Journal of the Brazilian Computer Society, 2026, 32:1, doi: 10.5753/jbcs.2026.5550
This work is licensed under a Creative Commons Attribution 4.0 International License.

Web xKaliBurr: An Online Platform for Information Gathering in Pentest for Internet Applications

Daniel R. Barros [**Federal University of Ceará** | ***daniel.rezende@lsbd.ufc.br***]
Lucas Cabral [**Federal University of Ceará** | ***lucas.cabral@lsbd.ufc.br***]
João V. A. Oliveira [**Federal University of Ceará** | ***joao.alves@lsbd.ufc.br***]
Felipe M. Castro [**Federal University of Ceará** | ***felipe.moura@lsbd.ufc.br***]
Lucas L. Soares [**Federal University of Ceará** | ***lucas.lopes@lsbd.ufc.br***]
José M. Monteiro [**Federal University of Ceará** | ***jose.monteiro@lsbd.ufc.br***]
Joaquim Bento [**Federal University of Ceará** | ***joaquim.bento@lsbd.ufc.br***]
Lincoln S. Rocha [**Federal University of Ceará** | ***lincoln@dc.ufc.br***]

Systems and Databases Laboratory (LSBD), Federal University of Ceará, Fortaleza, CE, Brazil.

Received: 14 February 2025 • **Accepted:** 13 July 2025 • **Published:** 15 April 2026

Abstract The Information Gathering stage in web Pentests is crucial as it lays the foundation for all subsequent activities. However, comprehensive information gathering requires the manual use of various tools that demand advanced technical knowledge. In this context, we propose Web xKaliBurr, an open-source web tool that automates the information gathering stage of web Pentest. With a simple and user-friendly interface, the proposed tool performs extensive scans from the site's URL, providing a wide range of information and recommendations, allowing users without advanced knowledge to assess their site's security and detect potential flaws or vulnerabilities. To evaluate Web xKaliBurr, we applied the System Usability Scale (SUS) questionnaire to measure aspects of usability in accordance with the user's subjective assessment and the Net Promoter Score (NPS) method to measure user satisfaction and willingness to recommend it to others. This study involved 10 respondents. The SUS method had a score of 80, which indicates a good to excellent product, and the results of using NPS reached a value of 70%, reflecting a very good level of user satisfaction. Besides, we performed an evaluation with 3 experts in web Pentests.

Keywords: Cybersecurity, Offensive Security, Pentesting, Intelligence Gathering, OSINT Tool.

1 Introduction

Cybersecurity is an area of paramount importance in the contemporary global technological landscape, given the exponential increase in virtual threats and the continuous discovery of vulnerabilities across a wide range of technologies. The growing interconnectivity between systems, fueled by advancements in cloud computing, the Internet of Things (IoT), and artificial intelligence, has expanded the attack surface, making security breaches more frequent and sophisticated [Fadziso *et al*., 2023].

Vulnerabilities are defined as weaknesses or flaws in processes, components, or procedures that can be exploited by malicious actors to gain unauthorized access, manipulate data, disrupt services, or compromise the confidentiality, integrity, and availability of digital systems [Force, 2018]. These security flaws may stem from inadequate software development practices, misconfigurations, outdated protocols, or even social engineering tactics that exploit human error.

Historically, various private companies, government entities, and even public institutions have faced Cyberattacks and catastrophic scenarios due to vulnerabilities and misconfigurations in their digitalized environments.

One of the most notorious cases of Cybersecurity incidents, described by [Mohurle and Patil, 2017], was the WannaCry ransomware attack, which affected Windows 7 and Server 2008 R2 operating systems in May 2017. This attack exploited a critical vulnerability in the Windows operating system, resulting in the encryption of infected machines' files and demanding a ransom in Bitcoin for their release. As a result, the threat quickly spread, compromising hundreds of thousands of computers in over 150 countries, affecting everyone from individual users to large corporations and government institutions. The impact of the attack was devastating, disrupting essential services such as hospitals, telecommunications companies, and transportation systems.

As cyber threats evolve in complexity, organizations must adopt proactive security measures, such as vulnerability assessments, continuous monitoring, and adherence to robust cybersecurity frameworks. Effective risk management strategies and timely software patching play a crucial role in mitigating potential attacks and safeguarding critical infrastructures from cyber incidents. All these activities and responsibilities fall upon the so-called information security professionals [Jones *et al*., 2018].

As highlighted by Walker [2013], information security professionals are responsible for ensuring the protection of computer systems against a wide range of cyberattacks. To achieve this objective, advanced strategies and methodologies are employed, aimed at developing and enhancing additional layers of protection within systems. Among these approaches, Penetration Testing (Pentest) stands out, as it enables the identification and remediation of vulnerabilities before they can be exploited by malicious actors.

The *Pentest* (penetration testing) is a controlled and systematic simulation of cyber-attacks, conducted with the objective of identifying, analyzing, and mitigating security vulnerabilities before they can be exploited by malicious actors [Weidman, 2014]. This process involves the application of specialized techniques and tools, allowing security professionals to assess the resilience of systems against potential threats, such as unauthorized access, data breaches and service disruptions.

Performing penetration tests on web systems is a fundamental practice for ensuring the confidentiality, integrity, and availability of information in online environments. By proactively detecting weaknesses in applications, network configurations, and authentication mechanisms, organizations can strengthen their cybersecurity posture and implement preventive measures to mitigate potential risks effectively [De Jimenez, 2016].

The typical phases of a *Pentest* include Planning, Information Gathering, Threat Modeling, Vulnerability Analysis, Exploitation, Post-Exploitation, and Technical Reporting [Weidman, 2014], as illustrated in Figure 1. Among these phases, the Information Gathering stage is crucial as it involves collecting data about the target, such as domains, IP addresses, technologies used, and potential weak points. This initial phase lays the foundation for all subsequent *Pentest* activities, making the breadth of collected information vital to the success of the assessment [Stuttard and Pinto, 2011].

The Information Gathering phase is facilitated by the availability of open-source tools that allow the collection of data from web applications. However, existing tools present limitations. First, each tool collects specific information, meaning that performing a comprehensive information gathering process requires manually running multiple tools sequentially. Second, most of these tools are executed through command-line interfaces in terminal environments, requiring advanced technical knowledge [Najera-Gutierrez and Ansari, 2018]. Finally, the interpretation of the obtained results and the identification of potential vulnerabilities depend heavily on the information security professional's experience.

Given this context, this work proposes an open-source web tool for automating the Information Gathering phase of *Pentesting* websites, called **Web xKaliBurr**[1], a direct evolution of our previous work [Barros *et al*., 2023]. This tool provides an easy-to-use interface that enables extensive information scanning of websites, requiring only the domain name and communication protocol as input. Additionally, the collected data is presented alongside security recommendations to inform users about potential vulnerabilities. Thus, the tool allows users with no advanced knowledge of cybersecurity to perform an initial assessment of the attack surfaces present in their domains.

In order to evaluate Web xKaliBurr, we applied the System Usability Scale (SUS) questionnaire to measure usability based on users' subjective assessments. The SUS score obtained was 80, indicating a good to excellent level of usability. Additionally, we used the Net Promoter Score (NPS) method to assess user satisfaction and willingness to recommend Web xKaliBurr to others. The NPS reached a value of 70%, reflecting a high level of satisfaction and strong user loyalty. These results demonstrate that Web xKaliBurr provides a valuable and user-friendly experience for its target audience. Finally, we performed a practical evaluation with 3 experts in web Pentests.

The remainder of this work is structured as follows: In session 2, we will introduce some basic concepts and definitions about the resources that were used for the development of this work, providing clarity and a better understanding of the processes addressed during the ideas proposed in the article. In Section 3, we present a review of related works, highlighting tools with functionalities similar to Web xKaliBurr, focused on performing security tests and checks in online environments. In Section 4, we detail the architecture of the proposed tool, its main features, and the execution flow of its processes. In Section 5, we discuss the results obtained after applying the tool in a controlled testing environment, evaluating its effectiveness in identifying vulnerabilities and the proposed mitigation strategies. In Section 6, we introduce the tool in a real-world scenario, where new tests were conducted and the findings resulting from the execution of Web xKaliBurr were analyzed. From this analysis, we generated comprehensive technical reports containing security recommendations specific to the vulnerable environments identified. Finally, in Section 7, we present the conclusions of the work, suggest directions for future research, and outline the next steps for the ongoing development of our proposal.

The main contributions of this work are: (i) the development of an open-source web-based tool to automate the Information Gathering phase in Pentests; (ii) the orchestration of multiple CLI tools in a single automated flow accessible to non-experts; and (iii) the presentation of technical reports with security recommendations, validated in real-world scenarios.

