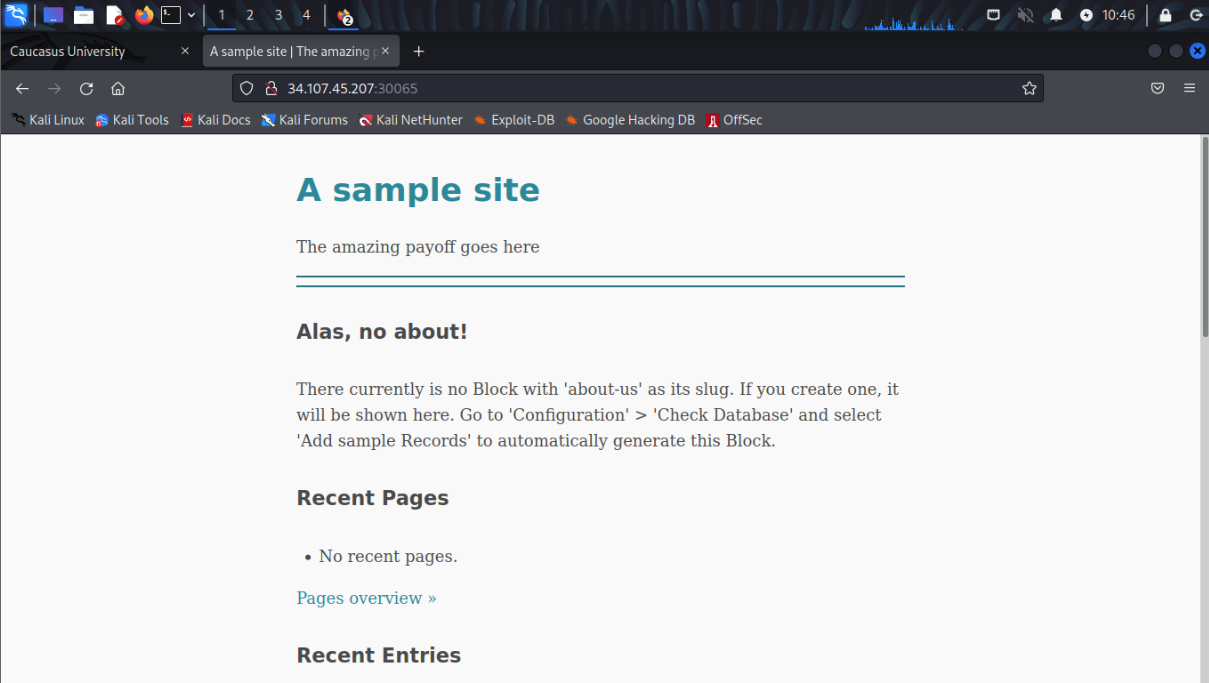
სიმულაციური აქტივობა 1

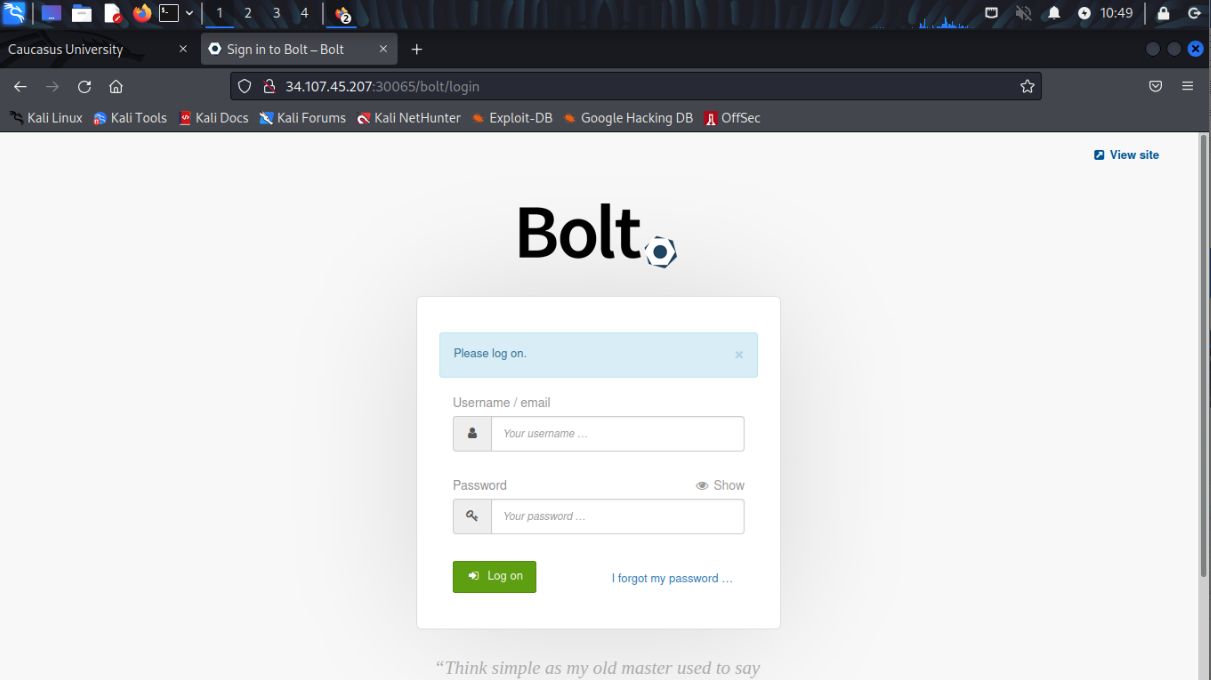
ლანა ეგიკოვა

Lab: Bolt

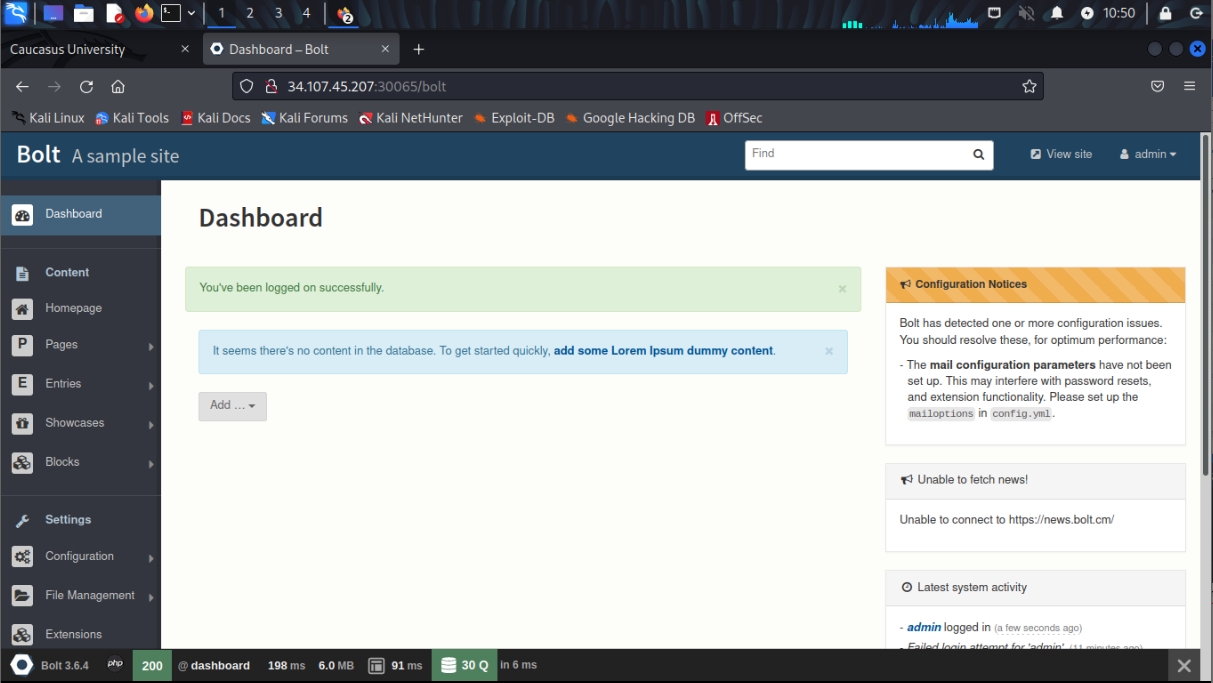
შევდივარ მითითებულ აიპი ადრესზე და მხვდება ასეთი გვერდი



