iGNITE Technologies

BURP SUITE FOR PENTESTER

HACK BAR

TABLE OF CONTENTS

Abstract

Isn't it a bit time consuming and a boring task to insert a new payload manually every time for a specific vulnerability and check for its response?
So, today in this publication we'll explore one of the best burp suite's plugins **"Hack Bar"** which will speed up all of our manual payload insertion tasks and will work with almost all the major vulnerabilities.

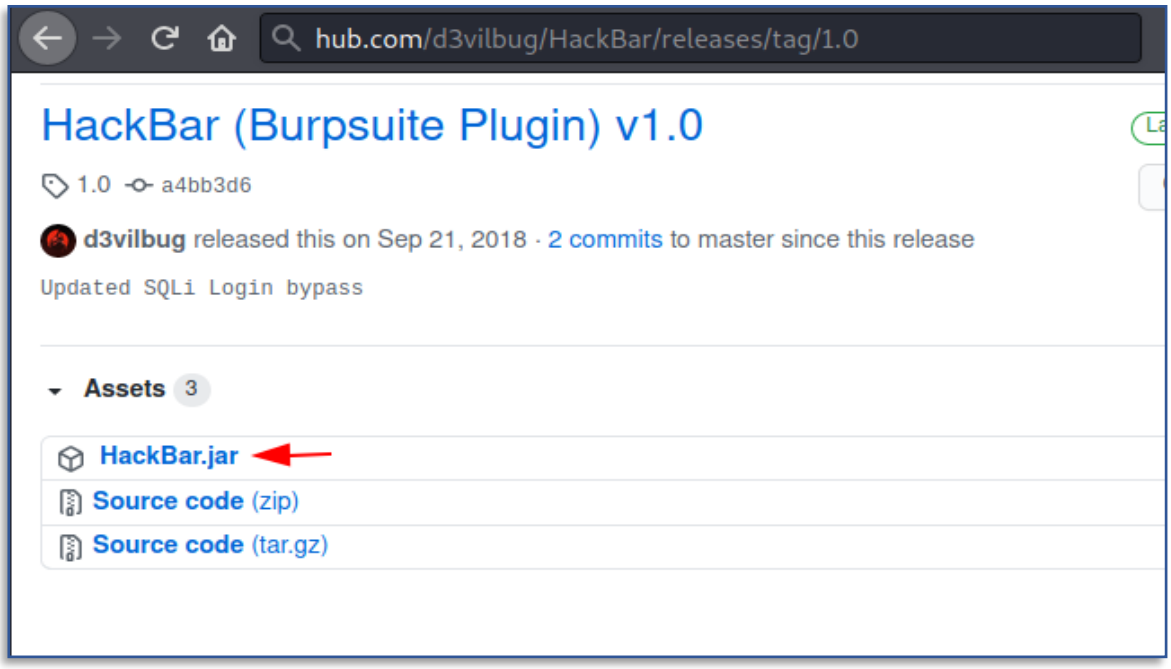
Introduction to Hack Bar

What is Hack Bar?

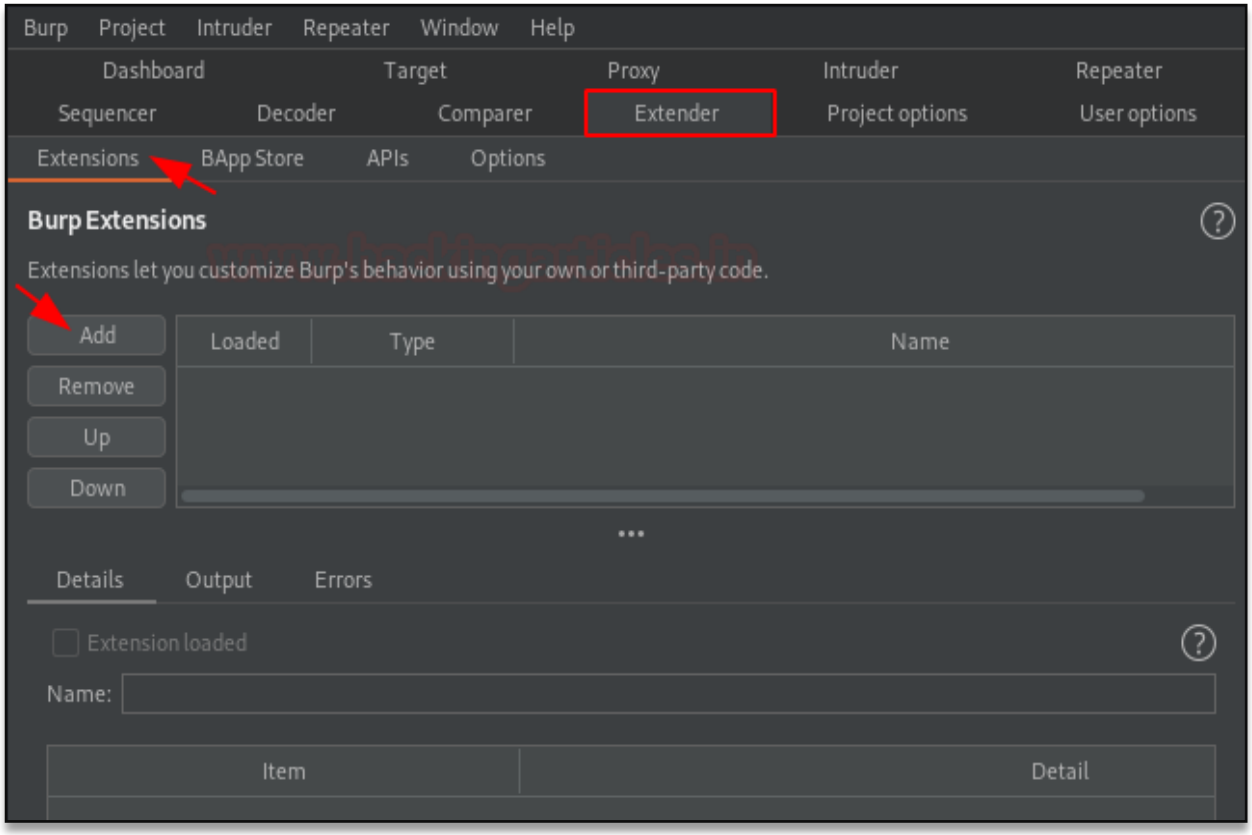
Hackbar is a plugin designed for the penetration tester such in order to help them to **speed their manual testing procedures.** However, the hackbar are specifically built for the **browser's extensions**, which contains a number of dictionaries according to the vulnerability type whether its SQL Injection, Cross-Site Scripting, or URL Redirections. This hackbar are designed somewhat similar to the **address bars** in the browsers.
The Burp's Hack Bar is a **Java-based Burpsuite Plugin** which helps the pen-testers to insert any payload by opting from a variety of different dropdown lists. Although it works the same as the browser's hackbar, its design and implementation are totally different.
Scroll your mouse down and you'll get to know about it.

Hack Bar Installation

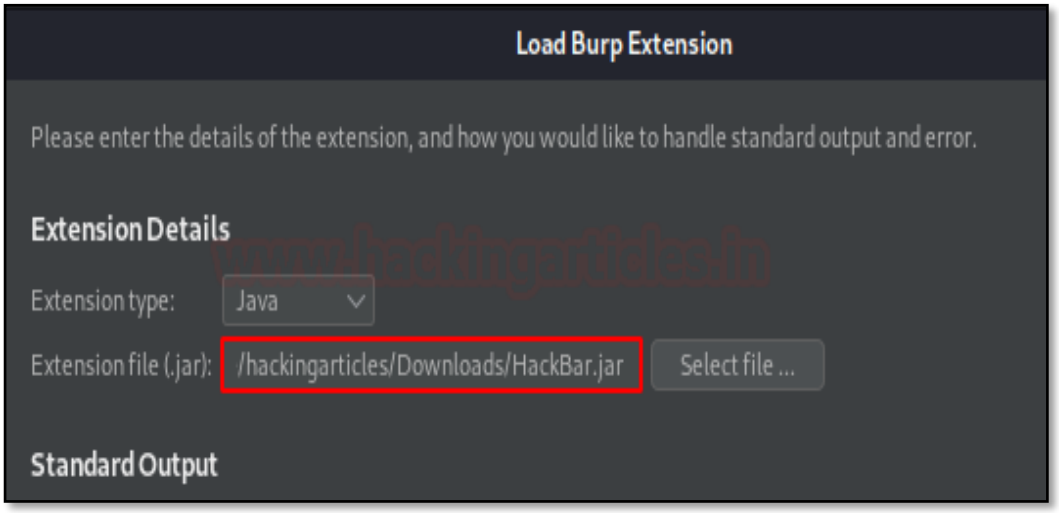
You might not find this great plugin over at the bApp store neither in the professional version or the community one. So, how will you set this up? In order to make this Hackbar a part of our pentesting journey, we need to **download its jar file** from the **GitHub repository.**



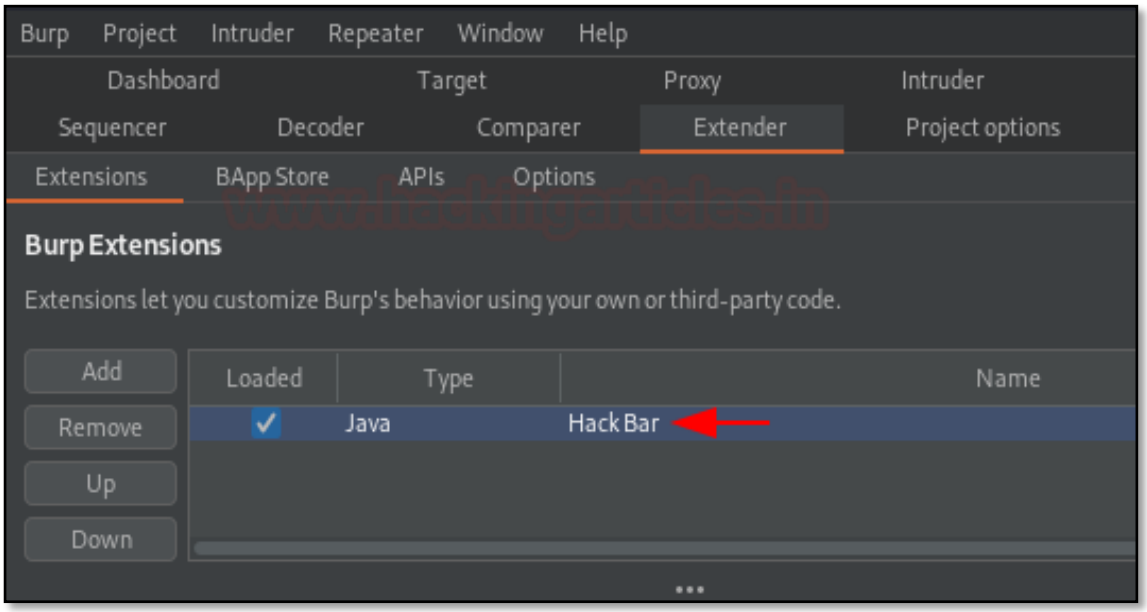
As soon as the file gets downloaded, we'll tune back into our burpsuite monitor and will navigate to the **Extensions section** in the **Extender tab**. There we'll hit the **Add button** in order to pull the **"Load Burp Extension"** window.



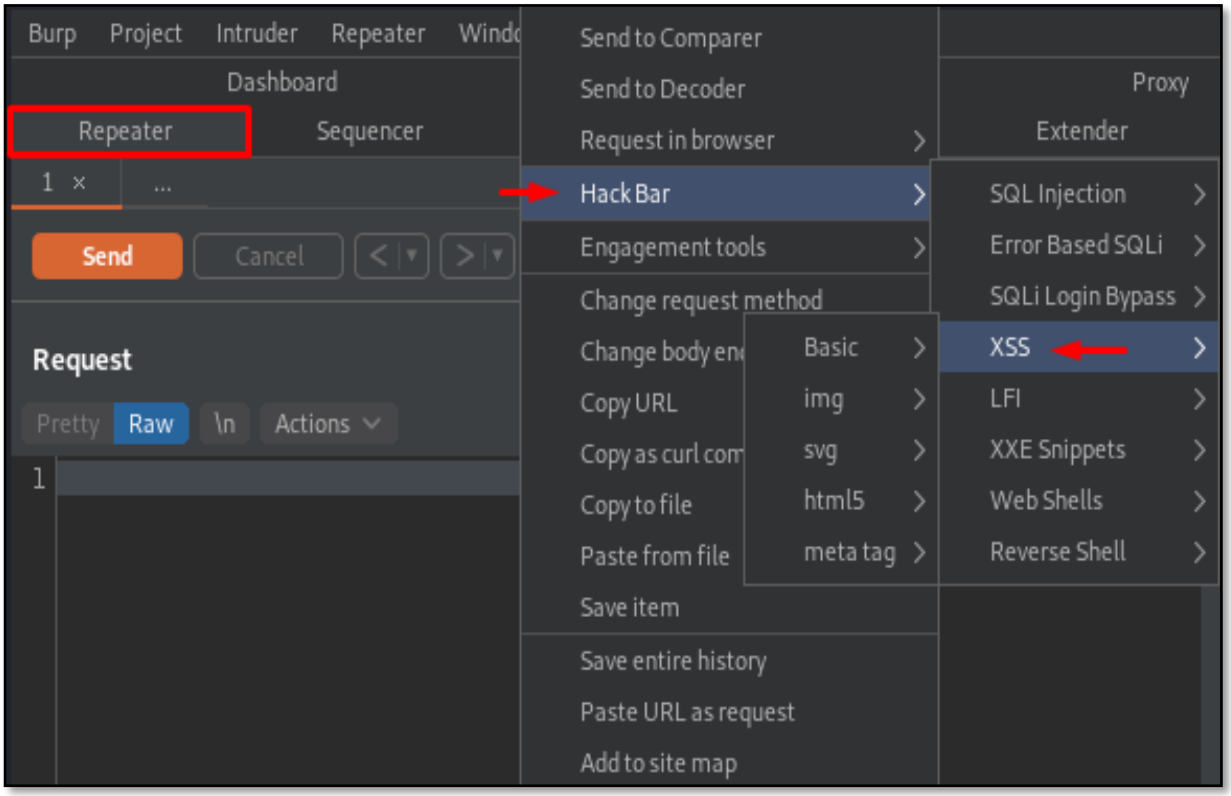
Let's now set the extension type to **"Java"** and opt the downloaded file. Further, we'll hit **"Next"** to initiate the installation.



Once the installation ends up, we got our payload listed into the **"Burp Extensions"** section.