2 Related Works

There are several online platforms, both in academic literature and industry practice, dedicated to cybersecurity analysis, which have significant relevance to the Web xKaliBurr tool. As described by [Tabatabaei and Wells, 2017], the operation of these tools is *based on the OSINT (Open-Source Intelligence) philosophy*, an open-source intelligence methodology that involves collecting, analyzing, and utilizing publicly available information for various purposes, such as investigations, cybersecurity, business intelligence, and investigative journalism. In the context of cybersecurity, OSINT is primarily used to identify vulnerabilities, monitor threats, and gather information on potential attacks.

These platforms, often accessible via the internet, are designed to assist cybersecurity professionals, providing practical tools to assess and diagnose vulnerabilities in web systems. The main advantage of these tools lies in their user-friendly interfaces, which typically require minimal input from the user to return the desired analysis results. These platforms generally work by scanning websites and applications for common security flaws, such as SQL injection vulnerabilities, cross-site scripting (XSS), and weak configurations.

Some of these online tools are designed to provide quick

[1]Repository: https://github.com/xKaliBurr2024/SBSeg2024

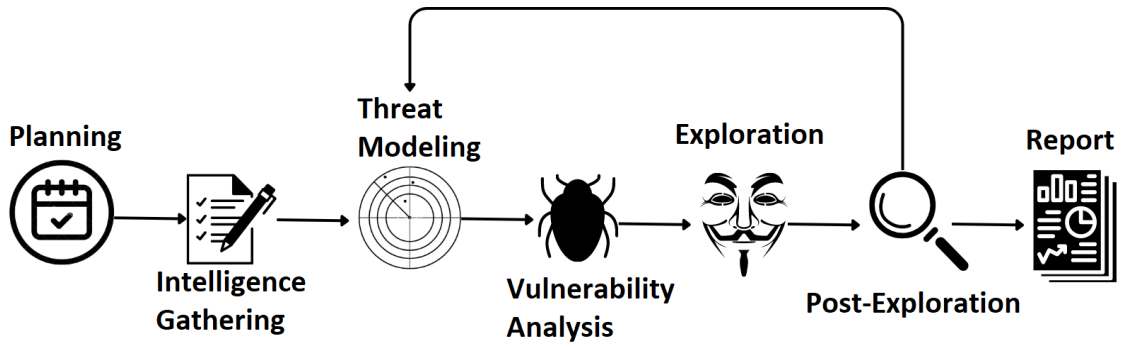

Figure 1. Pentesting Steps

and general reports, while others offer deeper insights, including the identification of advanced threats, misconfigurations, and compliance with industry security standards. Although these platforms can significantly enhance the efficiency of vulnerability assessments, their analysis may often be limited by the complexity of the environment being tested and the depth of the security tests they perform. Additionally, some of them offer partial free functionality, providing only a simplified and incomplete test for free, and offering full access to their features in exchange for a paid service plan.

In contrast to other tools, the Web xKaliBurr proposal automates the Information Gathering phase, one of the steps involved in the Pentest process, integrating additional functionalities aimed at more detailed investigations and offering a broader scope of analysis of the types of vulnerabilities that can be detected. Furthermore, unlike many online platforms that rely on predefined scanning protocols, Web xKaliBurr offers greater practicality in executing its evaluations, performing specific scans according to the configuration of the target system's web infrastructure. This entire process is made possible through the provision of open-source code, which is completely free and easy to use.

The tool developed by [Probely, 2025] is a free and widely accessible online resource designed to evaluate the presence and effectiveness of Security Headers in communications between clients and servers. By analyzing packet traffic over the Internet, the tool identifies which security headers are implemented and assesses their configurations to determine whether they adhere to best-security practices. This verification process is crucial for web applications, as improperly configured or missing security headers can expose systems to various cyber threats. Security headers are great mechanisms that enable servers to send specific instructions to browsers, reinforcing protection against a wide range of common attacks, including XSS (Cross-Site Scripting), Clickjacking, and code injection attacks. These headers are embedded in HTTP responses and play a key role in defining strict security policies that regulate browser behavior when interacting with web applications. By enforcing these policies, security headers help mitigate the risk of data breaches, unauthorized access, and malicious exploits, thereby enhancing the overall security posture of web platforms.

The service [DeHashed, 2025] is a powerful search engine designed for digital asset tracking and data mining across the *deep web*. It specializes in identifying, collecting, and indexing leaked information, including compromised user login credentials, entire system databases, and other sensitive records that may have been exposed due to security breaches. By continuously scanning underground forums, dark web marketplaces, and publicly accessible data dumps, the service provides cybersecurity professionals, ethical hackers, and organizations with valuable insights into potential threats and vulnerabilities affecting their digital assets. In addition, it allows users to proactively monitor compromised credentials associated with their domains, allowing timely remediation actions to mitigate risks. This service plays a crucial role in detecting and preventing Data Leakage, a term used to describe the unauthorized diversion or exposure of confidential or private information to individuals, systems, or organizations that lack the proper authorization to access it. Data leakage can manifest itself in numerous ways and in various environments, affecting corporations, government institutions, and individuals alike. The compromised data may include highly sensitive information, such as personally identifiable details, financial records, healthcare data, intellectual property, or security-related assets. Left unaddressed, such leaks can lead to identity theft, financial fraud, espionage, and other severe consequences, highlighting the importance of continuous monitoring and robust cybersecurity measures.

The system developed by [Shodan, 2025] is an advanced search engine specifically designed for indexing internet-connected devices, being the first tool of its kind focused on the Internet of Things (IoT) and digital infrastructure. Unlike the approach of Web xKaliBurr, which focuses specifically on exploits targeting web pages and applications, Shodan enables the discovery of a wide range of devices, including servers, routers, security cameras, industrial control systems, exposed databases, and even smart home devices such as refrigerators and virtual assistants. Its operation is based on the continuous scanning of the internet for publicly accessible services, collecting detailed information on IP addresses, open ports, service banners, and protocols in use. In addition to serving as a valuable tool for cybersecurity researchers and network administrators, ShodanMonitor allows for real-time monitoring of asset exposure, helping organizations identify vulnerable devices before they can be exploited by malicious actors. The Shodan enhances security beyond the traditional perimeter, enabling the detection of data leaks in the cloud, compromised servers, phishing websites, and exposed infrastructures, ensuring a proactive approach to cybersecurity

defense.

Sherlock is a popular tool developed by [Sherlock, 2025], designed for searching usernames across various social networks and websites. It is widely used in investigations and security testing to verify the presence of a username on multiple online platforms, facilitating the recognition of digital identities and the discovery of public information about a person or organization. Sherlock allows users to enter a username, and it automatically searches for its presence across numerous social platforms and popular websites. It checks whether the username is available on social networks such as Facebook, Instagram, Twitter, LinkedIn, GitHub, among others. This tool makes it easier to uncover information about individuals that could be exploited in social engineering attacks, such as phishing. During penetration testing, Sherlock can be used to map information regarding the online presence of an organization or individual.

The TheHarvester tool, created by [TheHarvester, 2025], was the software that most closely resembled the operating mode of Web xKaliBurr. It is an OSINT (Open-Source Intelligence) tool used to collect publicly available information about domains, companies, or individuals. TheHarvester is employed by cybersecurity professionals, pentesters, and intelligence analysts to conduct both passive and active reconnaissance. It was designed to be used during the reconnaissance phase in Red Team assessments or penetration tests. The tool performs open-source intelligence (OSINT) gathering to assist in identifying a domain's external threat landscape. Additionally, it enables the retrieval of names, emails, IP addresses, subdomains, and URLs from various public sources. However, although the functionality and objectives are quite similar, Web xKaliBurr utilizes different resources for information gathering, allowing for a more comprehensive data collection on the analyzed target. Another important characteristic to highlight is that TheHarvester operates exclusively through command-line interfaces, which can be a limitation for users with less technical knowledge. Thus, Web xKaliBurr aims to accommodate a larger number of users by offering a more intuitive interface while also providing its services and functionalities online.

SpiderFoot is an open-source intelligence (OSINT) automation tool designed to collect and correlate information from over 200 specialized modules. It provides an intuitive web interface, as well as full command-line support, offering great flexibility for use by both analysts and automated systems. Key features include subdomain enumeration, metadata collection, leak analysis, detection of public storage buckets, exposure checks on social networks, and integration with external APIs such as Shodan, HaveIBeenPwned, and SecurityTrails. It even supports dark web searches via TOR. Its module-based architecture enables automatic chaining of collected data, allowing for deep target analysis. In addition to the open-source version, the project also offers SpiderFoot HX, a cloud-based enterprise edition with advanced monitoring and collaboration features [SpiderFoot, 2025].

