VAAST (Vocal and Automated Shortcut Tool)

Submitted in partial fulfilment of the requirements for the award of the degree of

Six Months Training

BACHELOR OF TECHNOLOGY

(Information Technology)

Submitted By:-

Vanshika Chaudhary
University Roll no:- 1905412
Class Roll no:-1921112

GURU NANAK DEV ENGINEERING COLLEGE
LUDHIANA-141006, INDIA

Abstract

The “VAAST- Vocal and Automated Shortcut Tool “ has been built over the shell script and contain a lot of features like a calculator and shortcut to sites and also the shortcut to the tools that are predefined in the “VAAST” by us during the scripting time. In this there are some funny packages fo the fun purpose. The cyber tools are used for the different purposes like the information gathering and also perform the phishing attacks but only for the educational purposes. Also, there is an option for bug bounty incase the user want to find out any bug inside the site or rthe page over the internet then, you can perform it with just a single click. All the features are enclosed inside the security system that is built to keep the system secure from the invaders. For, the security purpose the passwords are commonly there but they are cracked easily with the help of brute force attack in which the error and trial method is commonly used. And we know that time is most important thing while cracking the password. So, it contains some additional features to increase the time consumption while cracking the password by automatically shutting down after receiving the wrong username or password.

Acknowledgement

We are highly grateful to Dr. Sehijpal Singh, Principal, Guru Nanak Dev Engineering College (GNDEC), Ludhiana, for providing this opportunity to carry out the minor project work at making 'VAAST ' using unity. The constant guidance and encouragement received from Dr. K.S. Mann, H.O.D., IT Department, GNDEC Ludhiana has been of great help in carrying out the project work and is acknowledged with reverential thanks. We would like to express a deep sense of gratitude and thanks profusely to Pf. Mohanjit Kaur , without his wise counsel and able guidance, it would have been impossible to complete the project in this manner. We express gratitude to other faculty member of Information Technology department of GNDEC for their intellectual support throughout the course of this work Finally, we are indebted to all whosoever have contributed in this report work.

Vanshika Chaudhary (URN 1905412)

Certificate

Solitaire infosys Simple Creative innovative

S.No. 296343

Certificate of Training

This certificate has been awarded to Mr/Ms **Vanshika Chaudhary** from **GNE Ludhiana** who has undertaken an internship program of **6 Months** from **22/08/2022** to **31/01/2023** in **Networking with Cyber Security** Department from Solitaire Infosys Pvt. Ltd.

During the tenure of this internship with us, we found the candidate self-starter and hardworking. Also he/she had worked sincerely on the assignments and his/her performance was satisfactory to be part of the team.

We wish the Candidate success for all the future endeavors.

For Solitaire Infosys Pvt. Ltd.

CERTIFIED ISO 9001 2015 COMPANY

Human Resources Department

Note: To check the authentication of certificate, please visit www.slinfy.com

Contents

1 Introduction

1.1 Introduction to Project (Kali Linux)

Operating System is the main system software which is responsible for the flawless working of the machine. Some Operating Systems are designed for some specific purposes. Though we could use them for anything we want to, but they have some special tools or services available feasibly to its users which makes it a good OS for the specific purpose. Like we generally prefer Windows in case of gaming as most of the games are available for windows itself. Likewise, we prefer mac OS for designing related purposes as most of the designing software is easily available for mac and can be used flawlessly. In the same way when we have an OS for Network Security, Digital Forensics, Penetration testing, or Ethical Hacking named Kali Linux.
Kali Linux is a Debian-derived Linux distribution that is maintained by Offensive Security. It was developed by Mati Aharoni and Devon Kearns. Kali Linux is a specially designed OS for network analysts, Penetration testers, or in simple words, it is for those who work under the umbrella of cybersecurity and analysis. The official website of Kali Linux is Kali.org. It gained its popularity when it was practically used in Mr. Robot Series. It was not designed for general purposes; it is supposed to be used by professionals or by those who know how to operate Linux/Kali. To know how to install Kali Linux check its official documentation. Kali Linux is to be used by those who are professional penetration testers, cybersecurity experts, ethical hackers, or those who know how to operate it. In simple words, if you know how to use Linux and its terminal commands, architecture, system, and file management then you are good to go with Kali Linux. And if you are not, then we will recommend you first start with Ubuntu distribution and get your hands on Linux and after sufficient practice, you could give Kali Linux a try. This will not only save your time of searching on the internet but also will make you use it with ease. However, if you're a professional penetration tester or studying penetration testing, there's no better toolkit than Kali Linux.

1.2 Why Kali Linux ??

If you are interested in penetration testing or cybersecurity stuff you need some specific tools to perform some tasks which come pre-installed and settled up in Kali Linux so you may directly use them without doing any configuration. Or in case if one wants to check the vulnerabilities on a website or want to know security-related bugs in any application then it is great to go with Kali Linux. Many people think that Kali is a tool for hacking or cracking social accounts or web servers. This is one of the biggest myths about Kali Linux. Kali Linux is just another Debian distribution with a bunch of networking and security tools. It is a weapon to train or defend yourself not to attack anyone. Kali Linux was designed mainly for professionals. It is for those who want to get their hands in Penetration Testing, Cyber Security, or Ethical Hacking. It is a powerful tool and in case, not used properly, it may lead to losses even.

1.3 Project Category (Internet based, Application or System Development, Research based, Industry Automation, Network or System Administration)

Our project category is Internet based VAAST.

1.4 Kali Linux Tools

Kali Linux is a Linux based operating system, mostly used in penetration testing. Kali.org has recently released its new update with some extra functionalities. There are different types of tools that are present in Kali Linux to perform different operations.

1. **Information Gathering**
   These software or applications have a job of collecting and formatting the data in a form that could further be used. This is similar to cookies used by different websites or your browsing history used by Google to personalize every advertisement and providing the best services to you. Kali operating system provides these tools to the developer and penetration testing community to help in gathering and formulating captured data.
   Some of the tools are:
   • Nmap
   • Zenmap
   • Stealth scan
   Nmap is the most famous in these tools. Go to "Applications" then in "Information Gathering", you will find these tools.

2. **VulnerabilityAnalysis:**
   Vulnerability is a state or condition of being exposed to the possibility of being attacked or harmed in one or the other way. These tools are used to check a system or machine for any kind of flow and vulnerability available in them, which could lead to any security breach and data loss. These tools also help in fixing those vulnerability as identification make the user aware of the flow. For example: If windows release its new operating system, before providing it into the end-user they send for vulnerability analysis and fixes.
   Some of the tools:
   • Bed
   • Ohrwurm
   • Powerfuzzer
   • Sfuzz
   • Siparmyknife
   All these tools are very common in the community. Go to "Applications" then in "Vulnerability Analysis", you will find these tools.

3. **Web Application Analysis:**
   Web Application is a dynamic response web page that helps in a better and interactive client-server relationship. These tools identify and access websites through the browser to check any bug or loophole present, which could lead any information or data to lose. For example, there is a website with a payment gateway then these web analyzers check if sufficient authentication and authorization present of the site. These web application uses:
   • SQL injections
   • Denial of service
   • URL manipulation
   Some of the tools are:
   • Burpsuite
   • Sqlmap
   • Webscarab
   • Wpscan
   Burpsuite, vega, and web scarab are some most famous tools. Go to "Applications" then in "Web Application Analysis", you will find these tools.

4. **Database Assessment:**
   These applications are made to access the database and analyze it for different attack and security issues. These assessment shows some opportunities for improvement and changes. They develop a report of the analysis done on the database system. They perform:
   • Configuration checking

• Examining user account
• Privilege and role grants
• Authorization control
• Key management
• Data encryption

Some of the tools are:

• Bbqsl
• Jsql injection
• Oscanner
• Sqlmap
• Sqlninja

Sqlmap is the most famous database assessment tool. This tool injects SQL injection for scanning, detecting, and exploitation. Go to "Applications" then in "Database Assessment", you will find these tools.

5. **Password Attacks:**
   These are basically a collection of tools that could handle the wordlist or password list to be checked on any login credentials through different services and protocols. Some tools are wordlist collectors and some of them are the attacker. Some of the tools are:
   • Crewl
   • Crunh
   • Hashat
   • John
   • Johnny
   • Medusa

   John the Ripper and Medusa are the most famous tools. Go to "Applications" then in "Password Attacks", you will find these tools.

6. **Wireless Attacks:**
   These tools are wireless security crackers, like breaking wifi – routers, working and manipulating access points. Wireless attacks are not limited to password cracking these are also used in information gathering and knowing behavior of victims over the internet. For example, the Victim is connected to a compromised access point or a fake access point then it can be used as a Man-in-The-Middle attack. Some of the tools are:
   • Aircrack-ng
   • Fern- wifi –cracker
   • Kismet
   • Ghost Phisher

   Aircrack-ng and Ghost Phisher are the most famous tools. Go to "Applications" then in "Wireless Attacks", you will find these tools.

7. **Reverse Engineering:**
   Reverse Engineering is to break down the layers of the applications or software. This is used in creating cracks and patches for different software and services. These tools reach the source code of the application, understand its working and manipulate according to needs. For example, Reverse engineering tools are also used by High-End companies to know the logic and idea behind the software. Some of the tools are:
   • Apktools
   • Ollydbg
   • Flasm

   Most famous tools are ollydbg and apltools. Go to "Application" then in "Reverse Engineer-

ing", you will find these tools.

8. **Exploitation Tools:**
   These tools are used to exploit different systems like personal computers and mobile phones. These tools can generate payloads for the vulnerable system and through those payloads information from the devices can be exploited. For example, the Victim's system is compromised using payloads over internet or installing it if physically accessible. Some of the tools are:
   • Armitage
   • Metasploit
   • Searchsploit
   • Beef xss framework
   • Termineter
   The most famous tool is Metasploit (there are courses to learn Metasploit alone). Go to "Applications" then in "Exploitation Tools", you will find these tools.

9. **Sniffing and Spoofing:**
   Secretly accessing any unauthorized data over network is sniffing. Hiding real identity and creating fake identity and use it for any illegal or unauthorized work is spoofing. IP spoofing and MAC spoofing are two famous and mostly used attacks. Some of the tools are:
   • Wireshark
   • Bettercap
   • Hamster
   • Driftnet
   The most used tool is Wireshark. Go to "Applications" then in "Sniffing and Spoofing", you will find these tools.

10. **Post Exploitation:**
    These tools use back doors to get back to the vulnerable system i.e. to maintain access to the machine. As the name suggests these are useful or mostly used after an attack has previously been made on the victim's machine. For example, After an attack victim removed the vulnerability from the system, in this situation if attacker wants to access data again, then these tools are helpful. Some of the tools are:
    • MSF
    • Veil –Pillage framework
    • Powersploit
    The most famous tool is Powersploit. Go to "Applications" then in "Post Exploitation Tools", you will find these tools.