Let's check that out, whether it's working or not!!
Follow up at the repeater tab and make a right-click anywhere at the screen. Over with that, we can see a new option lined up as **"Hackbar".**

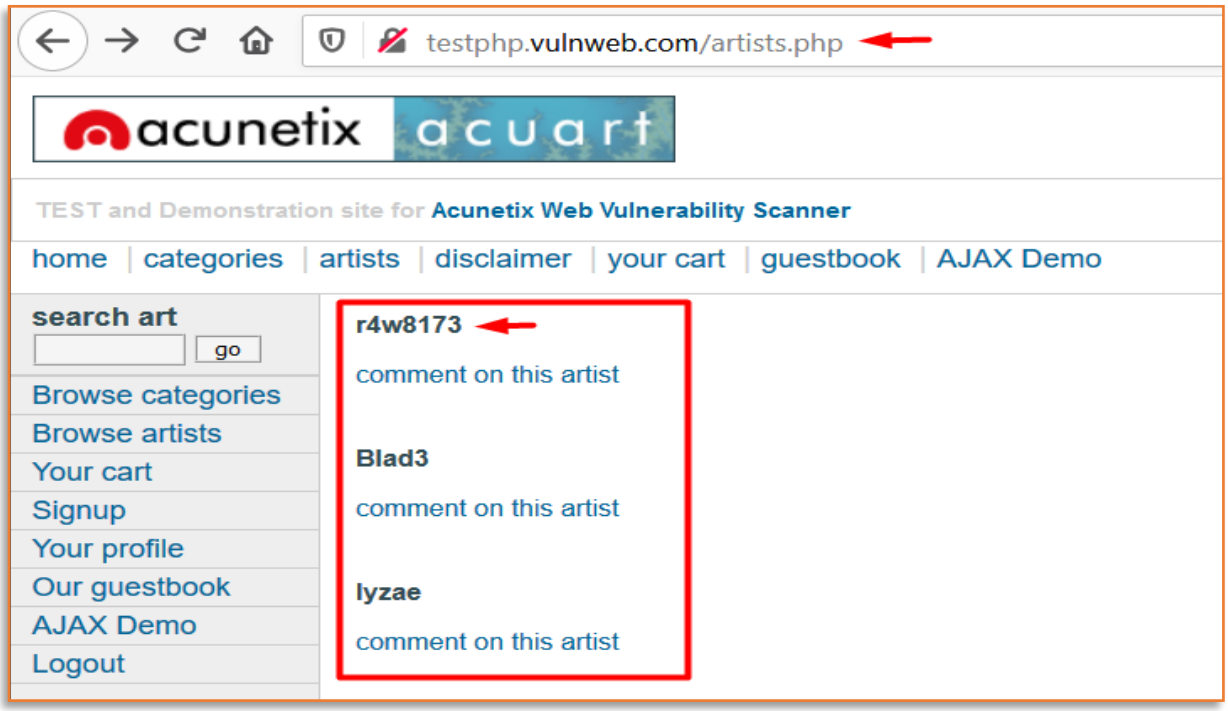


Exploiting Vulnerabilities with Hack Bar

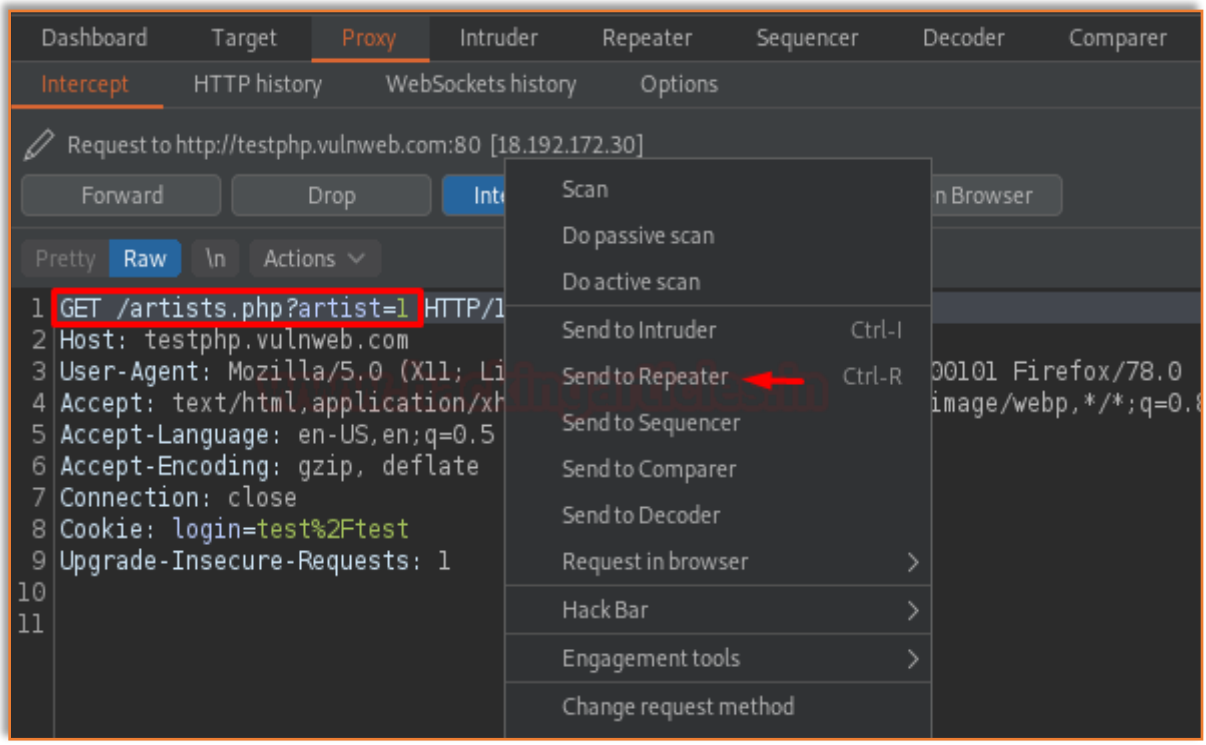
Hackbar has been designed in such a way to hit a number of crucial vulnerabilities as the dictionaries within it are segregated according to the type they belong too. However, we can use this hackbar or its dictionaries wherever we wish to, whether it's at the **Repeater tab** while manipulating the requests or at the **Proxy tab** during their interception.
So, for the time being, let's explore it and exploit the vulnerabilities exits up in bWAPP & Acunetix(test.vulnweb) vulnerable applications.

SQL Injection

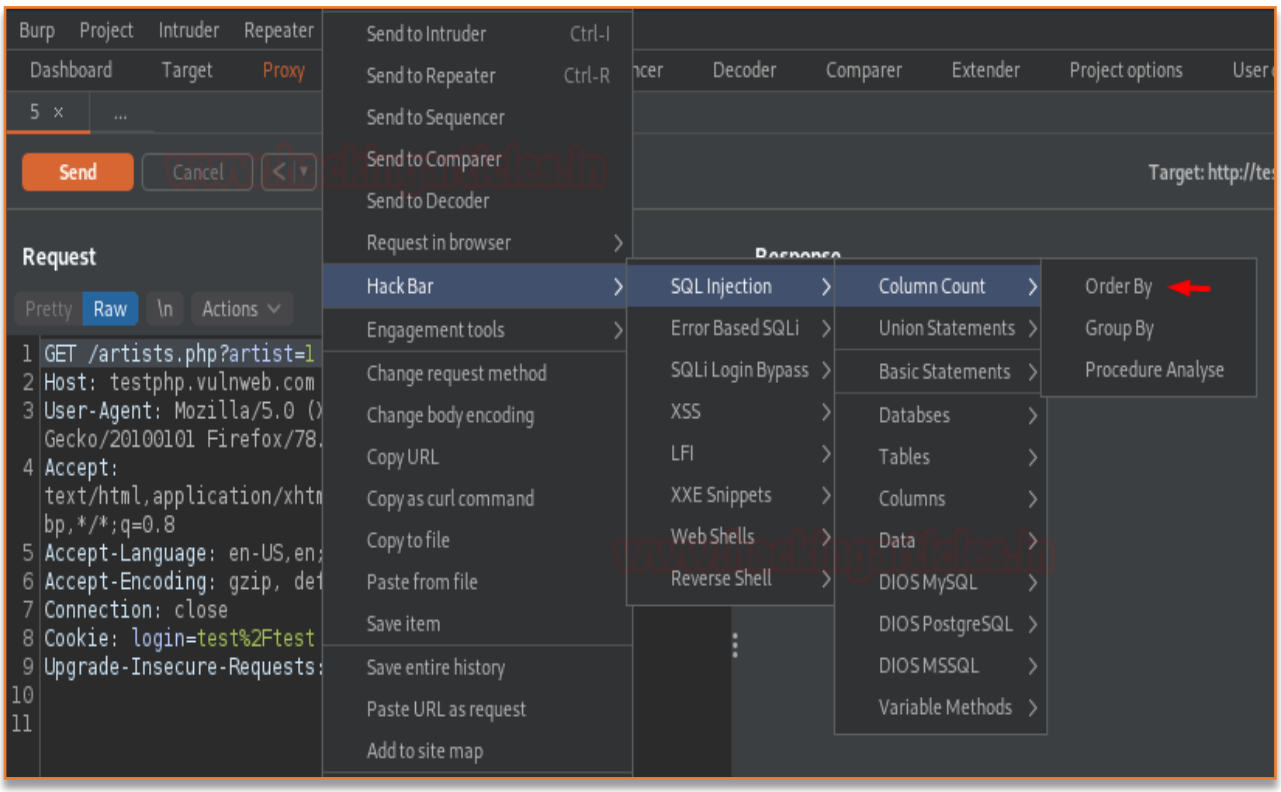
SQL Injection is one of the most crucial vulnerabilities exists over the web as almost every dynamic web-application carries a database within it. Thus with this, the attacker could bypass the authentication, access, modify or delete data within a database. You can learn more about it from **here**.
However, the automated tools that are designed to exploit this vulnerability need some of **the manual detection for the injection points**. And up till now, we know this thing that the manual pentesting can be best done with our hackbar, so let's try it out.
Initiating with test.vulweb, let's login inside it and check the artists within it.



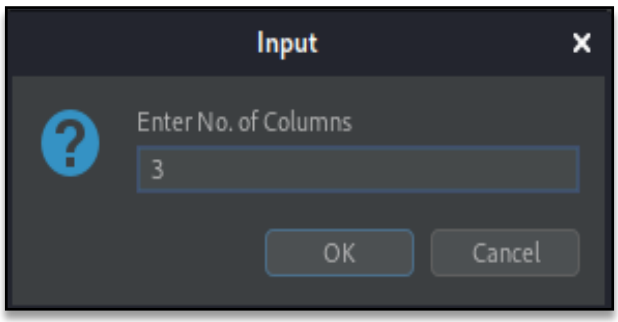
Now, time to analyse what it offers. Let's **capture the request** for the first artist over in our burpsuite monitor and then we'll further share it with the **Repeater.**



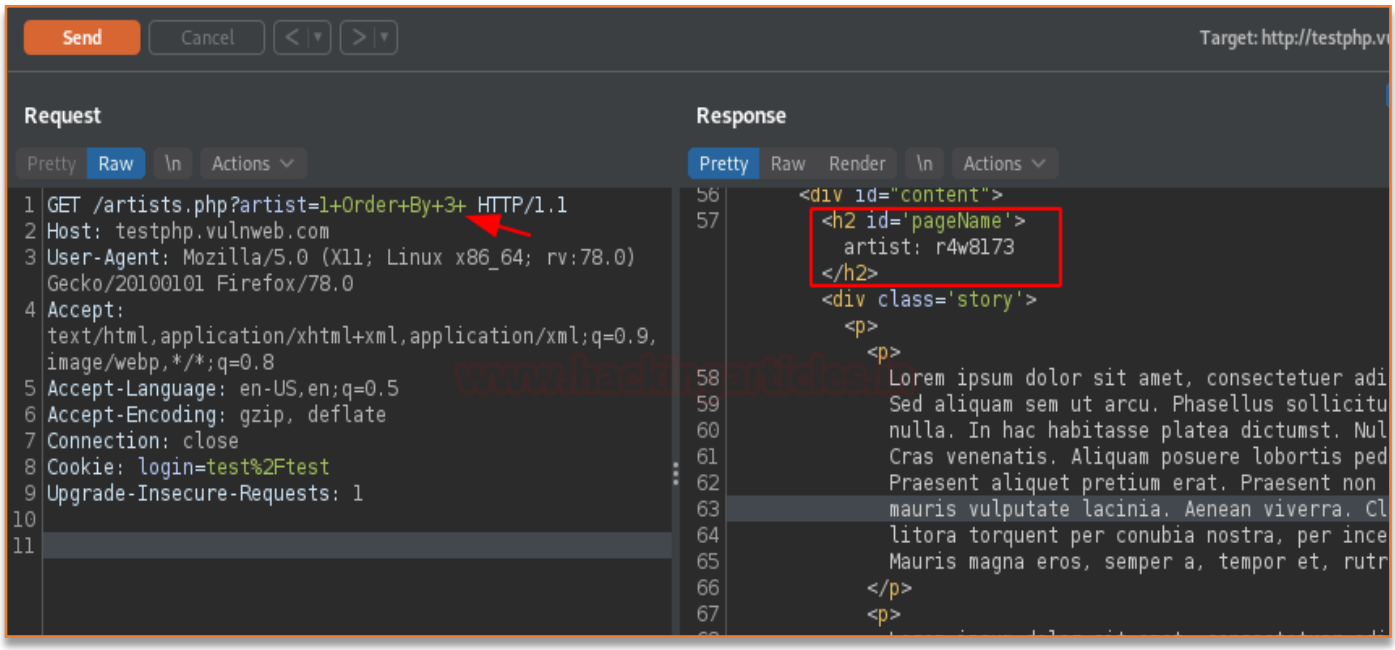
As soon as we do so, we'll hit right-click after **"artist=1"** and then will navigate to **Hack Bar -> SQL Injection -> Column Count -> Order By** in order to determine the number of records it consists of.



