Hacking a Website, because I had nothing better to do!

Jaden Furtado

Sem 4, computer engineering

University of Mumbai

Foreword:

Just as the title suggests, I had nothing to do, so I put on my Instagram story whether you'll would want to see me hack something or build something. As the majority of you'll said hacking, I put on my hoodie and started hacking away at some random target.

Just like the previous times that I have written a writeup on my exploits as a hacker, here is another one of those moments that I have documented. The unfortunate thing about this hack, however, is that I was not able to document it like I usually would. Why? The reason to this I will get to later on in the writeup.

About the target, in this case I took down a company which supplies medical equipment. The flaws are SSRF and Vulnerable SSH and FTP ports.

I have tried to explain the exploit as best as I can. I have assumed that the reader is familiar with web-development and basic computer science. If not, you can always google the terms I have used or you can ask me to explain parts that you have not understood.

I have not disclosed the company name or any personal/business data. The website in its entirety belongs to said company. I do not own the company or the website. I have not been hired by said company and do not work for them. All the images(screenshots) you see here, of the website, are real. They were taken by me at the time of doing the hack. I have blurred the name of the company to protect their privacy until they can fix the problems I have flagged.

Jaden Furtado

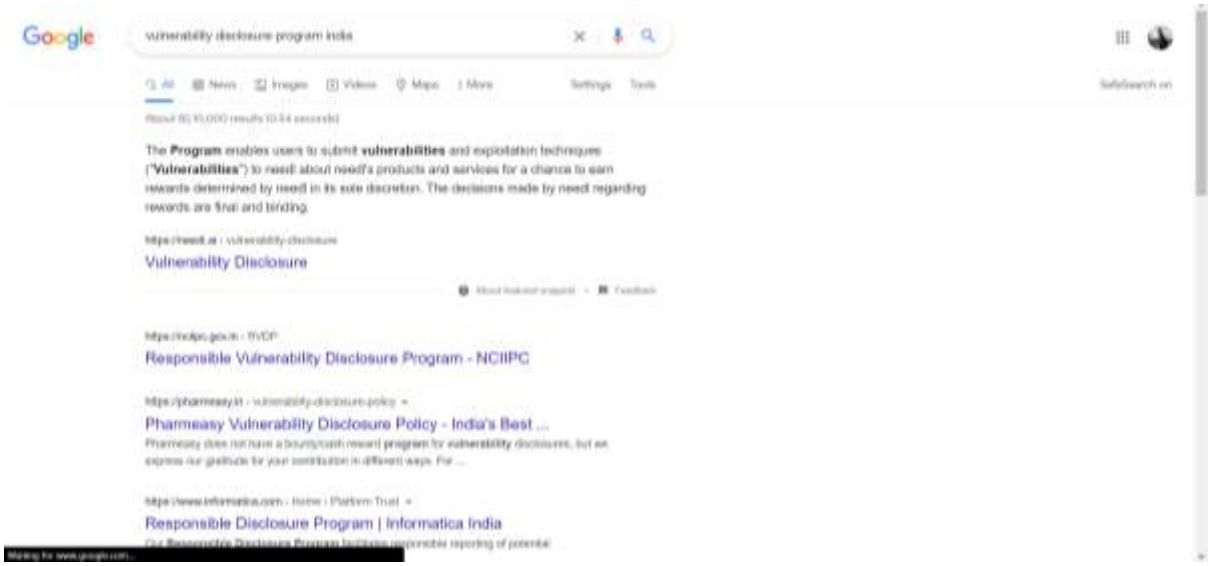
NOTE: This is a step-by-step documentation of how to hack the website. I have shown how a hacker can use this site for malicious purposes, in this document. I respect the privacy of the agencies involved and I don't intend any breach of law or harm in any way. That being said, I feel the need to document this to get the agencies to take me seriously and fix the underlying issues.

Note: This document is to be used only for the purpose of documentation and fixing this site. This document does not stand as evidence and cannot be used as evidence in any court of law in India or any other country.