11. **Forensics:**
    These tools are used by forensic specialist to recover information from any system or storage devices. This helps in collecting information during evidence searching for any cybercrime. Some of the tools are:
    • Autopsy
    • Binwalk
    • Hashdeep
    • Volafox
    • Volatility
    The most famous tool is Autopsy, it has also been used by security forces, many judicial and investigating officials. Go to "Applications" then in "Forensics", you will find these tools.

12. **Reporting Tools:**
    After all the assessment and vulnerability testing analysts have to report all those to the client in an organised and authenticated way. These tools develop statistics and information to help in analysing. Some of the tools are:
    • Dradis
    • Faraday IDE
    • Pipal
    • Magictree
    Most famous tools are faraday, Dradis, and Pipal. Go to "Applications" then in "Reporting Tools", you will find these tools.

13. **Social Engineering:**
    As the name suggests these tools generate similar services that people use in daily life and extract personal information using those fake services. These tools use and manipulate human behavior for information gathering. For example, Phishing is one of the example of social engineering, in this, a similar looking home page of any social platform is created and then login details are compromised. Some of the tools are:
    • SET
    • Backdoor-f
    • U3-pwn
    • Ghost Phisher
    The most famous social engineering tool is SET. Go to "Applications" then in "Social Engineering Tools", you will find these tools.

1.5 Bash Script

Bash is a command-line interpreter or Unix Shell and it is widely used in GNU/Linux Operating System. It is written by Brian Jhan Fox. It is used as a default login shell for most Linux distributions. Scripting is used to automate the execution of the tasks so that humans do not need to perform them individually. Bash scripting is a great way to automate different types of tasks in a system. Developers can avoid doing repetitive tasks using bash scripting. Bash scripting supports variables, conditional statements, and loops just like programming languages.

Different users can be configured to use different shells. But most users prefer to stick with the current default shell. The default shell for many Linux distros is the GNU Bourne-Again Shell (bash). Bash is succeeded by Bourne shell (sh).

when you first launch the shell. it uses a startup script located in the .bash or file which allows you to customize the behaviour of the shell.
when a shell is used interactively, it displays a dollar sign when it is waiting for a command from the user. This is called the shell prompt.
([username@host])dollar sign
If shell is running as root, the prompt is changed to . The superuser shell prompt looks like this:
([root@host])
Bash is very powerful as it can simplify certain operations that are hard to accomplish efficiently with a GUI. Remember that most servers do not have a GUI, and it is best to learn to use the powers of a command line interface (CLI).

1. **Applications of Bash Scripts**
    • Manipulating files
    • Executing routine tasks like Backup operation

- Automation

2. **Advantages of Bash Scripts:**
   - It is simple.
   - It helps to avoid doing repetitive tasks.
   - Easy to use
   - Frequently performed tasks can be automated
   - A sequence of commands can be run as a single command.

3. **Disadvantages of Bash Scripts:**
   - Any mistake while writing can be costly.
   - A new process launched for almost every shell command executed.
   - Slow execution speed.
   - Compatibility problems between different platforms.

1.6 Objectives

1. To automate software development tasks.

2. To manage research systems.

3. To ensure the privacy of information, the correctness of data, and access to authorized users.

1.7 Problem Formulation

A voice is a tool that transports us into the future. A future that has more possibilities and more solutions. A voice is a tool that can be used for standing up for what is right, rather than what is easy. A voice gives your opinions a platform, and gifts you with the opportunity to have perspective and knowledge on things that matter. No two voices are the same, each voice has something different to say. And in a world that needs to represent freedom and democracy, a voice is a powerful symbol of this. It is what has allowed people to protest injustice, to sing for freedom, or simply speak the truth. A voice can be a source of hope in difficult times. So we prepare this model, to make the information gathering tools more simple and user friendly.

1.8 Identification/Reorganization of Need

To use the site, calculator or kali linux tools, i am using bash programming language because it is most widely used language in cyber security.

1.9 Existing System

In this, we are using various tools of kali linux and these all tools i came up with one platform where it is easily usable and i also came up with the idea of adding calculator and various types of sites.

2 Requirement Analysis and System Specification

2.1 Feasibility study (Technical, Economical, Operational)

1. **Economical:** Images, code and algorithms for training the model follow all the free libraries available. No cost will be charged in developing and deploying this project. All the coding part will also be done on freely available software and editors. The cost of hardware and software will be zero, and it can be performed on any operating system. Therefore, from these details, we can say this project is economically feasible.

2. **Technical:** The tools will be obtained from browser or any other software. Framework will be used for. The Data Integration aspect of the data manipulation will be taken care of with the use of data frame such as kali linux. The primary language used to code out the exploration is bash. Thus, this project is also technically feasible.

3. **Operational:** Operational feasibility is dependent on human resources available for the project and involves projecting whether the system will be used if it is developed and implemented. This project can be deployed on any specific domain after allotment.

2.2 Software Requirement Specification Document

1. **Data Requirement-** Data required for this project is available online on a site. Data is required for this project for training of the model. For training the dataset was divided into training and testing dataset in the ratio of 80-20.

2. **Functional Requirements-** Functional requirements in an SRS document (software requirements specification) indicate what a software system must do and how it must function; they are product features that focus on user needs. Our model will function as it is a encryption decryption techniques to hide confidential data.

3. **Performance Requirement-** System performance is the most important quality in non functional requirements and affects almost all the other preceding ones. Our model can produce the output of the data of image and about the private key in just a few milliseconds and helps to give suggestion.

4. **Dependability Requirement-** Dependability is the probability and percentage of the software performing without failure for a specific number of uses or amount of time. This feature defines the amount of time the system is running, the time it takes to repair a fault, and the time between lapses. Users can get their output of the desired one.

5. **Maintainability requirement-** Maintainability defines the time required for a solution or its component to be fixed, changed to increase performance or other qualities, or adapted to a changing environment. If our model's prediction become unavailable, they can be under maintenance for approximately one or two hours. This feature is defined as the ability to control a system efficiently and keep it fully operational.

6. **Security requirement-** Security measures ensure your software's safety againstespionage or sabotage. These features are necessary even for stand-alone systems; you don't want anyone to have access to your sensitive data. As we had deployed on vscode, and anyone can access it but no one can change the backend part so it is fully secured.

7. **Look and feel requirement-** No external requirements will affect our model. It will just run on the backend irrespective of the environment conditions. All system components must follow a common and standard set of exchange formats to exchange data; the lack of interoperability happens when people do not follow standards. Feel indicates how effectively they can learn and use a system.

2.3 Validation

This model will validate the art of concealing the fact that connection is taking place, by hiding data in other information. Some different carrier file structure can be used, but digital images are the important because of their frequency on the computer network.

2.4 Expected hurdles

While making this model we came across many hurdles, some of them are listed below: First, it was difficult to put the correct technique like there are different types of tools for example- redhawk, usserrrecon, nmap, bug bounty and many more. On this basis, this was quite difficult to implement this, in our code.

2.5 SDLC model to be used

We used a waterfall model in making this project.

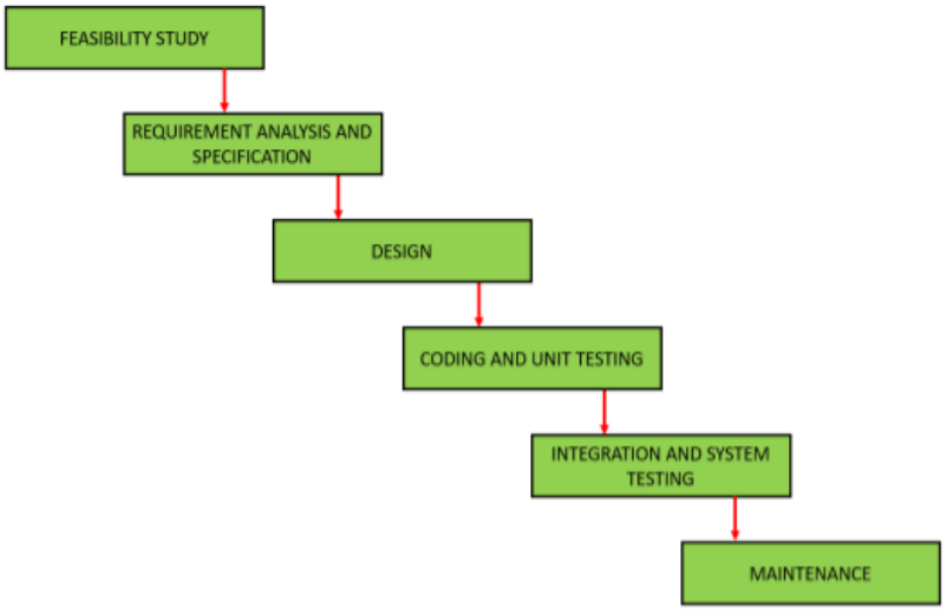

Fig 2.1 waterfall model

..

1. **Requirement Gathering and analysis -** In this phase we have collected resarch work, code, where to implement.

2. **System Design -** We have been using different algorithm and code for the design of the system and then compared all the algorithm based . Implementation We implemented the model using like vscode, online editor, anaconda, android studio.

3. **Integration and Testing -** After that we have tested our model on vscode . Deployment of system For the deployment phase we have integrated on vscode, with image, secrect message, and key.

4. **Maintenance -** The maintenance part is also easy as it is easy to edit code part because we had used proper comments to tell what we want to do in that part of the code section.

3 System Design

3.1 Kali inside VMware (Guest VM)

1. Open chrome and visit to vmare.com. Then, click on workplace on the top of tab.Then, option will appear on the right of tab in which select, the “Workstation Player”.