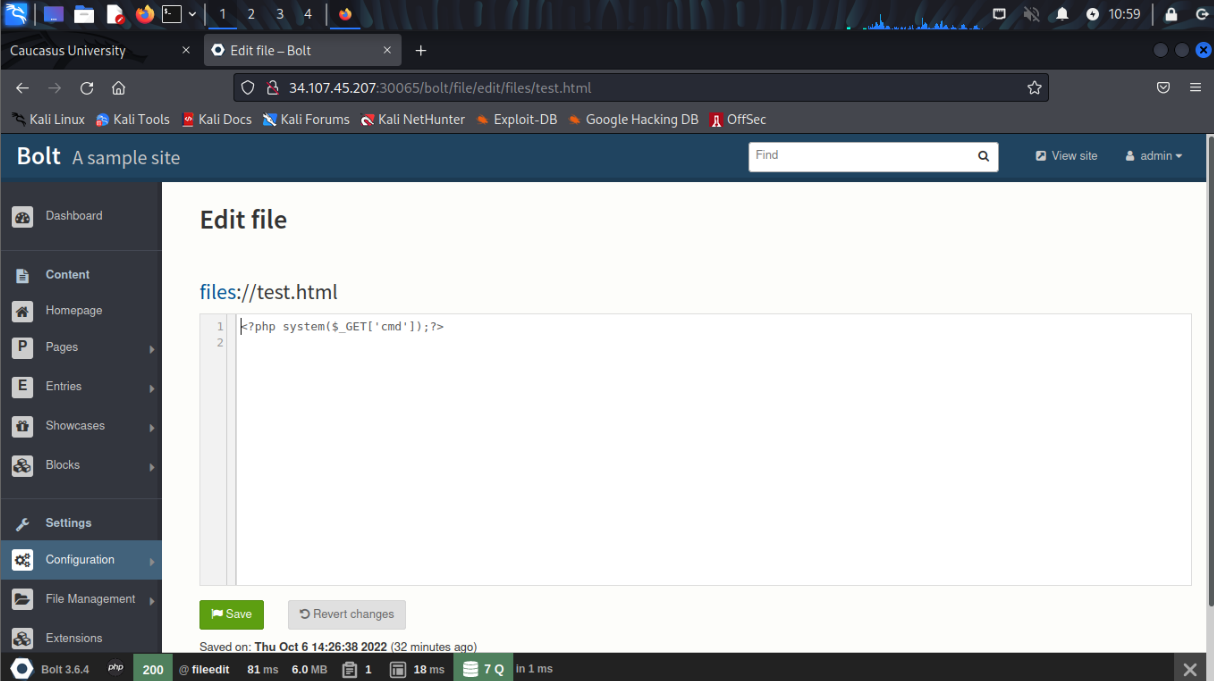
ვნახულობ გვერდის სორსს, შემდეგ url-ში ვამატებ სიტყვა bolt-ს რადგან სორსში არის მითითება href=bolt.cm და ამის შემდეგ იხსნება ასეთი გვერდი სარეგისტრაციო პანელით