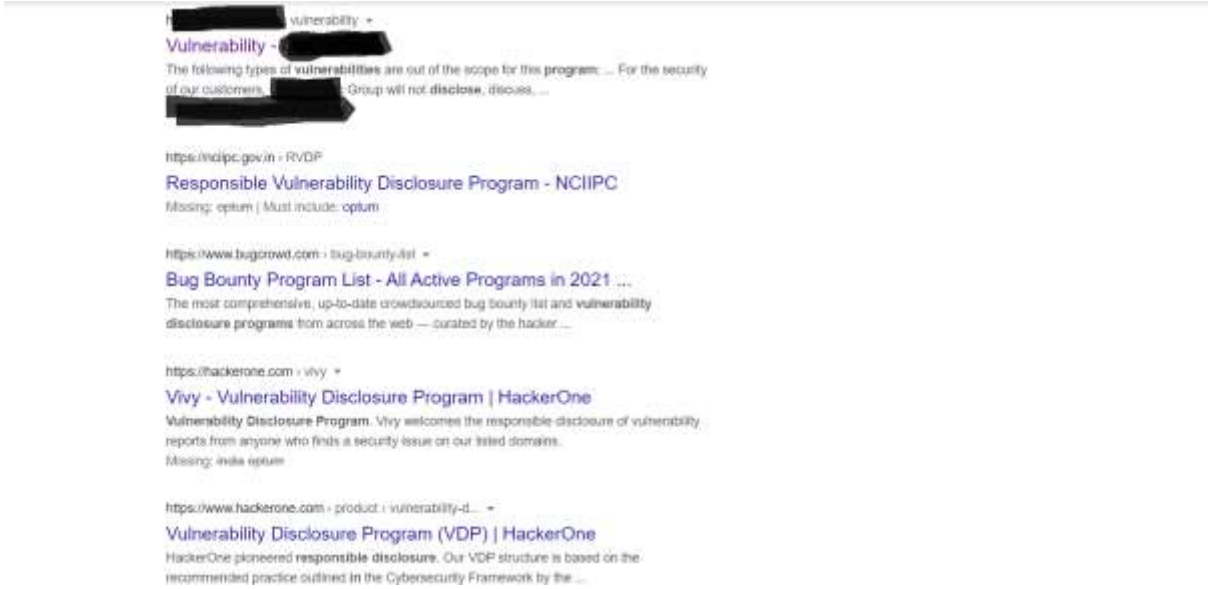
Disclaimer: Hacking is a punishable offense in India with imprisonment up to 3 years, or with fine up to two lakh rupees, or with both. Chapter IX Section 43 of IT act, 2000 prescribes a penalty for the damage to computer or computer system. It is a common thing which happens whenever a computer system is hacked. Black hats damage the system that they hack and steal the information. This enumerative provision includes a lot of activities. I do not encourage hacking in any way and am doing this solely with an educational objective. I am doing this in the best interest of the company.

Step 1: Pick a target

The task of picking a target should be fairly simple. We start by going to the best friend of a hacker, “Google”. Let’s enter a good search phrase, like



However, I don’t really want to go the 1st few results as I am doing this for fun, not for profit. I eventually came across a good enough website with a disclosure policy.



I picked the one that I have censored as you can see. Let’s go through their policy regarding hacking and vulnerability disclosure:

Vulnerability Reporting Policy

Introduction

#######takes the protection of our customer and member data seriously. We are grateful for investigative work into security vulnerabilities that is carried out by well-intentioned, ethical security researchers. We are committed to collaborating with the information security community to investigate and resolve security issues within our web sites, online services, and mobile applications that are reported to us in accordance with this Vulnerability Reporting Policy. If you have information related to potential security vulnerabilities of ####### Group, ###### or Company_name products or services, we want to hear from you.

Scope

This program is not intended for submitting complaints about ####### Group, #######, Company_name, or its subsidiaries' (hereafter referred to as "######### Group") services or products, or for inquiries regarding the availability of company web sites or online services.