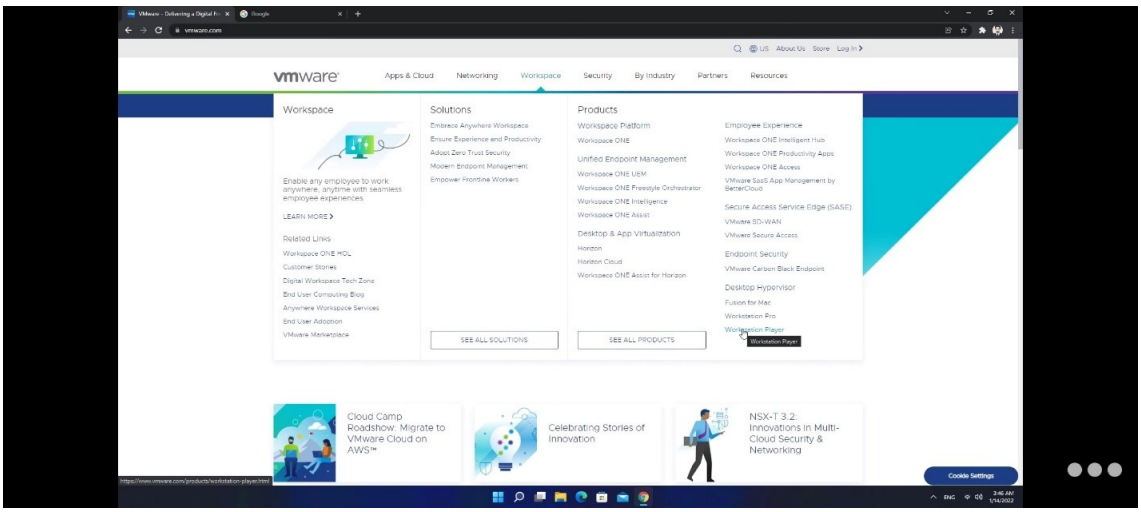

Fig 3.1 Vmware installation

2. After,select the Workstation Player. A new tab will appear in which select the “Download for free” option.

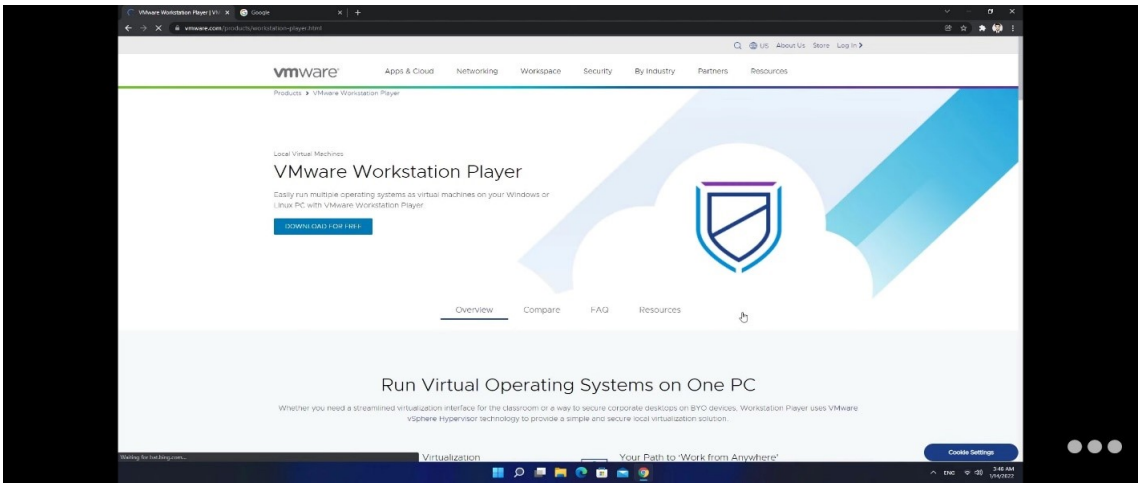

Fig 3.2 Installation process

3.Then, select “Go to Download” option.

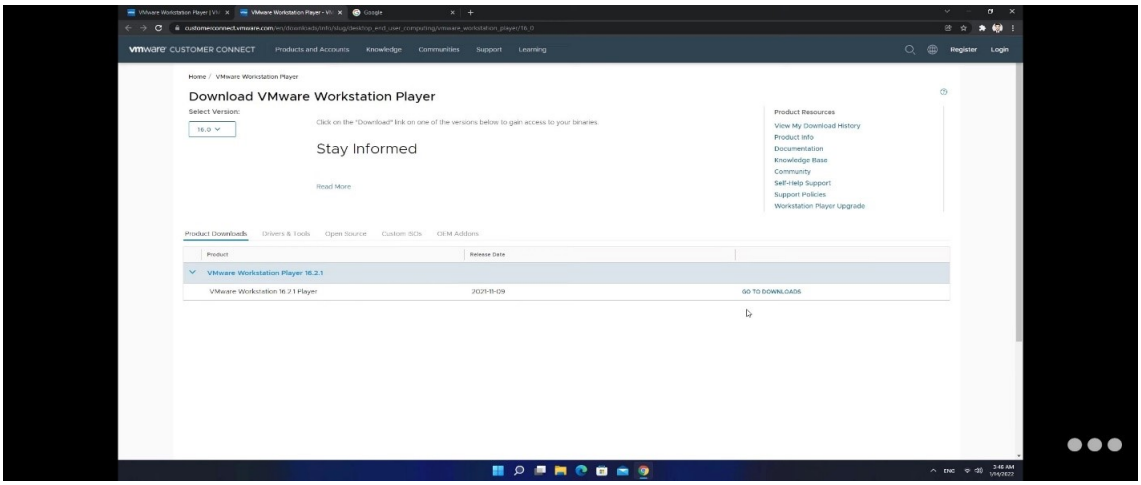

Fig 3.3 Installation process

4. In next tab, there will be two option regarding the OS you are using like window 64 bit and Linux 64 bit Select acc. to your choice and wait until the download is complete.

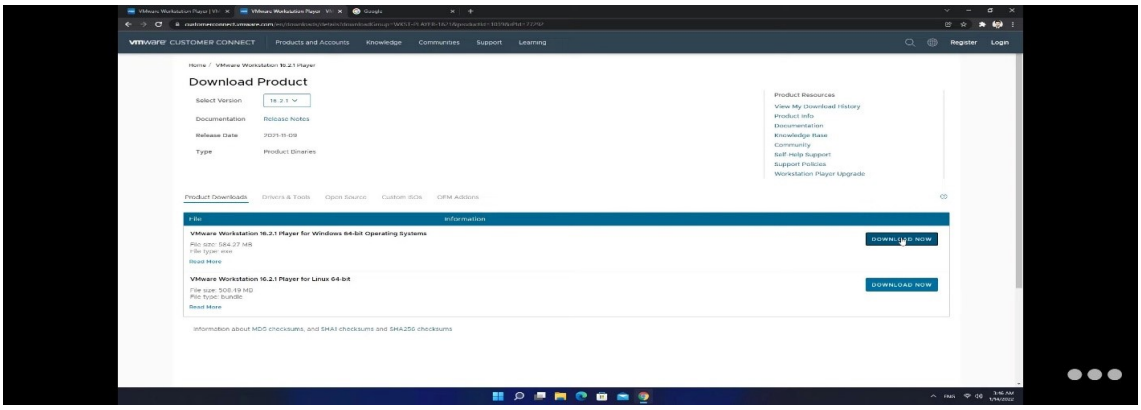

Fig 3.4 Installation process

5.After, the downloading is complete. Then, right click on it and select, as “run as administrator” option. So, that it will run as a administrator.

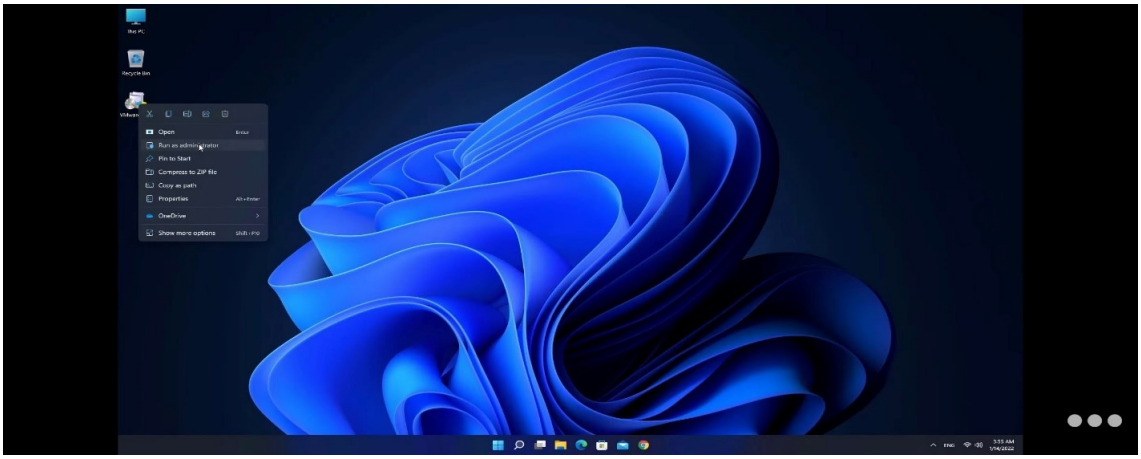

Fig 3.5 Installation process

6.Complete the installation process and wait till the VMware is installed.

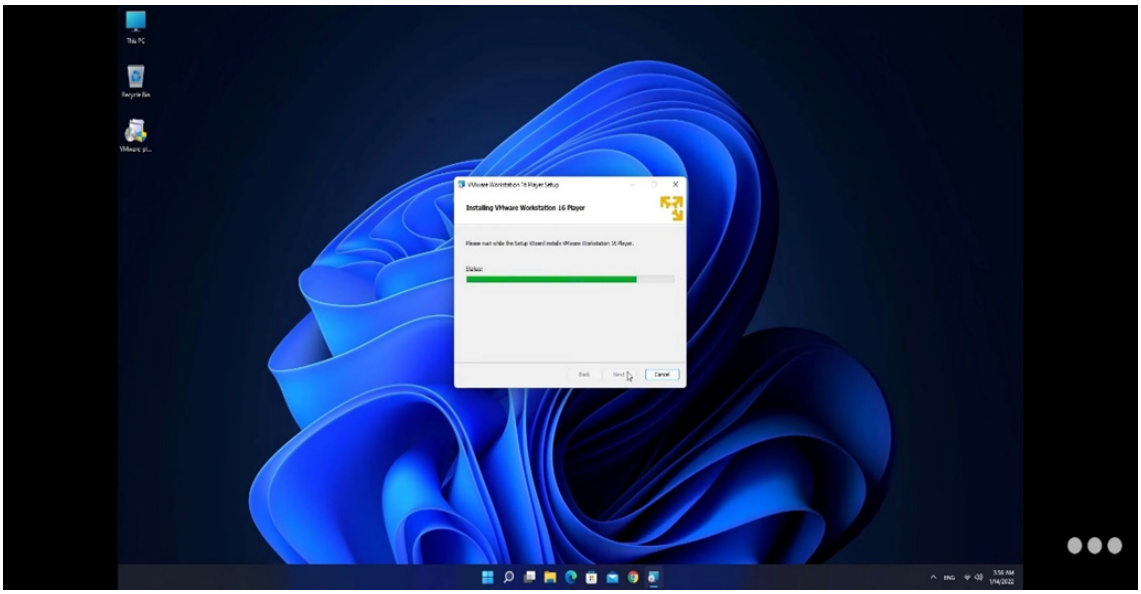

Fig 3.6 Installation process

7. Open chrome and visit “Kali.org”. Then, you will visit the official page of kali as shown below in the fig. .Over there, select the “Download” option.