OWASP Amass is an open-source tool aimed at attack surface mapping and external asset discovery on the internet. Using passive and active reconnaissance techniques and open-source intelligence (OSINT) collection, Amass allows advanced subdomain enumeration, WHOIS data gathering, ASN analysis, DNS resolution, among other capabilities. It is widely used by security professionals to identify exposed assets, assess infrastructure, and uncover potential attack vectors. The tool operates primarily via the command line and integrates with various external data sources, also supporting containerized execution via Docker. Its data model, known as the Open Asset Model (OAM), facilitates the organization and analysis of information in large environments, making it a robust choice for both point-in-time assessments and continuous digital asset monitoring [OWASP Amass Project, 2025].

The Ax Framework is an open-source tool geared toward offensive security professionals, such as bug hunters and penetration testers, operating in cloud environments. Its main goal is to simplify the construction and management of ephemeral, highly scalable infrastructure for large-scale security testing. Using Packer provisioner files (JSON or HCL), Ax enables the configuration of base images with specific security tools, which can be replicated across dozens or hundreds of instances deployed to providers such as AWS, Azure, GCP, DigitalOcean, among others. The framework also includes utility scripts that streamline tasks like parallel command execution, file transfers, backups, and automation of continuous scanning pipelines. This on-demand infrastructure approach allows for fast, secure, and repeatable operations, optimizing time and cost in offensive scanning campaigns and automated security testing [Ax Framework, 2025].

Sn1per is an automated vulnerability assessment and reconnaissance platform used by security professionals to perform offensive scans and attack surface mapping. Developed as an open-source tool, Sn1per integrates several popular utilities from the security ecosystem such as Nmap, Nikto, Metasploit, WhatWeb, and many others, allowing comprehensive testing with minimal manual effort. Sn1per offers multiple operation modes, including recon, scan, exploit, and report, and even supports collaborative environments through a web interface (in the Professional version). Its features include subdomain enumeration, service detection, identification of known vulnerabilities, brute-forcing of directories, and generation of customizable technical reports. Widely adopted in penetration testing, Sn1per stands out for its ability to gather, automate, and organize large amounts of scan data in a centralized and efficient manner [Sn1per, 2025].

In Edwards [2019], the authors highlight the growing shortage of qualified professionals in Red Teams, specialized groups focused on Offensive Security, responsible for simulating real attacks to identify vulnerabilities and strengthen organizations' security posture. Due to this lack of experts, the work emphasizes the need to create automated tools capable of addressing this gap, enabling penetration tests to be conducted more efficiently and at scale. To this end, the authors propose the development of a comprehensive pentest simulator, designed to replicate the processes and techniques employed by a human professional during an attack. However, the described tool operates within a White Box testing context, meaning the evaluators already possess detailed information and privileged access to the target system, such as credentials, architecture, and source code. This characteristic makes the simulator suitable for controlled and specific environments where deep system analysis is possible, but also im-

poses prerequisites for its operation, limiting its use in Black Box scenarios. Thus, the work contributes to the automation of offensive security testing by offering an alternative to mitigate the shortage of specialized professionals, albeit with some operational constraints inherent to the adopted model.

In the study conducted by Laxmi Kowta *et al*. [2021], the authors provide a detailed analysis of the Information Gathering phase within offensive security processes, emphasizing its role as the initial and critical step in any penetration test. The work presents practical examples of various exploratory tools that are executed independently to collect relevant data about the target, including subdomains, open ports, active services, and potential vulnerabilities. The authors stress that Information Gathering goes beyond mere data collection, encompassing the identification, classification, and recognition of sensitive information that can facilitate subsequent stages of the test, such as exploitation and post-exploitation. Thus, the study highlights the need for careful planning and the combined use of specialized tools to maximize the effectiveness and depth of the initial reconnaissance in real-world offensive security environments.

Table 1 presents a comparative analysis between the tools discussed in this section and the solution proposed in this work. To the best of our knowledge, Web xKaliBurr is the only open-source platform specifically developed for automating the Information Gathering phase in websites, a fundamental step in the Pentest process. Our tool represents a direct evolution of the proposal introduced by [Barros *et al*., 2023], distinguishing itself by offering online access, eliminating the need for local installation and command-line execution, and incorporating a detailed report generation mechanism. Additionally, Web xKaliBurr not only identifies potential vulnerabilities but also provides specific security recommendations for each detected finding, facilitating the understanding and mitigation of risks associated with the target. As a result, the tool becomes an accessible and intuitive solution, allowing even users with limited technical knowledge to perform preliminary security assessments of their online environments.

3 Web xKaliBurr

In this paper, we present a tool called Web xKaliBurr, designed to accelerate and simplify tasks involved in the information-gathering phase of web penetration tests. The proposed tool was fully developed in Flask and Python. In this section, we will describe in depth the Web xKaliBurr's architecture and the information gathering flow used internally.

3.1 Architecture

The foundation for the architecture of Web xKaliBurr relies on resources found in the Kali Linux operating system[2], a GNU/Linux distribution based on Debian, recognized as a comprehensive environment for penetration testing and security analysis.

The Web xKaliBurr can be executed locally via Docker or accessed remotely through a web interface. The execution environment only requires Docker and internet access. The user interacts with a simple form and receives the results on screen or as a downloadable PDF report.

The proposed tool employs a client-server architecture, where the *back-end* consists of a Docker *container* with a custom image based on Kali Linux. This *container* encapsulates all the software and tools executed by Web xKaliBurr to perform exploration activities. An API serves as the communication channel between the tools in the *back-end* and the *front-end*. The API was built using the Python programming language and the Flask *framework*, leveraging its native features to execute calls for each tool that comprises the system. This architecture is illustrated in **Figure 2**.

3.2 Information Gathering Flow

The Web xKaliBurr tool provides a service that performs extensive Information Gathering on online applications, taking the communication protocol and the site's domain name as input. Based on this input, Web xKaliBurr initiates its exploration *pipeline*, conducting scans to gather additional information about the specified target, revealing potential attack surfaces. Figure 3 illustrates the information gathering flow used by Web xKaliBurr.

The complete exploration process comprises seven steps, each described in detail hereafter. Each step explores different Kali tools. According to Figure 3, these seven steps are as follows:

- **Step 1:** The exploration process begins with the user supplying two input parameters: the website's domain name and the communication protocol (HTTP or HTTPS).
- **Step 2:** Next, the identification of IP addresses associated with the target domain is performed using the WhatWeb and Host tools. The WhatWeb, developed by urbanadventurer aka Andrew Horton and GPLv [2025], is a reconnaissance tool used to identify technologies and software on websites, detecting web servers, frameworks, plugins, and other relevant information. It is widely employed by penetration testers and security analysts, allowing for quick or detailed scans for vulnerability analysis. On the other hand, Host is a command-line utility used to query DNS information for a domain or IP address, including name resolution, DNS records, and other predefined data. It is particularly useful for network diagnostics and DNS configuration analysis.

[2]https://www.kali.org/

Table 1. Comparisons Between Pentest Tools

OSINT Tool	Open-Source	Completely Free	User Interaction	Functionality
[Probely, 2025]	No	Yes	Graphical Interface	Security Header Analysis
[DeHashed, 2025]	No	No	Graphical Interface	Sensitive Data Leak
[Shodan, 2025]	No	No	Graphical Interface	Device Search and Monitor
[Sherlock, 2025]	Yes	Yes	Command Line	Searching Usernames
[TheHarvester, 2025]	Yes	Yes	Command Line	Information Gathering
[SpiderFoot, 2025]	Yes	Yes	Web Interface or CLI	Automated OSINT
[OWASP Amass Project, 2025]	Yes	Yes	Command Line	Attack Surface Mapping
[Ax Framework, 2025]	Yes	Yes	Command Line	Cloud Distributed Recon
[Sn1per, 2025]	Yes	No	Command Line (GUI in Pro)	Scanning Failures
[Edwards, 2019]	No	No	Command Line	Pentest Simulator
[Laxmi Kowta *et al.*, 2021]	No	No	Command Line	Data Collection
Web xKaliBurr	Yes	Yes	Web Graphical Interface	Information Gathering