The following types of vulnerabilities are out of the scope for this program:

- *Volumetric vulnerabilities (e.g., Denial of Service or Distributed DoS);*
- *Reports of non-exploitable vulnerabilities and violation of "best practices" (e.g. missing security headers);*
- *Transport Layer Security (TLS) configuration weaknesses (e.g., support for "weak" cipher suites);*
- *Fingerprinting/banner disclosure on common/public services;*
- *Self-cross-site scripting (XSS);*
- *Internal IP disclosure;*
- *Cross-site request forgery (CSRF);*
- *Un-exploitable HTTP Methods (e.g., OPTIONS or HEAD);*
- *Error-messages with non-sensitive data; and*
- *Lack of secure/HTTP-only flags on non-session cookies.*

#######may at any time update this policy, including the foregoing list of out-of-scope vulnerabilities.

Reporting a Vulnerability

If you have discovered an issue that you believe is an in-scope vulnerability, please email VulnerabilityReportingl@company_name.com. Please include the following, as applicable:

- *A detailed description of the vulnerability*
- *The full URL*
- *A Proof of Concept (POC) or instructions (e.g. screen shots, video, etc.) on how to reproduce the vulnerability or steps taken*
- *Entry fields, filters, or other objects involved*
- *Risk or exportability assessment*
- *Instructions for how to reach you with follow up questions*

Offering a solution is encouraged but not required. Lack of detailed vulnerability explanation may result in delays in our response and subsequent potential actions on the finding.

Bug Bounties

#######does not currently offer a bug bounty program. However, we appreciate the efforts of security researchers who take time to investigate and report security vulnerabilities to us in accordance with this policy.

What to Expect

Upon receipt of the vulnerability report, #######may send an automated response as acknowledgement. #######may contact reporter(s) if additional information is needed to assist with the investigation. For the security of our customers, #######will not disclose, discuss, or confirm security issues.

Public Notification

In order to protect our customers, #######requests security researchers not post or share any information about a potential vulnerability in any public setting until we have researched, responded to, and addressed the reported vulnerability and informed customers and stakeholders as needed. The time to address a valid, reported vulnerability will vary based on impact of the potential vulnerability and affected systems.

Guidance

This policy prohibits the performance of the following activities:

- *Hack, penetrate, or otherwise attempt to gain unauthorized access to #######software or systems;*
- *Active vulnerability scanning or testing;*
- *Disclose or use any proprietary or confidential #######information or data, including customer data; or*
- *Adversely affect the operation of #######software or systems.*

Security researchers must not violate any law, or access, use, alter or compromise in any manner any #######data.

If you have any questions regarding this policy or the guidance above, please contact our security team for guidance: VulnerabilityReporting@company_name.com.

Policy Definitions

***Vulnerability:** A weakness in the design, implementation, operation or internal control of a process that could expose the system to adverse threats from threat events.*

***Denial of Service (DoS):** An attack on a service from a single source that floods it with so many requests that it becomes overwhelmed and is either stopped completely or operates at a significantly reduced rate.*

***Distributed Denial of Service (DDoS):** An attack on a service from multiple compromised computer systems that floods it with so many requests that it becomes overwhelmed and is either stopped*

completely or operates at a significantly reduced rate, thereby denying service to legitimate users or systems.

Transport Layer Security (TLS): *A protocol that provides communications privacy over the Internet. The protocol allows client/server applications to communicate in a way that is designed to prevent eavesdropping, tampering, or message forgery.*

Self-Cross-Site Scripting (XCSS): *A social engineering attack to gain control of a victim's web accounts via the victim unknowingly running malicious code on their own web browser.*

Cross-Site Request Forgery (CSRF): *A type of malicious exploit of a web site where unauthorized commands are transmitted from a user that the web site trusts. This is also known as a one-click attack or session riding.*