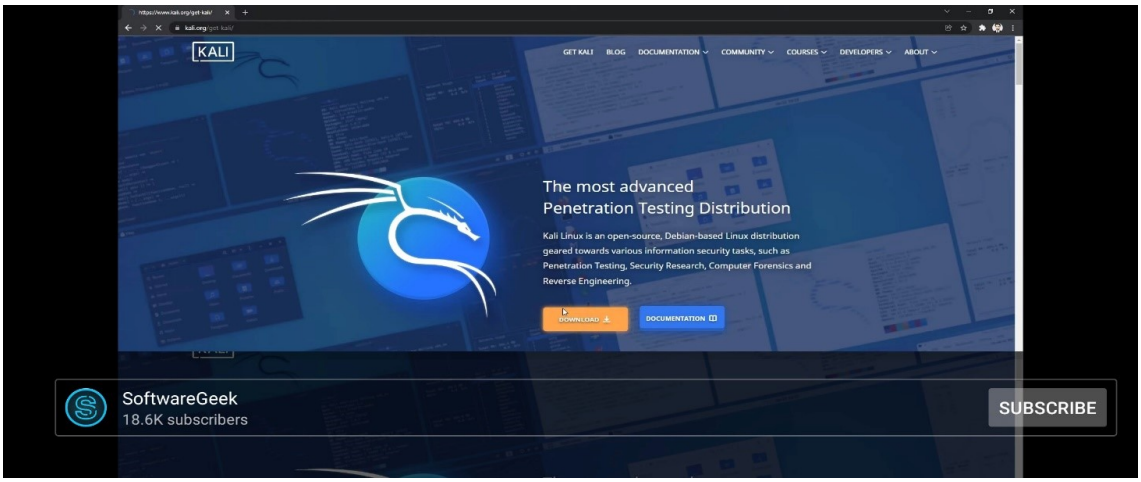

Fig 3.7 Kali Linux installed

8.After, selecting the “Download” option. Different options are shown like the device you want to use kali for. In which select the virtual machine option. Then different images are shown of kali that can be installed in virtual machine. Among all of them, select the “VMware 64-bit”.

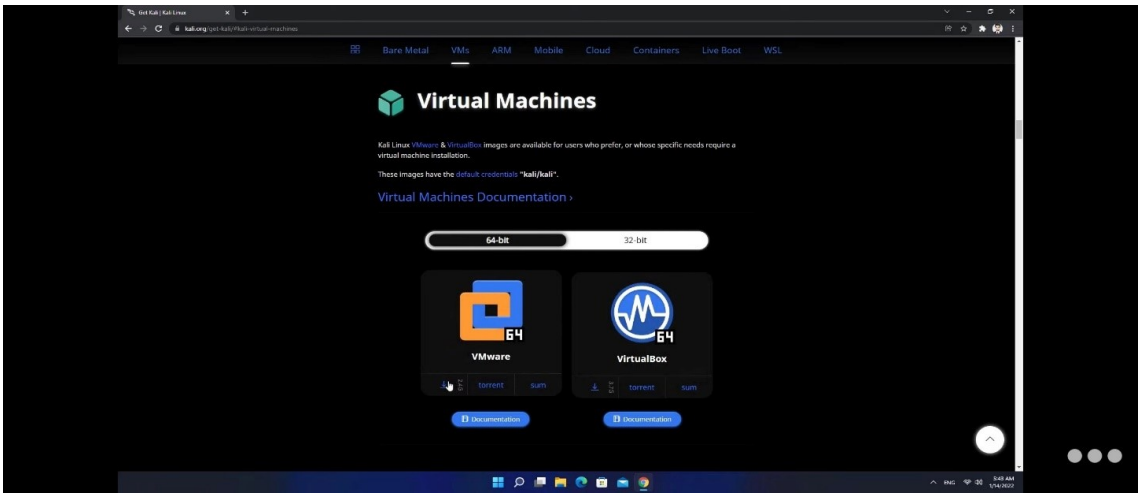

Fig 3.8 Kali linux in Vmware

9.After selecting , the “VMware 64-bit” option downloading starts. Wait until the downloading process is not completed and then, open the zip file. Select the “Extract to” option and select the location where you want to extract the file.

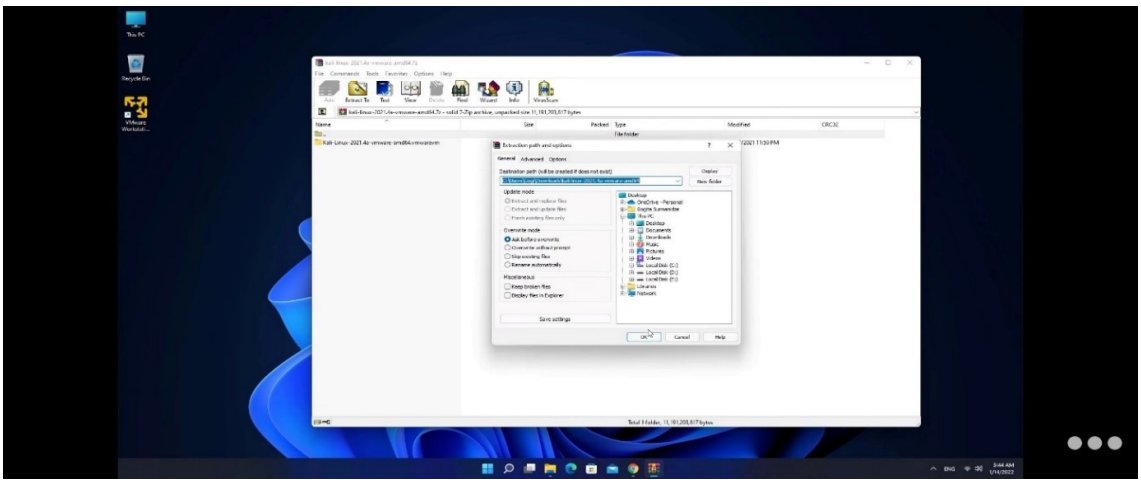

Fig 3.9 Process

10.Then, the extraction process will be started. It will take some time just wait for its completion.

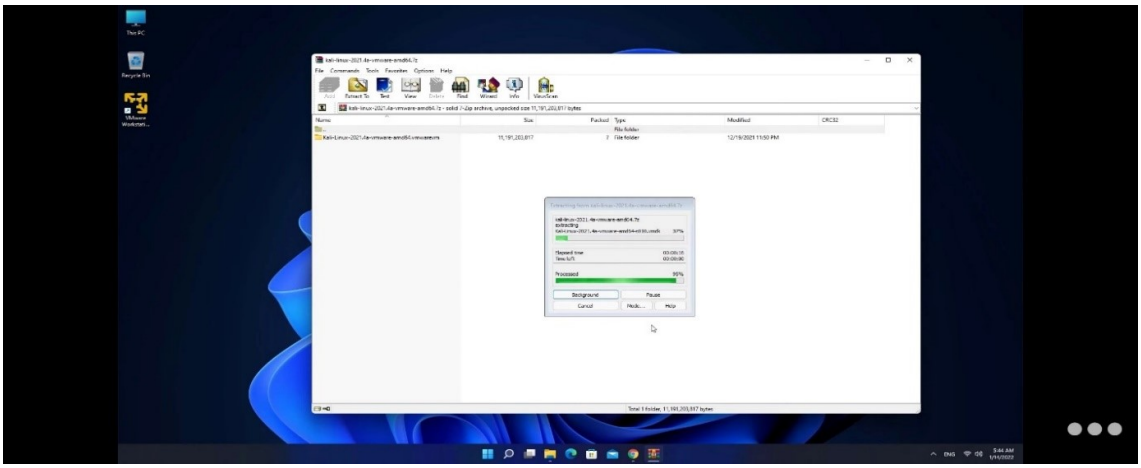

Fig 3.10 Installation process

11. Open the extracted file and select the 4th file having type “VM ware Virtual Machine” of size 4KB and run it with VMware workstation.

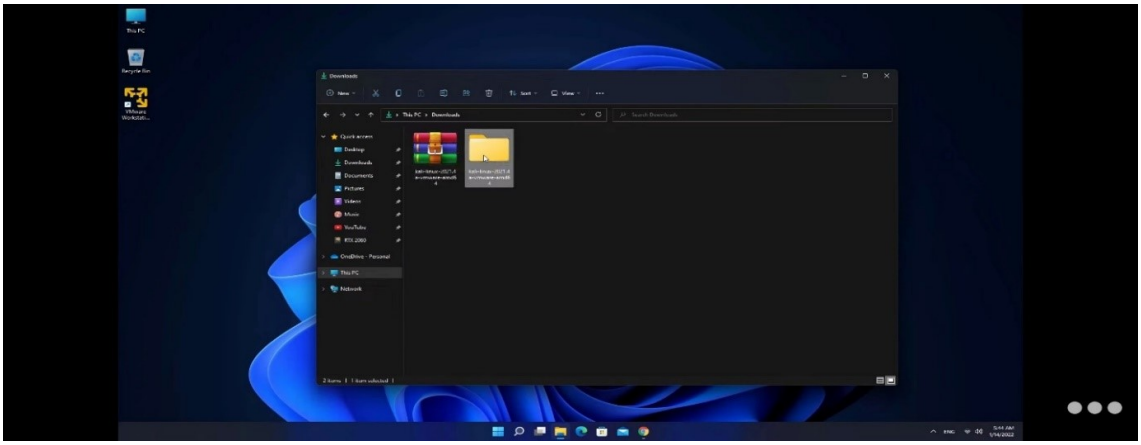

Fig 3.11 Installation process

12.. Over next tab, select “I copied It”. Kali will be going to start in VMware.