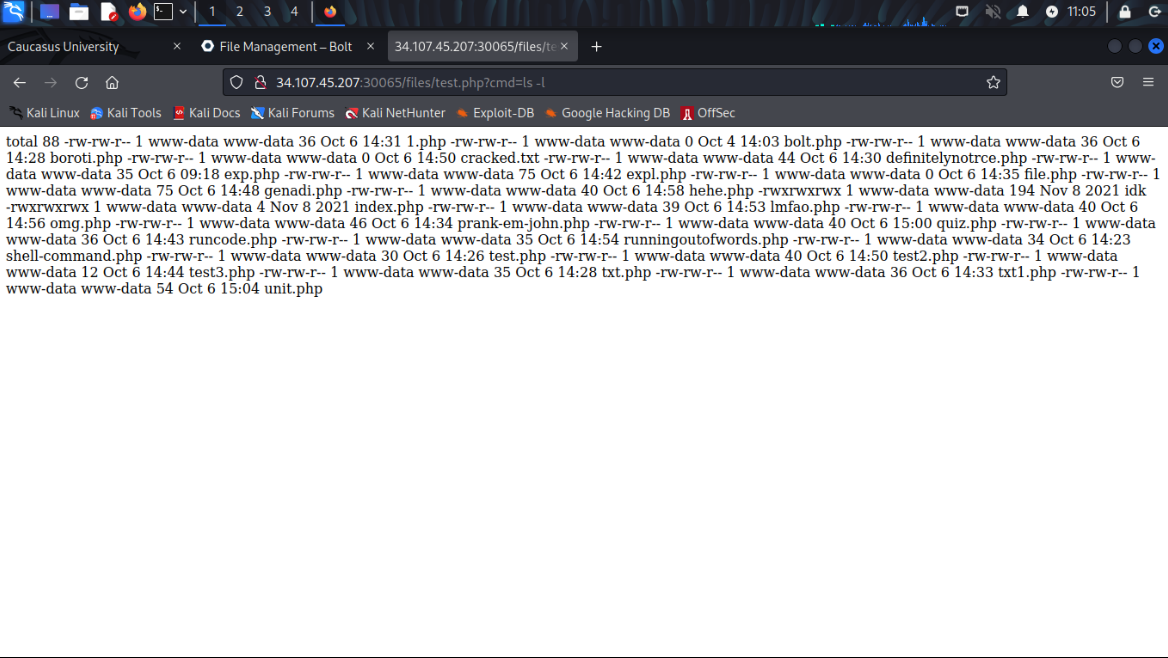
შემდეგ ვვარაუდობთ შესაძლო იუზერს და პაროლს და რამდენიმე ცდის შედეგად იხსნება გვერდი



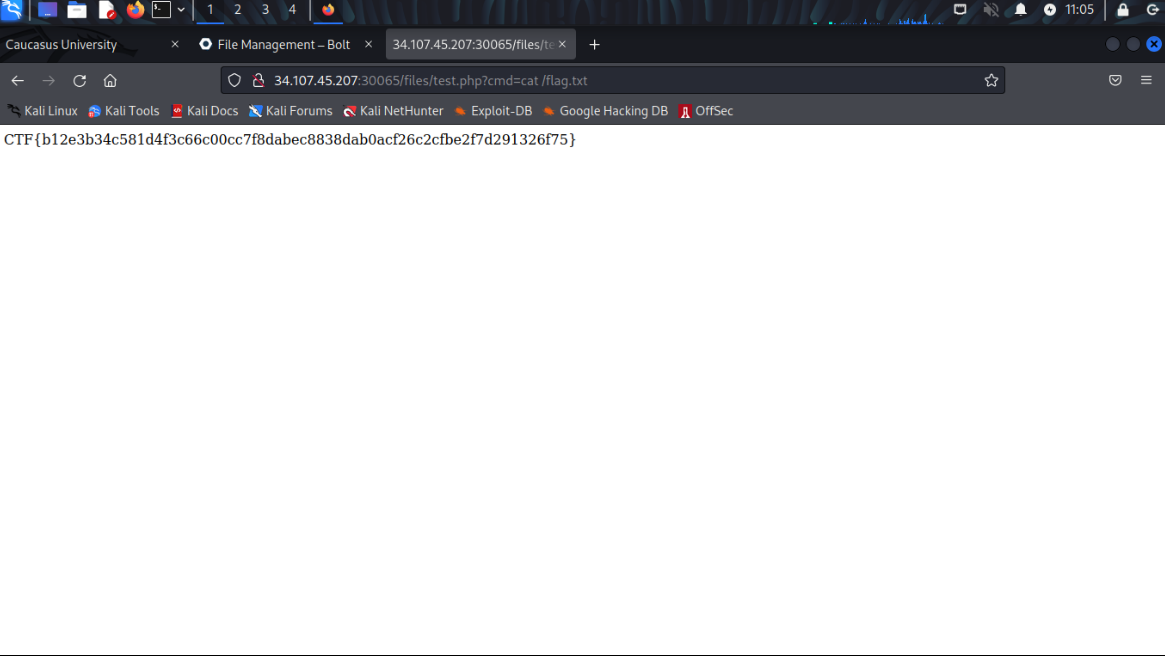
ამის შემდეგ File management-ში გადავდივართ რო ვნახო შეგვიძლია თუ არა რაიმე php ფაილის ატცვირთვა, ამისთვის ვქმნით ფაილს test.php , მაგრამ გვერდი ვერ აღიქვამს ამ ფაილს, ექსთენშენის გადარქმევის შედეგად php->html ატვირთვა შესაძლებელი გახდა



შემდეგ უკვე ატვირთულ html ფაილს ვუცვლით php-ზე და ვაკლიკებთ მასზე, თუმცა გვერდი მაინც ცარიელია ამიტომ ვცდი ასეთი ბრძაწნების მიწერას

