Server Side Request Forgery

Miss.Singh Shivani[1], Dr.Ravi Sheth[2]

[1] Student, *School of Information Technology and Cyber Security, Raksha Shakti University, Lavad, Gandhinagar, Gujarat, India*

[2] *Assistant Professor, School of Information Technology and Cyber Security, Raksha Shakti University, Lavad, Gandhinagar, Gujarat, India*

***Abstract*- In this paper firstly I introduced the Basic of SSRF Attack and how this attack was worked. Then what are types of SSRF Attack and the impacts of SSRF attack which are basically affected by any web application as well as the mobile applications. And also how the attackers can compromise the server of any application as we know that the server is a very important part of any applications because it stores entire applications data. So we should at least know how the entire process is done that will be covered here. And we know that the security of web application as well as mobile applications is very important nowadays. While SSRF Attack is one of the top most vulnerabilities in web security so the analysis of this attack and how it works, what are the types of SSRF Attacks than common exploitation of these attacks all are very important points to know.**

***Index terms*- Port scanning, Blind SSRF, Internal N/W scanning, IP Formatting, Bypass IP whitelisting**

I.INTRODUCTION

January 10, 2019. Server Side Request Forgery (SSRF) is a one type of attack that can be carried out to compromise a server. The exploitation of a SSRF vulnerability enables attackers to send requests made by the web application, often targeting internal systems behind a firewall.

In a Server-Side Request Forgery (SSRF) attack, the attacker can abuse functionality on the server to read or update internal resources.

The attacker can supply or a modify a URL which the code running on the server will read or submit data to, and by carefully selecting the URLs, the attacker may be able to read server configuration such as AWS metadata, connect to internal services like http enabled databases or perform post requests towards internal services which are not intended to be exposed.

II. TYPES OF SERVER SIDE REQUEST FORGERY

- The one which displays response to attacker (Basic)
- The one which does not display response (Blind)

In Basic it means the server fetches the particular URL which is asked by the attackers for him.

For example:

These are some files which are open by the server for us.

- http://localhost:8080/?url=/etc/passwd(Will open etc/passwd and response to serve)
- http://localhost:8080/?url=https://google.com (Will request google.com on server and show response).

Blind SSRF vulnerabilities arise when an application can be induced to issue a back-end HTTP request to a supplied URL, but the response from the back-end request is not returned in the application's front-end response.

Blind SSRF is generally harder to exploit but can sometimes lead to full remote code execution on the server or other back-end components.

III. HOW IT WORKS

For example :-

GET /?url=http://google.com/ HTTP/1.1

Host: example.com.

Here example.com fetch http://google.com from its server.

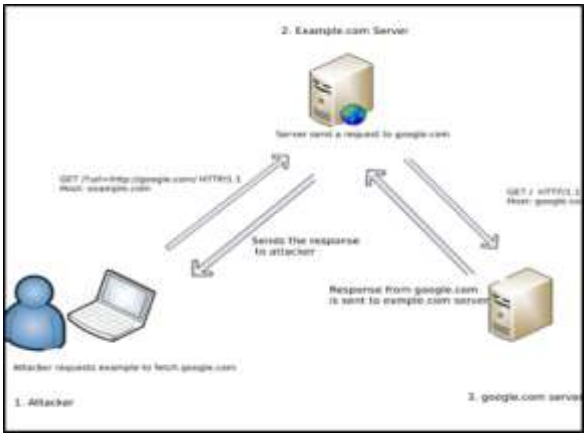

Fig - SSRF attack

1 Attacker Sends the Request the Example.com (Which is the Vulnerable Application) server to fetch the request from "google.com".
2 Now Example.com's server is vulnerable to SSRF attack so it will fetch the response of google.com request.
3 Send a response back to the Attackers.

IV. IMPACT OF SSRF ATTACKS

If an attacker is able to control the destination of the server side requests they can potentially perform the following actions:-
Bypass IP whitelisting.
Bypass host-based authentication services.
Read files from the web server.
Retrieve sensitive information such as the IP address of the web server behind a reverse proxy.
Port Scans or Cross Site Port Attack.
Protocol Smuggling.
Server Side Rendering.
Sensitive data exposure.
Scan the internal network to which the server is connected to.
Sensitive data exposure.
Scan the internal network to which the server is connected to.

