A

MINOR PROJECT REPORT ON

Persistent Cookie Analyzer with a Graphical User Interface

Submitted in partial fulfillment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING (CYBER SECURITY)

BY

CH. AKHIL SAMPATH VINAY (21P61A6209)

Under the esteemed guidance of

DR. P. SUSHMA
Professor, Dept of CSE(CS)

Department of Computer Science and Engineering (Cyber Security)
VIGNANA BHARATHI INSTITUTE OF TECHNOLOGY
(Approved by AICTE, Accredited by NBA, NAAC, Permanently Affiliated to JNTU)
Aushapur (v), Ghatkesar (m), Medchal.dist, TELANGANA-501301

Academic Year 2024-2025

Disclaimer:

This project report, titled “Persistent Cookie Analyzer with a Graphical User Interface”, including its content, source code, diagrams, and documentation, is an original academic work submitted in partial fulfillment of the requirements for the degree of Bachelor of Technology in Computer Science and Engineering (Cyber Security) at Vignana Bharathi Institute of Technology.

All contents of this project are protected under the Indian Copyright Act, 1957 and applicable international copyright laws. No part of this work may be reproduced, distributed, or used for commercial purposes without prior written permission from the author.

The project may include software or components licensed under open-source licenses such as the MIT License and Creative Commons Attribution-NonCommercial 4.0 (CC BY-NC 4.0). Such components are used in accordance with their respective terms and conditions. Any external libraries, tools, or frameworks used are credited appropriately within the documentation.

A

MINOR PROJECT REPORT ON

Persistent Cookie Analyzer with a Graphical User Interface

Submitted in partial fulfillment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING (CYBER SECURITY)

BY

CH. AKHIL SAMPATH VINAY (21P61A6209)

Under the esteemed guidance of

DR. P. SUSHMA
Professor, Dept of CSE(CS)

Department of Computer Science and Engineering (Cyber Security)
VIGNANA BHARATHI INSTITUTE OF TECHNOLOGY
(Approved by AICTE, Accredited by NBA, NAAC, Permanently Affiliated to JNTU)
Aushapur (v), Ghatkesar (m), Medchal.dist, TELANGANA-501301
Academic Year 2024-2025

VIGNANA BHARATHI Institute of Technology
AUTONOMOUS Under UGC
Accredited by NBA NATIONAL BOARD of ACCREDITATION (ECE,EEE,CSE,IT,CE) NAAC NATIONAL ASSESSMENT AND ACCREDITATION COUNCIL A Grade

AUSHAPUR (V), GHATKESAR (M), MEDCHAL, DIST-501 301
Department of Computer Science and Engineering
(Cyber Security)

CERTIFICATE

This is to certify that the project entitled **"Persistent Cookie Analyzer with a Graphical User Interface"** is being submitted by **Ch. Akhil Sampath Vinay (21P61A6209),** in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology in Computer Science and Engineering (Cyber Security)** is a record of bonafide work carried out by them under my guidance and supervision during the academic year 2024-2025. The results embodied in this project report have not been submitted to any other University for the award of any degree or diploma.

Internal Guide
Dr. P. Sushma
Professor
Department of CSE(CS)

Head of the Department
Dr. P. Sushma
Professor
Department of CSE (CS)

Project Coordinator
Mrs. V. Madhumala
Mrs. V. Raja Lakshmi
Associate Professor
Department of CSE (CS)

External Examiner

VIGNANA BHARATHI Institute of Technology

AUTONOMOUS Under UGC

Accredited by NBA NATIONAL BOARD OF ACCREDITATION (ECE,EEE,CSE,IT,CE) NAAC NATIONAL ASSESSMENT AND ACCREDITATION COUNCIL A Grade

AUSHAPUR (V), GHATKESAR (M), MEDCHAL.DIST-501 301

Department of Computer Science and Engineering

(Cyber Security)

DECLARATION

I, **CH. AKHIL SAMPATH VINAY** bearing hall ticket number **21P61A6209,** hereby declares that the project report entitled **"Persistent Cookie Analyzer with a Graphical User Interface"** under the guidance of Dr. P. Sushma, Professor, Department of Computer Science and Engineering(Cyber Security), **VBIT**, Hyderabad, submitted in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering (Cyber Security).

This is a record of bonafide work carried out by us and the results embodied in this project have not been reproduced or copied from any source. The results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

CH. AKHIL SAMPATH VINAY (21P61A6209)

ACKNOWLEDGEMENT

First and foremost, we wish to express our gratitude towards the institution "**Vignana Bharathi Institute of Technology"** for fulfilling the most cherished goal of our life to do Bachelor of Technology.

It is a great pleasure in expressing a deep sense of gratitude to our Internal guide, **Dr. P. Sushma, Professor,** Department of Computer Science and Engineering (Cyber Security) for her valuable guidance and freedom she gave to us.

We also express our sincere thanks to **Mrs. V. Madhumala and Mrs. V. Raja Lakshmi, Associate Professor, Project Coordinator** for his encouragement and support throughout the project.

We are deeply grateful to **Dr. P. Sushma, Head of Department, Department of Computer Science and Engineering (Cyber Security)** for granting us the opportunity to conduct this project.

We take immense pleasure in thanking **Dr. P.V.S. Srinivas, Principal, Vignana Bharathi Institute of Technology**, **Ghatkesar** for having permitted us to carry out this project work.

Our utmost thanks also go to all the **FACULTY MEMBERS** and **NON - TEACHING STAFF** of the Department of Computer Science and Engineering (Cyber Security) for their support throughout our project work.

CH. AKHIL SAMPATH VINAY (21P61A6209)

ABSTRACT

The rise of web applications has revolutionized digital interactions, but it has also introduced security challenges related to persistent cookies. These cookies, essential for maintaining sessions and improving user experience, can inadvertently expose users to significant risks, including session hijacking, cross-site request forgery (CSRF), and unauthorized data access. This project addresses this gap by introducing the Cookie Analyzer Tool, a comprehensive solution for cookie risk assessment and management. The Cookie Analyzer Tool systematically captures persistent cookies stored in a browser and evaluates their security using critical attributes such as secure, httponly, samesite, expiry, and domain path. Through a detailed risk analysis, cookies are categorized based on their threat levels, providing users with actionable insights. The tool also facilitates immediate remediation by enabling the deletion of high-risk cookies with a single click and offering security recommendations to mitigate potential threats. Designed with both technical and non-technical users in mind, the tool features an intuitive interface that enhances usability and accessibility. Users can monitor newly stored cookies in real-time and receive instant notifications of potential risks, empowering them to proactively manage their privacy and security while browsing. By operating entirely on the user's device without internet connectivity, the tool ensures data confidentiality and offers a secure environment for comprehensive cookie management. In an increasingly web-centric world, the Cookie Analyzer Tool enhances transparency and control over browser cookies, bridging the gap between usability and security. By addressing the hidden risks of persistent cookies, this project contributes to improved user privacy and security, making it an essential tool for modern web browsing.

VIGNANA BHARATHI INSTITUTE OF TECHNOLOGY
Department of Computer Science and Engineering
(Cybersecurity)
COURSE OUTCOMES

Course: Major Project Class: IV B. Tech II Semester
AY: 2023-24

Course Outcomes
After completing the Projects, the student will be able to:

Code	Course Outcomes	Taxonomy
C424.1	Identify and state the problem precisely to prepare the abstract	Remember
C424.2	Analyze the existing system, and outlining the proposed methodology for effective solution	Analyze
C424.3	Use various modern tools for designing applications based on specified requirements	Apply
C424.4	Develop applications with adequate features and evaluate the application to ensure the quality	Create
C424.5	Prepare the document of the project as per the guidelines	Create

PROGRAM OUTCOMES

PO 1. Engineering knowledge:

Apply the knowledge of mathematics, science, engineering fundamentals,and an engineering specialization to the solution of complex engineering problems.

PO 2. Problem analysis:

Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO 3. Design/development of solutions:

Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and

the cultural, societal, and environmental considerations.

PO 4. Conduct investigations of complex problems:

Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO 5. Modern tool usage:

Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO 6. The engineer and society:

Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO 7. Environment and sustainability:

Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO 8. Ethics:

Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO 9. Individual and team work:

Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO 10. Communication:

Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO 11. Project management and finance:

Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO 12. Life-long learning:

Recognize the need for, and have the preparation and ability to engage in independent and

life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSO's)

PSO1: Proficiency in Cybersecurity Tools and Techniques

The development and implementation of the Persistent Cookie Analyzer demonstrate the ability to design and utilize tools for identifying potential security risks. By analyzing persistent cookies, the tool addresses cybersecurity challenges, such as privacy concerns and data misuse.

PSO2: Application of Software Development Practices

The project integrates advanced programming and GUI design practices to create an efficient and user-friendly tool. This showcases the ability to develop software solutions with a focus on real-world cybersecurity applications and usability.

PSO3: Problem-Solving in Digital Privacy and Security

The tool reflects problem-solving capabilities by addressing specific issues related to persistent cookies, such as identifying harmful cookies and assessing their safety. The ability to analyze cookies' metadata and expiration enhances privacy and security measures.

VIGNANA BHARATHI INSTITUTE OF TECHNOLOGY
Department of Computer Science and Engineering
(Cybersecurity)

COs Mapping with PO/PSP

Project Title: Persistent Cookie Analyzer with a Graphical User Interface

Name of the Supervisor: Dr. P. Sushma

Batch Details:

S.No	Regd No.	Student Name	Technology
01.	21P61A6209	CH. AKHIL SAMPATH VINAY	CYBER SECURITY
02.	21P61A6216	G. TRISHA REDDY	CYBER SECURITY
03.	21P61A6255	S. PRAVEEN	CYBER SECURITY

Note: Write your domain name in technology field (ex. ML, IOT, BC, Security, Cloud etc.)