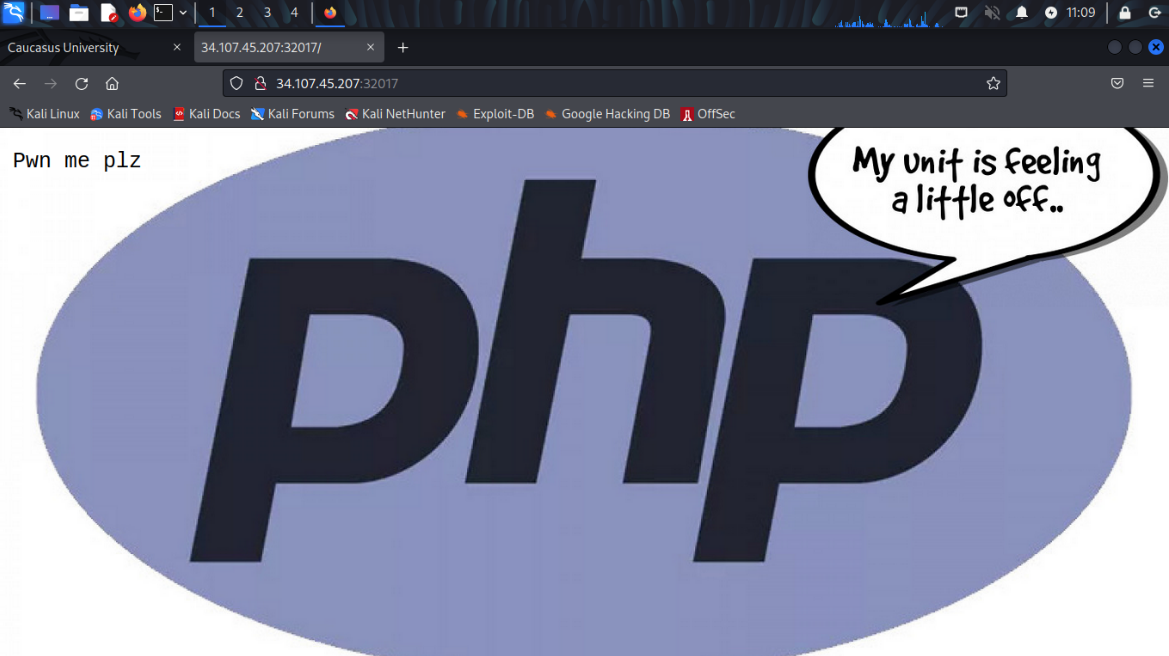


შემდეგ ამის და აი ჩვენი ფლეგიც

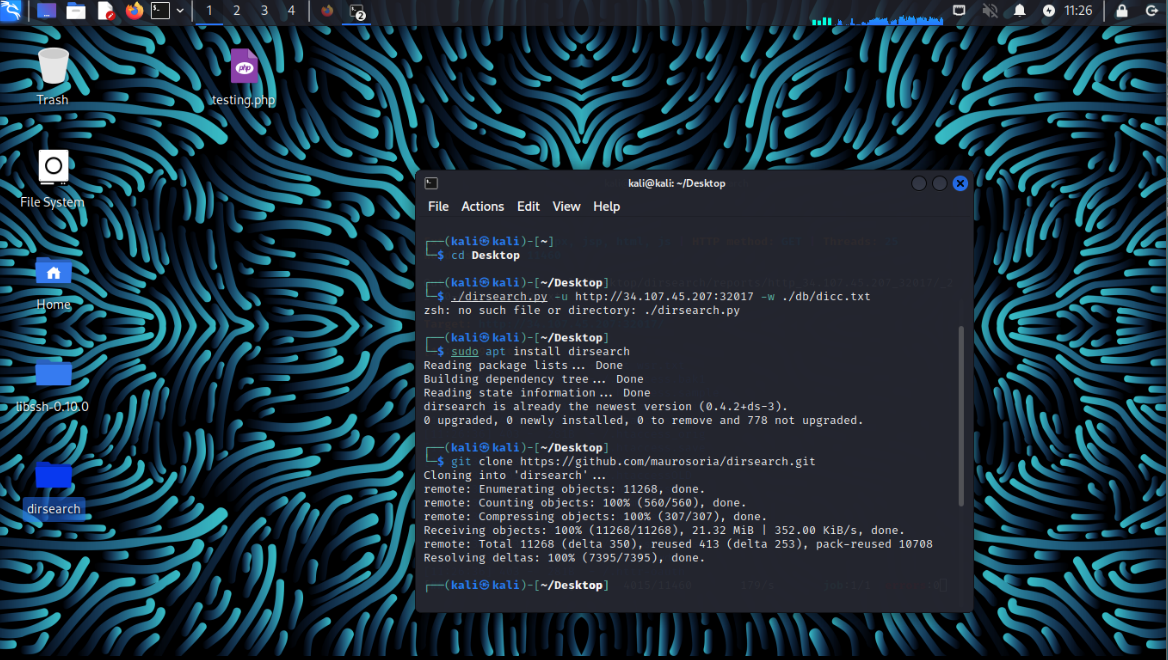


Lab: Php-unit

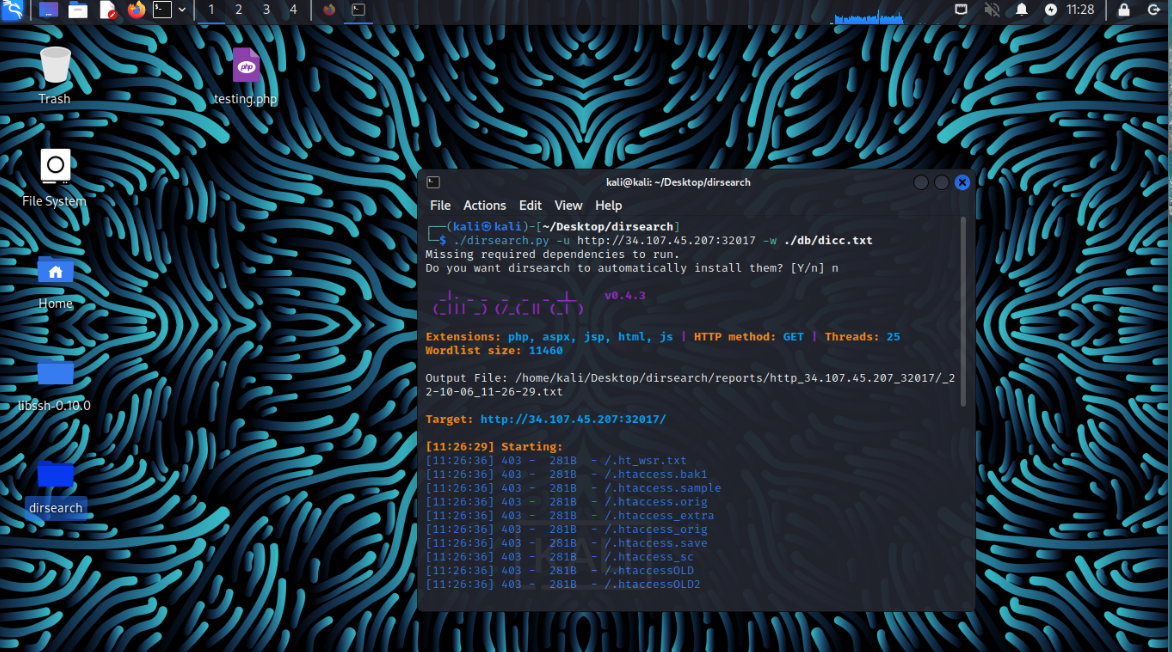
შევდივარ მითითებულ მისამართზე და მხვდება რაღაც გვერდი სურათით