V. COMMON SSRF ATTACKS

SSRF attacks often exploit trust relationships to escalate an attack from the vulnerable application and perform unauthorized actions. These trust relationships might exist in relation to the server itself, or in relation to other back-end systems within the same organization.

SSRF attacks against the server itself
1).First goto this website web-security-academy.net and select any particular product.

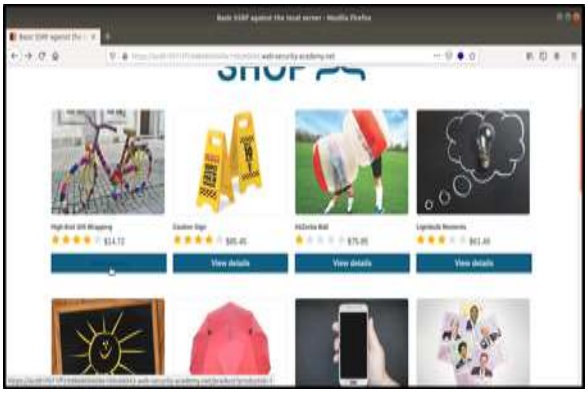

Fig -: 1

2). Click "Check stock" Button.

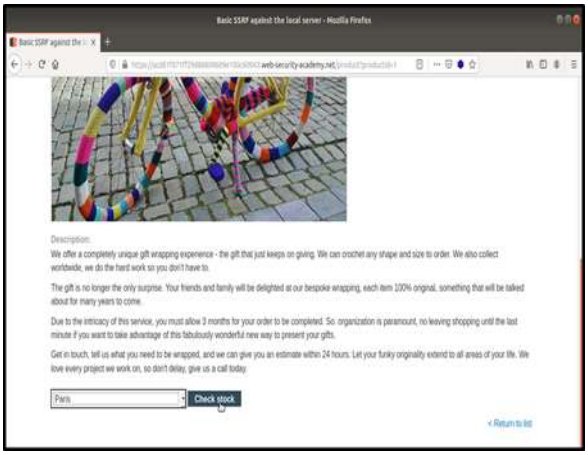

Fig -: 2

3). Intercept the following request using Burp Suite and observe the stockApi parameter.

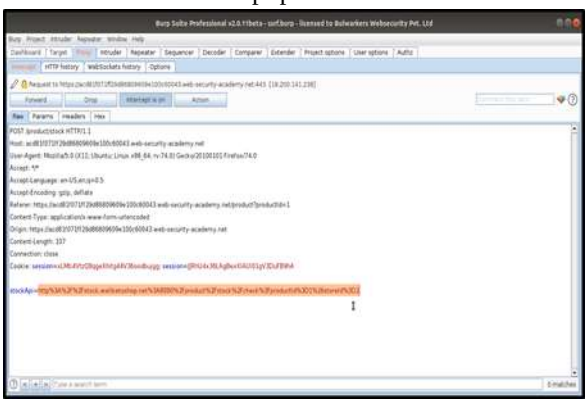

Fig -: 3

4). Now here change the URL in the stockApi parameter to http://localhost/admin and then click on forward Button.

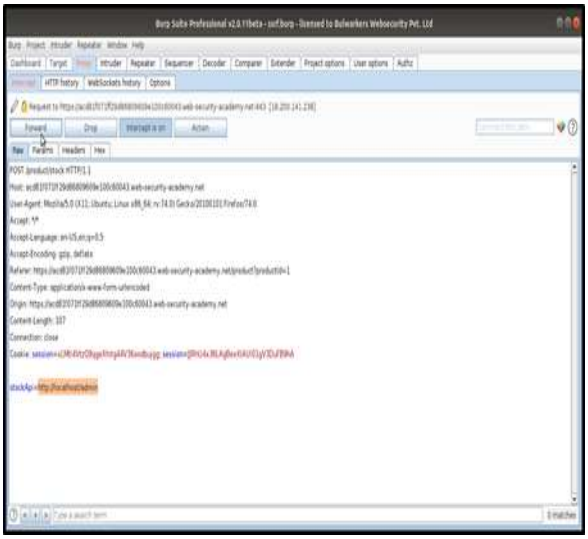

Fig -: 4

5). Now back to the website page and see we can successfully get the response of Admin Interface.