CO-PO Mapping for Major Project:
High -3 Medium -2 Low-1

PO/CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
C424.1	3	3	3	3	3	-	-	3	3	3	-	3	3	3	-
C424.2	3	3	3	3	3	-	3	3	3	3	-	3	3	3	3
C424.3	3	3	3	3	3	3	3	3	3	3	-	3	3	-	3
C424.4	3	3	3	3	3	3	3	3	3	3	-	3	3	3	3
C424.5	3	3	3	3	3	3	3	3	3	3	-	3	3	3	3
AVG	3	3	3	3	3	3	3	3	3	3	-	3	3	3	3

Guide Signature

INDEX

LIST OF FIGURES **PAGE NUMBER**

LIST OF TABLES

-

CHAPTER 01

INTRODUCTION

1. INTRODUCTION

Cookies are small text files stored on a user's device by websites to manage sessions, track user activity, and personalize online experiences. While they play a vital role in modern web browsing, cookies can also introduce significant privacy and security risks if mismanaged. These risks range from exposing sensitive user information to enabling unauthorized tracking across websites. As the digital landscape evolves, it becomes increasingly important to assess and manage cookies effectively to ensure both user safety and compliance with global privacy standards.

The Persistent Cookie Analyzer addresses this need by offering a comprehensive solution for cookie evaluation and management. This project is designed to provide users with a deep understanding of the cookies stored in their web browsers, particularly focusing on Google Chrome profiles. By extracting and analyzing cookies, the tool identifies potential vulnerabilities such as the absence of Secure and HttpOnly flags, improper SameSite configurations, and cookies that persist beyond reasonable limits.

In addition to assessing security risks, the tool evaluates cookies for compliance with regulations like the General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA). These privacy laws mandate strict handling of user data, and cookies are a critical aspect of this compliance. The Persistent Cookie Analyzer empowers users to identify cookies that may pose regulatory risks and take appropriate actions.

With an intuitive graphical interface, the tool simplifies cookie management by allowing users to filter, select, and delete cookies directly. It also generates detailed reports, making it a valuable resource for cybersecurity professionals, developers, and privacy-conscious individuals. By bridging the gap between technical analysis and user-friendly design, the Persistent Cookie Analyzer ensures that cookie management is accessible to both experts and everyday users.

1.1 MOTIVATION

In today's digital ecosystem, cookies play a pivotal role in enabling seamless web interactions, personalizing user experiences, and supporting online analytics. However, the widespread use of cookies has introduced significant challenges in privacy and security. Misconfigured or insecure cookies are frequently exploited for cyberattacks, such as session hijacking and cross-site scripting (XSS). Additionally, the lack of transparency and user awareness about cookie risks amplifies vulnerabilities and contributes to data misuse. This necessitates a solution that addresses both security and compliance concerns effectively.

The **Persistent Cookie Analyzer** is motivated by the need to bridge the gap between user accessibility and technical sophistication in cookie management. Many end-users and small organizations lack the tools or expertise to analyze cookie behavior comprehensively, leaving them exposed to risks. While web browsers offer limited cookie visibility, they fail to highlight vulnerabilities or regulatory compliance. This project seeks to empower users with actionable insights to assess cookie configurations, identify risks, and make informed decisions.

Another driving force behind this tool is the growing importance of data protection laws, such as the GDPR and CCPA. These regulations impose strict guidelines on how cookies are used and stored, with non-compliance resulting in severe penalties for organizations. The tool's compliance-checking features help users and businesses align with these legal requirements, reducing potential liabilities while enhancing trust.

Beyond compliance and security, this project is inspired by the broader mission of promoting cybersecurity awareness. By simplifying cookie analysis and offering a user-friendly interface, the tool encourages users to explore and understand cookie behavior, fostering proactive privacy and security practices. Empowering individuals to take control of their digital footprint aligns with the evolving demands for ethical and transparent data handling.

In summary, the **Persistent Cookie Analyzer** addresses the urgent need for robust cookie management solutions that balance technical depth with accessibility. By mitigating vulnerabilities, supporting regulatory compliance, and enhancing user

understanding, the tool contributes to a safer and more privacy-conscious digital landscape.

1.1.1 OVERVIEW OF EXISTING SYSTEM

The current landscape of cookie management tools and practices reveals a mix of capabilities and limitations. Most web browsers, such as Google Chrome, Mozilla Firefox, and Microsoft Edge, offer built-in mechanisms to manage cookies. These mechanisms allow users to view, delete, or block cookies on a site-by-site basis. While helpful, these tools are generally basic and do not provide detailed insights into the risks or compliance status of cookies. They focus primarily on user convenience and privacy settings rather than comprehensive analysis.

Additionally, browser-based cookie management often lacks depth in assessing the technical attributes of cookies, such as their security flags (e.g., Secure, HttpOnly, SameSite) and expiration parameters. This leaves users and even some administrators unaware of potential vulnerabilities that could be exploited by attackers. For example, cookies without the Secure flag are susceptible to interception over unencrypted HTTP connections, and those without HttpOnly are exposed to JavaScript-based attacks.

In the organizational context, advanced cookie management tools are available, but these are often part of broader enterprise solutions. Examples include web application firewalls (WAFs) and data protection platforms that provide cookie scanning as a feature. However, such tools are costly, complex, and tailored for larger enterprises, making them inaccessible to individual users and small to medium-sized organizations.

From a compliance perspective, current systems provide limited support for addressing regulatory frameworks like the GDPR and CCPA. While some websites offer cookie consent banners to meet basic requirements, these implementations are often superficial and fail to ensure that all cookies comply with legal standards. This gap exposes organizations to the risk of non-compliance penalties and reputational damage.

Despite these tools, the existing systems lack a unified solution that integrates risk assessment, privacy compliance checks, and user-friendly features. Most solutions address only one aspect—security, compliance, or visibility—without holistically

empowering users to take control of cookie management. This fragmentation creates a pressing need for a comprehensive, accessible, and affordable tool that addresses the limitations of existing systems.

1.1.2 DRAWBACKS OF EXISTING SYSTEM

Despite the availability of various cookie management tools and practices, the existing systems have notable shortcomings that limit their effectiveness and usability. Some key drawbacks include:

1. Limited Risk Assessment: Most tools, including those integrated into web browsers, fail to evaluate cookies for potential security risks. They do not analyze critical attributes such as Secure, HttpOnly, or SameSite flags in-depth, leaving users unaware of vulnerabilities like cross-site scripting (XSS) or session hijacking.

2. Lack of Compliance Analysis: Current cookie management solutions do not adequately address legal and regulatory requirements like GDPR, CCPA, or other data protection laws. While cookie consent banners are commonly implemented, they often fail to ensure that cookies adhere to compliance standards, increasing the risk of penalties for non-compliance.

3. Insufficient User Control: Browser-based cookie tools offer limited functionality, such as enabling or disabling cookies globally or per site. They do not provide granular control for analyzing, selecting, or removing specific cookies based on risk or compliance factors.

4. Absence of Detailed Reports: Existing systems rarely generate comprehensive reports on cookie activity, security, and compliance. This lack of documentation prevents users and organizations from having a clear overview of their cookie-related vulnerabilities and areas of improvement.

5. High Complexity and Cost in Enterprise Tools: Advanced cookie management solutions tailored for organizations are often expensive and complex to deploy. These tools are designed for large-scale enterprise environments, making them inaccessible to small businesses, independent developers, and individual users.

6. Reactive Rather than Proactive Approach: Most tools only allow users to delete or block cookies after they have been set, rather than proactively identifying and mitigating risks associated with cookies before they are utilized.

7. Inadequate User Education: Current systems do not educate users on the technical implications of cookies or their potential security and privacy impacts. This knowledge gap prevents informed decision-making and leaves users vulnerable to exploitation.

8. Fragmented Solutions: No single tool offers an integrated approach to cookie management, combining features like risk scoring, compliance checks, and secure deletion in an intuitive interface. Users often need to rely on multiple tools, which leads to inefficiency and inconsistencies.

1.1.3 OVERVIEW OF PROPOSED SYSTEM

The **Persistent Cookie Analyzer with a Graphical User Interface** aims to address the drawbacks of existing systems by offering a comprehensive, user-friendly solution for analyzing, managing, and securing cookies. This system integrates advanced features designed to enhance the security, compliance, and usability of cookie management processes.

Key Features of the Proposed System:

1. Comprehensive Risk Assessment: The tool analyzes live HTTP cookies, evaluating attributes such as Secure, HttpOnly, SameSite, expiration time, and domain information. It identifies vulnerabilities and assigns a risk score to each cookie, enabling users to understand the potential security implications.

2. **Privacy Compliance Checker:** By cross-referencing cookies against data protection regulations like GDPR and CCPA, the tool determines compliance status. It provides actionable feedback to help organizations adhere to legal and ethical standards, reducing the risk of non-compliance penalties.

3. **Detailed Reporting:** The system generates a thorough PDF report that includes cookie vulnerabilities, risk scores, and compliance assessments. This feature benefits

organizations by offering clear documentation for audits, regulatory submissions, or internal evaluations.

4. **Persistent Cookie Detection:** The tool identifies and flags persistent cookies that may retain user data for extended periods, potentially violating privacy norms or creating security risks.