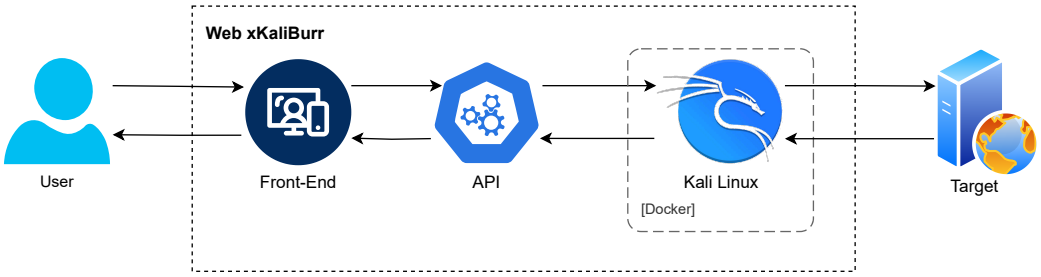

Figure 2. Web xKaliBurr Architecture

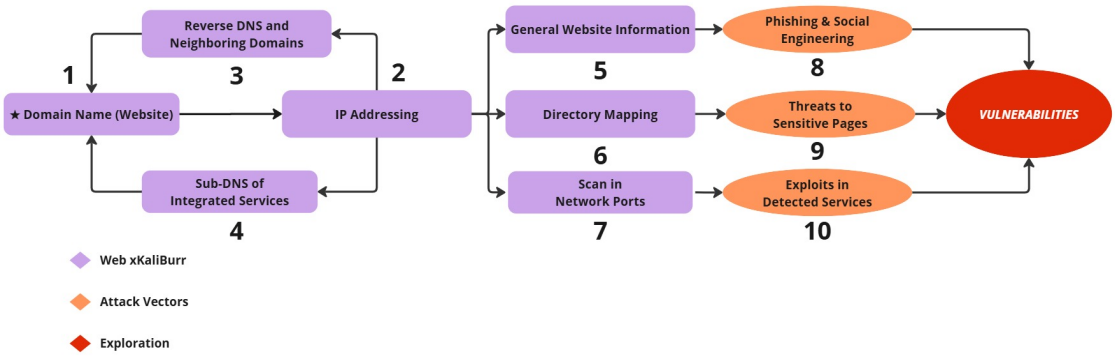

Figure 3. Web xKaliBurr Information Gathering Flow

- **Step 3:** Recursive explorations are carried out using reverse DNS to verify the existence of systems related to the main target. In this step, the DNSRecon tool is employed. DNSRecon is a DNS enumeration tool used to collect record information, zone transfers, and subdomains. It is widely used in security testing and network reconnaissance.
- **Step 4:** Recursive explorations are carried out using sub-DNS to verify the existence of systems related to the main target. In this step, the DNSRecon and DNSMap tools are employed. DNSMap is a tool used to discover subdomains of a target domain through brute-force attacks. It is useful in security testing and reconnaissance, helping to identify hidden subdomains that may expose vulnerable services.
- **Step 5:** General information is collected, including data such as owner names, companies or employees, contact emails, and other details that could serve as a basis for malicious attacks. The data search tools used in this step were WhatWeb, WhoIs, and CURL. The WhoIs tool is used to query information about domain registration, such as the owner, contact details, creation and expiration dates, and name servers (DNS). It is commonly used for domain investigation and availability verification on the Internet. On the other hand, CURL Stenberg [2025] is a software tool used to transfer data to a server using various protocols such as HTTP, HTTPS, FTP, among others. It is widely used for making web requests, testing APIs, downloading files, or interacting with servers via scripts.
- **Step 6:** Directory investigation is conducted through brute-force checks on possible web pages, using the DIRB and GoBuster tools to detect hidden pages susceptible to security flaws. DIRB is a brute-force tool used to discover hidden directories and files on web servers. It performs automated searches in URLs, attempting to

find paths and resources not publicly listed, making it highly useful in penetration testing and security audits. GoBuster, in turn, is a brute-force tool used to discover directories and subdomains on web servers. It is efficient, fast, and can be used for brute-force attacks on both directory URLs and subdomains, supporting HTTP, HTTPS, and DNS protocols. It is widely used in penetration testing to explore weaknesses in servers and networks.

- **Step 7:** Using the NMAP tool, the services employed in the target's infrastructure and their versions are identified. Potential security flaws can be detected by exploiting documented vulnerabilities, a strategy commonly used against systems running outdated services. Nmap (Network Mapper) Nmap Public Source License [2025] is an open-source tool widely used for network and security analysis. It allows the discovery of devices on a network, mapping open ports, identifying running services, and detecting vulnerabilities. Nmap is essential in penetration testing and security audits, offering features such as port scanning, operating system and service detection, as well as providing detailed information about hosts on the network.

It is important to highlight that Web xKaliBurr allows selective activation of different Kali tools via configuration parameters in the backend (currently under development for future exposure in the interface).

3.3 Exploration Scenarios

Web xKaliBurr begins its information gathering process by conducting investigations related to the identification of IP addresses (Internet Protocol), DNS enumeration (Domain Name System), and the subsystems integrated with the target under analysis, as depicted in steps 2, 3, and 4 of **Figure 3**. After gathering information about the services and IP addresses linked to the target, Web xKaliBurr proceeds with more in-depth searches, following four main exploration scenarios:

- **Information Leakage:** This scenario involves an attacker focusing their efforts on targeting employees and service administrators. These types of attacks can be characterized by Social Engineering tactics [Dewan *et al*., 2014]. In such cases, the malicious actor seeks to obtain privileged information by leveraging basic but genuine data that can be discovered through Information Gathering. The exploration flow {1, 2, 5} in **Figure 3** illustrates this scenario, which can lead to phishing and social engineering attacks.

- **Mapping of Sensitive Pages:** This scenario aims to uncover all hidden pages and directories related to the target system. A malicious actor can identify new attack surfaces by discovering pages that should not be accessible to regular users, such as employee login panels or system file transfer services. The exploration flow {1, 2, 6} in **Figure 3** illustrates this scenario, which can lead to threats to sensitive web pages.

- **Service Detection:** This scenario focuses on identifying and obtaining the versions of technologies used in the target's infrastructure. In this context, a malicious actor can gather this information and use it to attempt exploits targeting vulnerabilities in these services. The exploration flow {1, 2, 7} in **Figure 3** illustrates this scenario, which can lead to exploits in different services.

- **Neighborhood Identification:** This scenario focuses on detecting and identifying neighboring domains, meaning web pages and online applications within the same IP address range as the analyzed target. This allows an attacker to perform "lateral movement", when a direct attack on the target is not possible, but a connected domain is more vulnerable, increasing the attack surface for malicious actors. The exploration flow {2, 3, 4} in **Figure 3** illustrates this operation.

The results of each exploration scenario, obtained through the execution of different tools, are categorically organized and displayed on the Web xKaliBurr graphical interface. At the end of the execution, for each scenario, the identified potential vulnerabilities are presented, along with security recommendations and key points of concern, making it easier for users with less technical knowledge to understand.

4 Running Example

In order to illustrate the use of Web xKaliBurr, let us consider the execution of the information gathering phase of a web penetration test targeting the OWASP Juice Shop online environment[3]. This environment was developed by the OWASP (Open Web Application Security Project) organization to support training in information security. Juice Shop includes vulnerabilities from the entire OWASP Top Ten list [OWASP Foundation, 2021], which represents the most critical security risks for web applications. Due to its educational purpose, Juice Shop includes a series of challenges that involve executing different phases of a penetration test. Upon completing each challenge, the user's progress is recorded on the platform's scoreboard.

As an initial step, the user is required to enter the domain of the target website (https://juice-shop.herokuapp.com/) and the corresponding communication protocol (HTTPS), as illustrated in Figure 4.

Subsequently, Web xKaliBurr initiates its analysis. After performing the exploits on Juice Shop, Web xKaliBurr was able to solve 26 out of the 169 existing challenges, which correspond to the challenges related to the Information Gathering stage. These challenges would typically be solved manually, while the tool allowed for automating this process. The incomplete challenges correspond to subsequent stages of Pentest, many of which rely on information gathered in the stage performed by Web xKaliBurr. Additionally, it is important to emphasize that Web xKaliBurr was developed to perform a comprehensive scan, without taking into account prior knowledge of Juice Shop or any specific target. The **Figure 5** presents a percentage of challenges completed through

[3] https://juice-shop.herokuapp.com/

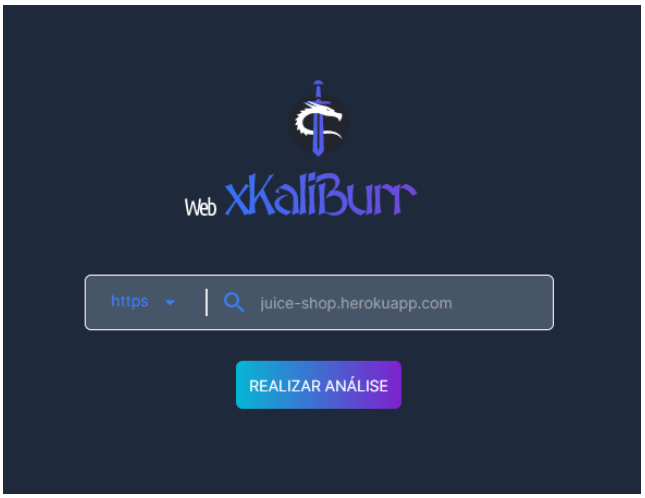

Figure 4. Illustrative Example of User Input on the Web xKaliBurr

the automated execution of Web xKaliBurr, as provided by the ranking display of the Juice Shop platform itself.