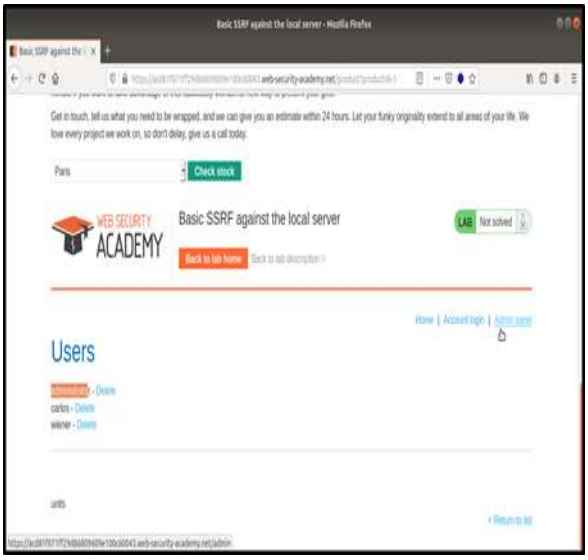

Fig -: 5

Here, the server will fetch the contents of the /admin URL and return it to the users. Then the attacker could directly visit the Admin interface and perform any malicious things So this is the First scenario of SSRF Attacks.

SSRF attacks against other back-end systems:-

1).First goto this website web-security-academy.net and select any particular product.

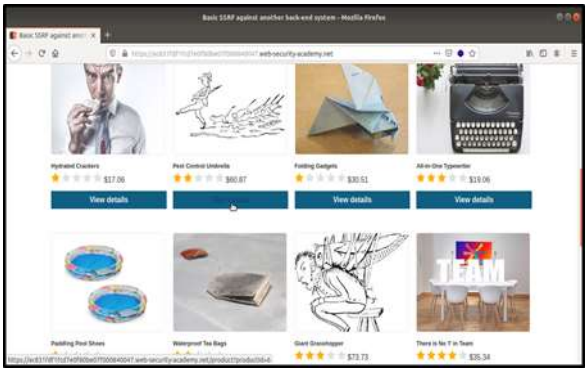

Fig -: 1

2). Click "Check stock" Button.

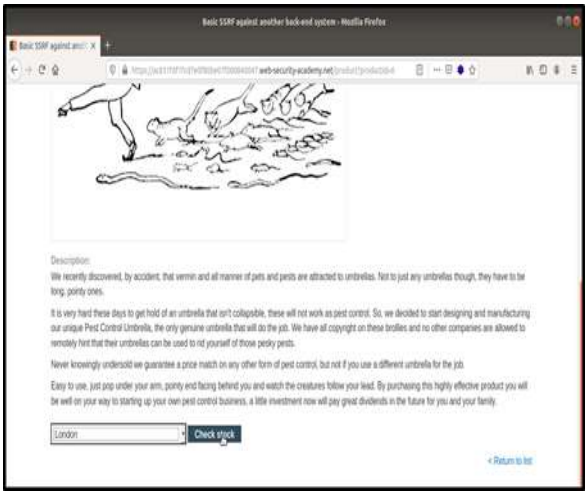

Fig -: 2

3). Intercept the following request using Burp Suite and observe the stock Api parameter.

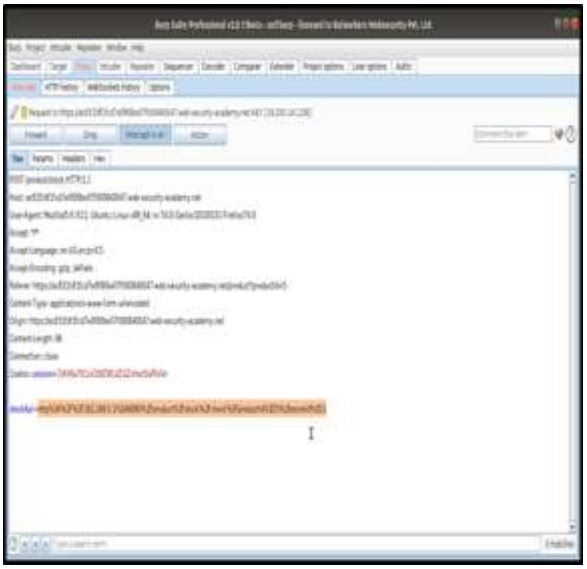

Fig -: 3

4). Now here replace the value of stock Api parameter by http://192.168.0.1:8080/admin and send it to “Repeater”.