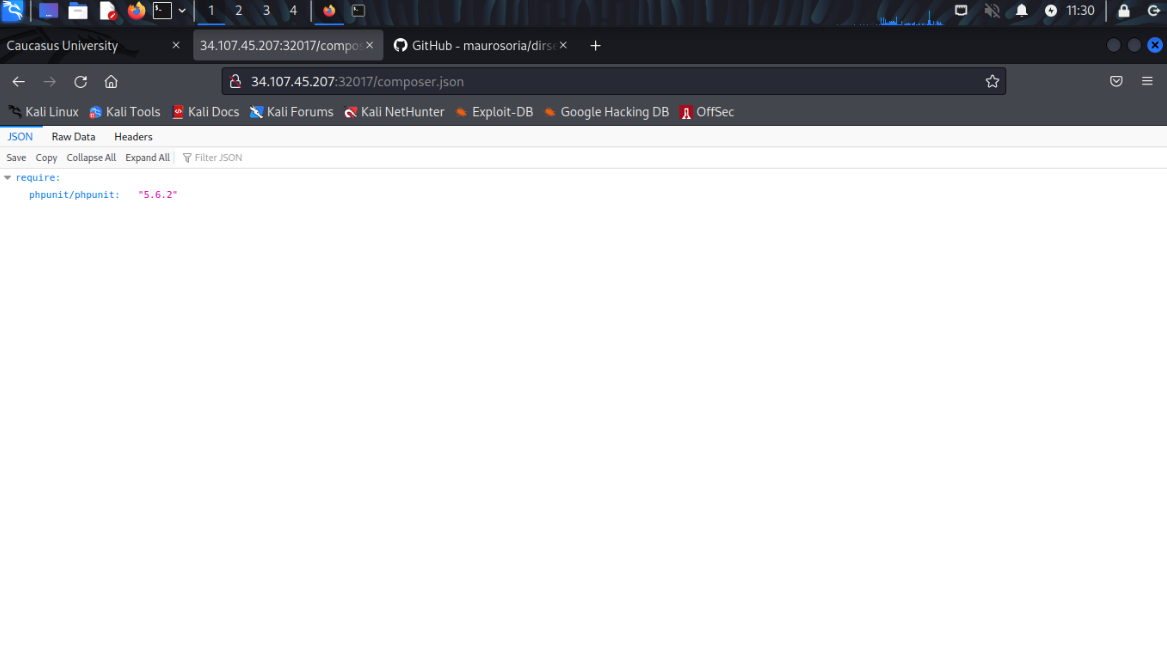
ამ ლაბში დაგვჭირდება dirsearch თული, ეს არის “advanced” command line რომელიც გვეხმარება ფაილების და ფოლდერების დაბრუტფორსებაში ვებსერვერებზე



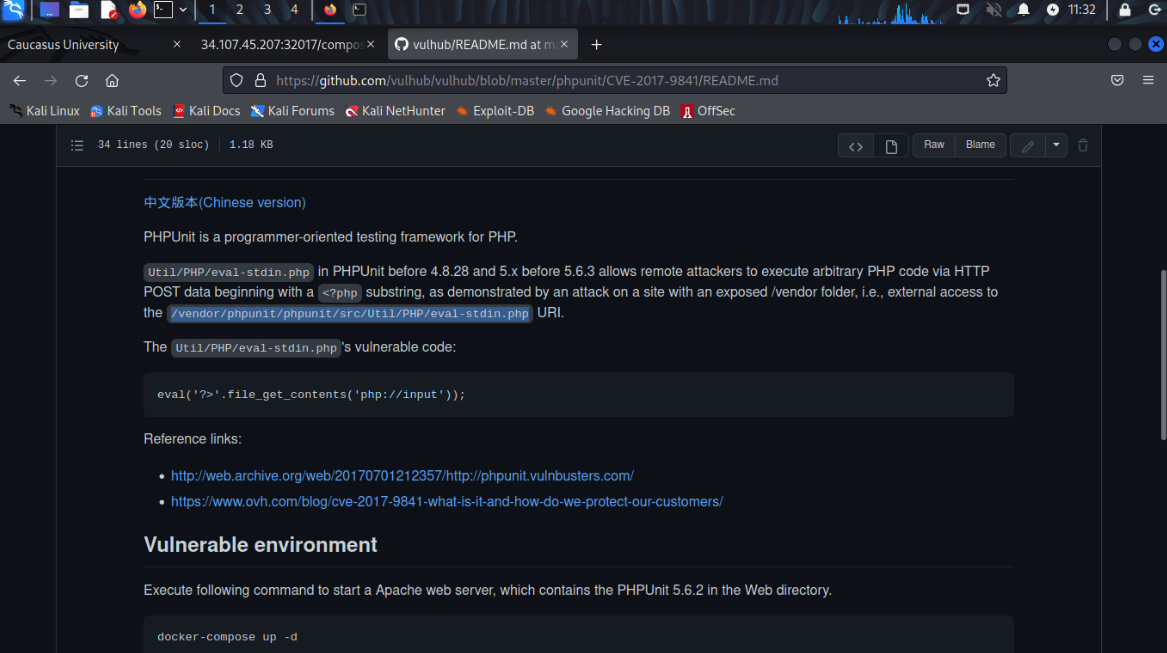
გადმოვწერეთ და ტერმინალში ვწერთ ასეთ ბრძანებას