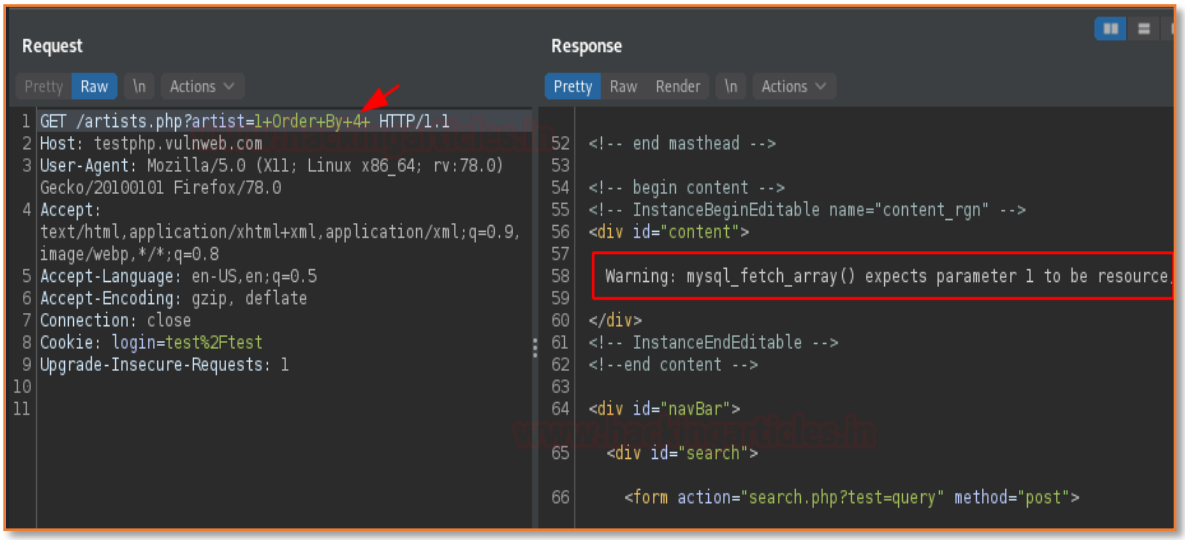
With the interception, let's try for **"3"** and check what it dumps.



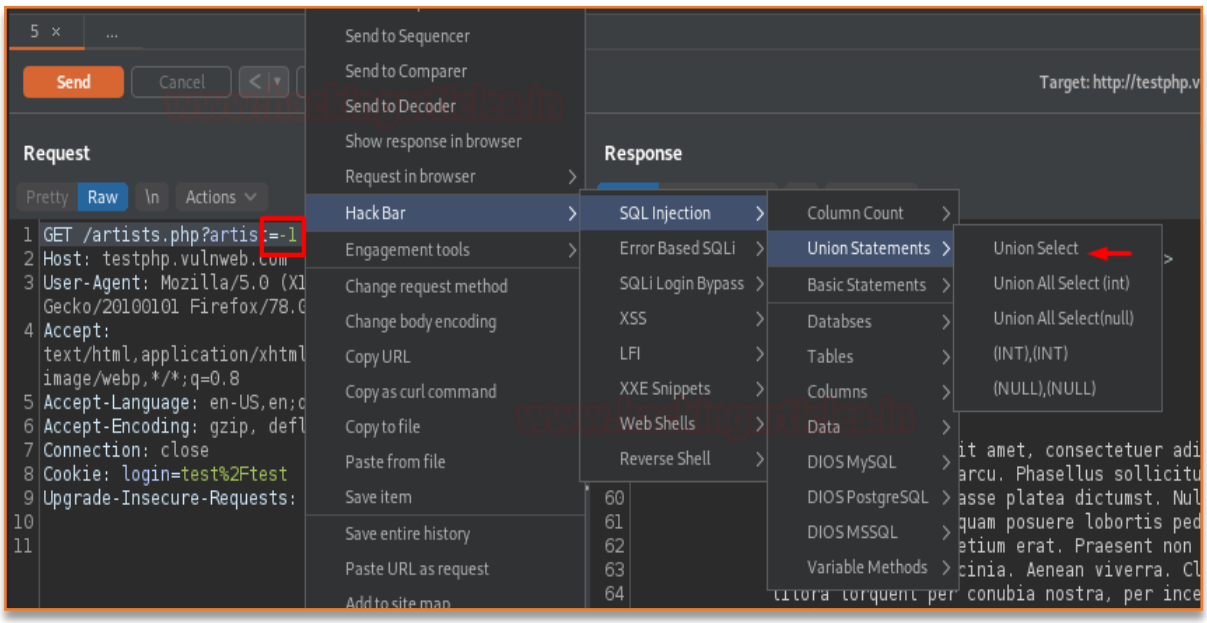
Over at the 3rd field, we're having an entry fed up as **"r4w8173".** Let's increment it will **1** i.e. **"4".**



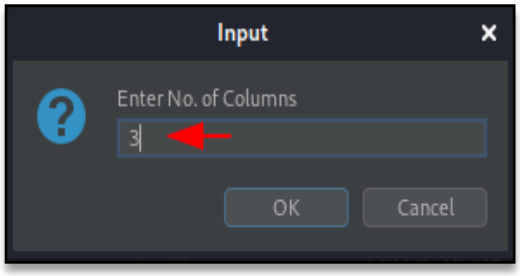
And there is an error for the 4th field, this confirms that it consists of only **three records.**



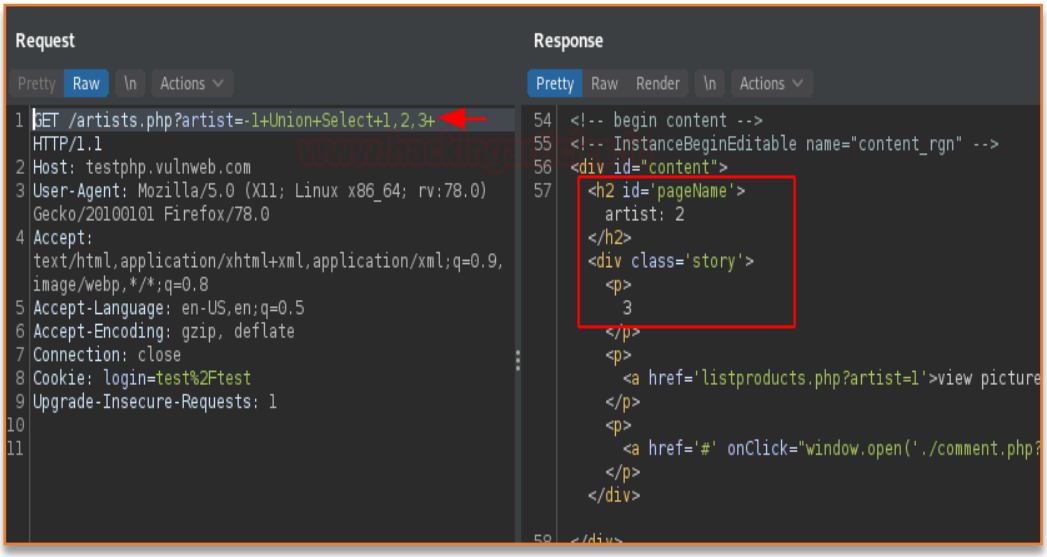
Let's penetrate more inside using **Union base injection** and even we'll pass wrong input into the database by replacing **artist=1** from **artist=-1**



As for the **Order By section,** we got that the records are 3, thereby we'll set the **No. of Columns** as **3** here too.

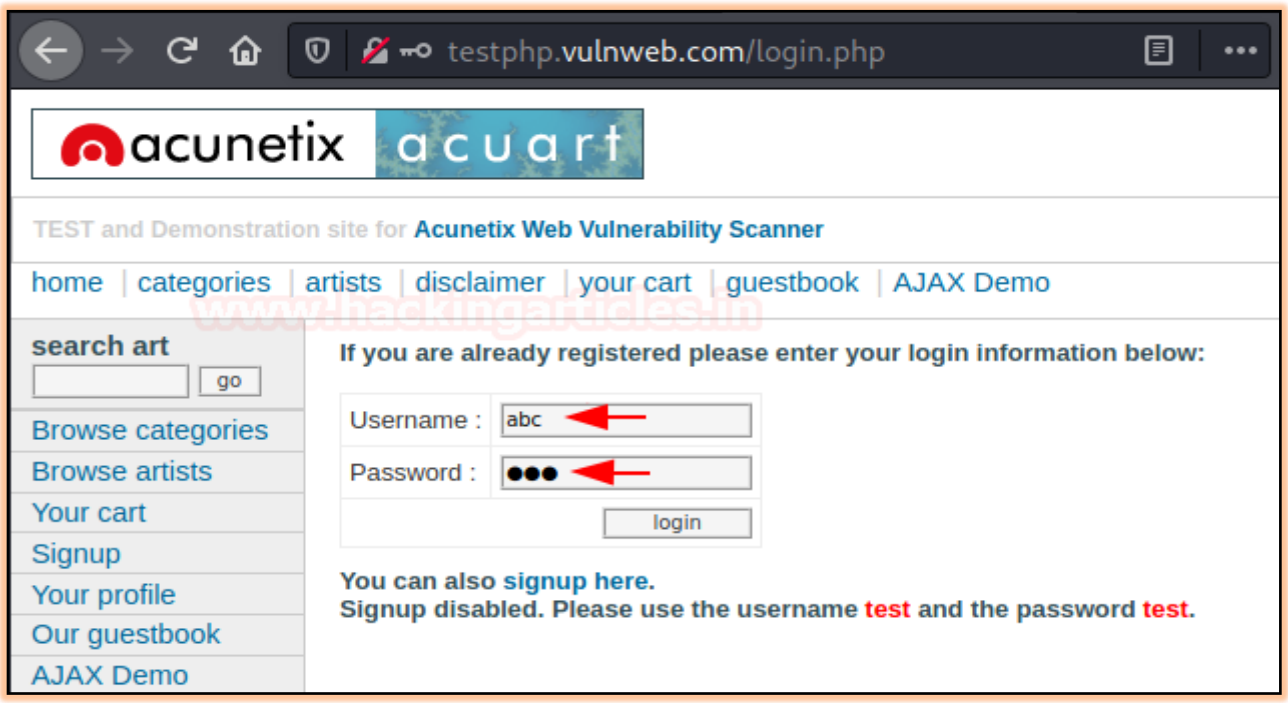


With the completion of the query and as we hit the **send button,** we got the result displaying the remaining two tables, which thus could be used to fetch the details within the database.
However, you can follow up more for manual SQL exploitation from **here**.

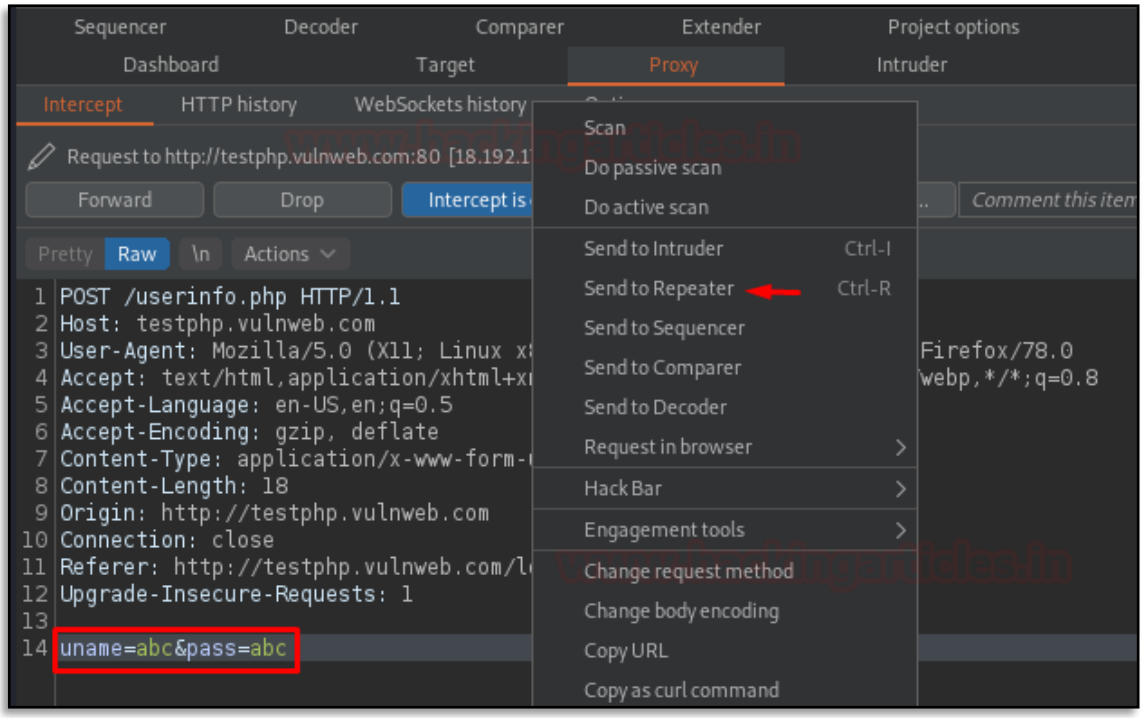


SQLi Login Bypass

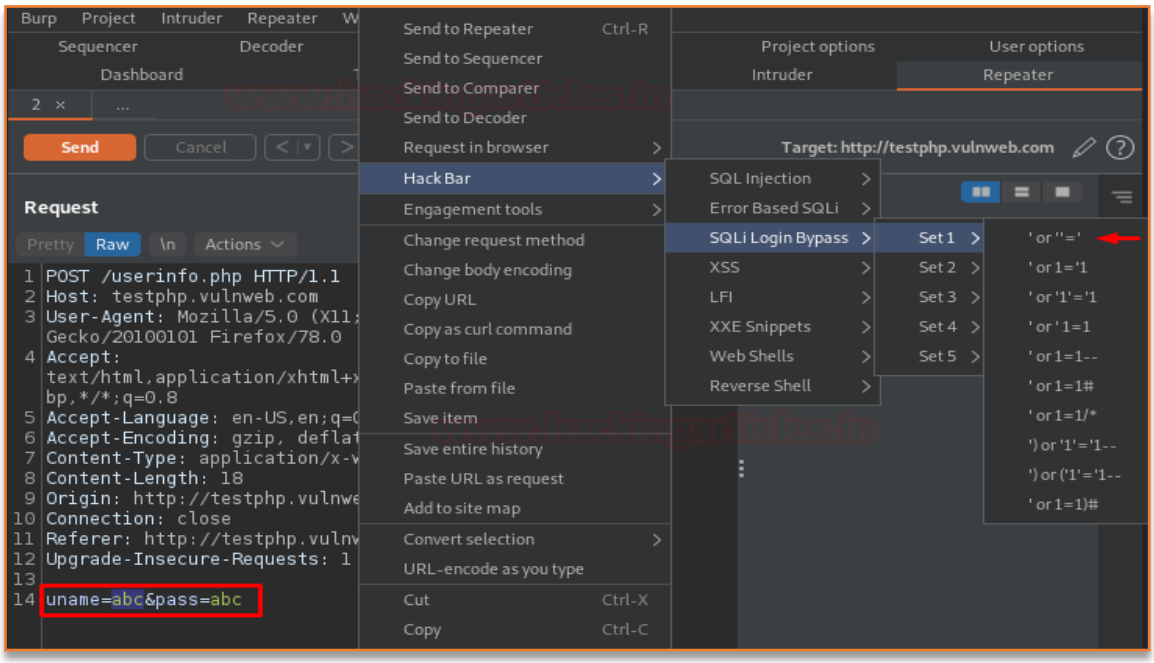
As discussed in the earlier section that over with the SQL Injection vulnerability the attacker tries to bypass the login portal so let's explore this exploitation with our Hack bar.
Login with some random credentials and capture the request into our **Burpsuite's Proxy tab**.