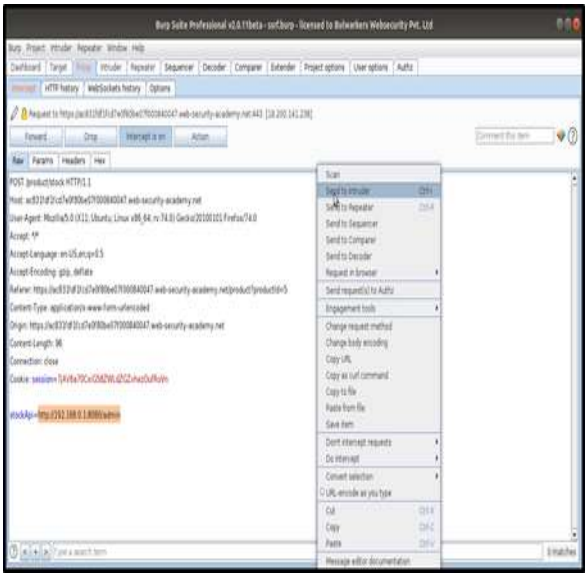

Fig -: 4

5).Then highlight the final octet of the IP address (the number 1), click the "Add §" button.

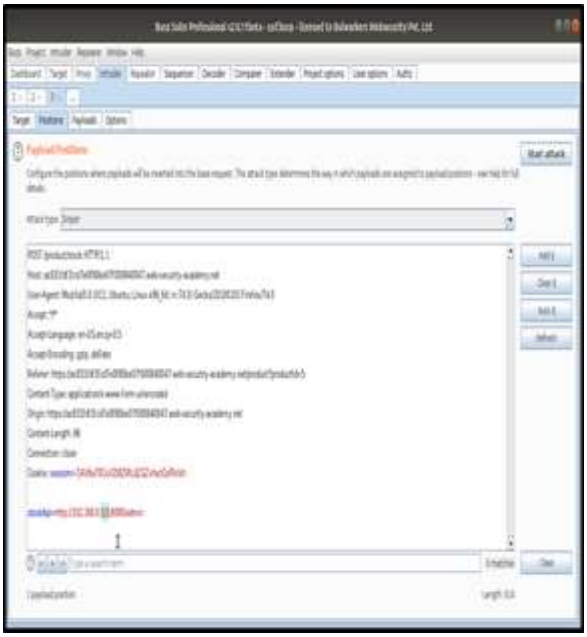

Fig -: 5

6).Now goto the Payloads tab, change the payload type to Numbers, and enter 1, 255, and 1 in the "From" and "To" and "Step" boxes respectively and click on “Start Attack” Button

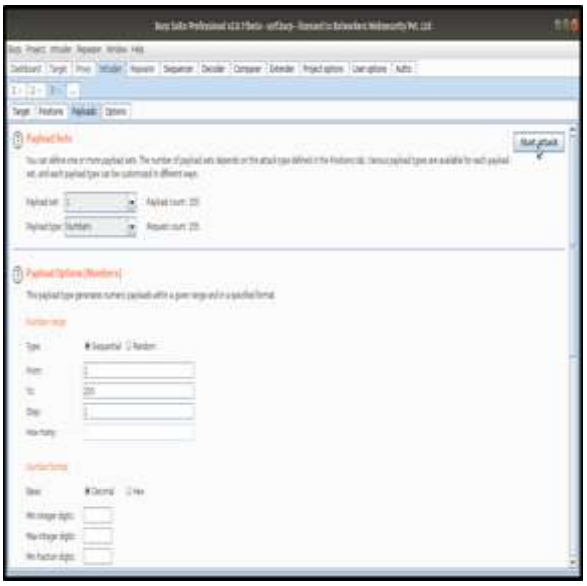

Fig -: 6

7). Here Click on the "Status" column to sort it by status code ascending. You should see a single entry with a status of 200, showing an admin interface click on that particular “Request” and send it “Burp Repeater”.