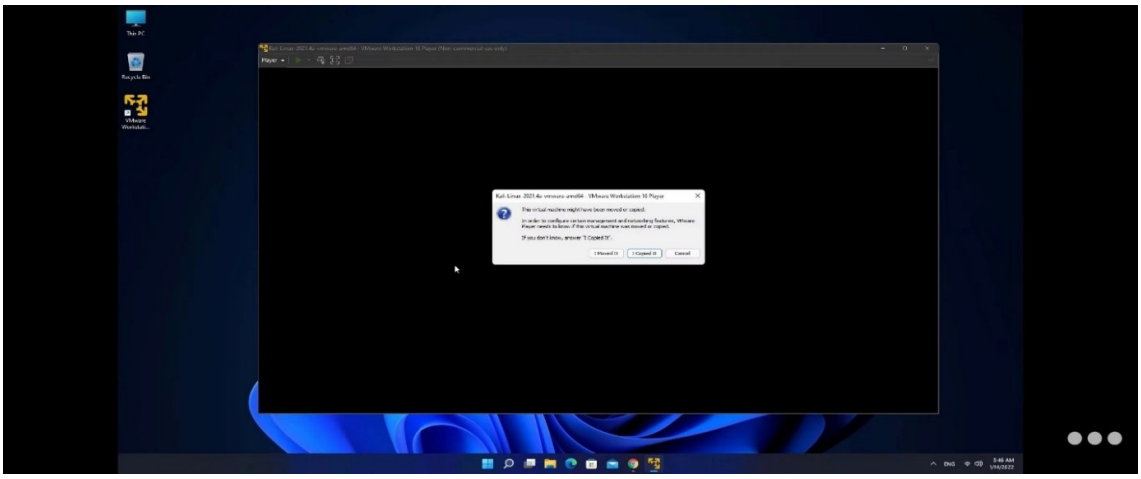

Fig 3.12 Installation process

13.. Kali will open in workstation and ask for user and password.By default, both are “Kali”. Enter user and password and work over kali.

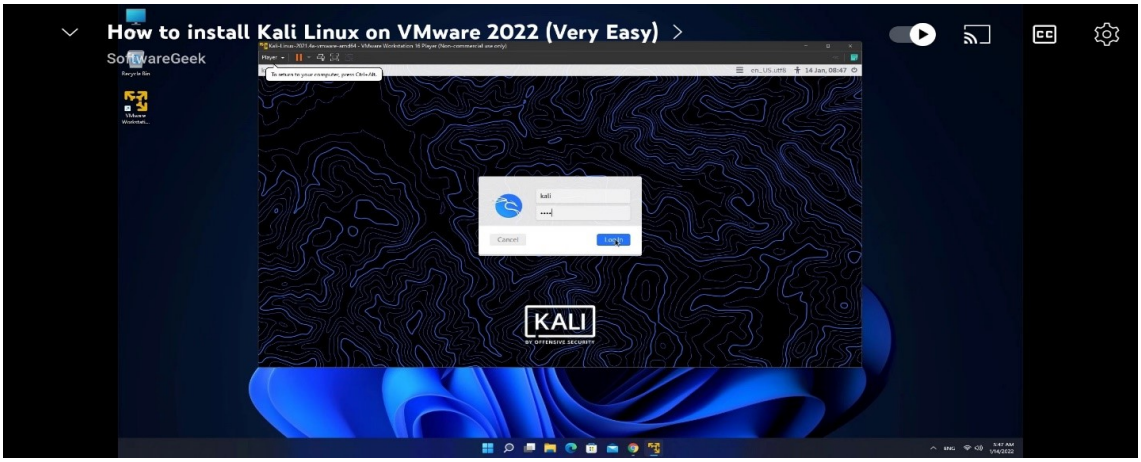

Fig 3.13 Entering password and username

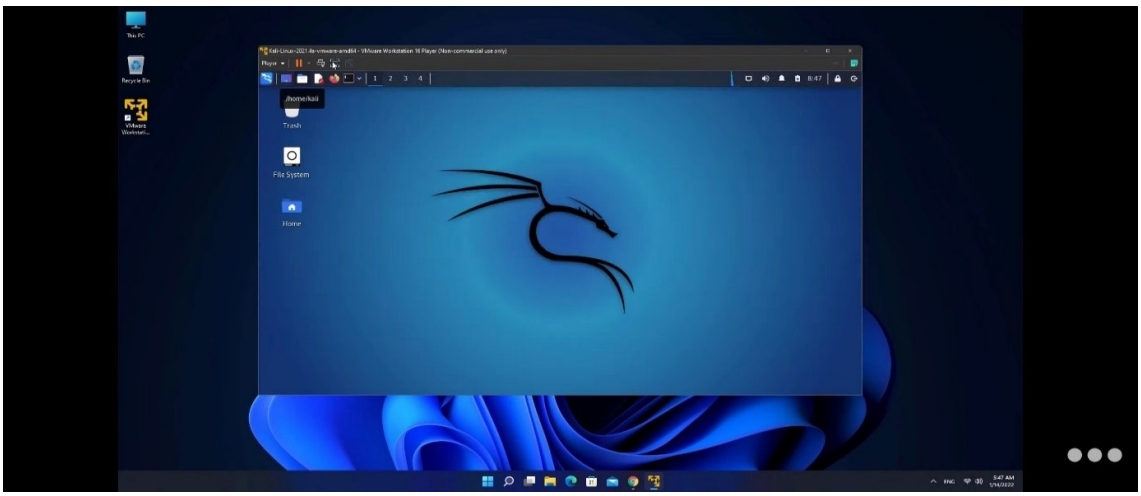

Fig 3.14 Kali linux opening

3.2 Introduction

A voice is a tool that transports us into the future. A future that has more possibilities and more solutions. A voice is a tool that can be used for standing up for what is right, rather than what is easy. A voice gives your opinions a platform, and gifts you with the opportunity to have perspective and knowledge on things that matter. No two voices are the same, each voice has something different to say. And in a world that needs to represent freedom and democracy, a voice is a powerful symbol of this. It is what has allowed people to protest injustice, to sing for freedom, or simply speak the truth. A voice can be a source of hope in difficult times.

1. **Security**
   In today's world the lidfe is running very fast and in this hactic schedule it is common to misplace or forget our things sometime. Then, it is common that our item is ge in other hands. Afterthat, it totally depends on the person to return to us or not. Even in friends, it is common to share the things but some are private so we need to secure them by add some security measures. Username and password are commonly there but, those are cracked by brute force attack in which we use hit and trial method. So we need something new and more secure something that consume the time so that it will be hactic to crack the security. This tool provide such protection. Like the username and password are verified to enter the system and if any one of the input is mismatched then the system will be automatically shut down.If you enter the correct username and password only then, system will proceed ahead.

2. **Talking Calculator.**
   In today's world the lidfe is running very fast and in this hactic schedule it is common to misplace or forget our things sometime. Then, it is common that our item is ge in other hands. Afterthat, it totally depends on the person to return to us or not. Even in friends, it is common to share the things but some are private so we need to secure them by add some security measures. Username and password are commonly there but, those are cracked by brute force attack in which we use hit and trial method. So we need something new and more secure something that consume the time so that it will be hactic to crack the security. This tool provide such protection. Like the username and password are verified to enter the system and if any one of the input is mismatched then the system will be automatically shut down.If you enter the correct username and password only then, system will proceed ahead.

3. **Shortcut to sites**
   In a day to day life there are some tasks that are daily repeatative. That we perform daily and are time consuming to open again and agin everyday. Like we have to open any site and then login on it and perform some task on the daily basis then, it will be very annoying. In this tool there are some of the defined sites just by entering the number you will automatically shift to the sign in page of the app or we can change the setting so that we can visit to some more sites or enter inside more complex and time consuming links. Just with the single click you now open the sites.

4. **Tools**
   The most important thing in the cyber security and ethical hacking is the tools that help us to perform the tasks according to our choice. There are different types of tools that perform the phishing attacks and information gathering. Some of them are installed and linked to the tool. Now you can practice some of them on a single platform with a single click. This tool is also link with some of the other tool that are for the enjoyment purpose and also it is linked with the site named open bug bounty where you can perform the bug bounty.

Steps-

1. Firstly, the tool welcome you and take you inside it in an interesting way.To know the way you have to be patience till you get a chance to enter the tool.

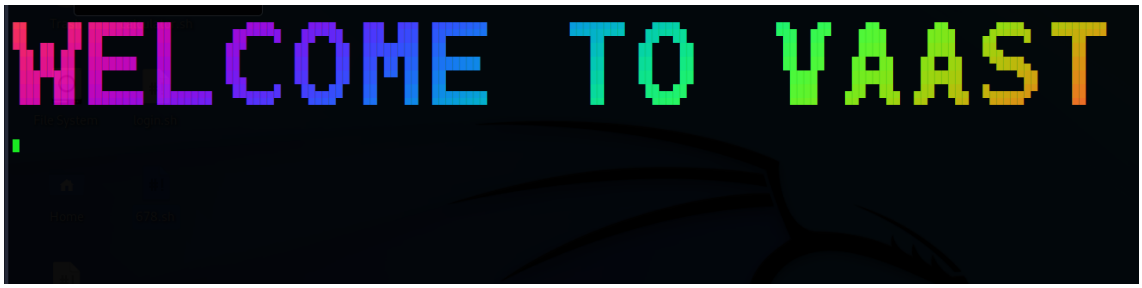

Fig 3.15 Model page

2. When you enter inside the tool the first tab you see the security page, the page will ask for the user name and the passward to move ahead. If you know the key to enter the tool then, you are welcomed otherwise you have to move out of the tool whether you want or not.