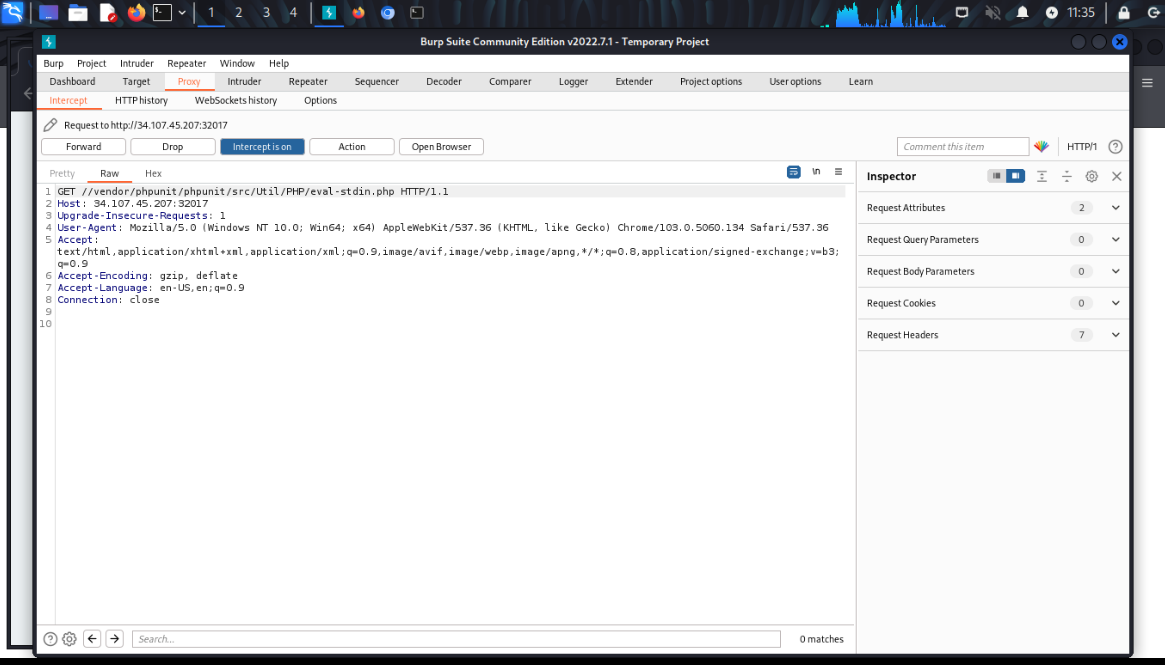
ახლა გვჭირდება გავიგოთ თუ php-unit-ის რომელი ვერსიაა, ამაში დაგვეხმარება composer.json



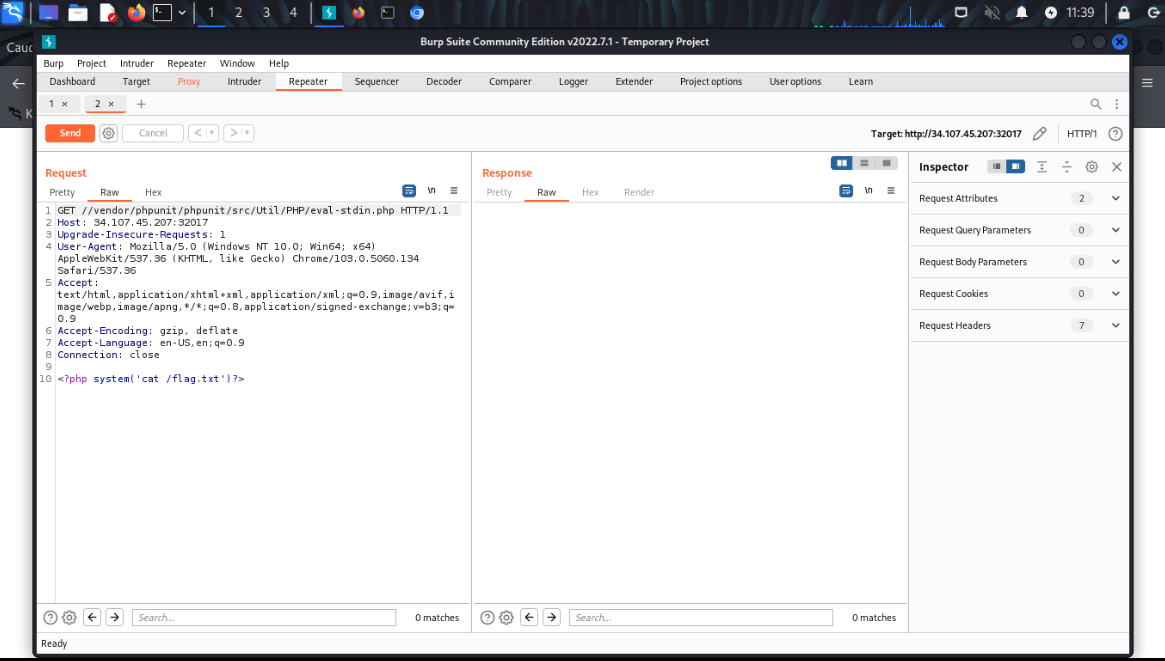
გავიგეთ ვერსია და ახლა გუგლში ვეძებთ ვულნერაბილითის, მონიშნულ ნაწილს ვუწერთ მისამართს