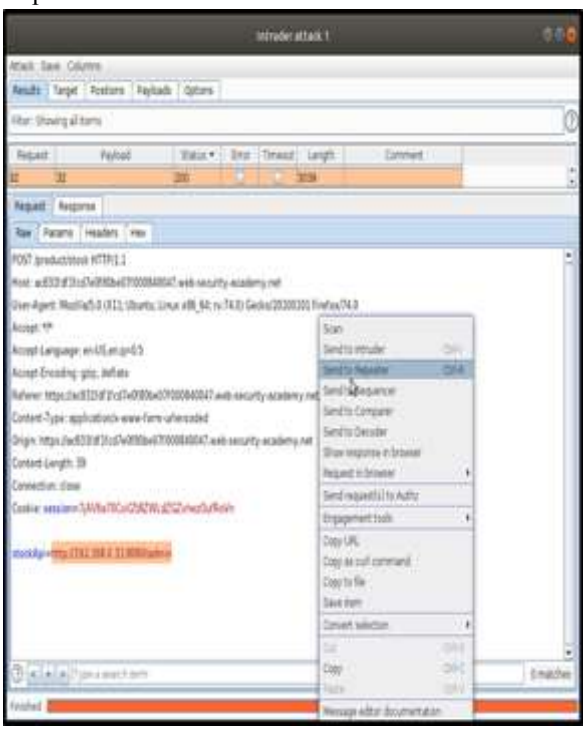

Fig -: 7

8). Now here see the response of that particular request: it will show you the Admin Panel.

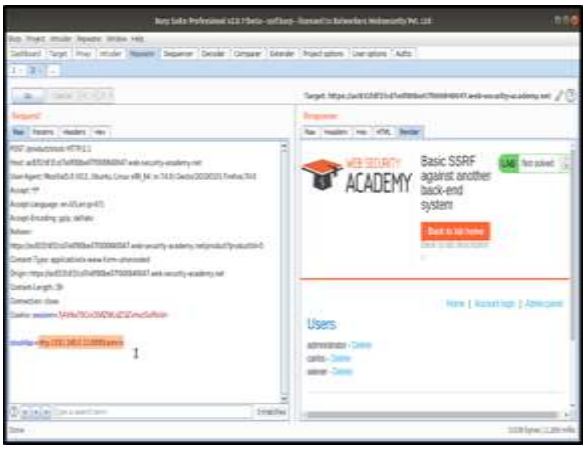

Fig -: 8

9).Here replace the stockApi parameter value with http://192.168.0.1:8080/admin/delete?username=carlos and see the response of this request.

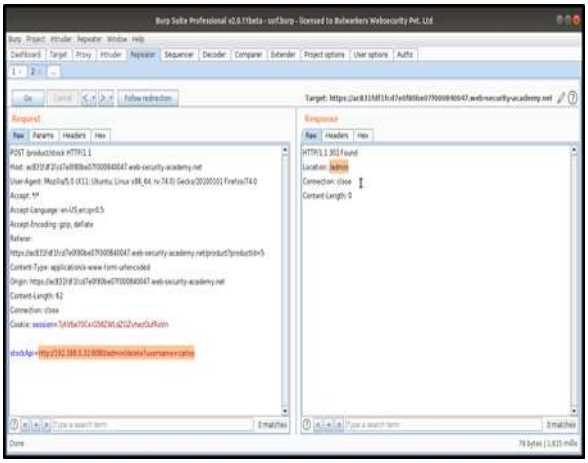

Fig -: 9

10). Now visit the Admin Panel then see successfully “Deleted” the “Carlos” user.

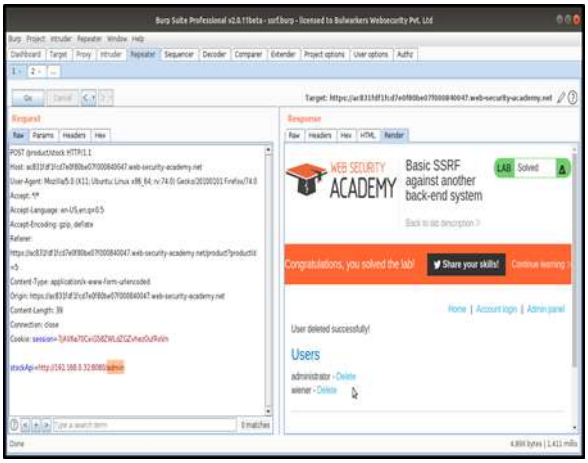

Fig -: 10

Exploiting XXE to perform SSRF attacks

1).First goto this website web-security-academy.net and select any particular product.