Fig 3.16 Enter password and username

3. Here is the options of different types that you want to select.

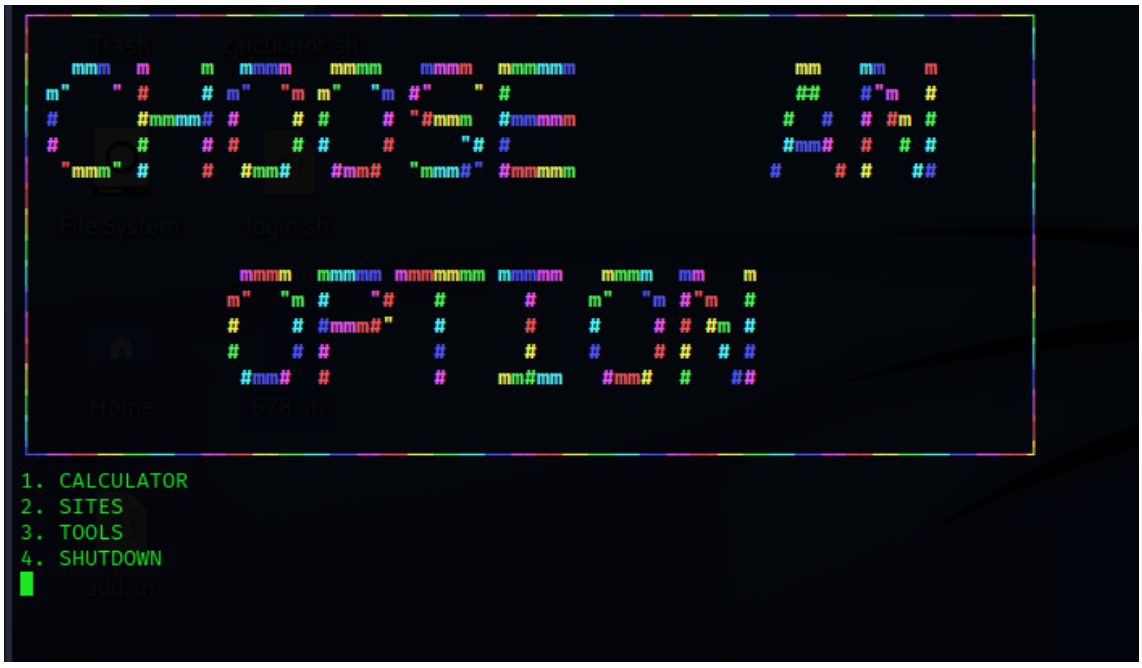

Fig 3.17 Available options

4. If you enter incorrect password, then your system automatically shutdown, here is the picture.

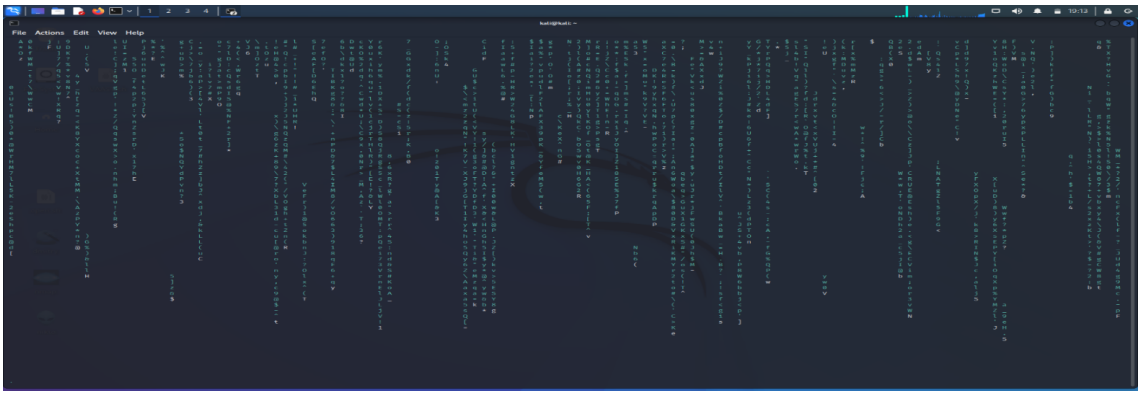

Fig 3.18 Shutting down

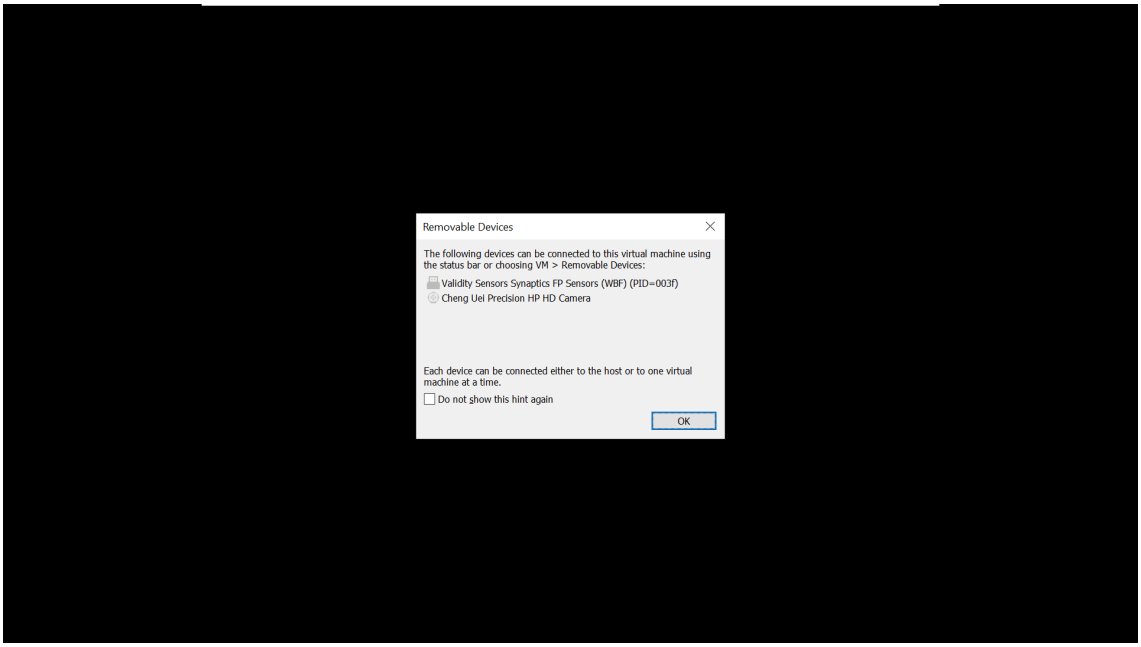

Fig 3.19 Shut down

4 Implementation

Implementation is one of the most important tasks in project implementation is the phase, in which one has to be cautious, because all the efforts undertaken during the project will be fruitful only if the software is properly implemented according to the plans made. The implementation phase is less creative than system design. It is primarily concerned with user-training, site preparation and file-sites, the tests of the network along with the systems are also included under implementation. Depending on the nature of the systems extensive user training may be required. Programming is itself a design works. The initial parameters of the management information systems should be modified as a result of programming efforts. Programming provides a real test for the assumption made by the analyst. System testing checks the readiness and accuracy of the system to access, update and retrieve data from the new files. Once the programs become available the test data are read into the computer and processed. In most conventions parallel run was conducted to establish the efficiency of the system. Implementation is used here to mean the process of converting a new or revised system in to an operational one. Conversion is one aspect of one implementation.

5 Testing

Testing is the process of evaluating a system by manual or automatic means to verify that it satisfies the specified requirements or to identify differences between the actual and expected results. During system testing, the system is used experimentally to ensure that the software does not fail. special test data are input for processing and the results are examined. If the program fails to behave as expected, then the conditions under which such a failure occurs are noted for debugging and correction. program testing represents the logical elements of the system. The testing ultimately leads to suffice the quality factors such as correctness, reliability, efficiency, usability, maintainability, portability,accuracy, error tolerance and expand ability.

5.1 Different Types of Testing

- Test Strategies -
  For testing software, various test strategies can be used such as Unit Testing Integration Testing, Black-box testing, White-box testing, regression testing, and acceptance testing etc.
- Unit Testing -
  Unit test focuses verification effort on the smallest unit of software design, the software components or modules. By testing in this method we should be very clear of the bugs occured. The module interfaces is tested to ensure that information property flows into and out of the program under test.
- Integration Testing -
  Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dedicated by design.
- Black-box Testing:
  Black box testing alludes to test that are conducted at the software interface. Although they are designed to uncover errors, black-box tests are used to demonstrate that software functions are operational, that input is properly accepted and the output is correctly produced. A black-box test examines some fundamental aspect of a system with title regard for the internal logical structure of the software.
- White box testing is predicated on close examination of procedural detail. Logical paths through software are tested by providing test cases that exercise specific set of conditions and loops.
- Regression Testing:
  Regression testing involves executing old test cases to test that no new errors have been introduced. This testing is performed when some changes are made to an existing system. The modified system needs to be tested to make sure that the new features to be added to work.
- Acceptance Testing:
  Acceptance testing involves planning and execution of functional tests, performance test to verify that implemented system satisfies its requirements. Acceptance tests are typically performed by the quality assurance and customer organization.

5.2 Preconditions

The following items are required before testing can take place:

- A complete and coherent functional specification of the System expressed as use cases and usage scenarios.
- A complete and validation-tested release of the System, delivered according to the delivery plan.
- An agreed-upon procedure for dealing with any anomalies that are discovered during the testing process.
- A set of test specifications describing how each functional area of the System is to be acceptance tested.
- An implemented test environment for the testing Sufficient, suitable resources to carry out the testing.
- Available standards for the acceptance testing.

5.3 Test Priorities

During testing of the System, the following qualities will be tested in order of priority:

- Functionality—whether the required functions are available and working as expected.
- Usability—how user-friendly and intuitive the System is
- Security—how well-protected and guaranteed corporate and user data is
- Performance—whether the response times are within acceptable limits
- Customization—how straightforward it is to use the application in new, unpredicted ways

5.4 Test Organization

5.4.1 Roles And Responsibilities

The following roles are defined:

- QA lead/test manager—responsible for planning and ensuring the smooth running of the test process
- Tester—carries out the tests according to the test plan, and then reports the results
- Product manager—ensures that the tests are carried out successfully from a user perspective
- Project sponsor/client—acts as main stakeholder, and ensures that the needs of the customer community as a whole are considered
- Test support—provides technical assistance, such as test environment configuration, and non-technical assistance, such as methodological support.

Weekly team meetings will be held involving the test manager, testers, and product managers. At these meetings, the progress of the testing process will be reported, any issues will be discussed, and actions will be agreed upon.

5.5 Test Environment

5.5.1 Hardware

The test environment will consist of:
• A laptop or a desktop with as much RAM and processor power you can arrange.
• A large HDD or SSD to store your tools and other important files.
• A host OS for your computer system. It can be Windows, Linux (any family, any flavor) or Mac OS depending on your choice.
• Latest security patches must be installed on your guest OS before you start.
• A WiFi adapter that supports monitor mode. (Optional)

5.5.2 Software

• Virtual Machine Player or Hypervisor: This will be used to host all the guest operating systems, vulnerable virtual machines, and test servers. There are many free and paid options for hypervisors provided by many vendors. For example, VMware has VMWare workstation, Oracle has Oracle VirtualBox and Microsoft has HyperV. You can choose any of these depending on your choice and budget.

• Guest Operating Systems: Guest operating systems will include unpatched versions of Windows and Linux. These will be installed to test for zero-days and other vulnerabilities for which patches, as well as exploits, have been released.

• Vulnerable VMs: Vulnerable Virtual Machines are developed intentionally for being highly vulnerable. Most of the VMs are parts of hacking events and are released later online. These VMs are usually CTFs with hidden strings that are to be found after compromising (pwning) the VM. Some popular vulnerable VMs are Metasploitable, OWASP broken web application, DVWA(Damn Vulnerable Web Application), BadStore, De-Ice, and Multidae, etc.