5. **Proactive Cookie Management:** Unlike existing systems, this tool empowers users to make informed decisions proactively. It suggests necessary actions to mitigate risks, such as enabling critical cookie attributes or removing high-risk cookies.

6. **User-Friendly Interface:** The system is designed to be intuitive, catering to technical and non-technical users alike. With a clear dashboard and easy-to-navigate features, the tool simplifies the complexities of cookie management.

7. **Real-Time Analysis:** By analyzing cookies in real time, the system ensures immediate detection of vulnerabilities and compliance issues as users interact with websites.

8. **Cost-Effective Solution:** The proposed tool is affordable and accessible, addressing the needs of both individual users and small to medium-sized organizations. It eliminates the high costs and complexities associated with enterprise-grade tools.

Benefits of the Proposed System:

1. Enhances user and organizational understanding of cookie behavior and risks.
2. Bridges the gap between security and regulatory compliance.
3. Provides an all-in-one solution for cookie management, eliminating the need for multiple tools.
4. Ensures a safer and more privacy-conscious web browsing experience.

1.2 PROBLEM DEFINITION

In the modern digital landscape, websites extensively use cookies to store user data, track activities, and enhance user experiences. While cookies are essential for functionality and personalization, they also pose significant risks to privacy and security. These risks include unauthorized access to sensitive data, exposure to tracking, and non-compliance with privacy regulations like GDPR and CCPA. Existing cookie management solutions often fall short due to their limited capabilities, lack of real-time analysis, and inadequate focus on regulatory compliance.

Organizations and users face several challenges:

1. **Lack of Comprehensive Analysis:** Existing tools fail to provide a detailed assessment of cookie attributes, such as HttpOnly, Secure, SameSite, and expiration details, leaving critical vulnerabilities undetected.

2. **Insufficient Focus on Privacy Compliance:** Ensuring adherence to privacy laws like GDPR and CCPA requires advanced checks, which most systems lack. Non-compliance can result in legal penalties and loss of user trust.

3. **Persistent Cookies and Data Retention Risks:** Many websites employ persistent cookies, which retain user data for extended periods, potentially violating user privacy and increasing security risks.

4. **Manual and Inefficient Processes:** Manual cookie analysis is time-consuming and prone to errors. Organizations struggle to keep up with the rapidly evolving regulatory landscape and cybersecurity threats.

5.**Lack of Reporting Mechanisms:** Without detailed and actionable reports, organizations cannot document or resolve cookie-related vulnerabilities effectively.

6. **User Accessibility Challenges:** Existing solutions often cater only to technically proficient users, making it challenging for non-technical individuals or small organizations to manage cookies effectively.

1.3 OBJECTIVE OF PROJECT

The primary objective of this project, the **Persistent Cookie Analyzer with a Graphical User Interface**, is to provide an automated, comprehensive solution for analyzing and managing cookies stored by web browsers, with a focus on security and privacy. This tool aims to:

1. **Automate Cookie Extraction and Analysis:** Automatically extract cookies from Chrome browser profiles and analyze their attributes, including expiration, security flags (Secure, HttpOnly), SameSite settings, and more. This process helps identify potential security vulnerabilities and privacy risks associated with each cookie.

2. **Risk Assessment and Scoring:** Assess each cookie's risk by evaluating its compliance with industry standards such as OWASP and NIST, as well as privacy regulations like GDPR and CCPA. The tool assigns a risk score to each cookie based on factors such as missing security flags, improper settings, and potential for abuse.

3. **Privacy and Compliance Checking:** Ensure that cookies meet the requirements of privacy regulations such as GDPR and CCPA by checking their compliance with essential attributes like data retention period, third-party access, and transparency.

4. **User-Friendly Interface for Non-Technical Users:** Provide an intuitive graphical user interface (GUI) that allows both technical and non-technical users to view, filter, and manage cookies effortlessly. The tool enables users to toggle cookies for deletion, apply filters based on active status or risk score, and even generate detailed reports for compliance auditing.

5. **Persistent Cookie Detection and Management:** Identify persistent cookies that store data for long periods and potentially violate user privacy by retaining unnecessary information. Allow users to delete or manage these cookies according to their security preferences.

6. **Comprehensive Reporting:** Generate detailed PDF reports summarizing the cookies analyzed, their associated risks, and the status of their compliance with privacy regulations. These reports are valuable for organizations that need to track, document, and address potential risks.

1.4 SCOPE OF PROJECT

The **Cookie Analysis and Risk Assessment Tool** aims to provide a solution for analyzing and securing cookies in the Chrome web browser, focusing on privacy and cybersecurity. The tool automates cookie extraction from various Chrome profiles, simplifying a process that could otherwise be complex and time-consuming. It allows users to evaluate cookie information such as expiration time, security flags, and SameSite attributes, ensuring cookies are secure and compliant with privacy regulations.

The tool assigns risk scores to cookies based on attributes like the presence of secure flags, HttpOnly settings, and SameSite values. This feature helps users identify and manage potentially risky cookies, essential for both individuals concerned with privacy and organizations seeking to mitigate cybersecurity risks.

Additionally, the tool supports compliance checks for privacy laws such as GDPR and CCPA, assessing whether cookies comply with legal requirements related to user consent, data retention, and third-party data sharing. This is particularly useful for businesses that need to ensure their websites and applications adhere to strict privacy regulations.

The user-friendly graphical interface allows easy filtering and management of cookies, enabling users to sort cookies by their activity status or risk score and delete selected ones. For organizations, the tool can generate detailed PDF reports summarizing cookie risks and compliance, which is valuable for auditing and compliance tracking.

In summary, this project targets both individual users and organizations seeking to better understand and manage cookies. By automating the extraction, risk analysis, and compliance checks, the tool enhances online security, protects privacy, and helps maintain compliance with cybersecurity standards and regulations.

CHAPTER 02

LITERATURE SURVEY

Title of the paper: HTTP Cookies: Overview and Threats.
Authors: L. Kohnfelder, A. Smith, D. Zhang, and J. Lee

In the paper "HTTP Cookies: Overview and Threats," Kohnfelder et al. explore the critical role of HTTP cookies in modern web applications and the security risks associated with their use. The paper provides an in-depth analysis of how cookies function in web browsers, highlighting their role in user authentication, session management, and tracking. The authors focus on various threats related to cookies, including **session hijacking**, where attackers steal session cookies to impersonate users, and **Cross-Site Scripting (XSS)**, which allows malicious scripts to access and steal cookies. They also discuss **Cross-Site Request Forgery (CSRF)**, where cookies are automatically sent in requests, potentially enabling attackers to perform actions on behalf of the user. To mitigate these risks, the paper emphasizes the importance of two key security mechanisms: the **Secure flag** and the **HttpOnly flag**. The **Secure flag** ensures cookies are transmitted only over secure HTTPS connections, while the **HttpOnly flag** prevents JavaScript from accessing cookies, reducing the risk of XSS attacks. A major contribution of the paper is its discussion on the **SameSite cookie attribute**, which limits the sending of cookies in cross-origin requests, helping prevent CSRF attacks. This research is pivotal for understanding cookie-related security vulnerabilities and is directly applicable to the development of tools like the Cookie Analysis and Risk Assessment Tool, which helps developers detect and address such vulnerabilities in cookies, promoting safer web applications.

Title of the paper: The Security of Cookies: Identifying and Mitigating Risks
Authors: H. P. Singh and A. K. Gupta

In the paper "The Security of Cookies: Identifying and Mitigating Risks," Singh and Gupta delve into the various security vulnerabilities associated with HTTP cookies and propose strategies for mitigating these risks. The paper highlights the widespread use of cookies in web applications for session management, user authentication, and personalization, while also emphasizing the potential threats posed by insecure cookie practices. The authors identify key security risks, including **session fixation**, where attackers can set a session ID for the victim before the session begins, and **cookie theft** through man-in-the-middle attacks, where an attacker intercepts cookies during transmission. They also discuss the risk of **cookie poisoning**, where malicious users can manipulate cookie data to compromise web applications. To mitigate these risks, the paper proposes several measures, such as the implementation of the **Secure flag** to ensure cookies are only sent over HTTPS, the use of the **HttpOnly flag** to prevent access to cookies via JavaScript, and the adoption of **token-based authentication** to strengthen session security. Additionally, the authors recommend utilizing **encryption** and **hashing** techniques to protect sensitive cookie data and **input validation** to prevent cookie manipulation. The paper also discusses the importance of **auditing** cookie usage

regularly to ensure compliance with security best practices. This research is crucial for understanding the different threats that cookies face and provides a foundation for building robust cookie security mechanisms, making it highly relevant to tools aimed at analyzing cookie risks, such as the Cookie Analysis and Risk Assessment Tool.

Title of the paper: Cookie Consent Mechanisms in Web Development.
Authors: R. A. Anderson and J. D. Smith

In their paper "Cookie Consent Mechanisms in Web Development," Anderson and Smith explore the importance of obtaining user consent before collecting cookies in accordance with global privacy regulations like the **General Data Protection Regulation (GDPR)** and the **California Consumer Privacy Act (CCPA)**. The paper discusses the evolution of cookie consent mechanisms and the growing emphasis on **user privacy** in the digital era.The authors highlight the challenges that developers face when implementing cookie consent banners on websites, especially in ensuring that they are both **compliant with regulations** and **user-friendly**. They examine various consent models, such as the **explicit consent model** (where users must actively agree to cookie usage) and the **implicit consent model** (where consent is assumed by default unless the user opts out). The paper also provides insights into the design and functionality of **cookie consent banners** that ensure compliance with laws while maintaining a seamless user experience. The paper emphasizes the need for clear and concise **cookie policies** that inform users about the types of cookies being used, their purposes, and how to manage cookie preferences. The authors stress the importance of offering users the ability to **accept**, **reject**, or **customize** cookie settings. They also discuss the technical aspects of implementing consent mechanisms, including the **storage of consent records** and the **use of cookie categorization** to allow users to opt into specific types of cookies. This research is relevant to the development of privacy-compliant tools like the Cookie Analysis and Risk Assessment Tool, as it underscores the necessity of integrating effective cookie consent mechanisms that respect user privacy and comply with international regulations.