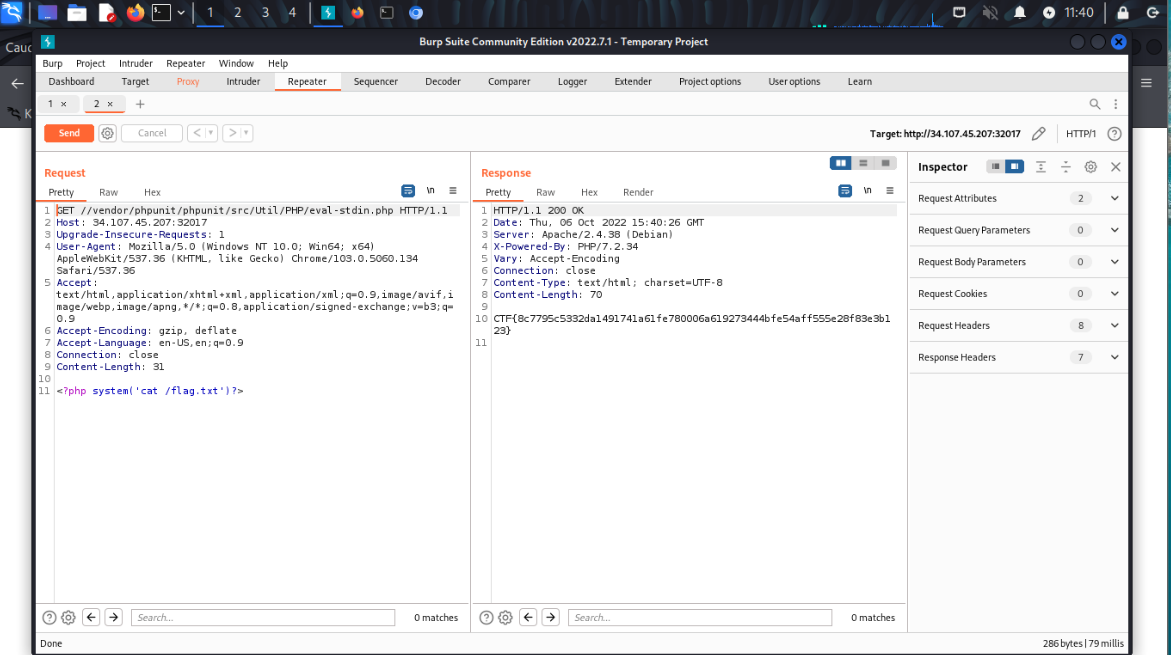
ახლა დაგვჭირდება burpsuite, მისი მეშვეობით ხელახლა შევდივართ ამ მისამართზე