5.6 Essential Tools

Once you have found and installed your favorite vulnerable assets, it is now time to get the tools required for pwning them. Install these tools on your computer to get started.

1. Metasploit Framework (MSF): An open-source version of the Metasploit tool is used extensively for exploiting known vulnerabilities in systems and software. The exploit list is updates regularly with exploits of most recent findings that went public.

2. WireShark: It is a tool used by network administrators but you can use it to supplement your hacking tools arsenal. For you as a hacker (ethical, of course) this tool will help in network penetration by the same basic feature of network monitoring: it can help you harvest sensitive data like plaintext passwords over unencrypted connections (http, telnet), analyze malware behavior by figuring out the endpoints it tries to connect, and many more.

3. Nmap: One tool to rule 'em all, it is used by almost every penetration tester. It is a port scanner with a set of additional utilities like OS detection and network mapping (nmap stands for "network mapper"). It can be automated by writing scripts in NSE (nmap scripting environment). Port scans are used to enumerate services and applications on the target. These enumeration data can be really useful in some cases for pwning the target. John The Ripper:

It is a free and open-source password cracking tool which is highly popular among penetration testers. Popularity is the reason why it is available on fifteen platforms. The tools were initially designed for cracking UNIX password hashes. However, the latest stable release from May 2019 supports Windows NTLM, Kerberos and hundreds of other hashes.

4. Burpsuite or OWASP ZAP: Both are great all in one tool for penetration testing web applications. Learning about hacking web applications is crucial for an aspiring (ethical) hacker since most of the services are provided online. These two tool-sets contain all the tools you will need for hacking (ethically) into a web application. Kali Linux: It is an operating system developed primarily for white hat hackers and penetration testers. This OS has a wide array of tools for almost every task before, during and after a penetration testing session. It contains all the tools mentioned above (No need for installing them manually).

5.7 Test Management

Tests shall be managed according to the corporate test management standards, which cover:

1. Conduct of tests
2. Reporting of test results
3. Defect tracking and resolution
4. Configuration management of the test environment
5. Configuration control of test deliverable.

6 Critical Evalution

Kali is better than Ubuntu is like thinking a Dentist is better than a General Physician. Its not fair to compare them on a same level. A General Physician is where you would normally go for any health related issues, but you go to the Dentist only to for your dental issues. You can't expect a Dentist to treat your fever!

Similarly, Ubuntu is the General Physician or in this case a General Purpose Operating System. Anybody can use it as their primary OS. But Kali (The Dentist!) is a Special Purpose Operating System. Its main objective is Pen Testing and Network Security Testing. And if you use the OS you will surely find so many things that it can't do that Ubuntu can.

Therefore, Kali is not a first choice to be used as a primary OS. Also if you are interest in Kali is because you are told Hackers use it, let me tell you, its not always true. Hackers will need to master all types of OSes.

So in summary, Ubuntu is your General Purpose OS and Kali is your Special Purpose OS. Choose the one that you think is favorable to you. Also Kali has a smaller software repository than Ubuntu, which means not all the software you may use will be available or supported by Kali.

So i prepare this project to make the tools usuable easily and user friendly. Basically, the model for Vaast is shown on the following figure:

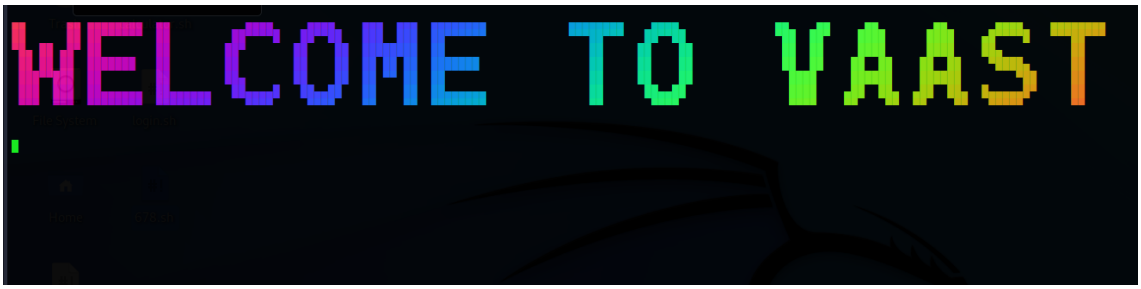

Fig 6.1 Model page

7 Result

(a) Therefore, the result is that, after opening calculator

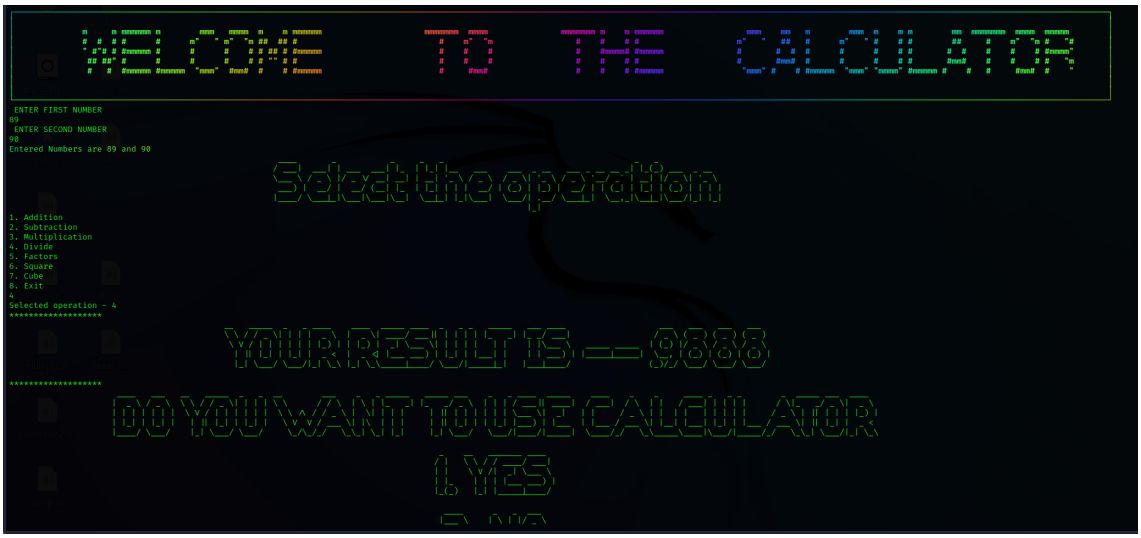

Fig 8.1 Pic 1

(b) By choosing second option that is site and here is the result of that

Fig 8.2 Pic 2

(c) The instagram page is open.

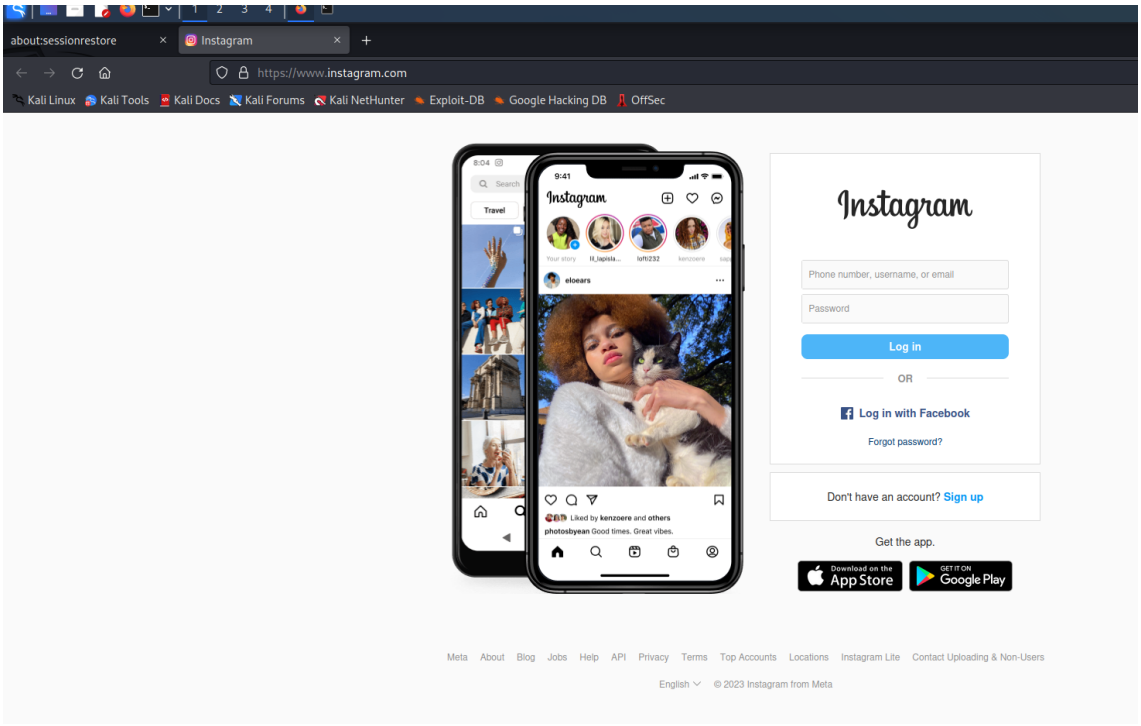

Fig 8.3 Pic 3

(d) Now, as choosing the third option, here is the result.