Title of the paper: Evaluating Web Security: A Comprehensive Study of Cookie Handling and Security Best Practices
Authors: T. S. Lin and G. M. Lee

In their 2018 paper "Evaluating Web Security: A Comprehensive Study of Cookie Handling and Security Best Practices," Lin and Lee offer an in-depth analysis of **cookie security** and the role of cookies in modern web applications. They investigate common vulnerabilities associated with cookies and the **security risks** posed by improper cookie handling in web applications. The paper highlights key issues, such as **cookie theft**, **session hijacking**, and **cross-site scripting (XSS)** attacks, which exploit insecure cookie configurations. The authors stress the importance of **secure attributes** like Secure, HttpOnly, and SameSite flags in preventing these attacks. They discuss how **session cookies** can be particularly vulnerable if not managed properly, especially in cases where cookies are transmitted over **unencrypted channels** or without **appropriate expiration times**. The paper also emphasizes the importance of following **security best practices** when setting and handling cookies in web applications. These practices include using **strong encryption** for sensitive cookie data, implementing **secure cookie storage mechanisms**, and enforcing **proper cookie expiration policies**. The authors provide a comprehensive guide on how to implement these measures and how developers can test and audit their web applications for **cookie-related vulnerabilities**. This study is significant for the development of tools like the Cookie Analysis and Risk Assessment Tool, as it outlines key security issues that must be addressed when analyzing cookies for compliance and risk management. By incorporating the recommendations from this paper, developers and security professionals can better safeguard their web applications against cookie-related vulnerabilities and ensure compliance with **security best practices**.

CHAPTER 03

SYSTEM ANALYSIS

3. SYSTEM ANALYSIS

3.1 SYSTEM ARCHITECTURE:

3.1.1 ARCHITECTURE DIAGRAM

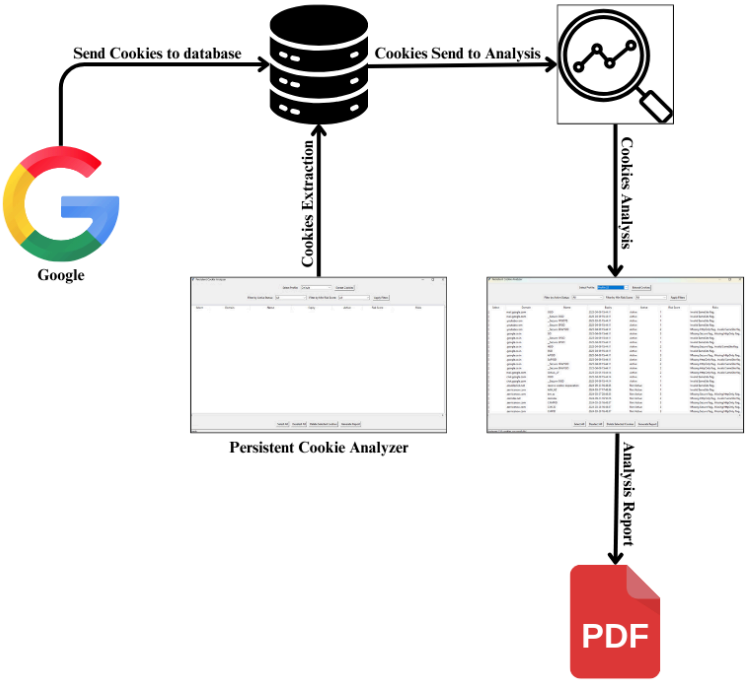


Fig 3.1.1 System Architecture

3.1.2 SYSTEM FLOW CHART

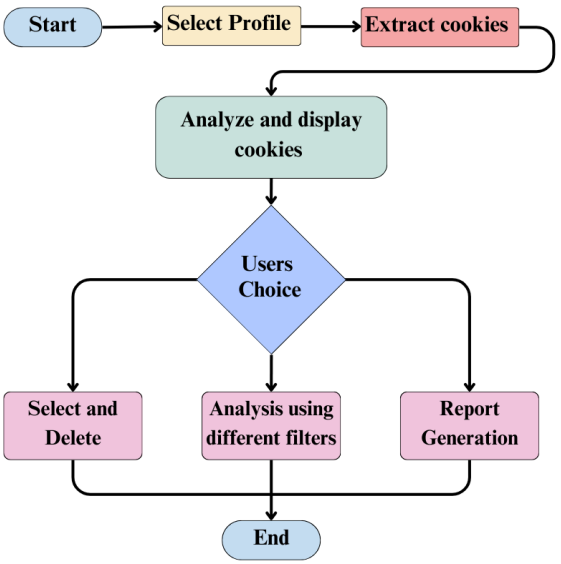


Fig 3.1.2 System Flow Chart

3.2 OPERATING REQUIREMENTS

3.2.1 HARDWARE REQUIREMENTS

1. **Processor:** Intel Core i3 or higher
2. **RAM:** Minimum 4GB
3. **Storage:** Minimum 1GB
4. **Monitor:** Full HD (1920x1080) resolution

3.2.2 SOFTWARE REQUIREMENTS

1. **Operating System:** Windows 7
2. **Python:** Version 3.6+ for scripting and automation
3. **Libraries and Frameworks:** Pandas, Sqlite3, FPDF, Os, Tinkter
4. **Web Browser:** Chrome
5. **Pdf Viewer**

3.3 SYSTEM ANALYSIS

System analysis is a critical phase in the software development lifecycle, focusing on understanding and documenting the requirements of the system. For the Persistent Cookie Analyzer with a Graphical User Interface (GUI), the analysis phase identifies system functionality, user needs, and potential challenges. In this context, the system aims to analyze cookies from a browser profile, identify vulnerabilities, and offer tools for security and compliance.

3.4 REQUIREMENTS ANALYSIS

The first step in system analysis involves gathering both functional and non-functional requirements. Functional requirements for this project include the ability to extract cookies from the browser, analyze cookie details, check the cookies for compliance with privacy regulations (such as GDPR), and allow users to delete or manage cookies. Non-functional requirements involve ensuring the system is fast, reliable, and user-friendly.

1. **Functional Requirements:**
   - 1.1. **Cookie Extraction:** The system should be able to extract cookies from the browser (e.g., Chrome) profiles.
   - 1.2. **Cookie Display:** The system should list cookies, showing details such as domain, name, expiry date, and any risk associated with each cookie.
   - 1.3. **Risk Scoring and Compliance Checks:** The system should assess the security risk level and privacy compliance of each cookie based on established standards.
   - 1.4. **Cookie Management:** Users should have the ability to delete selected cookies or update the cookie list.

1.5. **Reporting:** The system should be able to generate detailed reports in PDF format, including the analysis of cookie risks and compliance status.

2. **Non-Functional Requirements:**
   2.1. **User Interface:** The GUI should be intuitive and user-friendly, making it easy for users to navigate the features of the system.
   2.2. **Performance:** The system should handle large numbers of cookies efficiently and be responsive during operations such as fetching, filtering, and reporting.
   2.3. **Reliability:** The system should be stable, with minimal risk of crashes during the execution of its core functions.
   2.4. **Security:** The system should be designed with strong data privacy protection, ensuring that sensitive information like cookies is handled securely.

3.5 FEASIBILITY ANALYSIS

Feasibility analysis assesses the practicality of the system in terms of technological, operational, and financial resources.

1. **Technological Feasibility:**
   1.1. The project uses Python, along with libraries such as os, json, and PyQt5 for GUI development. This combination is well-suited for building an efficient and functional application. Cookie extraction is done through parsing Chrome's local data files, which is a straightforward approach for the given use case.
   1.2. There are no major technological barriers in implementing this system, as Python libraries provide robust support for data extraction, analysis, and GUI creation.
2. **Operational Feasibility:**
   2.1. The system operates by extracting cookies from the local browser profile of a user. Since most users already store cookies locally on their systems, the tool's operation is compatible with their daily workflow.
   2.2. A major challenge may arise if the user has an unfamiliar browser profile setup or if certain cookies are encrypted, in which case additional error handling or user guidance would be required.
3. **Financial Feasibility:**
   3.1. The project relies on open-source tools and libraries, reducing the cost significantly. As it doesn’t require expensive software or infrastructure, it is financially feasible to implement and distribute.

3.6 DATA FLOW AND PROCESS FLOW ANALYSIS

The data flow analysis examines how data moves through the system from start to finish:

1. **Data Flow:**
   - 1.1. The system starts by extracting cookies from a selected Chrome profile, storing them in memory.
   - 1.2. The cookies are then displayed to the user, with all relevant information (e.g., expiry date, domain).
   - 1.3. The user can filter cookies based on risk scores or active status and choose to delete or manage them.
   - 1.4. After any modifications, the system generates a PDF report summarizing the analysis and actions taken on cookies.
2. **Process Flow:**
   - 2.1. The process begins when the user selects a browser profile and clicks the "Extract Cookies" button.
   - 2.2. The system fetches cookies, displays them in the GUI, and applies any filters the user requests.
   - 2.3. The user selects cookies for deletion and clicks the "Delete" button.
   - 2.4. A report is generated based on the selected cookies, displaying risks and compliance scores.