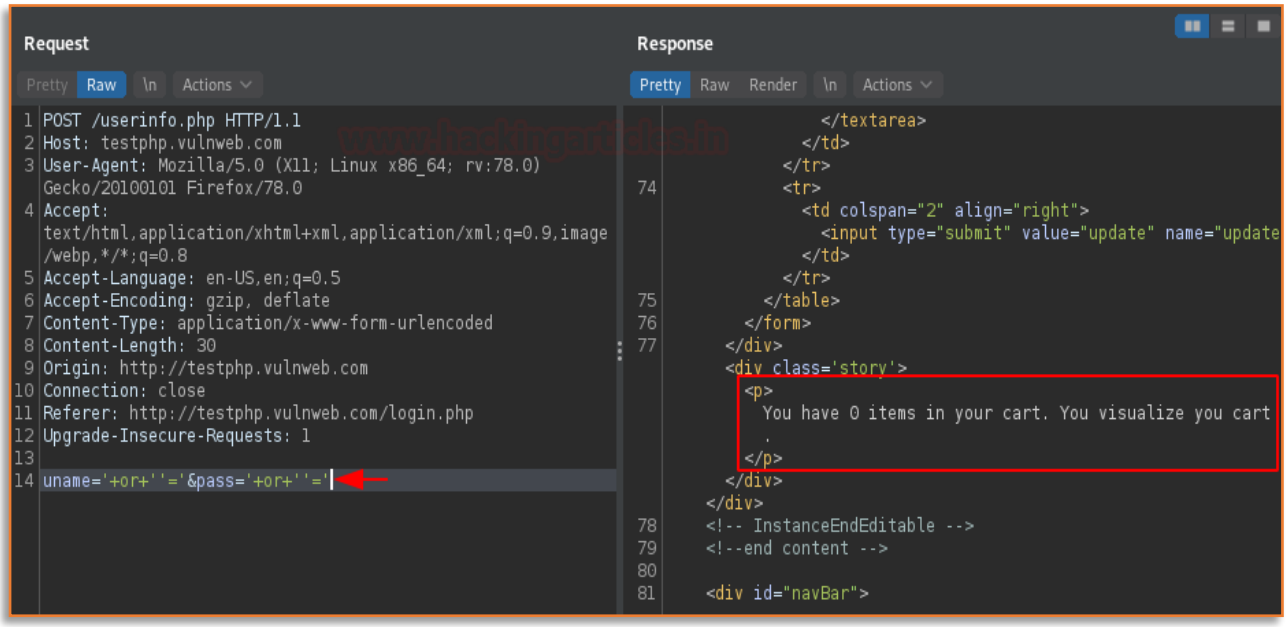
Once the Proxy **starts intercepting the request**, share it with the **Repeater**.



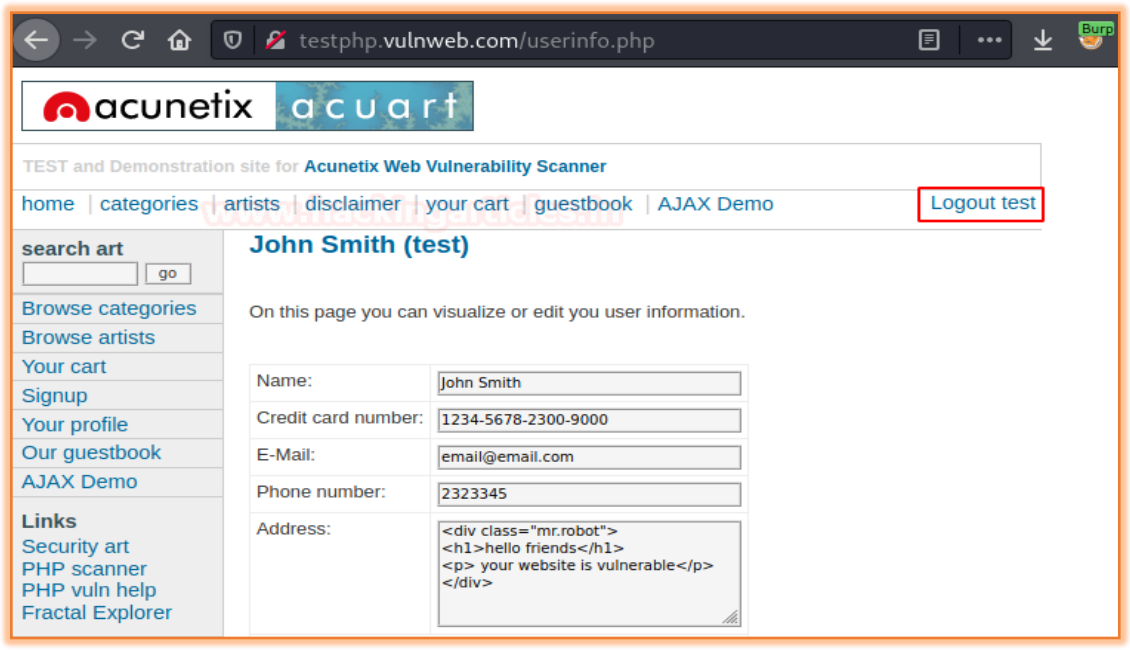
Here in the Request content, let's select the injection points **"uname"** and **"pass"** and then follow up with a right-click to **Hack Bar -> SQLi Login Bypass -> Set 1 -> 'or''='** dictionary value.



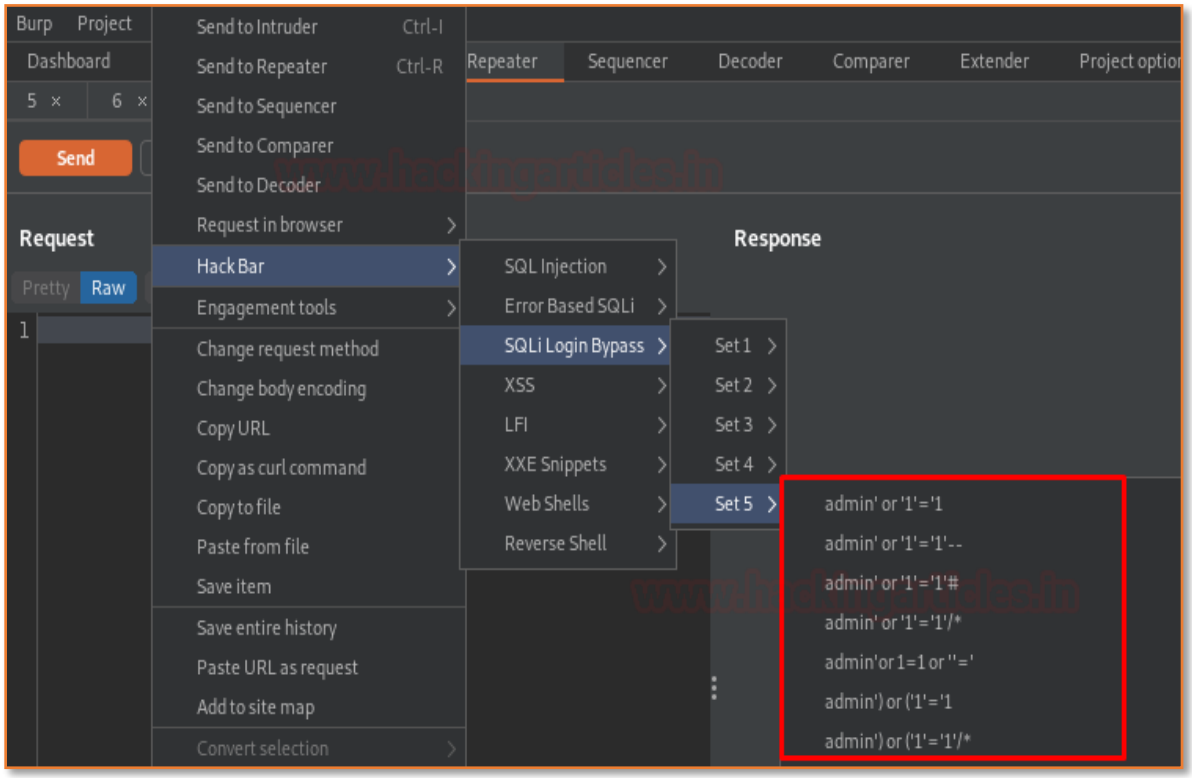
Hit the **Send** button to pass the values for authentication, and over at the right panel of the Response section, we can see some alterations. Let's check the same in the browser.



From the below image you can see that as soon as we paste the copied value generated with the **"Show Response in browser"** option, we got landed directly over at the dashboard.

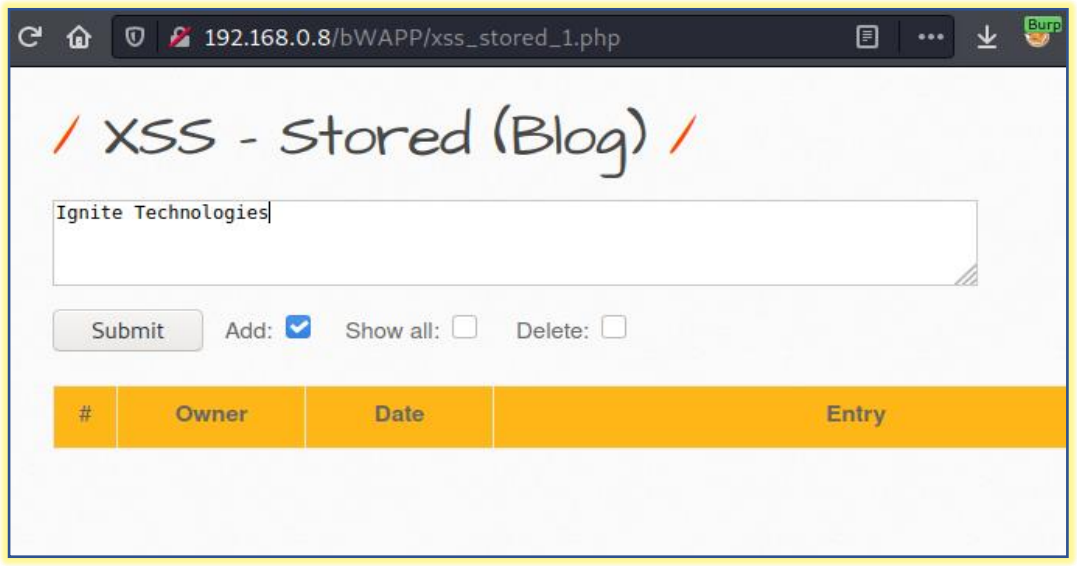


However, the SQLi Login Bypass contains a number of other dictionaries sets too, you can explore any of them if the payload within a specific dictionary is not working.

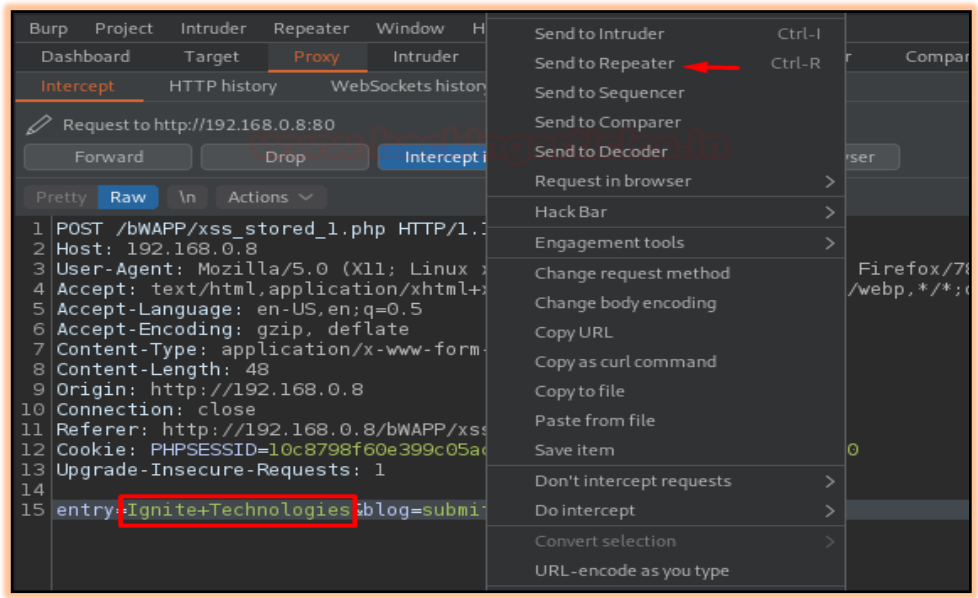


Cross-Site Scripting

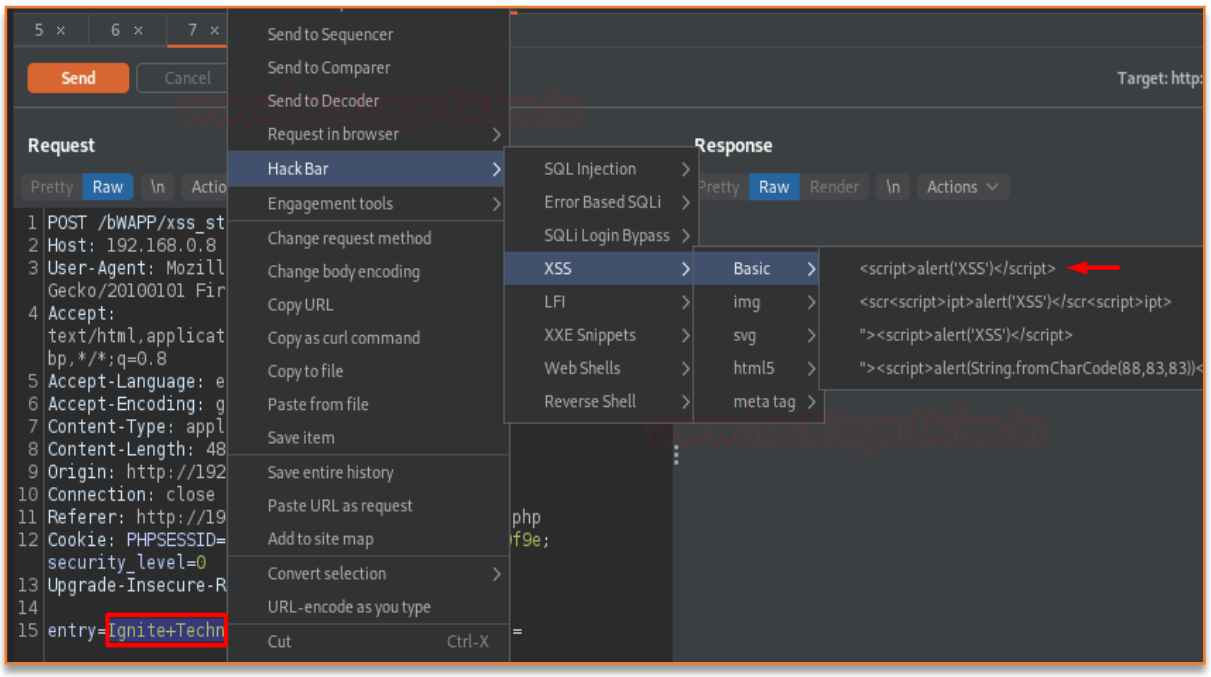
Cross-**S**ite **S**cripting or **XSS** is a client-side code injection attack where malicious scripts are injected into trusted websites and are triggered when the user visits the specific suffering web-page. You can learn more about it from **here**.
During an XSS exploitation, we majorly try to **inject payloads manually** at the injection points. But this manual exploitation sometimes didn't work due to typing error or blacklist implementation. Thereby in order to save our time and hit the vulnerability manually let's use our Hack Bar.
Open the target IP in the browser and login inside bWAPP as a **bee: bug**, further set the **"Choose Your Bug"** option to ***"XSS –Stored"*** and fire up the **hack button.**