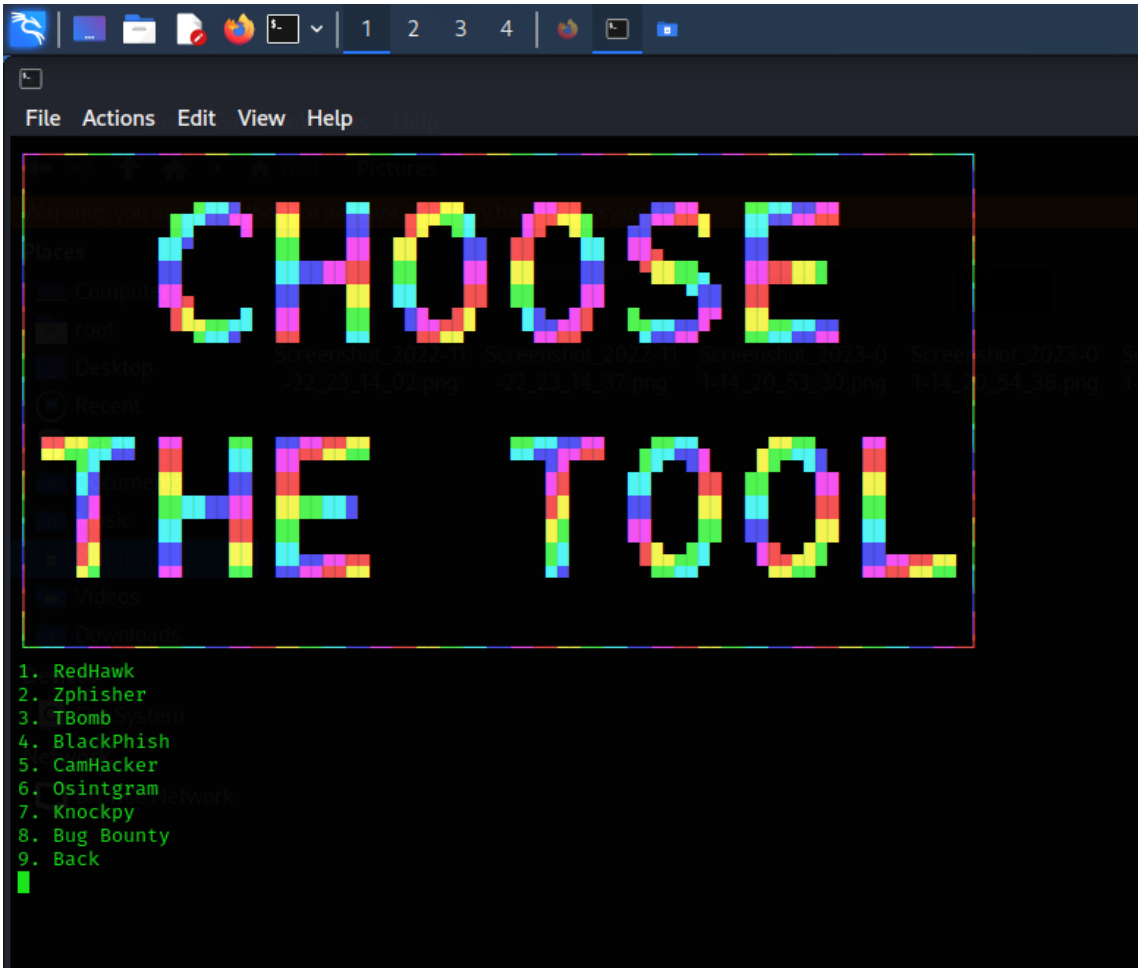

Fig 8.4 Pic 4

(e) Result of selected tool

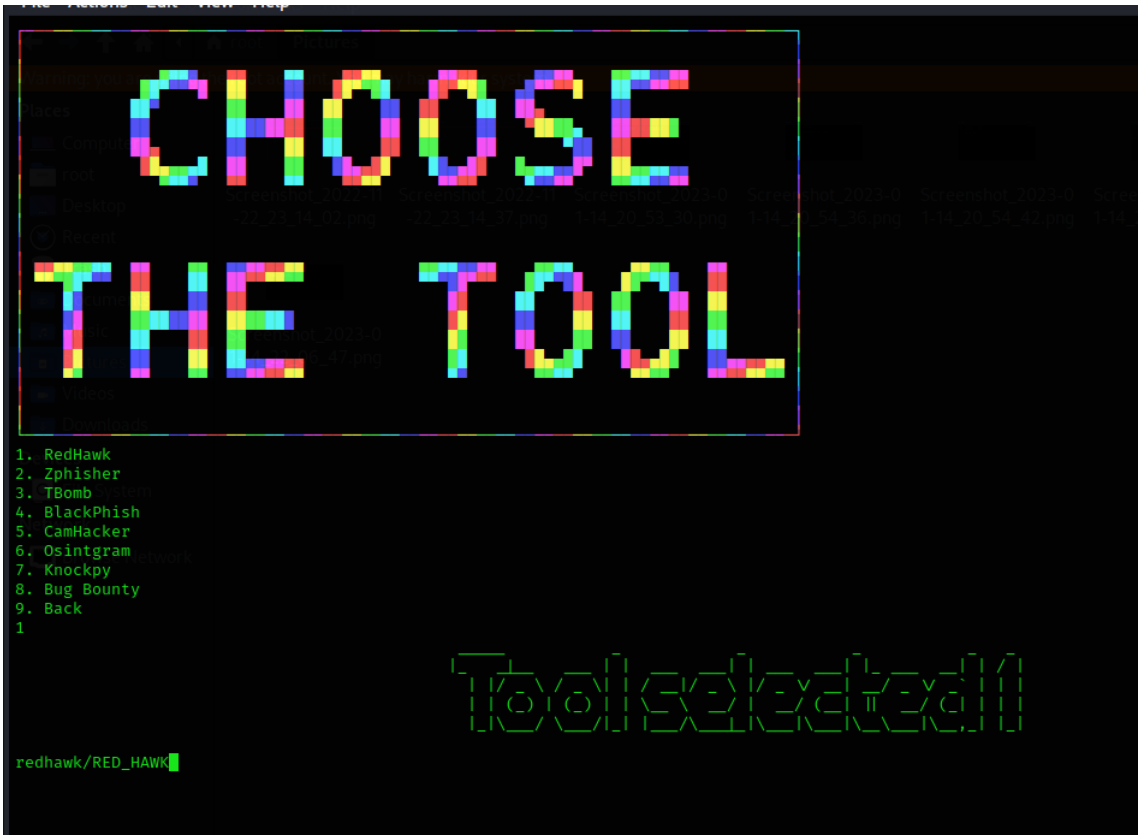

Fig 8.5 Pic 5

(f) Now, the selected tool is open

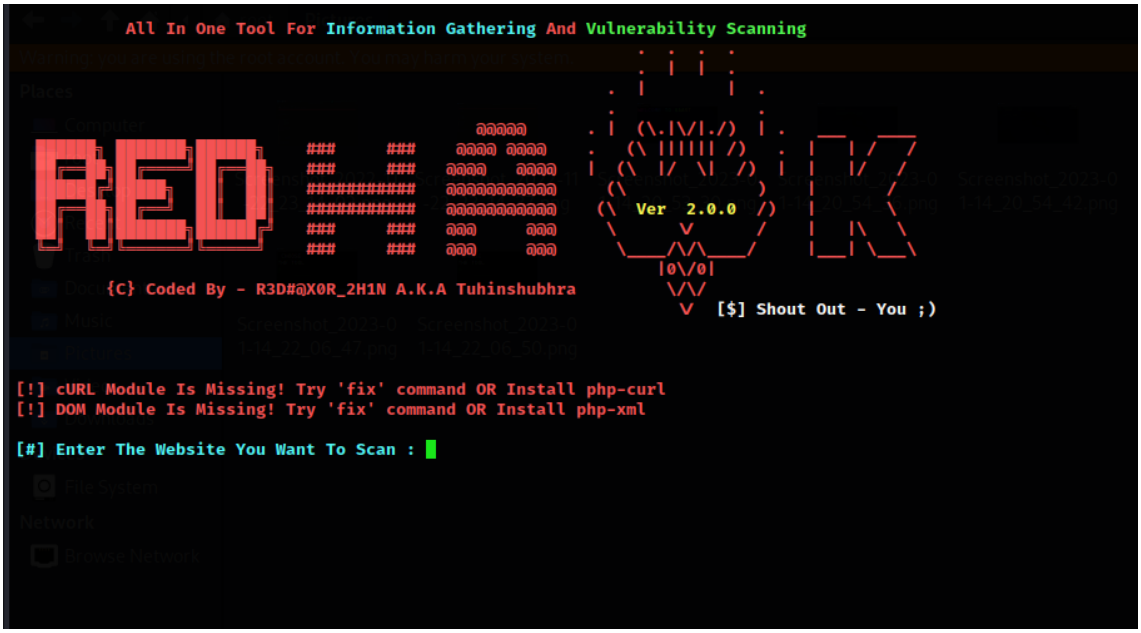

Fig 8.6 Pic 6

8 Future Enhancement And Conclusion

As we know that, shell is a medium provided to interact with the kernel, which is a Command Line Interface (CLI) to the Unix system. It is basically a command-line interface to the Unix system. It works by gathering the input from the users, executing the program based on the input from the users, and then displaying the output after the program's execution.

This is an important concept for DevOps or System administrators as it makes the work easy and reduces time. You can do almost anything and everything by writing a script and save your time; it is quite popular in the world of servers. You can manage multiple systems and servers at a time by using shell scripting. So, this is a great skill to learn.

Five functionalities for the future of the shell:

(a) **Distribution:** in the context of a shell, this means building a system capable of scaling beyond a single machine (for example, inserting compute resources at different stages of a shell command's execution to parallelize).

(b) **Incremental support:**if a shell script is changed slightly, but can reuse previous computation, a shell could strive to do so.

(c) **Heuristic support:** While transforming a shell script into a data flow graph can be facilitated by annotation languages, it would be costly to annotate every shell command. Ideally, the annotation of commands could be performed automatically (or with the support of automation).

(d) **User support:** A shell should take advantage of modern features like language servers. A formal specification for interacting with the shell can theoretically simplify interactions with the shell.

References

i. T.Morkel , J.H.P. Eloff, M.S.Olivier - "An overview of kali linux" - Information and Computer Security Architecture (ICSA) Research Group Department of Computer Science University of Pretoria, 0002, Pretoria, South Africa.

ii. F. L Bauer, "Decrypted Secrets - Methods and Maxims of Cryptology," Berlin, Heidelberg, Germany, Springer-Verlag (1997).

iii. Birgit Pfitzmannn, rInformation Hiding Terminologyr, in Proceedings of FirstWork- shop of Information Hiding, Cambridge, U.K. May 30 - June 1, 1996. Lecture Notes in Computer Science, Vol.1174, pp 347-350. Springer-Verlag (1996).

iv. Gustavus J. Simmons, "The Prisonersr Problem and the Subliminal Channel", in Proceedings of CRYPTO '83, pp 51-67. Plenum Press (1984).

v. Stefan Katzenbeisser and Fabien A. P. Petitcolas, "Information Hiding Techniques for Steganography and Digital Watermarking," Artech House (2000).

vi. Neil F. Johnson and Sushil Jajodia, "Steganalysis of Images Created using Current Software/' in Proceedings of 2nd International Workshop on Information Hiding, April 1998, Portland, Oregon, USA. pp. 273 - 289.

vii. Ross J. Anderson and Fabien A.P. Petitcolas, ron the limits of kali linux,r IEEE Journal on Selected Areas in Communications (J-SAC), Special Issue on Copyright Privacy Protection, vol. 16 no. 4, pp 474-481, May 1998.

viii. Scott Craver, "On Public-key bash in the Presence of an Active Warden," in Proceedings of 2nd International Workshop on Information Hiding, April 1998, Portland, Oregon, USA. pp. 355 - 368.