CHAPTER 04

SYSTEM DESIGN

4. SYSTEM DESIGN

4.1 UML DIAGRAMS

UML stands for Unified Modelling Language. UML is a standardized general-purpose modelling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML comprises two major components: a Meta- model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modelling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software systems, as well as for business modelling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS

1. The Primary goals in the design of the UML are as follows:

2. Provide users a ready-to-use, expressive visual modelling Language so that they can develop and exchange meaningful models.

3. Provide extendibility and specialization mechanisms to extend the core concepts.

4. Be independent of particular programming languages and development processes.

5. Provide a formal basis for understanding the modelling language.

6. Encourage the growth of the OO tools market.

7. Support higher level development concepts such as collaborations, frameworks, patterns and components

TYPES OF UML DIAGRAM

Each UML diagram is designed to let developers and customers view a software system from a different perspective and in varying degrees of abstraction. UML diagrams commonly created in visual modelling tools include:

1. Class Diagram: Shows the static structure of a system by depicting classes, attributes, operations, and relationships between classes.

2. Object Diagram: Represents instances of classes and their relationships at a specific point in time, providing a snapshot of the system.

3. Use Case Diagram: Illustrates the interactions between actors (users or external systems) and the system to capture functional requirements.

4. Sequence Diagram: Describes how objects interact in a particular scenario, showing the sequence of messages exchanged between objects.

5. Collaboration Diagram (Communication Diagram): Similar to a sequence diagram but focuses on the interactions between objects and their links.

6. Activity Diagram: Represents the flow of control or activities within a system, showing actions, decisions, and transitions between states.

7. State Diagram: Depicts the lifecycle of an object, specifying different states an object can be in and the events that trigger state transitions.

8. Component Diagram: Illustrates the components of a system and their dependencies, showing how the system is structured at a high level.

9. Deployment Diagram: Shows the physical deployment of software components on hardware nodes, including servers, devices, and networks.

10. Package Diagram: Organizes and shows dependencies between packages in a system, providing a high-level view of the system's structure.

These diagrams are used in software development to model, visualize, and document various aspects of a system, aiding in communication and design.

A. USE CASE DIAGRAM

The **Persistent Cookie Analyzer**'s use case diagram illustrates the interactions between the system and its users. The primary actor, the user, interacts with the system through the graphical user interface to perform various actions, such as uploading a file or entering a URL to scan for cookies. The system identifies persistent cookies, analyzes them for potential risks, and provides actionable insights. Additional use cases include generating reports and displaying cookie attributes, such as duration and location, ensuring a user-friendly and secure experience.

In addition to the primary interactions, the use case diagram for the Persistent Cookie Analyzer also includes other supporting functionalities such as logging, error handling, and settings management. These interactions ensure that the system can handle edge cases like invalid inputs or errors during the analysis process. Furthermore, the system maintains a seamless integration with external data sources and provides users with options to customize their scanning preferences, enhancing flexibility and user control. The diagram captures these additional aspects, demonstrating a comprehensive user-system relationship.

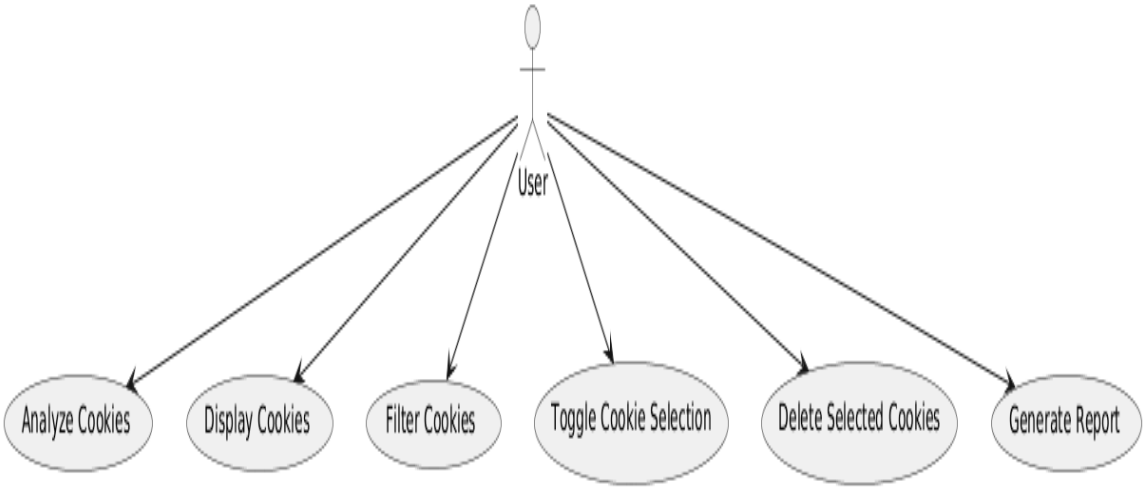


Fig 4.1 Use Case Diagram

B. CLASS DIAGRAM

The **Class Diagram** for the Persistent Cookie Analyzer showcases the structural design of the system. Core classes include PersistentCookieAnalyzer for handling the analysis logic, Cookie for encapsulating attributes like name, expiry, and location, and GUIManager for managing user interactions. Relationships between these classes highlight the dependencies, such as the GUI class invoking the analyzer's methods to process data and display results, ensuring a seamless integration of analysis functionality with the graphical interface.

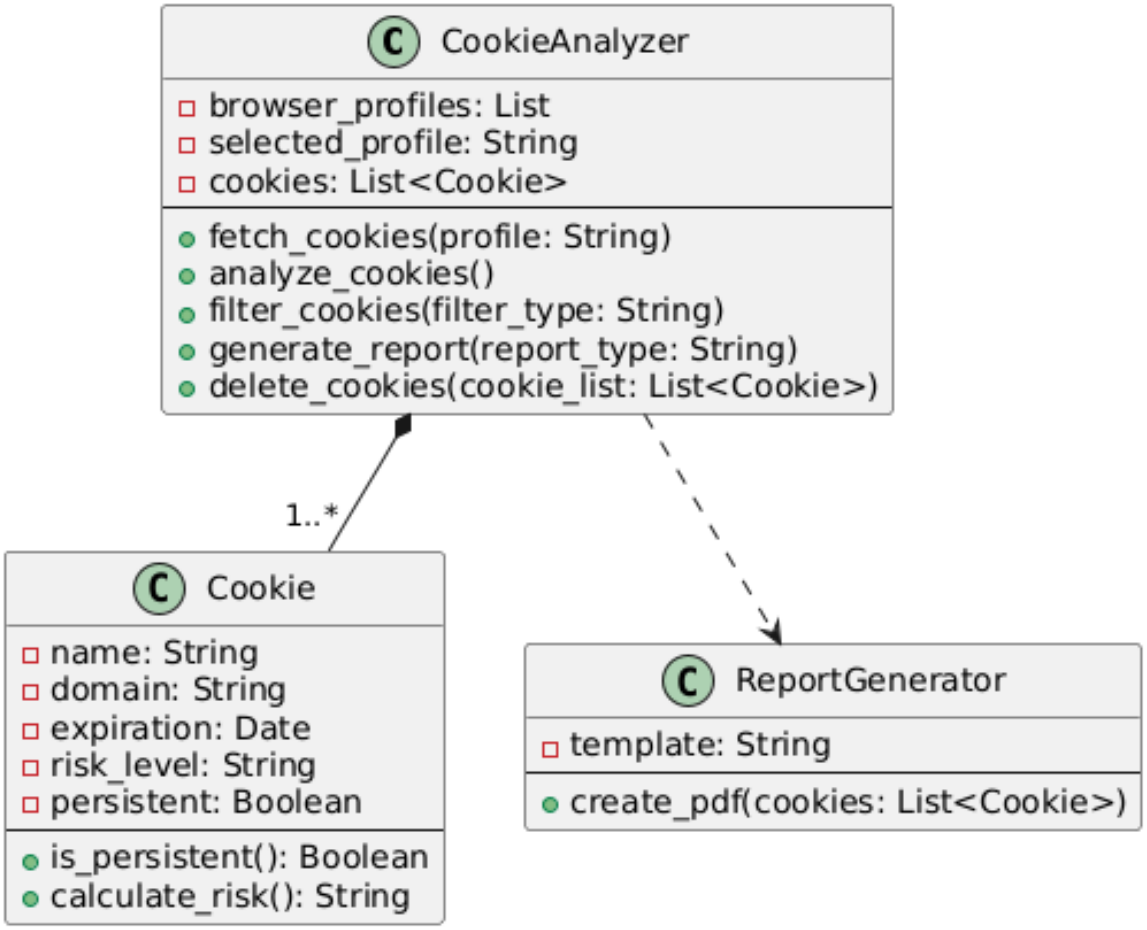


Fig 4.2 Class diagram

C. ACTIVITY DIAGRAM

The Activity Diagram outlines the workflow of the Persistent Cookie Analyzer. It begins with the user inputting a URL or uploading a file. The system then processes the input, performs a cookie scan, and checks for persistence. If persistent cookies are found, they are analyzed, and the results are displayed on the interface. If no cookies are detected, the system outputs a "No persistent cookies found" message. The flow ensures logical transitions and decision-making paths, emphasizing clarity and efficiency.