Although the number of challenges solved is not a valid metric for assessing the tool's performance, as many of the challenges fall outside its scope, we consider this a satisfactory result to illustrate its functionality. The completed challenges can also be analyzed according to the following vulnerability categories from OWASP Juice Shop:

- `Sensitive Data Exposure:` Scenarios in which confidential or private information is inadvertently revealed or exposed to unauthorized individuals. The identification of this vulnerability was carried out through steps 3, 4, 6, and 7 of Web xKaliBurr.
- `Improper Input Validation:` This vulnerability occurs when a system does not properly validate user-provided data before processing it, allowing the injection of malicious data. The exploitation of this flaw occurred by manipulating parameters in HTTP requests, using the tools present in step 6 of Web xKaliBurr.
- `Broken Access Control:` This happens when a system does not correctly enforce access restrictions to protected resources, allowing unauthorized access to functionalities or information. Vulnerable areas were identified during the Directory Mapping phase in step 6 of Web xKaliBurr.
- `Invalid Redirects:` This occurs when a web application redirects the user to an untrusted or invalid URL without performing proper checks. The CURL tool, present in step 5 of Web xKaliBurr, can perform these URL redirects improperly on the target.
- `Broken Authentication:` This scenario arises when authentication measures are misconfigured, allowing attackers to bypass security mechanisms and access protected resources. These vulnerabilities were exploited using the NMAP tool in step 7 of Web xKaliBurr's exploitation process during network port scanning of the target infrastructure.
- `Security Misconfiguration:` This scenario occurs when there are improper configurations that make the system more susceptible to exploitation by attackers. The WhatWeb tool, used in step 5 of the exploration, detected the exposure of administrative interfaces in the target system.

In addition to the automatic completion of challenges, Web xKaliBurr also gathered information related to the infrastructure characteristics of the Juice Shop domain, pertaining to the four exploitation flows executed by the platform, as shown in **Figure 6**. Through the analysis of this information, it is possible to identify the configurations and technologies used by the Juice Shop hosting domain itself, collecting hidden information about the platform solely from the public data related to the website's domain name. **Figure 6** also illustrates the tool's recommendation functionality, where actions and points of attention regarding security best practices are advised, making the analysis easier for less experienced users.

5 Evaluating the Web xKaliBurr Tool

The usability nature of Web xKaliBurr is a key aspect to achieve the acceptance of expert and non-expert users. In this section, we will detail the usability and satisfaction assessment performed to evaluate Web xKaliBurr.

Usability is defined as the extent to which a specific user in a certain context can use a product to achieve a defined goal effectively, efficiently, and satisfactorily. Satisfaction is related to how the users believe or feel positively that the product meets their requirements.

5.1 Usability Tests

Usability is the "extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use". So, usability is measured through user interaction while using software, products, or services, seeking to achieve goals with efficacy, efficiency, and user satisfaction according to Bevan *et al*. [2016]. The word "usability" relates to the methods used to facilitate the usage of a product or service, during the design process, where every detail is strategically thought out and built Nielsen *et al*. [2012]. The evaluation results for the Web xKaliBurr tool present a positive outlook regarding usability and user satisfaction. The Net Promoter Score (NPS) reached an excellent mark of 70% indicating that a large majority of users are willing to recommend the tool to colleagues. This reflects a high level of loyalty and overall satisfaction. Complementing this, the System Usability Scale (SUS) assessment showed an average of 80 points, which indicates a good to excellent product. This demonstrates that users find the xKaliBurr interface easy to use, well-integrated, and effective for their needs, although there's still some room for improvement, as indicated by intermediate responses from some participants. The combination of these results suggests that the tool offers a consistent and valuable user experience, effectively serving its target audience, especially professionals looking to automate information gathering in penetration tests. It also has the potential for optimizations that could transform neutral or unsatisfied users into even more enthusiastic promoters. Usability evaluation ensures that products

Figure 5. Score Board Juice Shop Results

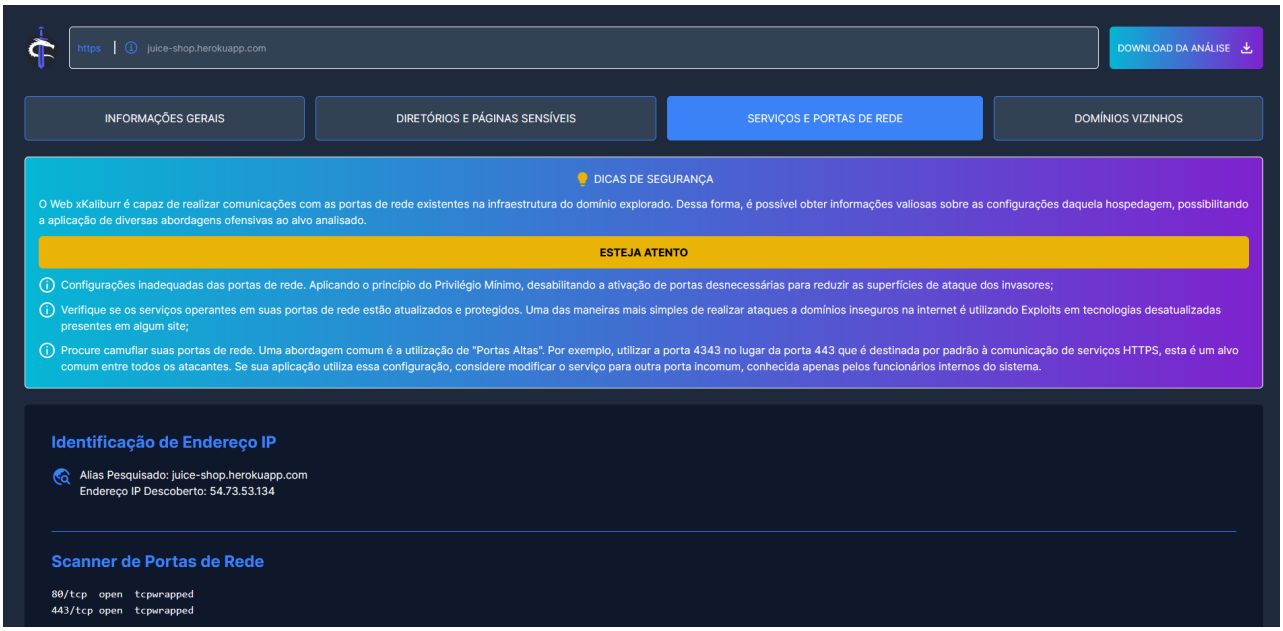

Figure 6. Example of Output Generated by Web xKaliBurr

or services are adapted to the users and their tasks. Its goal is to assess the degree of utility, efficiency, efficacy, learnability, accessibility, and satisfaction. Utility means the user's ability to use a particular product to achieve a specific goal. Efficacy is related to how well the system meets the tasks for which it was designed. Efficiency refers to the speed and precision with which the user achieves goals. Learning is the accumulated knowledge used by the user to handle a particular product. Accessibility consists of having access to products to achieve goals. Satisfaction refers to the user's perception of the product Charlton and O'Brien [2019]. Usability tests are user-centric design techniques used to evaluate a product or software in everyday situations. They allow feedback directly from users who work with or perform tasks with the analyzed object. It can measure how efficient and effective it is for predetermined goals. Besides, when carrying out the proposed tests, users often surprise the evaluators by taking unexpected actions while testing the software. To carry out a usability test, it is ideal to use already well-established methods, models, and artifacts. Some popular usability tests are: Nielsen's usability heuristics Nielsen [1995], System Usability Scale (SUS) Lewis [2018], Net Promoter Scores (NPS) Mandal [2014], Software Usability Measurement Inventory (SUMI) Kirakowski and Corbett [1993], Website Analysis and Measurement Inventory Questionnaire (Wammi) Claridge and Kirakowski [2011], and User Experience Questionnaire (UEQ) Schrepp [2015].

5.1.1 Net Promoter Score (NPS)

The NPS is a metric designed to measure the satisfaction of a customer or user so that companies or service providers can evaluate and improve their products and services Ras *et al.* [2017]. The concept of NPS relies on approaching customers or users on how likely they are to recommend products/services to their peers Korneta [2014]. The respondents give their answers on a scale of 0 (unlikely) to 10 (very likely), and they are labeled as "promoters", "passives" or "detractors". Users that answer with 9 or 10 are called promoters. Customers that give grades 7 or 8 are called passives (or indifferent), and those who give grades between 0 and 6 are called "detractors".