შემდეგ ვაგზავნით repeater-ში და ვუწერთ ასეთ ბრძანებას

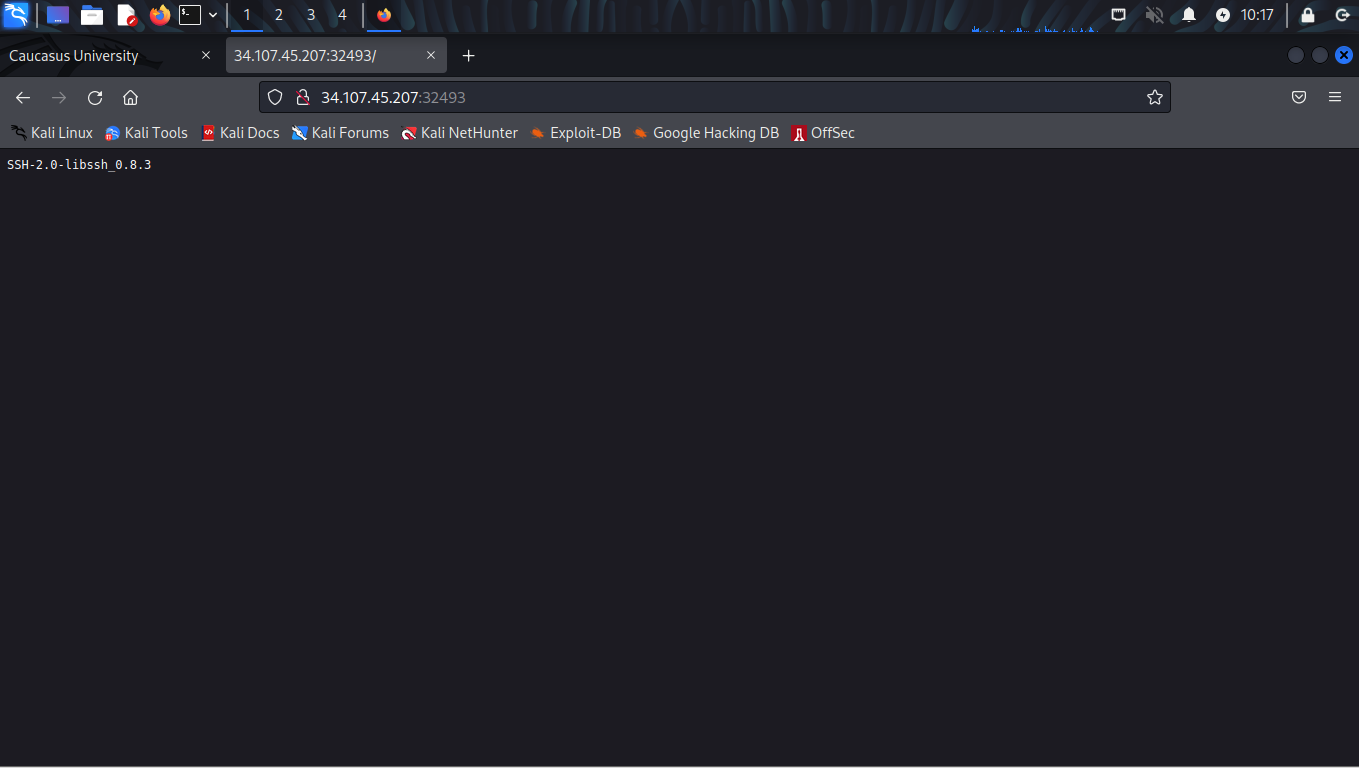


ვაჭერთ გაგზავნას და აი ჩვენი ფლეგიც

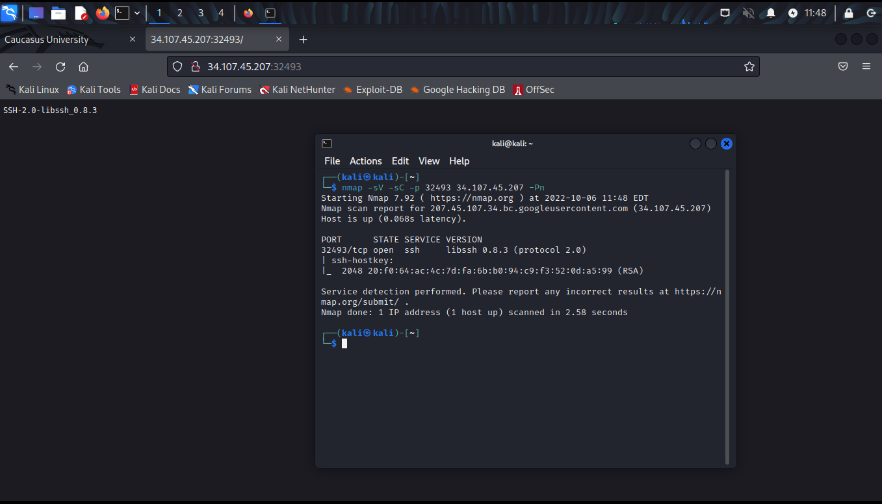


Lab: Libshh

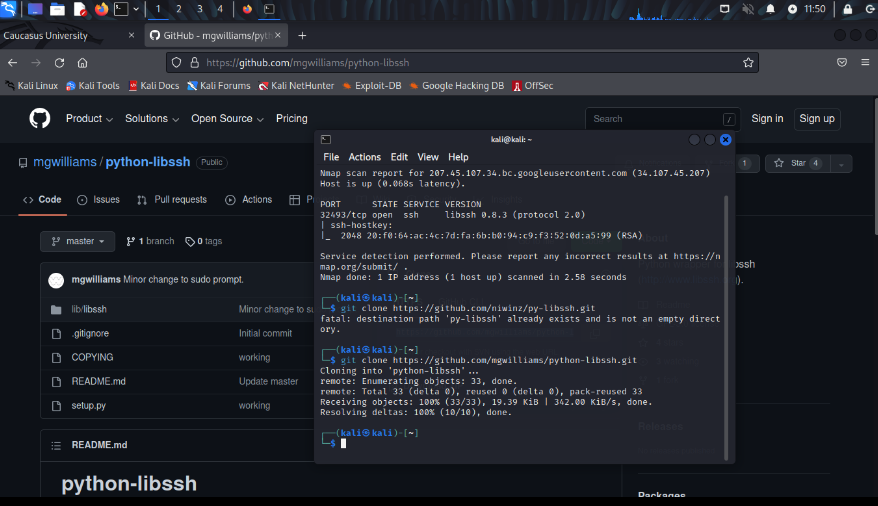
შევდივარ მითითებულ აიპი ადრესზე და მხვდება ასეთი გვერდი, libssh არის მულტიპლატფორმული C ლაიბრერი, რომელიც ახორციელებს SSHv2 პროტოკოლს კლიენტსა და სერვერის მხრიდან. ამ ლაიბრერით შეგვიძლია ჩავრთოთ პროგრამები, გადავაგზავნოთ ფაილები და ასე შემდეგ.



Nmap-ის საშუალებით ვასკანირებ პორტს ტერმინალში



ამის შემდეგ github-ზე ვეძებ libshh.py რადგან ფლეგის მოძებნაში ეგ დაგვეხმარება



და ვწერ ასეთ ბრძანებას რომელმაც უნდა მიმიყვანოს ფლეგამდე



თეორიული ნაწილი

2) Fuzzing არის ხარისხის უზრუნველყოფის (QA) ტექნიკა, რომელიც გამოიყენება

კოდირების შეცდომებისა და უსაფრთხოების ხარვეზების გამოსავლენად

პროგრამულ უზრუნველყოფაში, ოპერაციულ სისტემებში ან ქსელებში. ის

მუშაობს სისტემის დაქრაშვის მეთოდით ან რაიმე შეცდომების გამომწვევით,

როცა ხდება რანდომი ინპუტების შეყვანა. თუ აღმოჩენილია vulnerability, fuzz

ტესტირების პლატფორმა (ასევე უწოდებენ fuzzer) დაგვეხმარება მიზეზის

დადგენაში.

Website enumeration გულისხმობს რესურსების აღმოჩენას, რომლებსაც იყენებს ვებ

სერვერი, ისევე როგორც მთავარ, ძირითად ტექნოლოგიას, რომელზეც მუშაობს

ვებ სერვერი. ეს ინფორმაცია დაგვეხმარება ავარჩიოთ უფრო ეფექტური

თავდასხმის გზები, ისევე როგორც ასევე გამოიყენოს დაუცველობა ვებ სერვერის

პროგრამული უზრუნველყოფის კონკრეტულ ვერსიებში.

ჩვენ გვაქვს ასეთი თულები- Nmap, Metasploit, dirbuster და ა.შ. რაც გვეხმარება

Website enumeration-ისთვის

1)CVE-2018-10993 არის Libssh აუტენტიფიკაციის ექსპლოიტი, ხოლო libshh არის ლაიბერი. კონკრეტულად ეს ფექიჯი როგორც უწერიათ საიტზე, არის

მულტიპლატფორმული C ლაიბერი რომელიც ახორციელებს SSHv2 პროტოკოლს

როგორც კლიენტ ასევე სერვერ საიდზე. ამ ლაიბერით შეგვიძლია ჩავრთოთ

პროგრამები, გადავაგზავნოთ ფაილები, დავამენეჯოთ ფაბლლიქ ქიები და ა.შ.