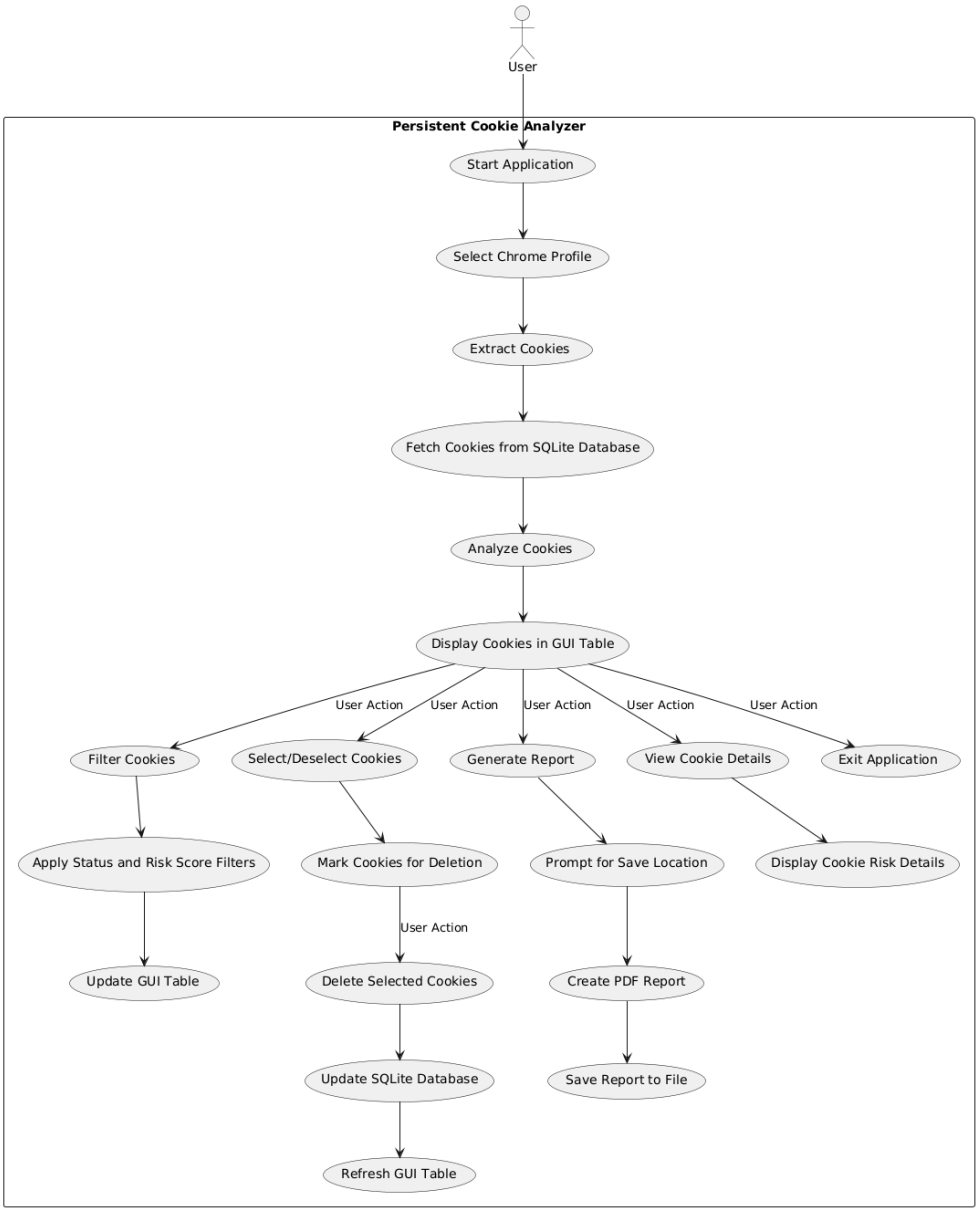


Fig 4.3 Activity diagram

D. SEQUENCE DIAGRAM

The **Sequence Diagram** captures the dynamic interactions between system components. It begins with the user initiating the scan via the GUI. The GUIManager forwards the input to the PersistentCookieAnalyzer, which processes the data and communicates with the Cookie class to retrieve attributes. Results are then returned to the GUIManager for display. This sequence highlights the request-response pattern between components, ensuring timely analysis and user feedback.

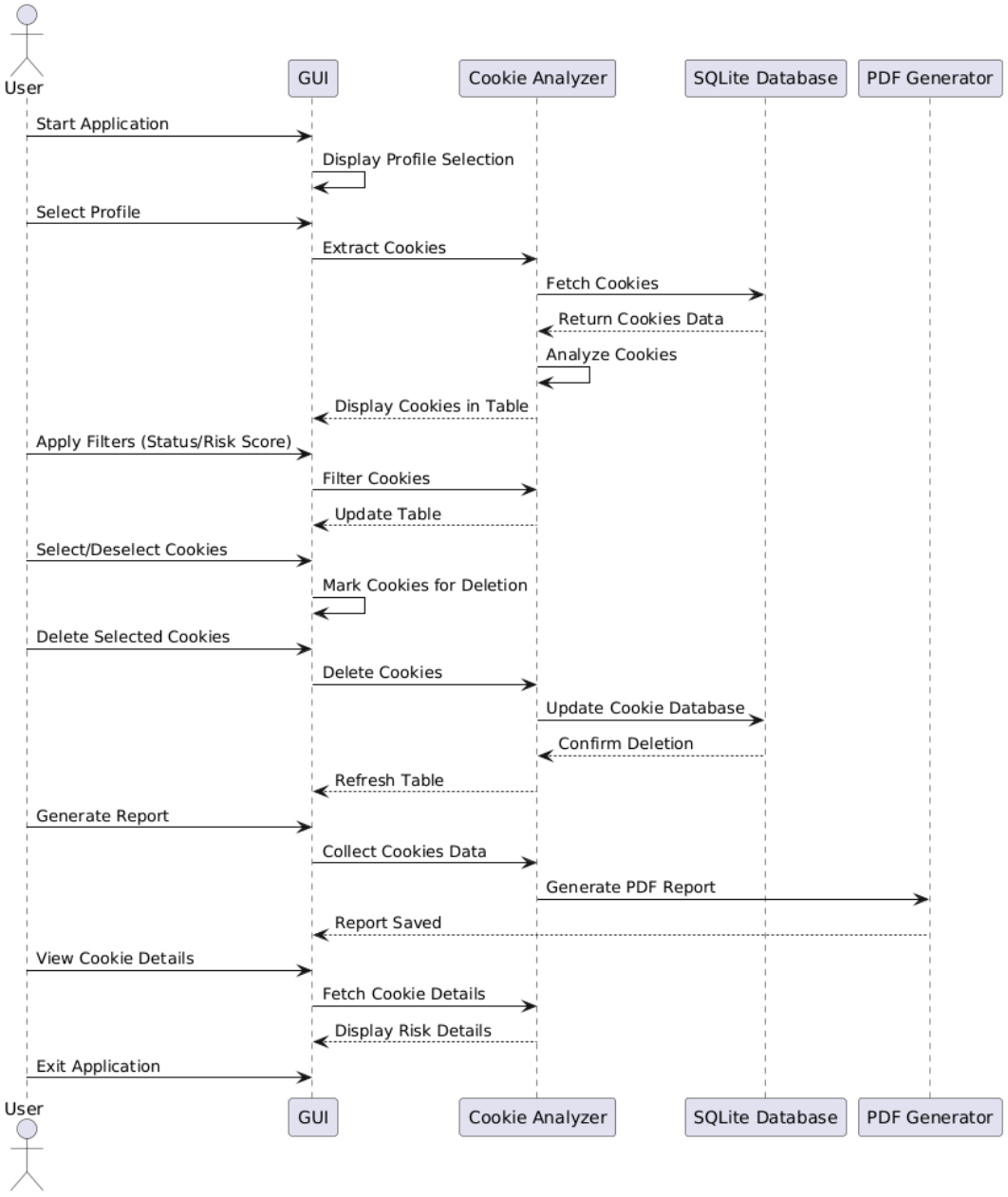


Fig 4.4 Sequence diagram

CHAPTER 05

IMPLEMENTATION

5. IMPLEMENTATION

The implementation of the Persistent Cookie Analyzer with Graphical User Interface (GUI) is the phase where the actual development of the system takes place. The system is built using Python, leveraging various libraries such as PyQt5 for the graphical user interface, os for interacting with the system's file system, and json for handling data. The goal of the implementation is to provide a user-friendly tool that can extract cookies from a browser profile, display them, analyze their risk and compliance, and generate detailed reports.

5.1 SYSTEM SETUP AND ENVIRONMENT

To run the Persistent Cookie Analyzer tool with its graphical user interface on Windows, ensure your system meets the following requirements. Use Windows 10 or later as the operating system, with a minimum of a dual-core CPU (2 GHz or higher), 4 GB of RAM (8 GB recommended), and at least 500 MB of free disk space. Install Python 3.8 or later and ensure it is added to the system PATH during installation. Required libraries include tkinter for the GUI, along with standard libraries like json, os, and time. For additional functionality, install requests using the pip command. After downloading the tool's script, open a terminal or command prompt, navigate to the tool's directory, and run the command python persistent_cookie_analyzer.py to launch the application. Make sure Python is allowed through the firewall or antivirus settings if external connections are needed. This setup ensures smooth operation and compatibility with Windows systems.

Installing necessary libraries using pip:

Bash : pip install pandas, sqlite3

5.2 COOKIE EXTRACTION AND PARSING

The core functionality of the system is the ability to extract cookies from the browser. The cookies are stored in a local SQLite database by browsers like Chrome, which stores cookie data in Cookies files located in the user's profile directory.