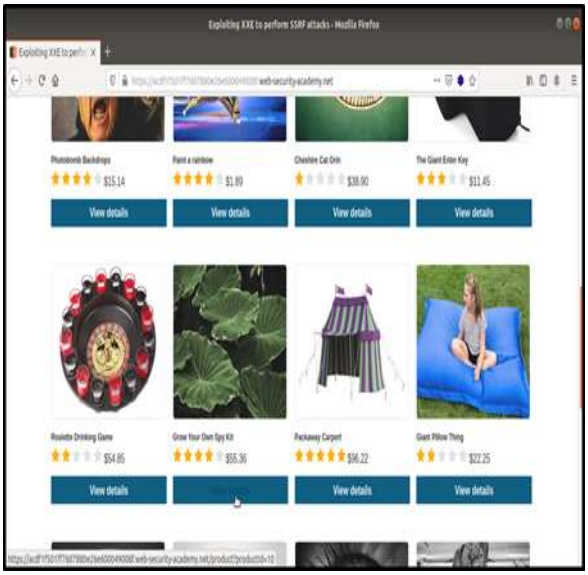

Fig -: 1

2). Click "Check stock" Button.

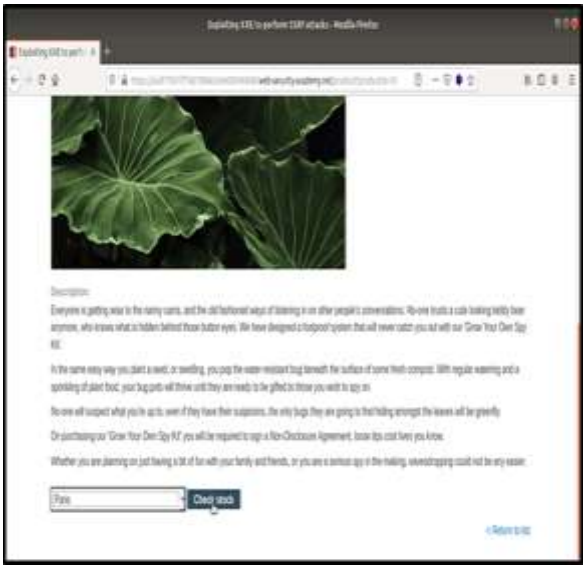

Fig -: 2

3). Intercept the following request using Burp Suite and Insert the following external entity definition in between the XML declaration and the stockCheck element:

<!DOCTYPE test [<!ENTITY xxe SYSTEM "http://169.254.169.254/">]> then send it to “Repeater”.

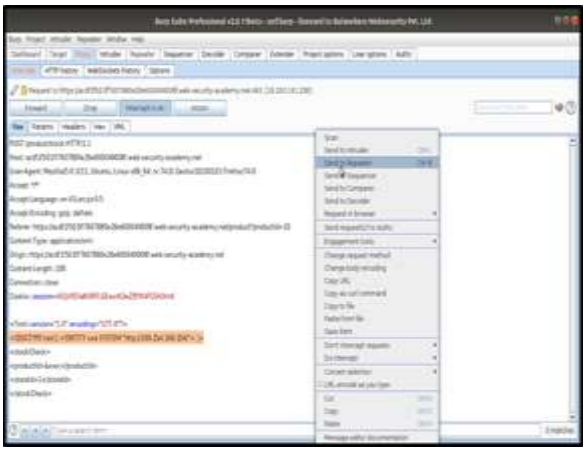

Fig -: 3

4).Click on “Go” Button and observe the “Response” The response should contain "Invalid product ID:" followed by the response from the “latest” endpoint, which will initially be a folder name.

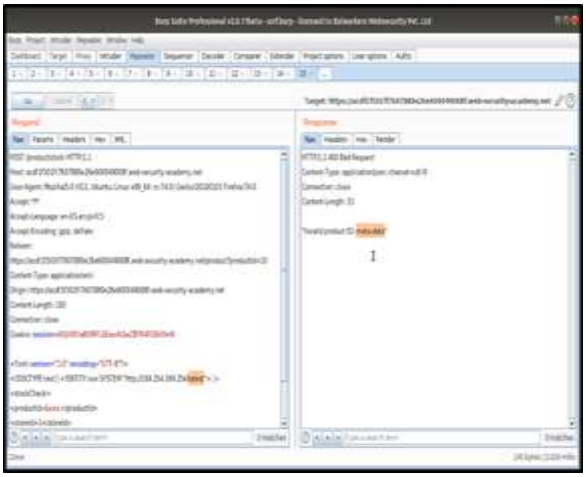

Fig -: 4

5).Here add the folder name in given path = http://169.254.169.254/latest/meta-data/.