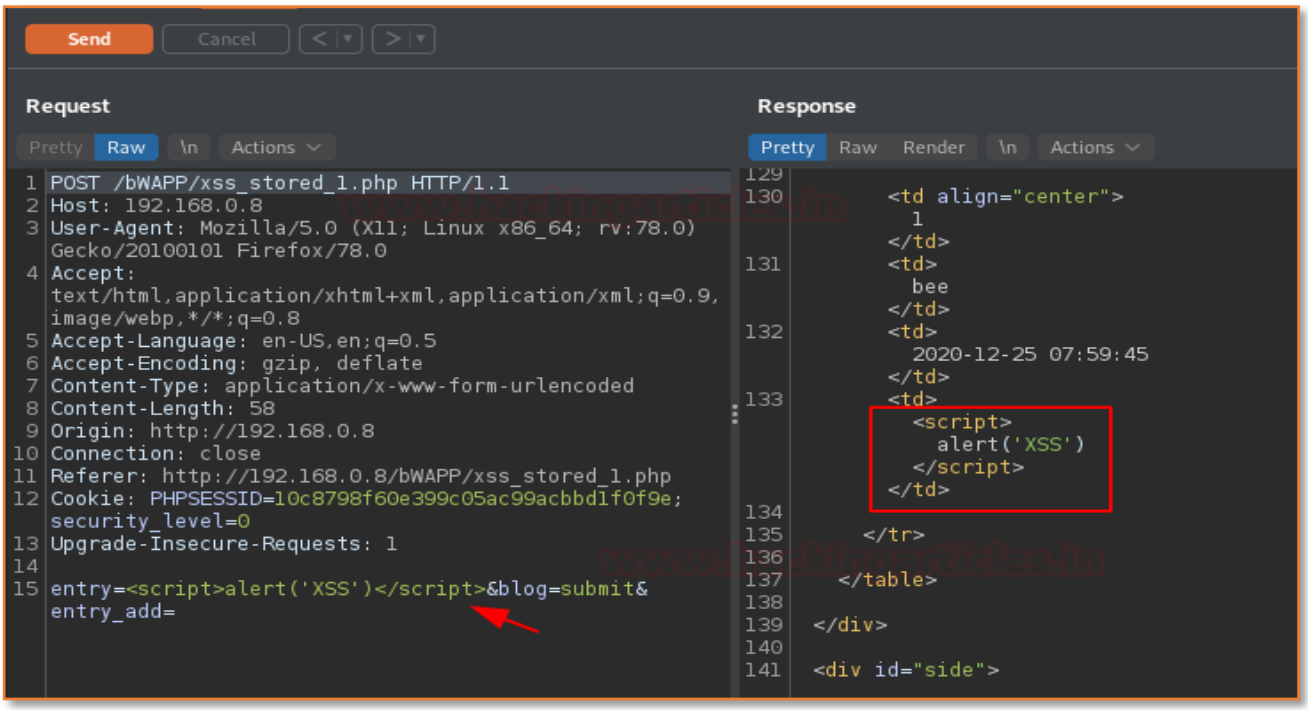
Before hitting the submit button, turn your burpsuite monitor and capture the ongoing HTTP Request. As soon as you got that, simply share it with the repeater for the manipulation part.



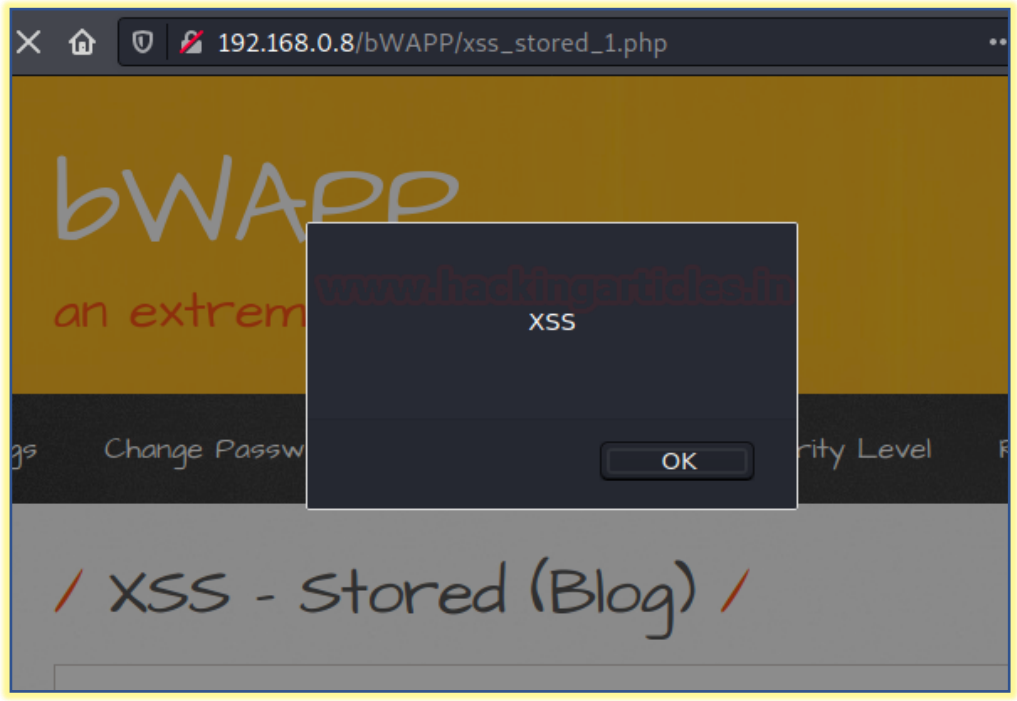
Time to go with the Hack Bar. Over at the Injection point, select it, and then navigate to **Hack Bar -> XSS -> Basic -> <script>alert('XSS')</script>**



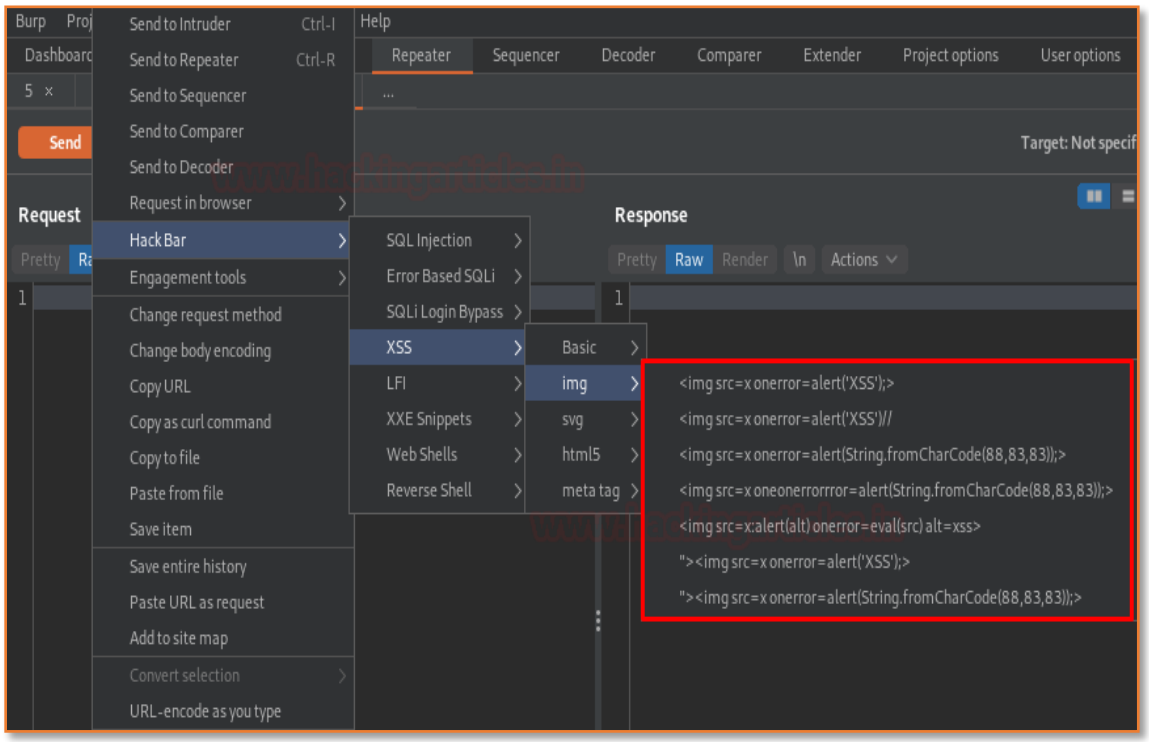
Once the payload gets injected up, hit the **Send** button and analyse the **Response.**



From the above image, you can see that our script has been embedded over into the webpage HTML content. Let's check the same in the browser.

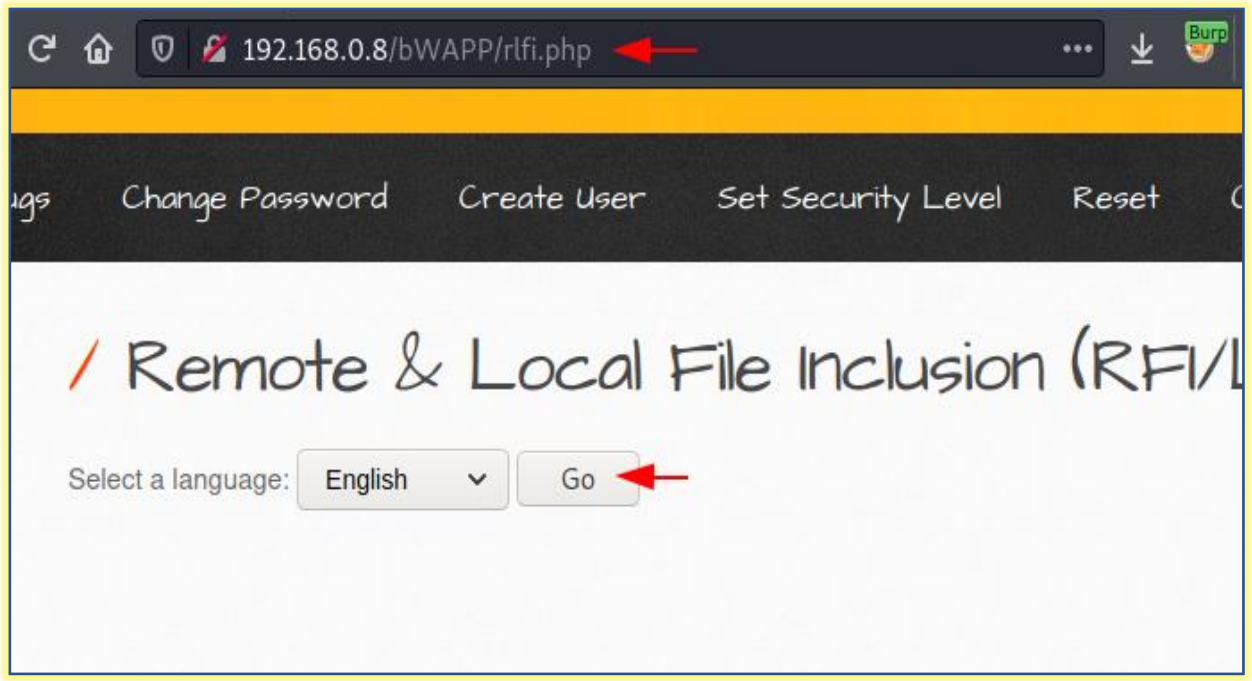


And there is a **Pop-up !!**
Similar to the SQL section, specific sets of dictionaries are also here. You can explore them according to your need.