Promoters are classified as loyal customers who will always provide product/service recommendations to third parties. On the other hand, those with a passive profile are classified as customers who are satisfied with the company's products/services but have the potential to accept other products/services offered by competitors. Finally, detractors are dissatisfied and disloyal customers, driving other people away from using the company's products/services. Finally, the NPS is calculated as the difference between the proportion of promoters and detractors and can thus lie between +100 (promoters only) and -100 (detractors only) percent. Values above 0 are considered "good", above 50 are classified as "very good", and above 70 are interpreted as "excellent", in terms of product and service quality Lee [2018]. It is important to note that the NPS calculation does not use passive users. Figure 7 illustrates the NPS calculation an the users profile classification.

5.1.2 System Usability Scale (SUS)

System Usability Scale (SUS) is a standardized questionnaire

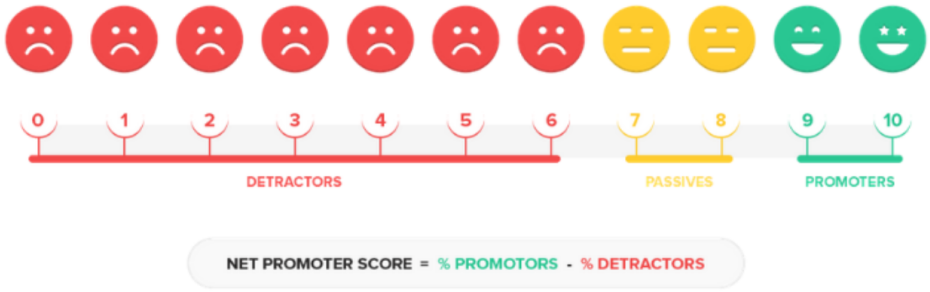

Figure 7. NPS Calculation

widely used to assess perceived usability. The survey consists of ten questions; each has a five-point Likert response continuum (from strongly agree to strongly disagree) Brooke [1996]; Lewis [2018]. Figure 8 shows these ten questions. Note that the odd items have a positive tone, while the even items have a negative tone.

The SUS scoring system requires ratings for all 10 items, so if a respondent leaves an item blank, they should receive a raw score of 3 (the center of the five-point scale). To calculate the SUS score, initially, the participant's scores (called raw item scores) for each question are converted to a new number (called adjusted scores or score contributions) as described next. For odd items, subtract one from the user response. For even items, subtract the user responses from 5. This process will scale all values from 0 to 4 (with four being the most positive response). Next, add up the converted responses for each user and multiply that total by 2.5. This converts the range of possible values from 0 to 100 instead of from 0 to 40. Though the scores are 0-100, they are not percentages and should be considered only in their percentile ranking. The following equation shows a more concise way to calculate a standard SUS score from a set of raw item ratings:

$$SUS = 2.5 * [[(Q1 + Q3 + Q5 + Q7 + Q9) + (Q2 + Q4 + Q6 + Q8 + Q10)]]$$

The SUS provides a score from 0 to 100. According to Bangor *et al*. [2009], a score of 85 or higher represents "exceptional" usability, a value between 72 and 85 denotes a "good" result, a score between 52 and 71 means "ok", and a value below 52 represents unacceptable usability. One of the main benefits of the SUS is that its output is an easy-to-understand score, ranging from 0 to 100, where the higher the SUS value, the better the usability of the product. This unitless score works very well for making relative comparisons. SUS allows you to evaluate a wide variety of products and services, including hardware and software.

5.2 Usability Assessment Settings

The usability assessment of Web xKaliBurr was designed to address two distinct user profiles: experts (individuals experienced in offensive security/penetration testing) and non-experts (individuals with minimal or no knowledge in the area). The tests were conducted remotely, with participants performing a series of tasks using the tool. Web xKaliBurr was set up to run locally on the participants' machines. This allowed users to access the tool without requiring an internet connection or specific server access. Before performing any tasks, participants completed a demographic form to map their respective profiles. After completing a set of information gathering tasks for a penetration test, they filled out the SUS (System Usability Scale) and NPS (Net Promoter Score) surveys.

5.2.1 Population

This usability test was attended by 10 participants, all with diverse academic and professional backgrounds including developers, researchers, UX designers, system analysts, and students, predominantly with undergraduate education. Their experience in relevant fields varied from beginners to professionals with up to 5 years of experience. The age of participants ranged from 22 to 30 years old.

5.2.2 Usability Assessment Interviews

To conduct the usability assessments, we initially scheduled individual interviews with each of the 10 participants. At the beginning, participants completed a demographic questionnaire collecting information such as age, academic background, professional experience, and familiarity with security and pentesting tools. Following this, each interviewee provided their availability for a subsequent session. In the second interview, participants executed the Web xKaliBurr tool locally on their machines. They were provided with a predefined dataset and instructed to perform a series of tasks designed to explore the main functionalities of the tool, including automated information gathering and analysis in pentesting contexts. After completing these tasks, participants responded to the System Usability Scale (SUS) and Net Promoter Score (NPS) questionnaires to assess their perceived usability and satisfaction with the tool.

5.3 Usability Test Results

This section presents and discusses the results of the usability tests, specifically focusing on the NPS and SUS metrics. The

System Usability Scale Questionnaire	Strongly Disagree				Strongly Agree
1. I think that I would like to use this product frequently.	1	2	3	4	5
2. I found the product unnecessarily complex.	1	2	3	4	5
3. I thought the product was easy to use.	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this product.	1	2	3	4	5
5. I found the various functions in the product were well integrated.	1	2	3	4	5
6. I thought there was too much inconsistency in this product.	1	2	3	4	5
7. I imagine that most people would learn to use this product very quickly.	1	2	3	4	5
8. I found the product very awkward to use.	1	2	3	4	5
9. I felt very confident using the product.	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this product.	1	2	3	4	5

Figure 8. Standard SUS Questionnaire

results are organized into three distinct user profiles: non-expert users, expert users, and a combined group including both profiles. This categorization allows for a clear comparison of usability perceptions across different user types. Additionally, it enables us to identify which profile demonstrates greater acceptance and satisfaction with the Web xKaliBurr tool. Ultimately, this analysis aims to answer the question: for which user profile is the Web xKaliBurr best suited?

5.3.1 NPS Results

This section presents the results obtained using the NPS method, organized as previously described.

The NPS responses were collected from ten participants. The Web xKaliBurr received 7 "promoters" (scores 9–10), 1 "passive" (score 7), and 2 "detractors" (scores 5 and 6). The NPS score is calculated as the difference between the proportion of promoters and detractors. Thus, NPS score = 70% - 20% = 50%. Scores between 0 and 50 indicate a "good" level of satisfaction. Therefore, Web xKaliBurr achieved a "good" satisfaction rating among users.

5.3.2 SUS Results

This section presents the results obtained using the System Usability Scale (SUS) method. To calculate the SUS score, participants answer 10 questions on a 5-point Likert scale (0 to 5), where odd-numbered items have a positive tone and even-numbered items have a negative tone. The initial responses are referred to as raw scores. These raw scores are then converted into adjusted scores: for odd-numbered items, one is subtracted from the user's response; for even-numbered items, the user's response is subtracted from 5. Finally, the adjusted scores are summed and multiplied by 2.5 to yield the standardized SUS score Lee [2018].

Table 2 displays the SUS responses provided by the ten participants. The final SUS score is the average of the individual SUS scores obtained from these participants. Thus, the final SUS score was **79.9**. It's important to note that a SUS score between 52 and 71 indicates "okay" usability, while a score ranging from 72 to 85 signifies "good" usability. Therefore, we can conclude that the SUS score of 79.9 for the Web xKaliBurr tool among participants indicates **good usability**.

5.4 Evaluation with Pentest Experts

This section details the evaluation performed with pentest experts using the Web xKaliBurr tool, offering a deep dive into its real-world performance. The primary objective of this test was to thoroughly assess the tool's effectiveness from the perspective of experienced offensive security professionals. We engaged three unique participants, all seasoned specialists in penetration testing, to put Web xKaliBurr through its paces. To ensure a realistic and comfortable testing environment, all evaluations were conducted remotely. Participants utilized their own machines, interacting directly with the locally installed Web xKaliBurr tool, mirroring how they would typically integrate such a solution into their daily operations. The evaluation focused on several key aspects crucial to a specialist's workflow. We aimed to understand precisely how

Table 2. SUS Responses from Participants and Scores

Question	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Raw Score	SUS Final
1	4	3	3	4	3	5	5	2	5	5		
2	1	1	1	1	3	1	1	2	1	2		
3	5	3	5	5	4	5	5	4	5	5		
4	2	5	1	2	5	3	5	2	1	1		
5	4	3	5	4	5	5	5	4	5	4		
6	1	3	1	2	1	1	1	1	5	2		
7	4	3	5	5	2	5	5	3	5	5		
8	1	2	1	1	3	1	1	3	1	1		
9	4	2	5	4	4	5	4	3	5	4		
10	2	3	1	1	4	1	1	2	1	1		
Raw Score	34	20	38	34	22	38	35	27	36	36	–	–
SUS Final	85.0	50.0	95.0	85.0	55.0	95.0	87.5	67.5	90.0	90.0	–	–
AVG											**32.0**	**80.0**

Web xKaliBurr integrates into and enhances their existing processes. To achieve this, participants were asked a series of targeted questions designed to gauge the tool's tangible impact on their efficiency during a pentest, the overall quality and depth of the findings it generated, and the comprehensiveness and accuracy of the reports it produced. This approach allowed us to collect qualitative and quantitative data, providing a holistic view of the tool's value in a professional context.