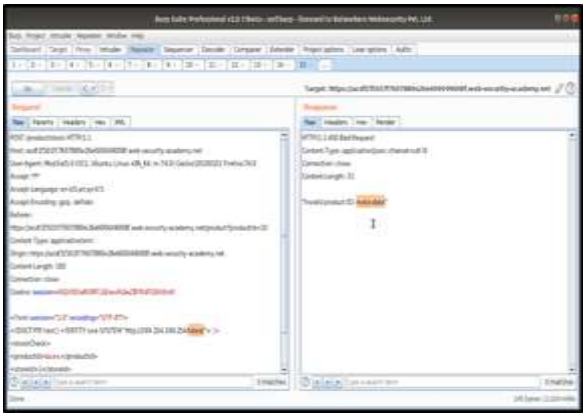

Fig -: 5

6).Here add the entire path like http://169.254.169.254/latest/meta-data/iam/security-credentials/admin and got the Secret Access Key of the host machine.

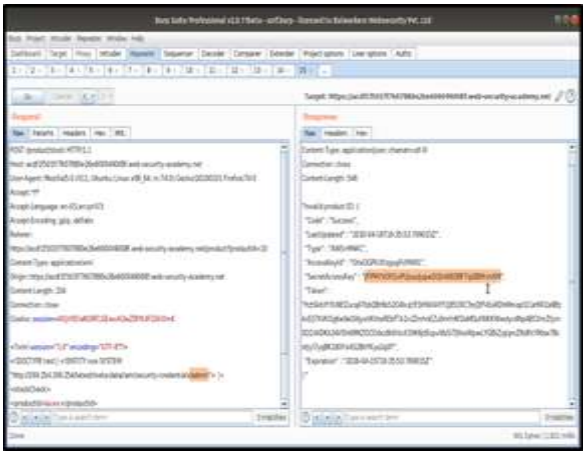

Fig -: 6

So here we exploit the XXE vulnerability to perform an SSRF attack that obtains the server's IAM secret access key from the EC2 metadata endpoint.

VI. PROBLEM STATEMENT

Nowadays attackers or say hackers all are focused on the server of any applications because server is a very important part of applications it stores entire data of applications. so if attackers are able to compromise the server of application(either it will be mobile application or web applications) then it might be possible they can compromise entire applications or will crash your applications and systems. So here my aim is to give you the brief introduction about “what exactly SSRF Attacks” and how the attackers used some mechanism to perform such types of attacks and compromise any application’s server and also propose some prevention methods by which we can prevent such types of attacks.

VII. CONCLUSION & FUTURE WORK

Here I conclude that SSRF attack is a top level vulnerability. Because In this, the Hackers are able to scan internal networks and also able to understand which network service is currently running on your system.May be read internal files or the important files of applications like ///etc/passwd.This “passwd” file is very important if attackers are able to read this file then they may crash your entire applications.

By this the attackers are also able to upload some malicious file into your servers which is basically referred to as a RCE (Remote Code Execution).
These Attacks are very dangerous so in future i want to present some strong mechanism for preventing these Attacks.

REFERENCES

[1] OWASP. About the open Web Application securityProject.https://www.owasp.org/index.php/About_The_Open_Web_Application_Securit_Project . 2018
[2] Andrew.Hann. LFI, RFI, PHP encapsulation protocol security problem learning, https://www.cnblogs.com/LittleHann/p/3665062.html, 2014.
[3] JobertAbma. SSRF Vulnerability Exploitation, https://www.hackerone.com/blog-How-To-Server-Side-Request-Forgery-SSRF, 2017.
[4] SanThosh. Server Side Request Forgery attack,https://medium.com/@madrobot/ssrf-server-side-request-forgery-types-and-ways-to-exploit-it-part-1-29d034c27978, 2019.
[5] Chandrakant Patil.SSRF Types and ways to exploit it https://hackersonlineclub.com/ssrf-server-side-request-forgery-types-and-ways-to-exploit-it-part-2/, 2019.
[6] port Swigger offers tools for web application security, testing & scanning, Emma Woollacott https://portswigger.net/daily-swig/scrapy-ssrf-to-rce-through-telnet-service-abuse, 2019