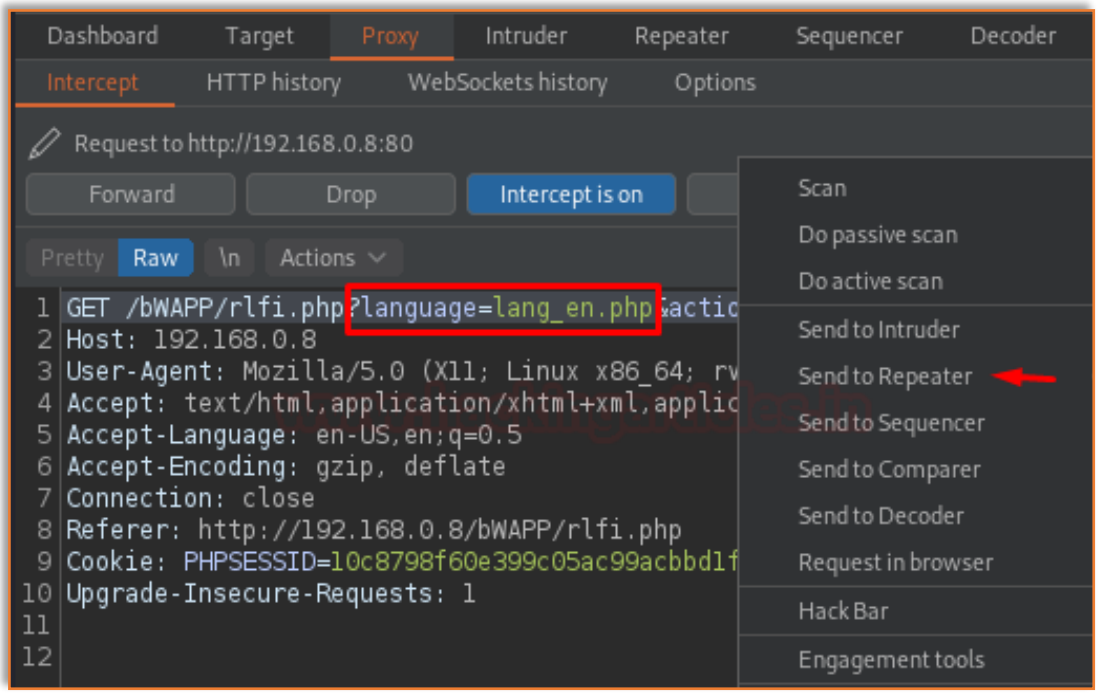


Local File Inclusion

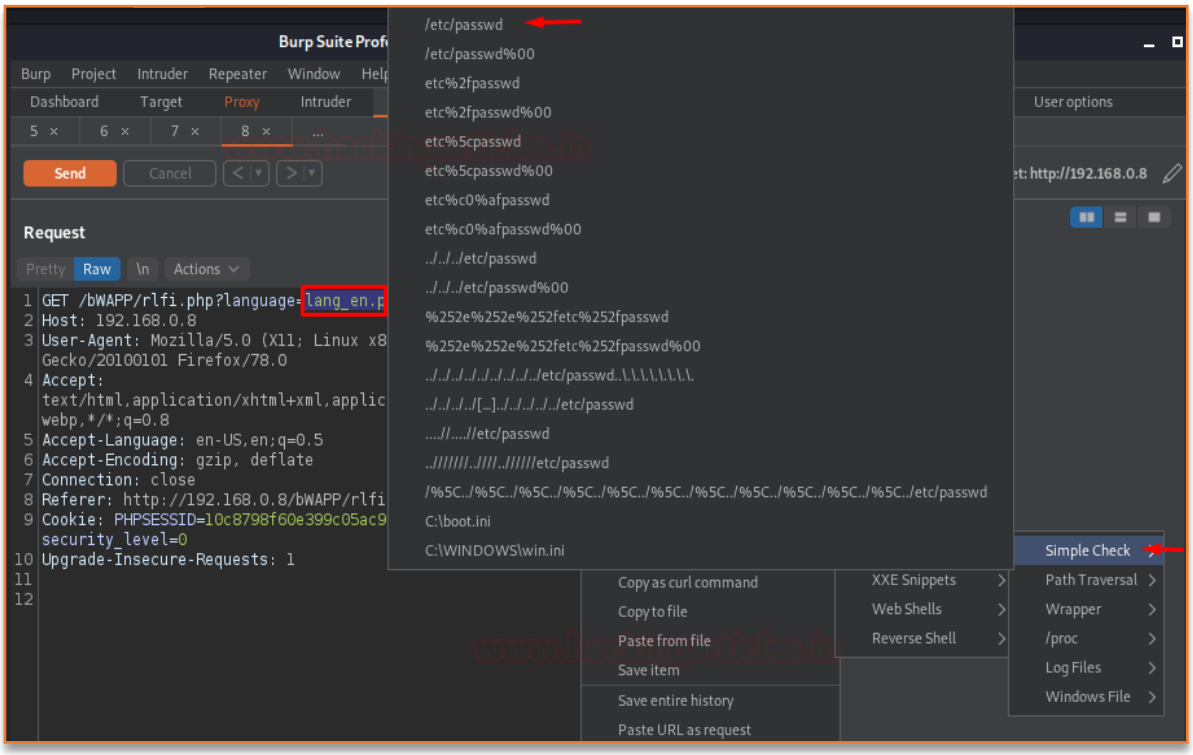
Local file inclusion is the vulnerability where the attacker tries to trick the web-application by including and calling the files that are already present locally into the server. This File Inclusion vulnerability is totally dependent on the type of injection point it carries up.
So, let's exploit its injection points with the Burpsuite's Hackbar.
Back into bWAPP switch to the **Remote & Local File Inclusion** vulnerability, and then opt **"English"** from the drop-down list and hit the **Go** button with the **Proxy service enabled.**



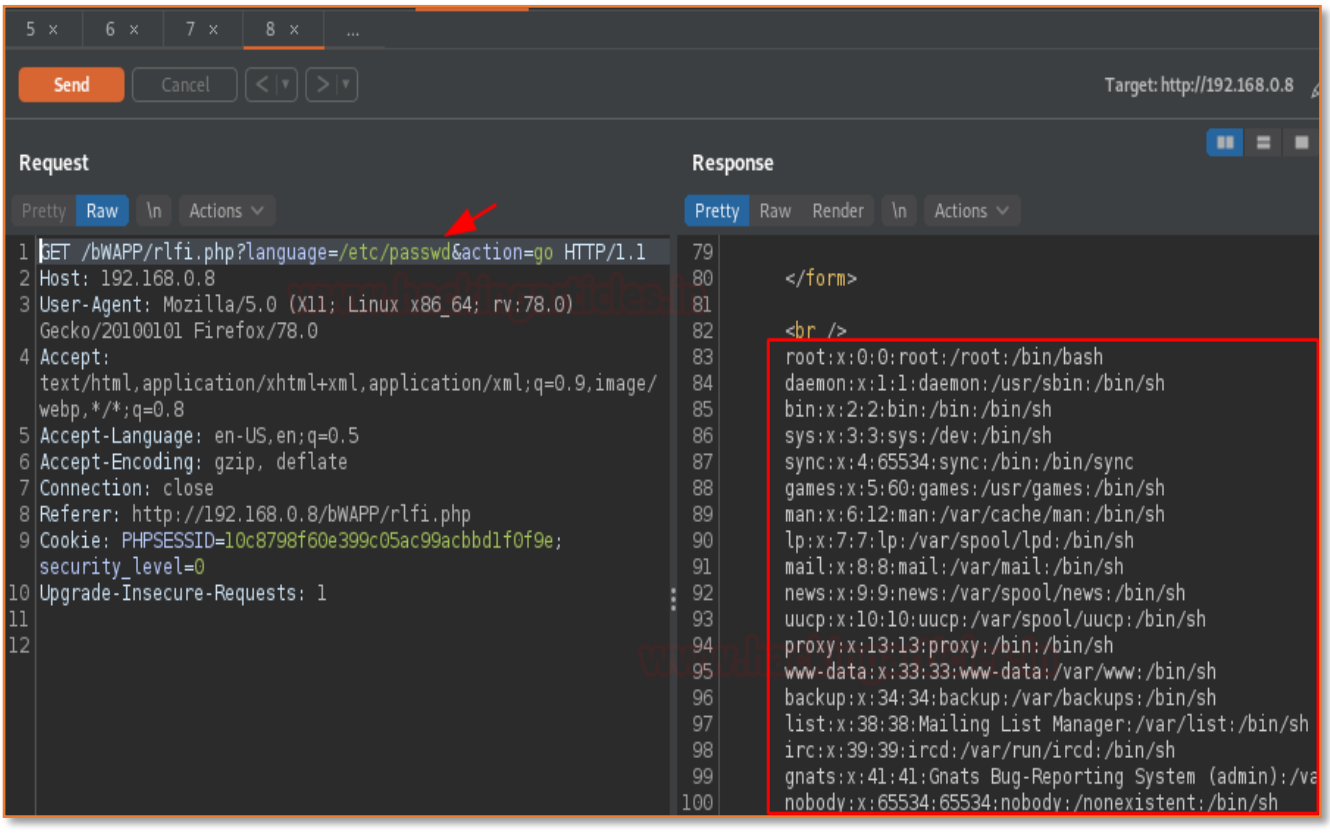
Once the request got captured by the burpsuite simply share it with the **Repeater.**



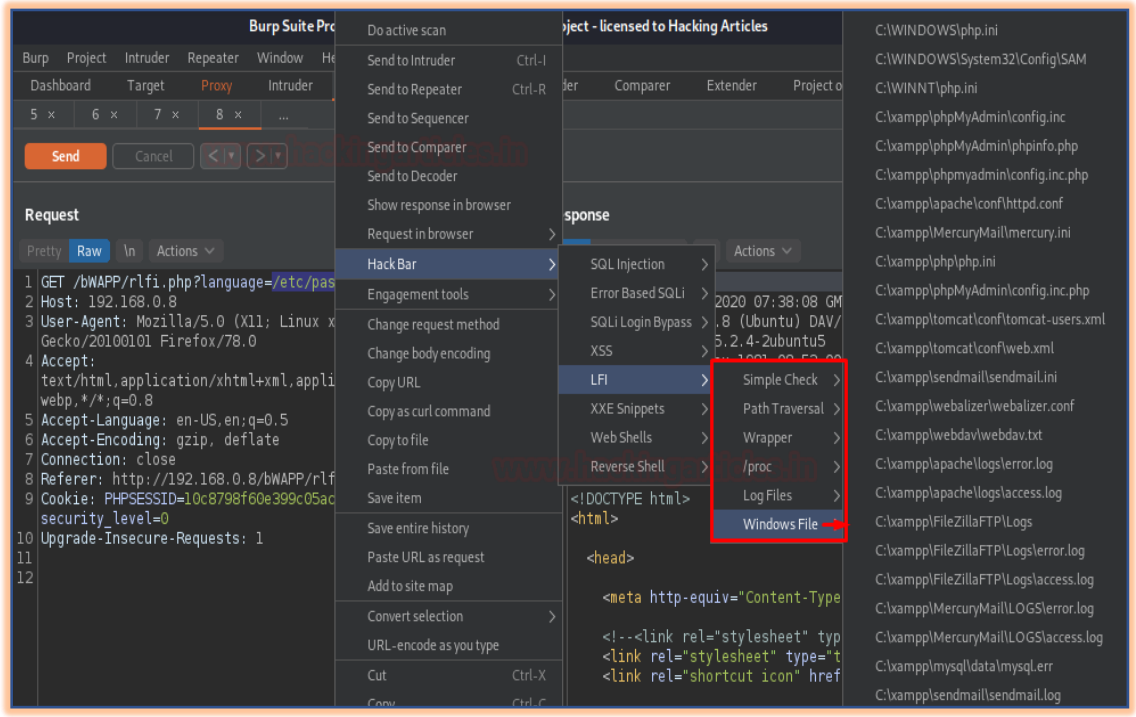
And I hope you know the next step. Navigate to **Hack Bar -> LFI -> Simple Check -> /etc/passwd**



As soon as we hit the **"Send"** button, we got our output listed over at the right panel.



However, the payloads for this file Inclusion vulnerabilities varies with the operating systems, thus the Hack Bar offers a number of payloads for **Linux and Windows**. It even carries some for the Path Traversal vulnerability.

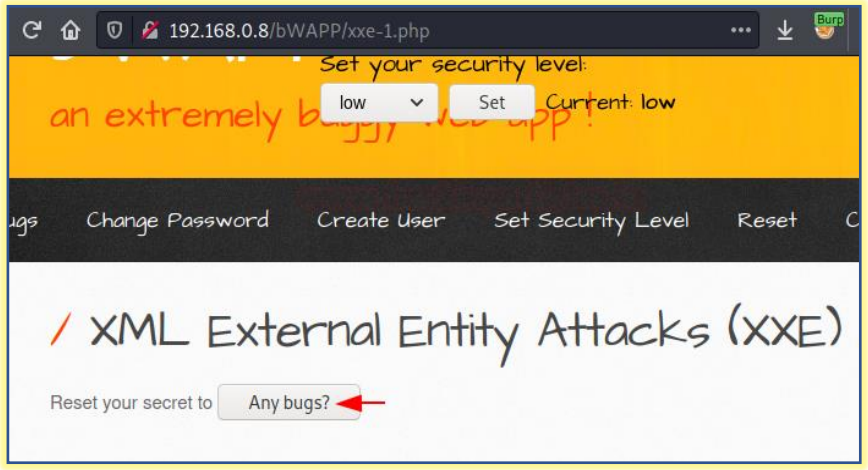


XXE Injection

XML e**X**ternal **E**ntity **(XXE)** attacks are the most common in today's era, as almost every application carries up XML inputs and parse them. Such XML attacks are possible as the input contains a reference to an external entity which is thus processed by a weakly configured XML parser. In order to learn more about it, check our **previous article**.

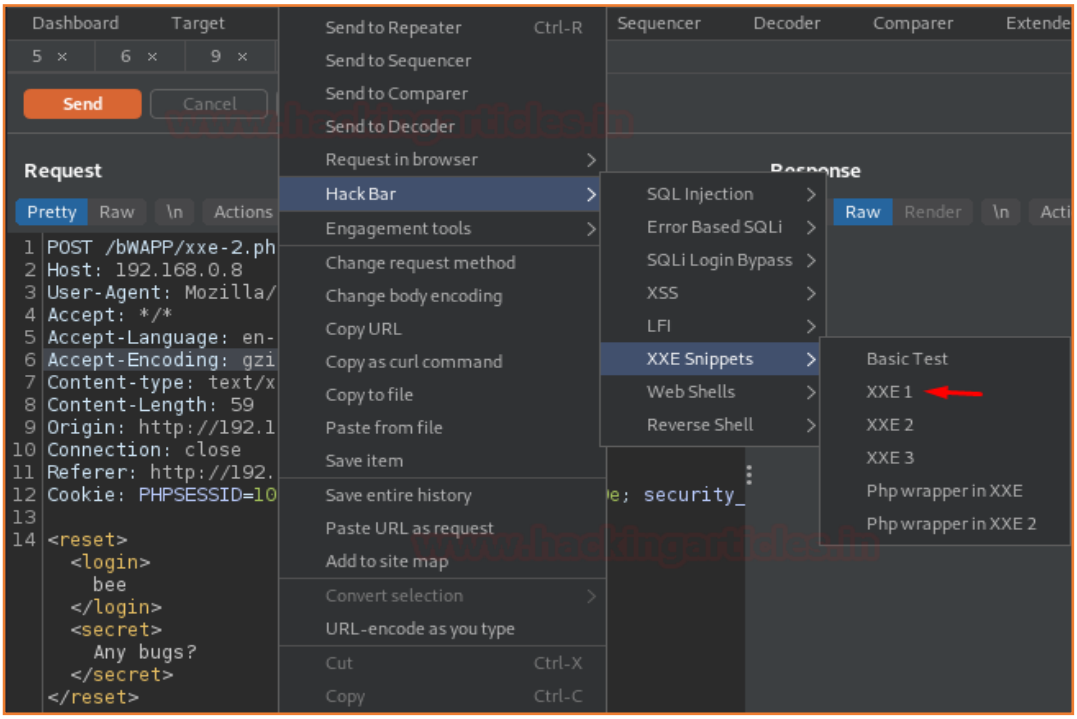
To exploit the XXE vulnerable applications, we need to type down the payloads. And yes we're a bit lazy to type the things down, thereby for this vulnerability too, hackbar is also having some great payloads. Let's check them out.