Effective Date

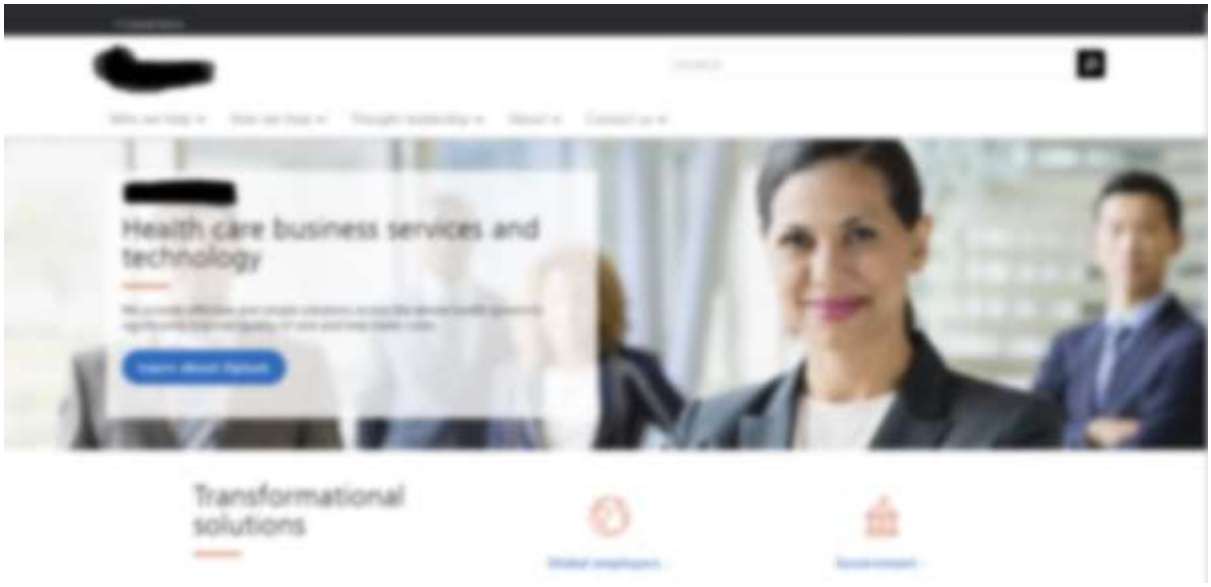
The effective date of this policy is April 1, 2019.

After taking a quick look at the above, I think I am well within my rights to try and hack them. And so step1 is complete. I have a target.

Step 2: Find vulnerabilities

A target has flaws, always, no matter how secure the software may seem! So as the hacker, it's your job to find them and exploit them.

Let's go to the home page:



The website is made using JSF. What is JSF? Java Server Faces (JSF) is a Java-based web application framework intended to simplify development integration of web-based user interfaces. JavaServer Faces is a standardized display technology, which was formalized in a specification through the Java Community Process.

I started by going to my best friend for web hacking, "ctrl+shift+I", i.e. open inspect element. Here I found, a few interesting links, including this one:

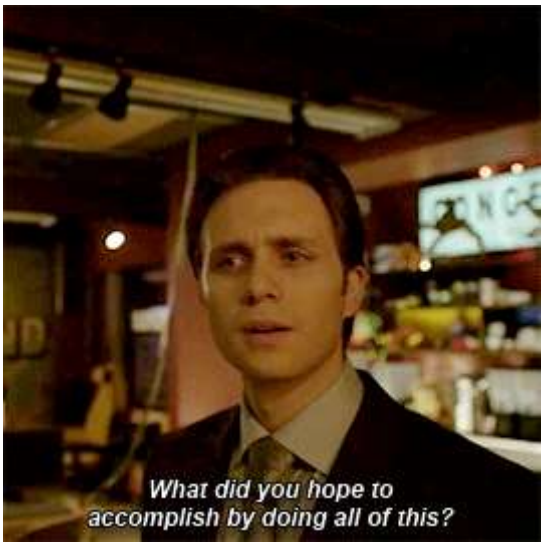
http://www.company_name.in/bin/company_name3/featuredarticles?path=https://www.company_name.com%2Fthought-leadership-topics%2Fhealthcare-operation

The above is a link to the API endpoint which returns JSON data:

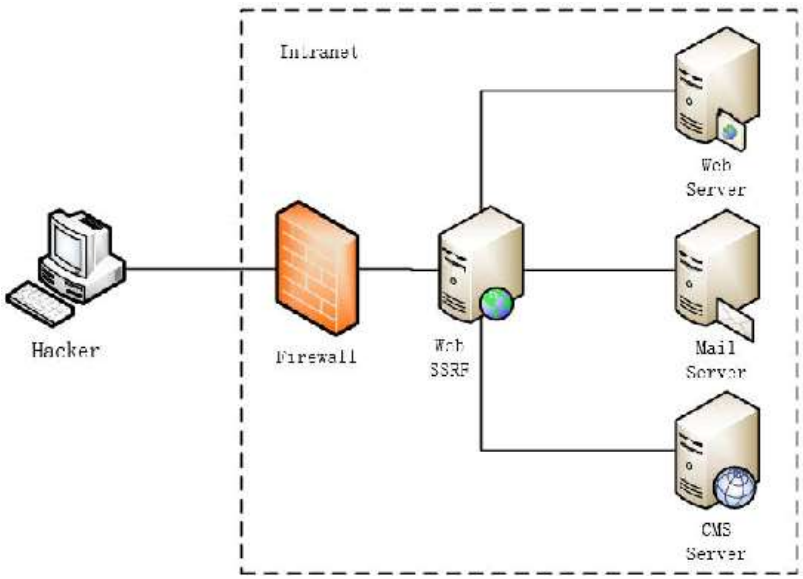

```
{"page":{"navTitle":"SEIQR Model Simulates COVID-19 Propagation & Spread in India","imagePath":"/content/dam/company_name/company_name/india/images/Covid-Base-card.jpg","title":"SEIQR Model Simulates COVID-19 Propagation & Spread in India","description":"In this white paper, company_name proposes a \"Susceptible - Exposed - Infectious - Quarantined - Recovered\" (SEIQR) model to simulate the spread of COVID-19 in India.","subtitle":"The SEIQR model simulates the spread of COVID-19 in India.","tags":["company_name3:en/thought-leadership-topics/technology","company_name3:en/thought-leadership-topics/data-and-analytics","company_name3:global/india/segment/payer","company_name3:en/articles","company_name3:en/articles/articles-blogs","company_name3:global/india/segment/government","company_name3:global/india/segment/global-employer"],"articleType":"Articles and blogs","video":"false"}}
```

Looking at the link, I got an idea, I replaces the “https://www.company_name.com%2Fthought-leadership-topics%2Fhealthcare-operation” with “https://localhost/” so that the link effectively became:

http://www.company_name.in/bin/company_name3/featuredarticles?path=https://localhost/



This is a classic example of an SSRF attack. By doing an SSRF attack, I am able to hit internal IP address of a company and get them to leak data on their internal servers.