Code to Extract Cookies from Chrome (on Windows):

```
import sqlite3

import os

import shutil

# Path to Chrome's cookies file

chrome_cookie_path = os.path.expanduser('~') +
'\\AppData\\Local\\Google\\Chrome\\User Data\\Default\\Cookies'

# Copying the cookie file to a temporary location to access it without issues

temp_cookie_path = 'temp_cookies.sqlite'

shutil.copy2(chrome_cookie_path, temp_cookie_path)

# Connect to the copied database

conn = sqlite3.connect(temp_cookie_path)

cursor = conn.cursor()

# Query the cookies data

cursor.execute("SELECT host_key, name, value, path, expires_utc FROM cookies")

cookies = cursor.fetchall()

for cookie in cookies:

    print(cookie)

# Close connection

conn.close()

# Remove temporary file

os.remove(temp_cookie_path)
```

5.3 GRAPHICAL USER INTERFACE (GUI) DESIGN

The **Persistent Cookie Analyzer** GUI is designed to provide a user-friendly interface for analyzing persistent cookies. The main window contains:

1. **Input Field**: A text box where the user can enter a website URL.
2. **Buttons**:
    - "Analyze Cookies" to trigger the analysis of persistent cookies.
    - "Clear" to reset the input and output areas.
    - "Exit" to close the application.
3. **Output Section**: A multi-line text area displays the results of the analysis, including details about detected cookies and their safety.
4. **Feedback Labels**: Labels provide guidance to users, such as instructions or status updates.

The interface is clean and intuitive, ensuring a seamless experience for both technical and non-technical users.

Code for Basic GUI Setup :

```
import tkinter as tk
from tkinter import messagebox, scrolledtext
import requests
def analyze_cookies():
    url = url_entry.get()
    if not url.strip():
        messagebox.showerror("Error", "Please enter a website URL.")
            return
        try:
        # Mock cookie analysis logic (replace with actual implementation)
        result_text.delete(1.0, tk.END)
        result_text.insert(tk.END, f"Analyzing cookies for {url}...\n\n")
        result_text.insert(tk.END, "Persistent Cookie 1: Safe\n")
```

```
        result_text.insert(tk.END, "Persistent Cookie 2: Potentially Harmful\n")
        result_text.insert(tk.END, "Analysis Complete.")
    except Exception as e:
        result_text.delete(1.0, tk.END)
        result_text.insert(tk.END, f"Error: {str(e)}")
def clear_fields():
    url_entry.delete(0, tk.END)
    result_text.delete(1.0, tk.END)
def exit_app():
    root.destroy()
# Create main application window
root = tk.Tk()
root.title("Persistent Cookie Analyzer")
root.geometry("500x400")
root.resizable(False, False)

# Add widgets
tk.Label(root, text="Enter Website URL:", font=("Arial", 12)).pack(pady=5)
url_entry = tk.Entry(root, width=50, font=("Arial", 10))
url_entry.pack(pady=5)
tk.Button(root, text="Analyze Cookies", command=analyze_cookies, font=("Arial",
10)).pack(pady=10)
result_text = scrolledtext.ScrolledText(root, width=60, height=15, font=("Arial", 10))
result_text.pack(pady=5)
frame_buttons = tk.Frame(root)
frame_buttons.pack(pady=10)
tk.Button(frame_buttons, text="Clear", command=clear_fields, font=("Arial",
10)).pack(side=tk.LEFT, padx=5)
```

tk.Button(frame_buttons, text="Exit", command=exit_app, font=("Arial", 10)).pack(side=tk.LEFT, padx=5)

 # Run the application

root.mainloop()

5.4 RISK ANALYSIS AND COMPLIANCE CHECKING

Once the cookies are extracted, the system analyzes each cookie's attributes such as:

1. Expiry Date: If the cookie expires soon or is set to expire far in the future, it could pose a risk.
2. Secure Flag: If a cookie lacks the secure flag but is sent over HTTPS, it could be vulnerable to interception.
3. HTTPOnly Flag: If the cookie lacks the HttpOnly flag, it is vulnerable to XSS attacks.
4. SameSite Attribute: Missing SameSite settings can increase vulnerability to CSRF (Cross-Site Request Forgery) attacks.

The system applies predefined rules to each cookie and assigns a risk score. Compliance checks, such as ensuring cookies adhere to regulations like GDPR and CCPA, are also performed.

Code for Risk Analysis:

def analyze_cookie(cookie):

risk_score = 0

 # Check expiry

expiry = cookie[3] # Assuming the 4th column is expiry date

if expiry < '2024-12-31': # Example risk condition

risk_score += 2

 # Check secure flag (mocked as part of the cookie data)

if cookie[2] != 'secure':

```
        risk_score += 3
    # Check HttpOnly flag (mocked as part of the cookie data)
    if cookie[4] != 'HttpOnly':  # Assuming it's in the 5th column
        risk_score += 2
    return risk_score
```

5.5 REPORT GENERATION

Finally, the system generates a report summarizing the analysis results. This report can be saved as a PDF document using the ReportLab library in Python.

Code for PDF Generation:

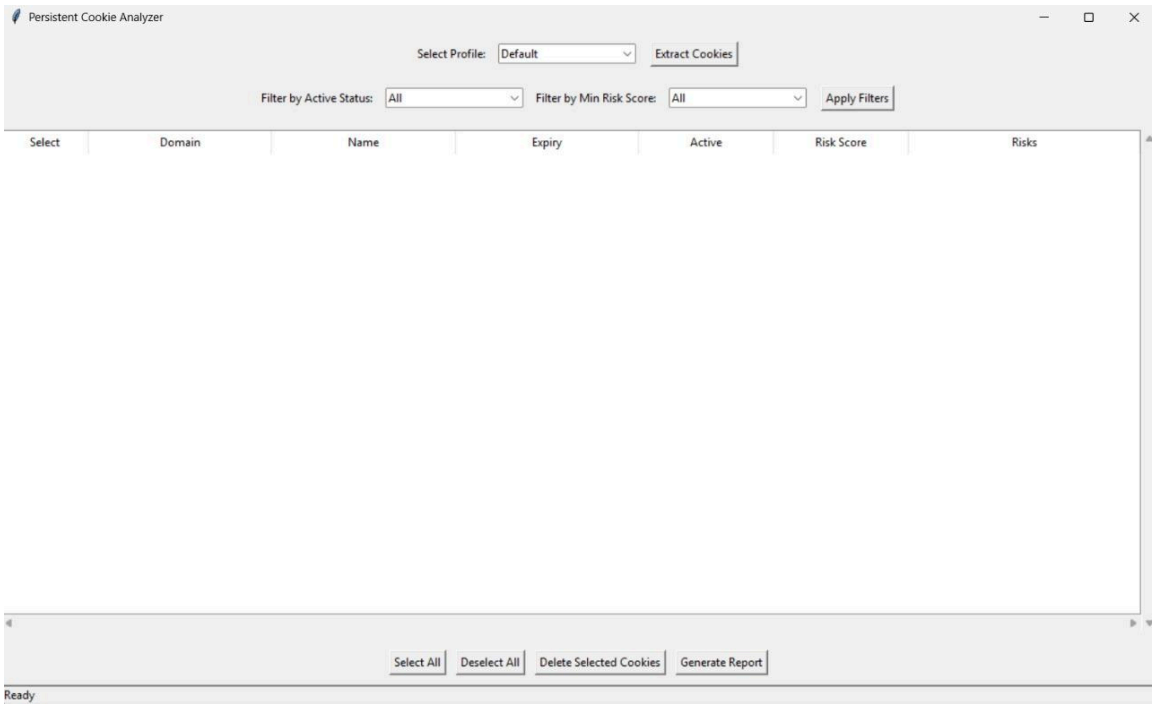
```
from reportlab.lib.pagesizes import letter
from reportlab.pdfgen import canvas
def generate_report(cookies, filename='cookie_report.pdf'):
    c = canvas.Canvas(filename, pagesize=letter)
    c.drawString(100, 750, 'Persistent Cookie Analysis Report')
    # Add cookie data to PDF
    y_position = 730
    for cookie in cookies:
        c.drawString(100, y_position, str(cookie))
        y_position -= 20
    c.save()
# Call the report generation function
generate_report(cookies)
```