5.4.1 Expert Questionnaire

To thoroughly assess Web xKaliBurr's impact, we designed a comprehensive set of seven questions for our expert participants. These questions aimed to capture both direct utility and perceived value across critical aspects of penetration testing. Specifically, we wanted to understand how the tool affected their workflow, the depth of their findings, and the quality of the outputs. Next, we will describe each question in deep:

- Question 1: Did using the Web xKaliBurr tool help accelerate the Pentest process? (Yes/No)

This question directly addressed the tool's efficiency in practical application, seeking to confirm if it genuinely saved time during a penetration test.

A resounding **100% of participants (3 out of 3)** unequivocally confirmed that using Web xKaliBurr significantly **accelerated their Pentest process**. This consistent feedback highlights the tool's substantial efficiency gains, enabling specialists to complete assessments more quickly without compromising thoroughness.

- Question 2: Did the Web xKaliBurr tool identify any aspect not observed in your manual assessment? (Yes/No)

Here, we sought to determine if the automated capabilities of Web xKaliBurr could uncover blind spots or provide additional insights that a human analyst might miss during a manual review.

Furthermore, **100% of participants (3 out of 3)** also reported that the tool **identified aspects not observed in their manual assessment**. This is a critical finding, as it indicates Web xKaliBurr's capability to act as a powerful complement to traditional manual methodologies, uncovering additional insights or vulnerabilities that might otherwise be overlooked by human analysts.

- Question 3: Did you find any vulnerability in your manual assessment that was not reported by the Web xKaliBurr tool? (Yes/No)

This critical question evaluated the tool's comprehensiveness and its ability to detect vulnerabilities without significant omissions, highlighting any areas where manual expertise remains uniquely necessary.

Regarding the tool's comprehensiveness in vulnerability detection, **66.7% of participants (2 out of 3)** stated that they **did not** find any vulnerability manually that was *not* reported by Web xKaliBurr. However, one participant did identify such an instance. While this strong majority signifies robust coverage, this single case suggests a minimal margin for refinement in the tool's detection capabilities or highlights specific edge cases where manual expertise remains crucial for identifying nuanced findings.

- Question 4: How do you evaluate the organization of the report generated by the Web xKaliBurr tool? (Scale of 1-5, where 1=Very Poor, 5=Excellent)

Beyond mere detection, the clarity and structure of the output are vital. This question assessed how well the information was presented in the generated reports.

Participants rated the organization of the generated reports with an average score of **3.67**. This indicates that while the report's structure is generally considered good and functional, there might be minor opportunities for refinement to achieve consistently excellent readability and intuitive navigation across all user perspectives.

- Question 5: Did using the Web xKaliBurr tool help improve the quality of the final Pentest result? (Yes/No)

This question focused on the overall impact, aiming to confirm if the tool enhanced the robustness, accuracy, or depth of the entire penetration test outcome.

Significantly, **100% of participants (3 out of 3)** concurred that using Web xKaliBurr **elevated the quality of their final Pentest results**. This powerful consensus reinforces the

tool's value proposition: it not only streamlines the assessment timeline but also genuinely enhances the thoroughness, accuracy, and overall effectiveness of the security analysis.

- Question 6: How do you evaluate the completeness of the information presented in the report generated by the Web xKaliBurr tool? (Scale of 1-5, where 1=Very Poor, 5=Excellent)

A thorough report provides all necessary details. This question gauged whether specialists felt the reports offered sufficient and exhaustive information.

The completeness of the information presented in the reports consistently received an average score of **4.0**. This highly positive result demonstrates that specialists perceive the reports as comprehensive, containing all necessary and relevant data to effectively interpret findings and inform subsequent actions.

- Question 7: How do you evaluate the correctness (accuracy and truthfulness) of the information contained in the report generated by the Web xKaliBurr tool? (Scale of 1-5, where 1=Very Poor, 5=Excellent)

Finally, trust in the data is paramount. This question aimed to ascertain the specialists' confidence in the precision and validity of the findings reported by Web xKaliBurr.

Crucially, participants rated the correctness (accuracy and truthfulness) of the information with an excellent average score of **4.33**. This outcome reflects a very high level of trust among specialists in the precision and validity of the data provided by Web xKaliBurr's reports, which is paramount for actionable security assessments.

The responses from the experts provided compelling evidence of Web xKaliBurr's effectiveness, revealing strong positive feedback across most evaluated metrics. These insights underscore the tool's practical utility in a professional penetration testing environment. Next, we will discuss the most relevant results.

These targeted questions allowed us to gather both qualitative insights and quantifiable feedback on the Web xKaliBurr tool's performance from the perspective of experienced professionals.

In summary, the specialist testing revealed that Web xKaliBurr is perceived as an **extremely valuable asset** in penetration testing. It consistently **accelerates workflows, identifies additional findings, and significantly enhances the quality of final reports**. While there's minimal room for enhancing report organization and addressing rare missed vulnerabilities, the overall feedback is overwhelmingly positive, cementing Web xKaliBurr's utility in a professional security context.

6 Limitations

In this section, we will discuss some limitations of the proposed tool. Initially, it is important to note that Web xKaliBurr allows the selective activation of different Kali tools solely through configuration parameters in the backend. As future work, we intend to expose this functionality through the graphical interface.

Besides, the output report is presented in a web page organized in four tabs, following the four exploration scenarios presented in Section 3.3. For each scenario, the identified potential vulnerabilities are presented, along with security recommendations and key points of concern, making it easier for users with less technical knowledge to understand. The user may download the report in PDF format, if desired. Currently, the report must be read and interpreted by a penetration testing specialist. Nevertheless, we are currently exploring the use of large language models (LLMs) to support non-specialist users in understanding the contents of the generated report.

7 Conclusion

This work proposed an open-source web tool aimed at performing Information Gathering in Penetration Tests for online environments, called Web xKaliBurr. The tool enables a comprehensive automated scan, capable of collecting characteristics and information not publicly available, which would normally be obtained manually through specific investigative activities. Web xKaliBurr provides an easy-to-use interface that enables extensive information scanning of websites, requiring only the domain name and communication protocol as input. Additionally, the collected data is presented in a technical report alongside security recommendations to inform users about potential vulnerabilities. Thus, the tool allows users with no advanced knowledge of cybersecurity to perform an initial assessment of the attack surfaces present in their domains.

In order to evaluate Web xKaliBurr, we applied the System Usability Scale (SUS) questionnaire to measure usability based on users' subjective assessments. The SUS score obtained was 80, indicating a good to excellent level of usability. Additionally, we used the Net Promoter Score (NPS) method to assess user satisfaction and willingness to recommend Web xKaliBurr to others. The NPS reached a value of 70% pointing high level of satisfaction and strong user loyalty. These results demonstrate that Web xKaliBurr provides a valuable and user-friendly experience for its target audience. Finally, we performed a practical evaluation with 3 experts in web Pentests.

As a plan for future work, the continuation of Web xKaliBurr's execution on the remaining websites in the same university network is intended, as well as in other susceptible environments. The goal is to enhance the tool's performance and gather more diverse reports on the vulnerabilities detected. With the gradual increase in the number of reports, the development of a personalized Database is planned, containing all the vulnerabilities found and pointed out by Web xKaliBurr. This database could be used in future studies, such as qualitative and quantitative analyses of these virtual threats, investigations on the behavior of security flaws, and even as benchmarks for optimizing tests with other tools under development. Another desired development line is the continuous improvement of Web xKaliBurr. The intention is to optimize its performance by adding new functionalities aimed at both cybersecurity experts and lay users who wish to conduct security checks practically and without complications in their environments. In this regard, the creation of a

mechanism for interpreting the generated technical reports is envisioned, capable of identifying and prioritizing activities needed to resolve the most severe security flaws. This way, the goal is to implement an artificial intelligence model focused and specialized in Web xKaliBurr's reports, acting as a bridge for understanding the best protection practices, essential to enhancing the security and reliability of all websites and applications on the global internet.