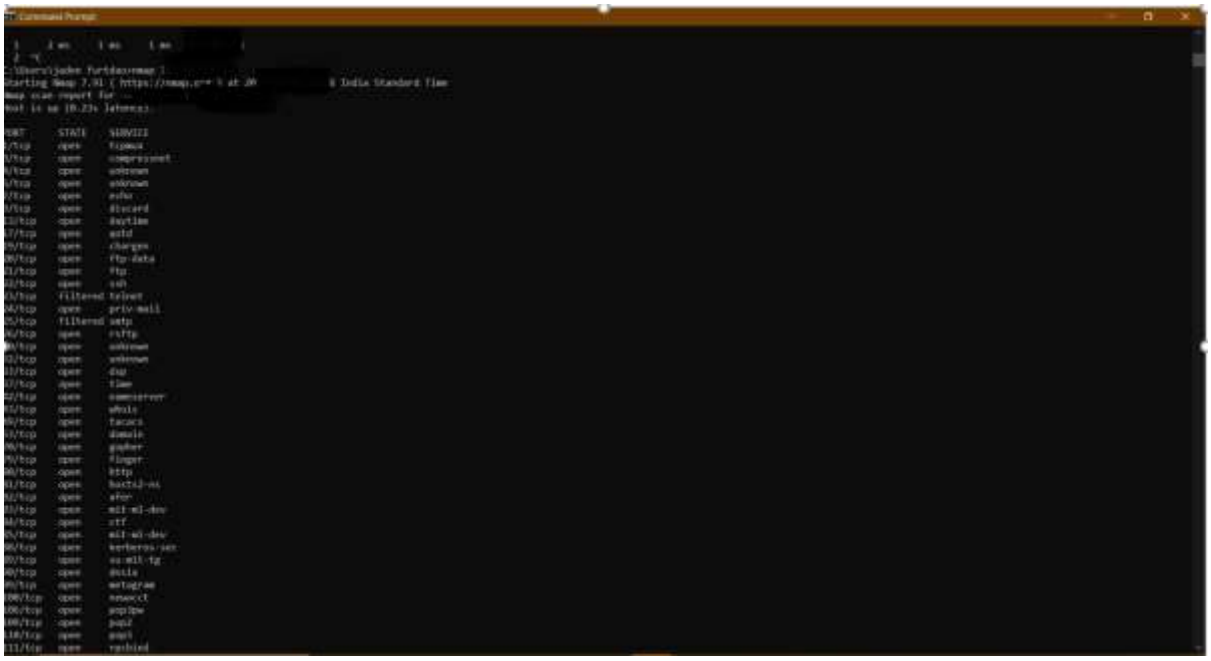
I can confirm that I hit an internal server, however, I am unable to retrieve any internal data of relevance. I am also able to hit the endpoint through my own local server, but again, I am unable to get any relevant data. Why? Because I don’t know how to do that!

Nothing too interesting so far, and yet, believe it or not, I spent a good 8-10 hours getting this far, but this is a dead end for me! But that is the reality of hacking.

Feeling down, I try and find another flaw, if I can.

I decided to try and scan the website's server's ports. So, I use Nmap to scan the ports, using the Ip address of the website.

This is what I found:



AHA! Here we go to the next step!

Step 3: Exploit said vulnerabilities!

The server is misconfigured. All the ports, except for a few, are open as you can see. This gives me a few ideas. Two ports that catch my eye are port no. 21 and 22. I.e, ftp and ssh.

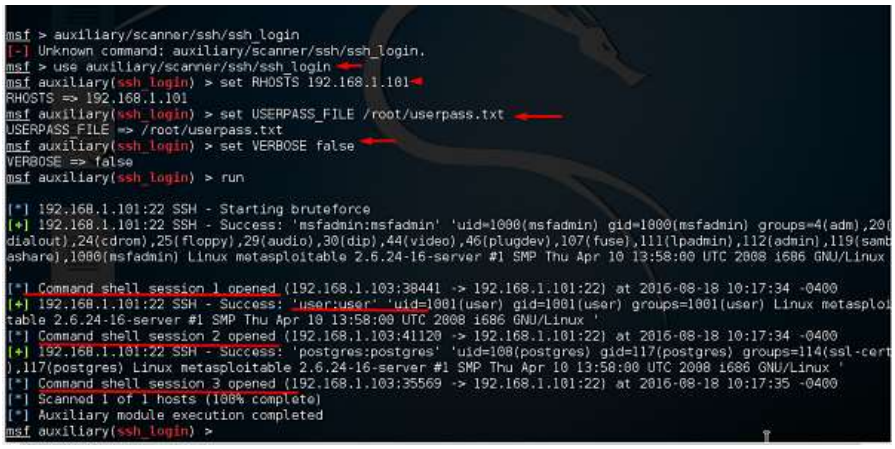
I tried to check if I am able to connect to the ssh port using telnet. And I am able to connect to it successfully. However, this by itself is not too interesting to me. I want to be able to send files across to the server. So, I need the SSH or FTP key to the server or the server's username and password.

So, we go to Metasploit and I will try and brute force the credentials.

Please note, I have changed the IP address, to protect the company.