This time switch to the **XML External Entity Attacks** web-page and push the **"Any bugs?"** button with the proxy service **ON.**

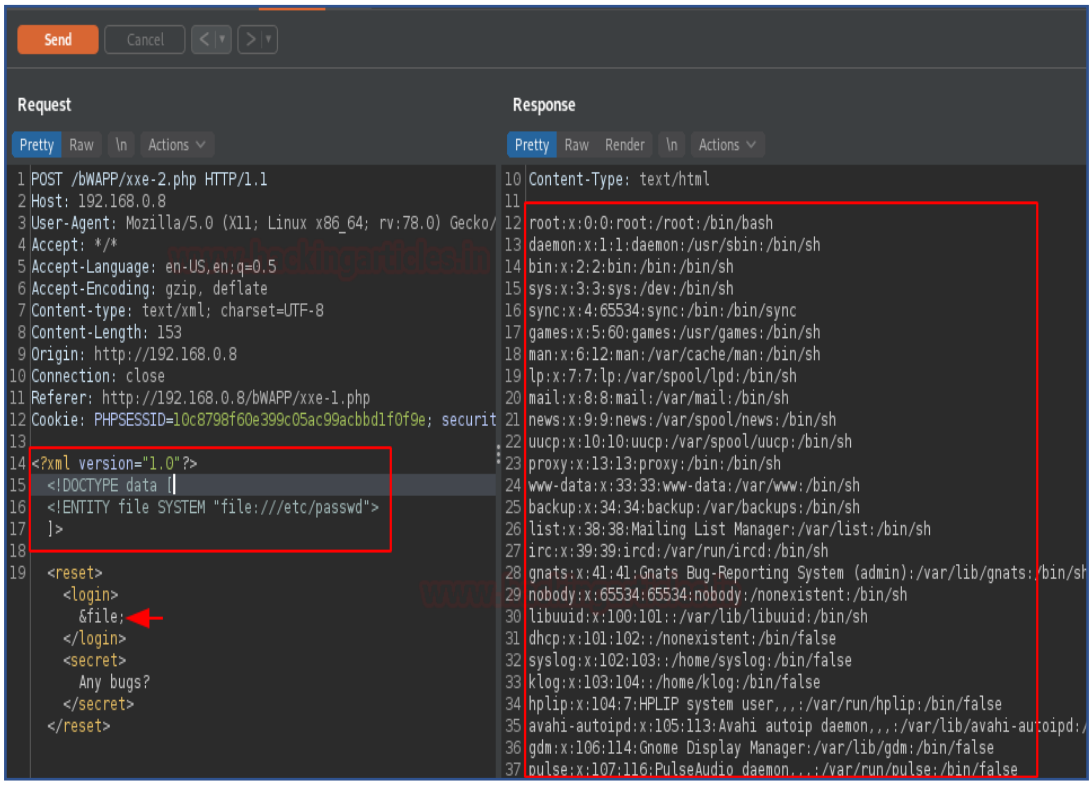


And our burpsuite did its work, the **request has been captured** again. Now, it's our turn to follow the next.

Share the captured request with the **Repeater** and hit right right-click just above the XML code and select **Hack Bar -> XXE Snippets -> XXE 1**

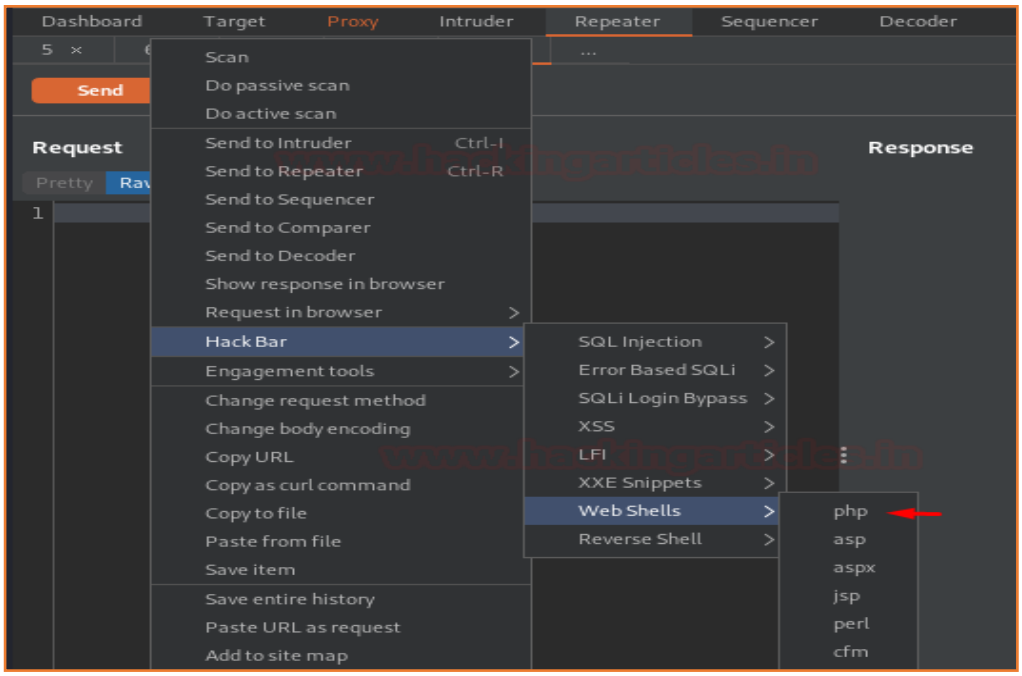


As the payload got injected, replace **bee** with the entity name (file) as **" &file; ",** and then fire the **Send** button. And within a few seconds, we got the password file over at our right eye.

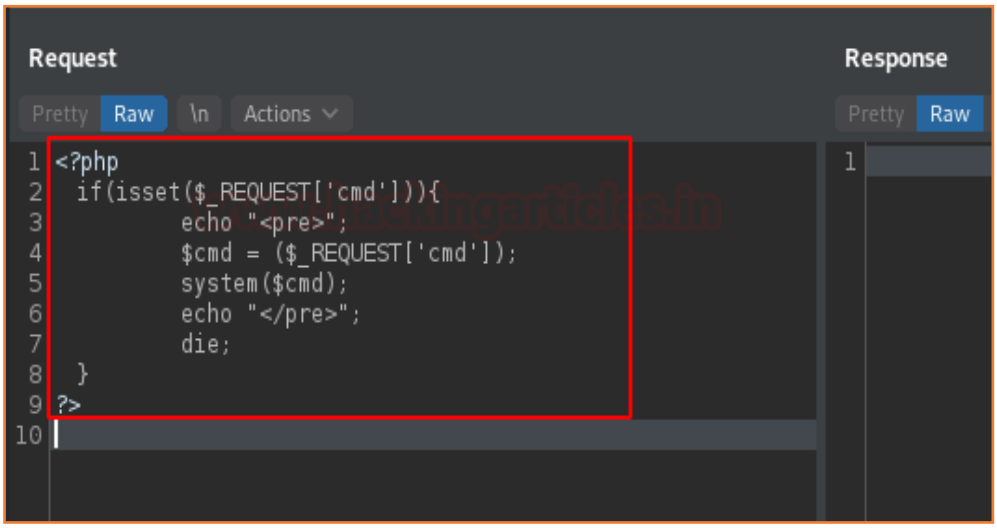


Unrestricted File Upload

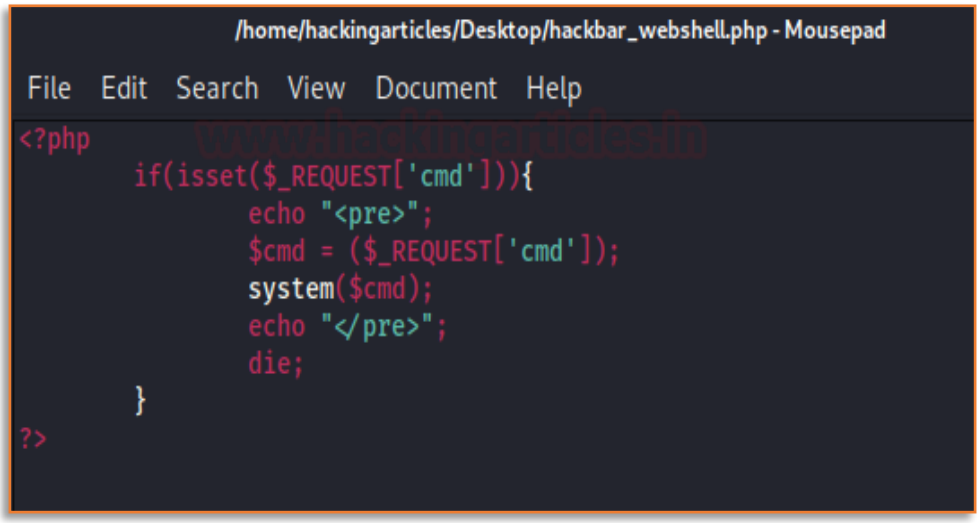
The File Upload vulnerability **allows an attacker to upload** a **file with malicious codes embedded within it**, which **thus could be executed on the server** directly resulting in **Information Disclosure**, **Remote Code Execution** and **Remote Command Execution**. Check out the article for File Upload impact.
However the uploading could not be done with this Hackbar, but it offers us the feature to create files with the malicious codes which were thus stored up into its dictionary. Let's check where they are.
Over in the Burpsuite's Repeater tab, open a new section and do a right-click at the empty portion of the Request bar and then follow to **Hack Bar -> Web Shells -> php**



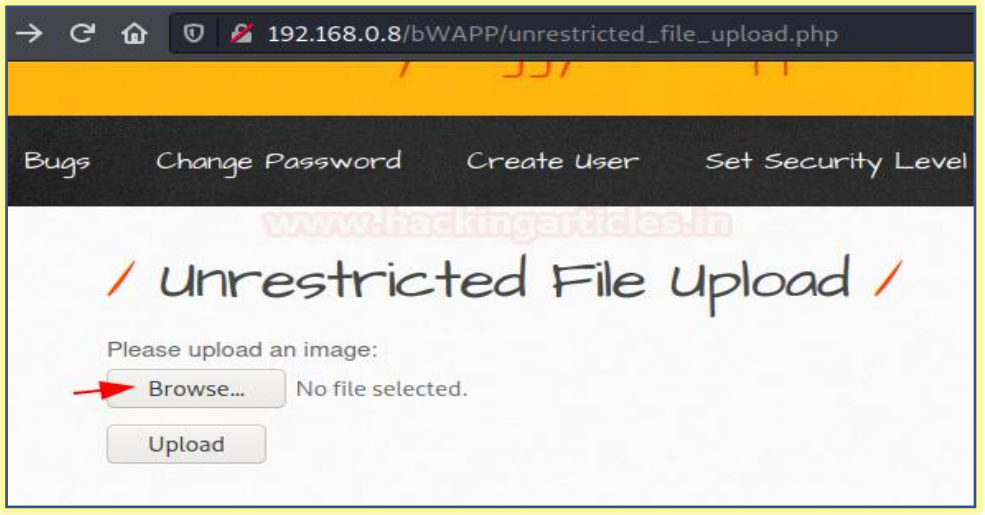
The empty section is thus filled with some code.



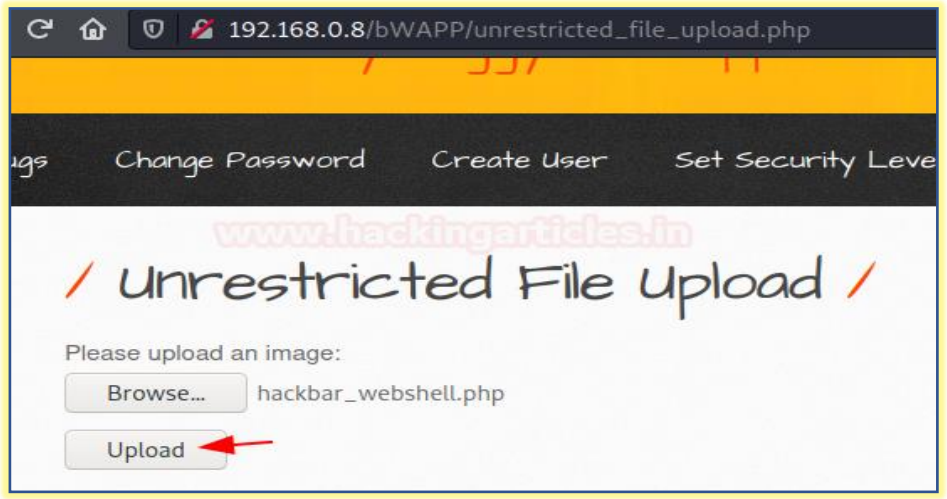
Let's copy that all and paste it into notepad, further saving it as **hackbar_webshell.php**



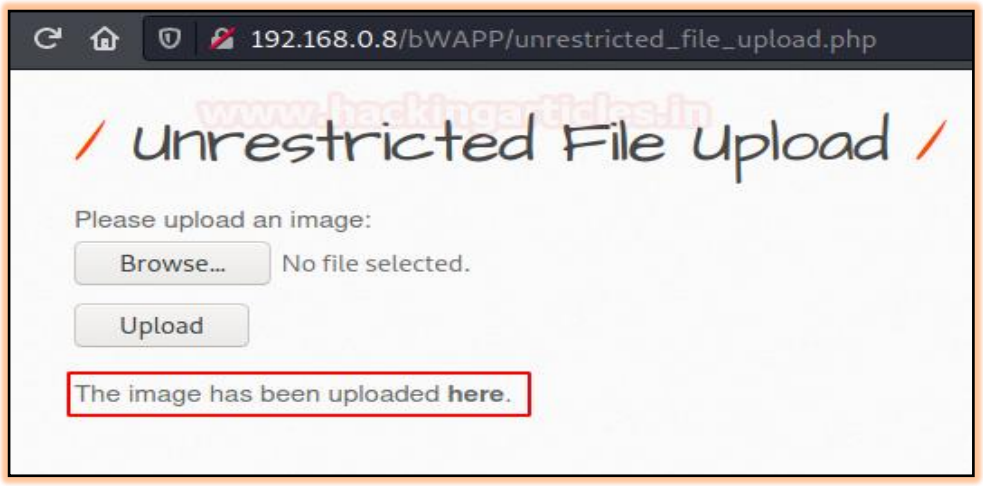
Now time to hit the vulnerability, back into the bWAPP application and opt **Unrestricted File Upload.**