Declarations

Authors' Contributions

DB proposed the conception of the central idea of the work, developed the MVP of the tool, conducted the experiments and contributed to the writing of the texts. LC had a significant role in the research and the development of the manuscripts. JA was responsible for the official coding of the tool's Back-End systems. FC worked on the Front-End integration. LS developed the visual identity, as well as all the artwork and animations present in the Front-End. JM, JB, and LR contributed to refining the ideas of the work, reviewed the written texts, and guided the conducted studies. DB is the main contributor and writer of this manuscript. All authors have read and approved the final version of the manuscript.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

This work was partially financed by Lenovo, as part of its investment in R&D under the Information Technology Law (Law nº 8,248/1991).

Availability of data and materials

The datasets (and/or softwares) generated and/or analysed during the current study will be made upon request.

References

Ax Framework, M. L. (2025). Ax framework. Available at:`https://ax.attacksurge.com/`.

Bangor, A., Kortum, P., and Miller, J. (2009). Determining what individual sus scores mean: Adding an adjective rating scale. *Journal of usability studies*, 4(3):114–123. Available at:`https://uxpajournal.org/determining-what-individual-sus-scores-mean-adding-an-adjective-rating-scale/`.

Barros, D. R., Pimenta, S. A., Rocha, L. S., and Monteiro, J. M. (2023). Exekaliburr: uma ferramenta exploratória auxiliar para o levantamento de informaçoes em pentests web. In *Anais Estendidos do XXIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*, pages 1–8. SBC. DOI: 10.5753/sbseg$_e$$stendido$.2024.242014.

Bevan, N., Carter, J., Earthy, J., Geis, T., and Harker, S. (2016). New iso standards for usability, usability reports and usability measures. In *International conference on human-computer interaction*, pages 268–278. Springer. DOI: 10.1007/978-3-319-39510-4$_2$5.

Brooke, J. (1996). Sus: a "quick and dirty'usability. *Usability evaluation in industry*, 189(3). Available at:`https://www.researchgate.net/publication/228593520_SUS_A_quick_and_dirty_usability_scale`.

Charlton, S. G. and O'Brien, T. G. (2019). *Handbook of human factors testing and evaluation*. CRC Press. DOI: 10.1201/9781003000815.

Claridge, N. and Kirakowski, J. (2011). Wammi: website analysis and measurement inventory questionnaire. *Retrived May*, 20(2013):57–66. Available at:`https://www.wammi.com/questionnaire.html`.

De Jimenez, R. E. L. (2016). Pentesting on web applications using ethical-hacking. In *2016 IEEE 36th Central American and Panama Convention (CONCAPAN XXXVI)*, pages 1–6. IEEE. DOI: 10.1109/CONCAPAN.2016.7942364.

DeHashed (2025). Dehashed. Available at:`https://dehashed.com/`. Take your employee security to the next level. Accessed at 02/05/2025.

Dewan, P., Kashyap, A., and Kumaraguru, P. (2014). Analyzing social and stylometric features to identify spear phishing emails. In *2014 apwg symposium on electronic crime research (ecrime)*, pages 1–13. IEEE. DOI: 10.1109/ecrime.2014.6963160.

Edwards, P. L. (2019). *Cyber Automated Red Team Tool*. PhD thesis, Monterey, CA; Naval Postgraduate School. Available at:`https://upload.wikimedia.org/wikipedia/commons/f/f3/CYBER_AUTOMATED_RED_TEAM_TOOL_%28IA_cyberautomatedre1094564145%29.pdf`.

Fadziso, T., Thaduri, U., Dekkati, S., Ballamudi, V., and Desamsetti, H. (2023). Evolution of the cyber security threat: an overview of the scale of cyber threat. *Digitalization & Sustainability Review*, 3(1):1–12. Avaialble at:`https://upright.pub/index.php/dsr/article/view/79`.

Force, J. T. (2018). Risk management framework for information systems and organizations. *NIST Special Publication*, 800:37. DOI: 10.6028/nist.sp.800-37r2.

Jones, K. S., Namin, A. S., and Armstrong, M. E. (2018). The core cyber-defense knowledge, skills, and abilities that cybersecurity students should learn in school: Results from interviews with cybersecurity professionals. *ACM Transactions on Computing Education (TOCE)*, 18(3):1–12. DOI: 10.1145/3152893.

Kirakowski, J. and Corbett, M. (1993). Sumi: The software usability measurement inventory. *British journal of educational technology*, 24(3):210–212. DOI: 10.1111/j.1467-8535.1993.tb00076.x.

Korneta, P. (2014). What makes customers willing to recommend a retailer-the study on roots of positive net promoter score index. *Central European Review of Economics & Finance*, 5(2):61–74. Available at:`https://yadda.icm.edu.pl/cejsh/element/bwmeta1.element.cejsh-element-000171322089`.

Laxmi Kowta, A. S., Bhowmick, K., Kaur, J. R., and Jeyanthi, N. (2021). Analysis and overview of information gathering

& tools for pentesting. In *2021 International Conference on Computer Communication and Informatics (ICCCI)*, pages 1–13. DOI: 10.1109/ICCCI50826.2021.9457015.

Lee, S. (2018). Net promoter score: Using nps to measure it customer support satisfaction. In *Proceedings of the 2018 ACM SIGUCCS Annual Conference*, pages 63–64. DOI: 10.1145/3235715.3235752.

Lewis, J. R. (2018). The system usability scale: past, present, and future. *International Journal of Human–Computer Interaction*, 34(7):577–590. DOI: 10.1080/10447318.2018.1455307.

Mandal, P. C. (2014). Net promoter score: a conceptual analysis. *International Journal of Management Concepts and Philosophy*, 8(4):209–219. DOI: 10.1504/ijmcp.2014.066899.

Mohurle, S. and Patil, M. (2017). A brief study of wannacry threat: Ransomware attack 2017. *International journal of advanced research in computer science*, 8(5):1938–1940. DOI: 10.26483/ijarcs.v8i5.4021.

Najera-Gutierrez, G. and Ansari, J. A. (2018). *Web Penetration Testing with Kali Linux: Explore the methods and tools of ethical hacking with Kali Linux*. Packt Publishing Ltd. Book.

Nielsen, J. (1995). How to conduct a heuristic evaluation. *Nielsen Norman Group*, 1:1–8. Available at:`https://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/`.

Nielsen, J. *et al*. (2012). Usability 101: Introduction to usability. Available at:`https://www.nngroup.com/articles/usability-101-introduction-to-usability/`.

Nmap Public Source License, G. G. (2025). Nmap: Discover your network. Available at:`https://nmap.org/`.

OWASP Amass Project, A. . L. (2025). Owasp amass project. Available at:`https://owasp.org/www-project-amass/`.

OWASP Foundation (2021). *OWASP Top 10:2021*. OWASP Foundation. Available at:`https://owasp.org/www-project-top-ten/`. Acessado em 04 de Junho de 2024.

Probely, S. H. P. (2025). Security headers powered by probely. Available at:`https://securityheaders.com/` Accessed at 02/05/2025.

Ras, Z. W., Tarnowska, K. A., Kuang, J., Daniel, L., and Fowler, D. (2017). User friendly nps-based recommender system for driving business revenue. In *International Joint Conference on Rough Sets*, pages 34–48. Springer. DOI: 10.1007/978-3-319-60837-2_4.

Schrepp, M. (2015). User experience questionnaire handbook. *All you need to know to apply the UEQ successfully in your project*. Available at:`https://www.ueq-online.org/Material/Handbook.pdf`.

Sherlock, C. S. D. M. (2025). Sherlock project. Available at:`https://sherlockproject.xyz/` Accessed at 02/05/2025.

Shodan (2025). Shodan monitor - search engine for the internet of everything. Available at:`https://www.shodan.io/`. Accessed at 02/05/2025.

Sn1per, L. A. E. (2025). Sn1per security. Available at:`https://sn1persecurity.com/wordpress/`.

SpiderFoot, M.-l. (2025). Spiderfoot: Attack surface exposure. Available at:`https://intel471.com/solutions/attack-surface-exposure`.

Stenberg, D. (2025). command line tool and library for transferring data with urls (since 1998). Available at:`https://curl.se/`.

Stuttard, D. and Pinto, M. (2011). *The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws*. Wiley. Book.

Tabatabaei, F. and Wells, D. (2017). Osint in the context of cyber-security. *Open Source Intelligence Investigation: From Strategy to Implementation*, pages 213–231. DOI: 10.1007/978-3-319-47671-1_14.

TheHarvester, C. M. (2025). Theharvester. Available at:`https://www.kali.org/tools/theharvester/`. Accessed at 02/05/2025.

urbanadventurer aka Andrew Horton and GPLv, B. C. L. (2025). What is that website. next generation web scanner. identify the technology stack that powers a website and explore the web of things. Available at:`https://morningstarsecurity.com/research/whatweb`.

Walker, M. (2013). *Certified Ethical Hacker Practice Exams*. McGraw-Hill Osborne Media. Book.

Weidman, G. (2014). *Penetration Testing: A Hands-on Introduction to Hacking*. Novatec. Book.