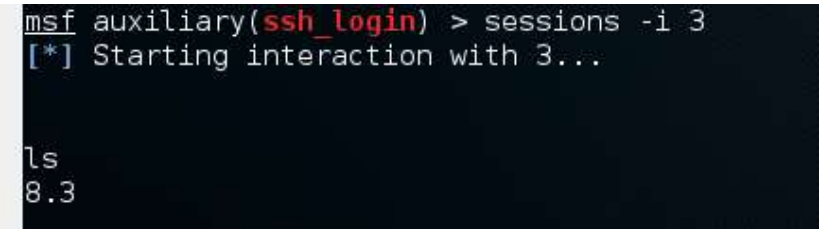
To attack the SSH service, we can use the auxiliary: **auxiliary/scanner/ssh/ssh_login**

As you can see in the following screenshot, we have set the RHOSTS to 192.168.1.101 (that is the victim IP) and the username list and password (that is userpass.txt). Then we apply the **run** command.



As can be seen in the above screenshot, three sessions were created. It means three combinations were successful. We have underlined the usernames.

To interact with one of the three sessions, we use the command **msf > sessions –i 3** which means we will connect with session number 3.

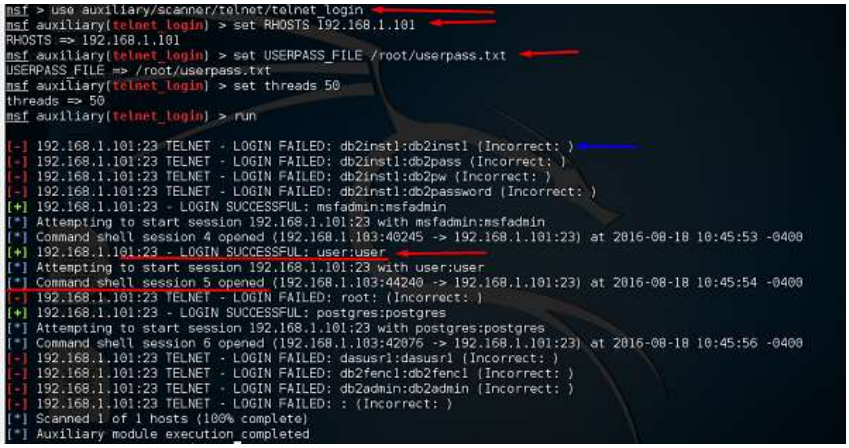


Attack the Telnet Service

The apply a brute-force attack on a Telnet service, we will take a provided set of credentials and a range of IP addresses and attempt to login to any Telnet servers. For this, we will use the auxiliary: **auxiliary/scanner/telnet/telnet_login**.

The process of using the auxiliary is same as in the case of attacking an FTP service or an SSH service. We have to use the auxiliary, set RHOST, then set the list of passwords and run it.

Take a look at the following screenshot. Highlighted in blue arrow are the incorrect attempts that the auxiliary did. The red arrows show the successful logins that created sessions.



In our case, the username of the server for SSH is “admin” and password is, well, the name of the company!

Not too secure, is it? I guess the only way to stop me is by not allowing me to access the website. **>:~>**

I am now able to send files and data to the server through the SSH port. So where are the screenshots? Weeeeellllllllllll……..

I tripped a security alarm while I was snooping around in the server and brute forcing the credentials. The next day, they detected an intruder, deleted the files I sent and changed the credentials and also closed off all server ports. And yes, banned my IP address (I spoke too soon). Before I could take a Screen Shot of what I had done!

No, need to worry, we exchanged emails and I was unbanned with immediate effect…. As for any evidence? This was their response

Conclusion:

SSRF:

As I already said, I was unable to determine the impact of SSRF bug I found, before they fixed it.

The SSH and other open ports:

Well, as I said, in the days after I tripped the alarm, they sealed off all ports to the server, changed the username and password and deleted the files that I had sent to the server via SSH.

The main takeaway from all this?

Always take a screenshot! And I am not just talking about chats!

When you are hacking, document everything that you do. This will let you come back to refer to it in the future and also help you in proving your claims should stuff that you did not anticipate go down, as in my case.

So yeah, if you have read this far, without taking a shortcut, kudos to you!