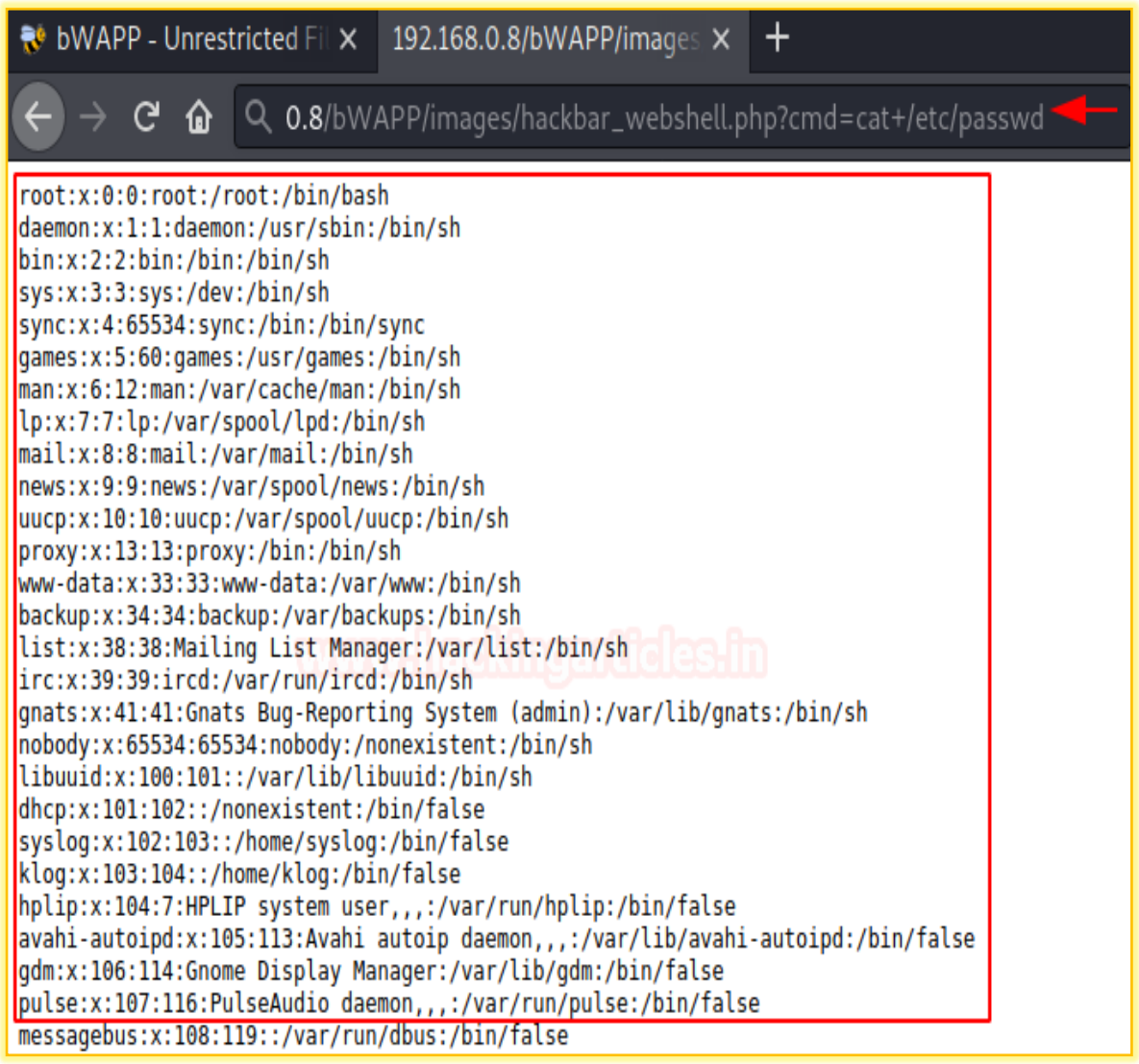
Further clicking on the **"Browse.."** button, select hackbar_webshell.php file.



As soon as the file got uploaded we got the redirection link, let's check that out.



However, the webpage was blank, as in order to execute the payload we need to call the **command** with **cmd** and that is with
http://192.168.0.8/bWAPP/images/hackbar_webshell.php?cmd=cat+/etc/passwd



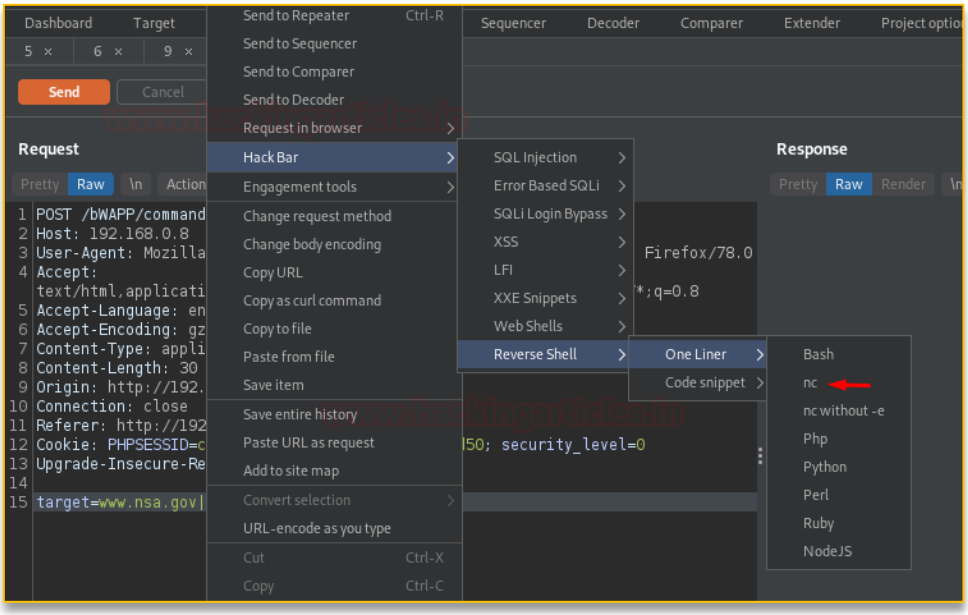
OS Command Injection

Remote Command Injection or OS Injection is the vulnerability where the attacker tries to perform system-level commands directly through a vulnerable application in order to retrieve information of the webserver. Learn more about this vulnerability from here.

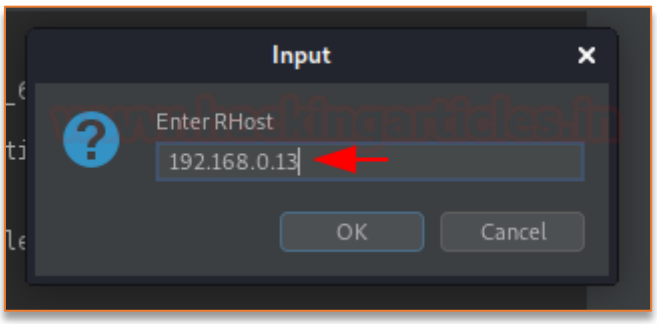
Similar to the web shells, hackbar offers us **Reverse shells** too which thus could be used over with netcat and command injection vulnerability. So let's dig them out.

For the last time, check you bWAPP and navigate to OS Command Injection and hit the **"hack"** button and capture the request there.

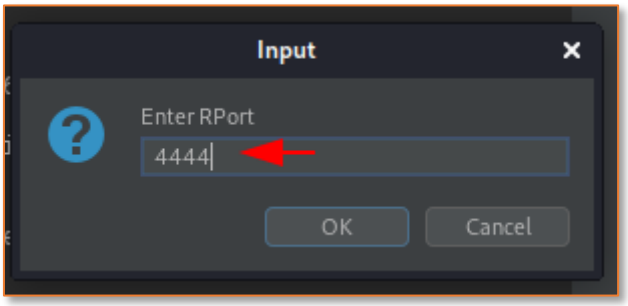
As soon as we share the captured request to the **Repeater,** we got remind off to hit right-click after **"www.nsa.gov"** and then choose **Hack Bar -> Reverse shell -> One Liner -> nc.** But, remember to set the meta-character between the two commands.



As we do so, we got the option to enter the RHost value, let enter our Kali Linux IP.

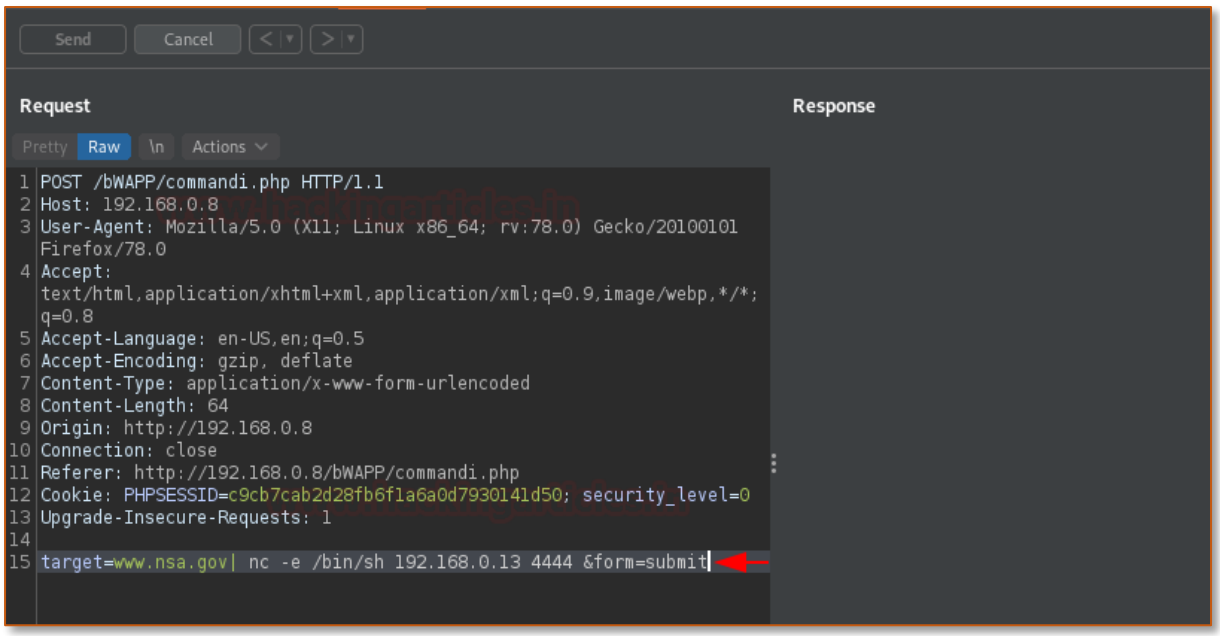


And now, our reverse shell needs a port, let's set it to **4444** our favourite one.

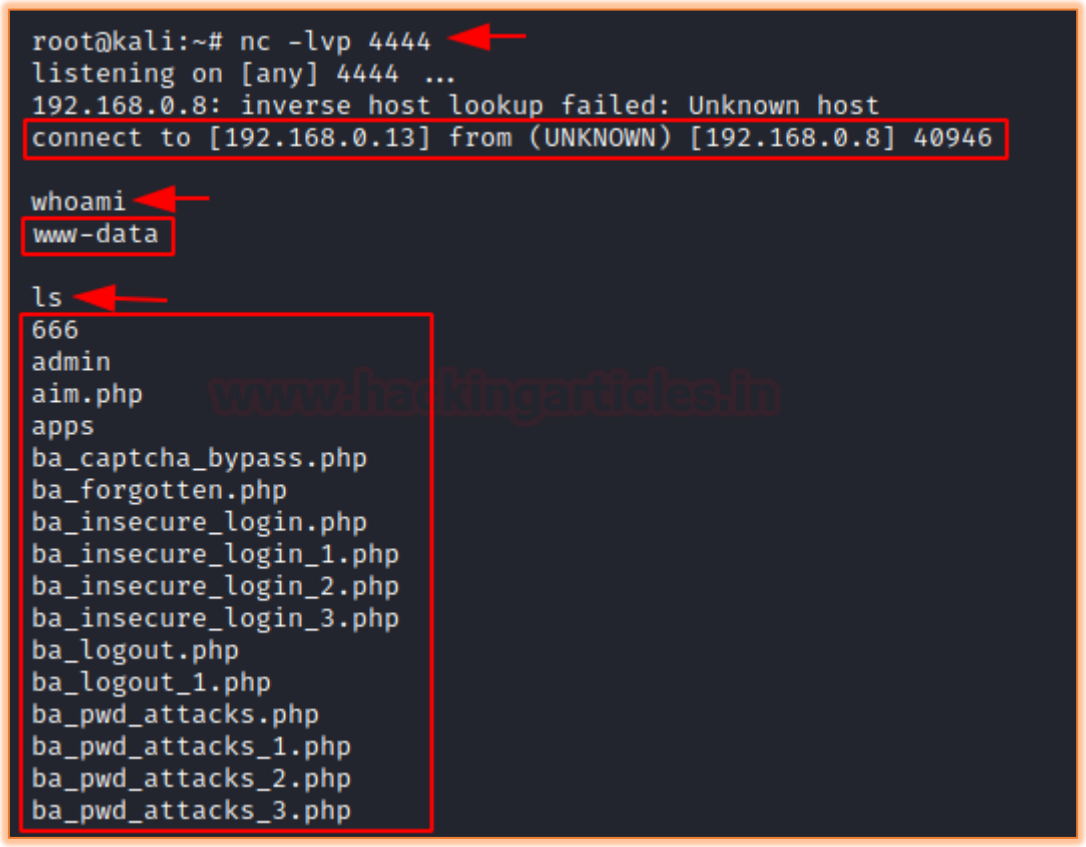


Before hitting the **"Send"** button let's initiate our netcat listener at our Kali machine with

nc -lvp 4444



As the **Send** button got pressed up, our listener fluctuates and we got the connection established. Time to dig into the web-server.



Reference

- https://www.hackingarticles.in/burp-suite-for-pentester-hackbar/
- https://www.hackingarticles.in/beginner-guide-sql-injection-part-1/
- https://www.hackingarticles.in/manual-sql-injection-exploitation-step-step/
- https://www.hackingarticles.in/comprehensive-guide-on-cross-site-scripting-xss/
- https://www.hackingarticles.in/comprehensive-guide-on-xxe-injection/
- https://www.hackingarticles.in/comprehensive-guide-on-unrestricted-file-upload/

Additional Resources

- https://github.com/d3vilbug/HackBar
- https://github.com/d3vilbug/HackBar/releases/tag/1.0

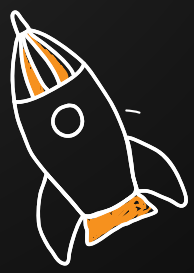
iGNITE Technologies

JOIN OUR TRAINING PROGRAMS

CLICK HERE

BEGINNER

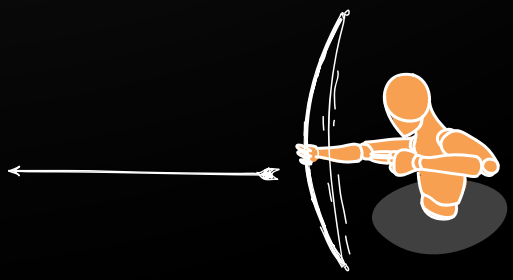
- Ethical Hacking
- Network Pentest
- Bug Bounty
- Wireless Pentest
- Network Security Essentials



ADVANCED

- Burp Suite Pro
- Android Pentest
- Web Services-API
- Advanced Metasploit
- Pro Infrastructure VAPT
- CTF
- Computer Forensics

EXPERT



- Red Team Operation
  - APT's - MITRE Attack Tactics
  - Active Directory Attack
  - MSSQL Security Assessment
- Privilege Escalation
  - Windows
  - Linux