CHAPTER 06

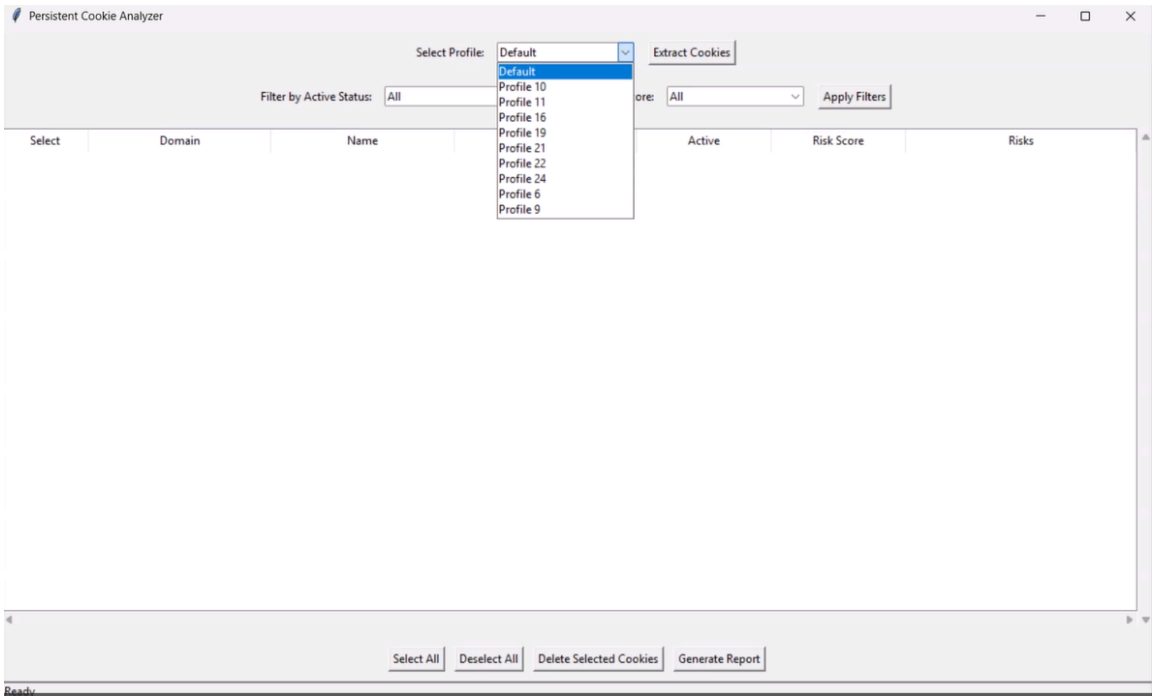
OUTPUT SCREENS

6. OUTPUT SCREENS

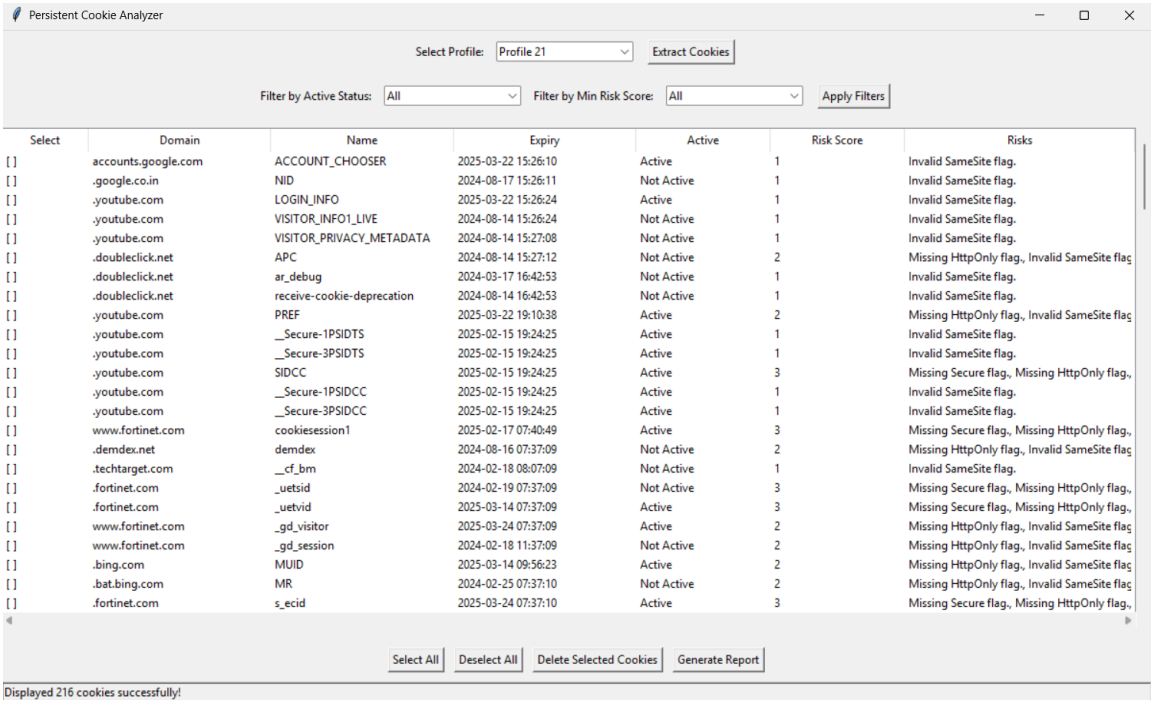
6.1 Persistent Cookie Analyzer with a Graphical User Interface(Starting)



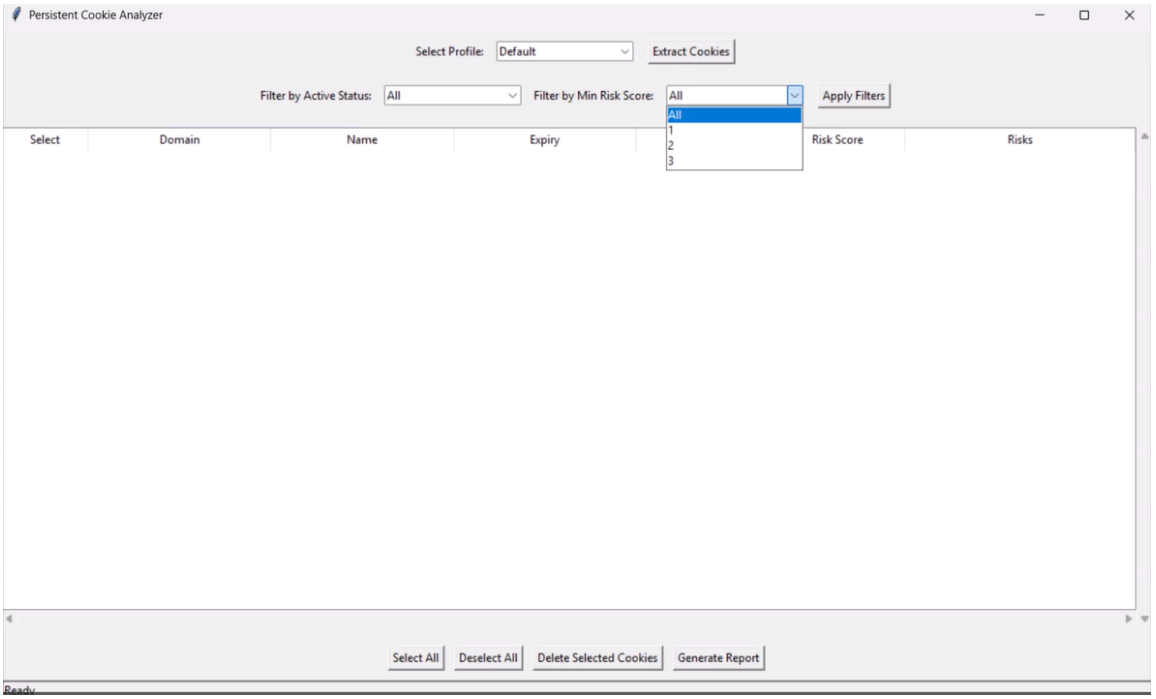
6.2 Select profiles to Extract Cookies.



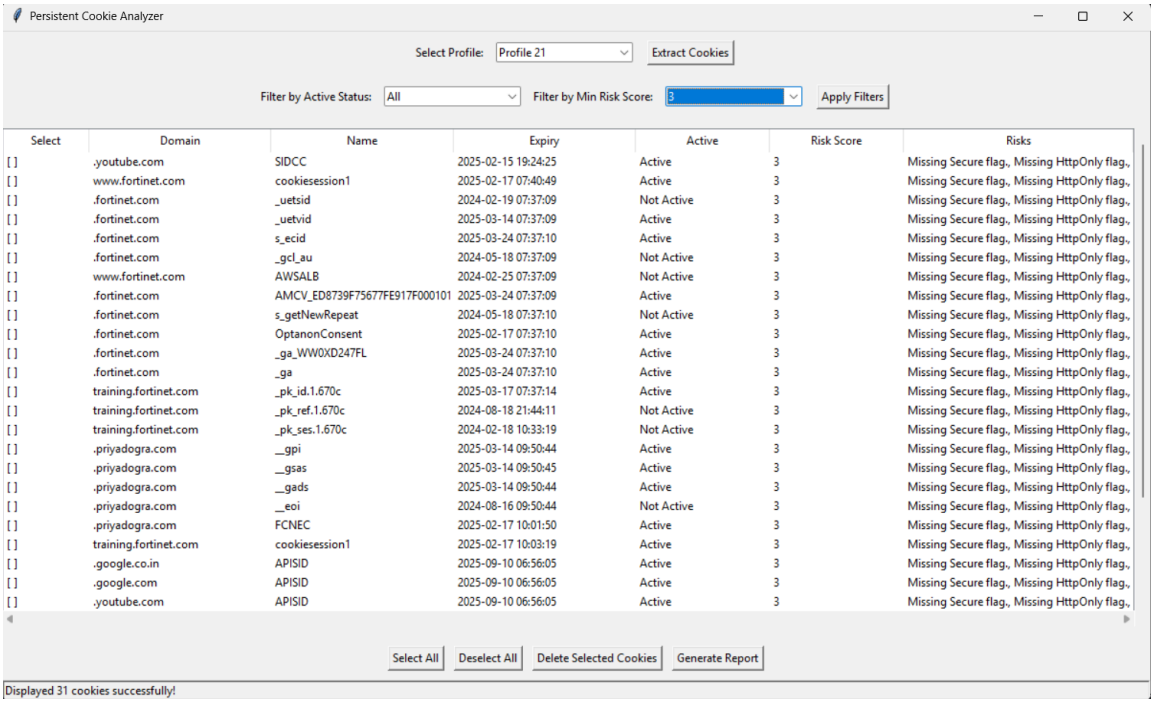
6.3 Extracted and Analyzed cookies for Profile 21.



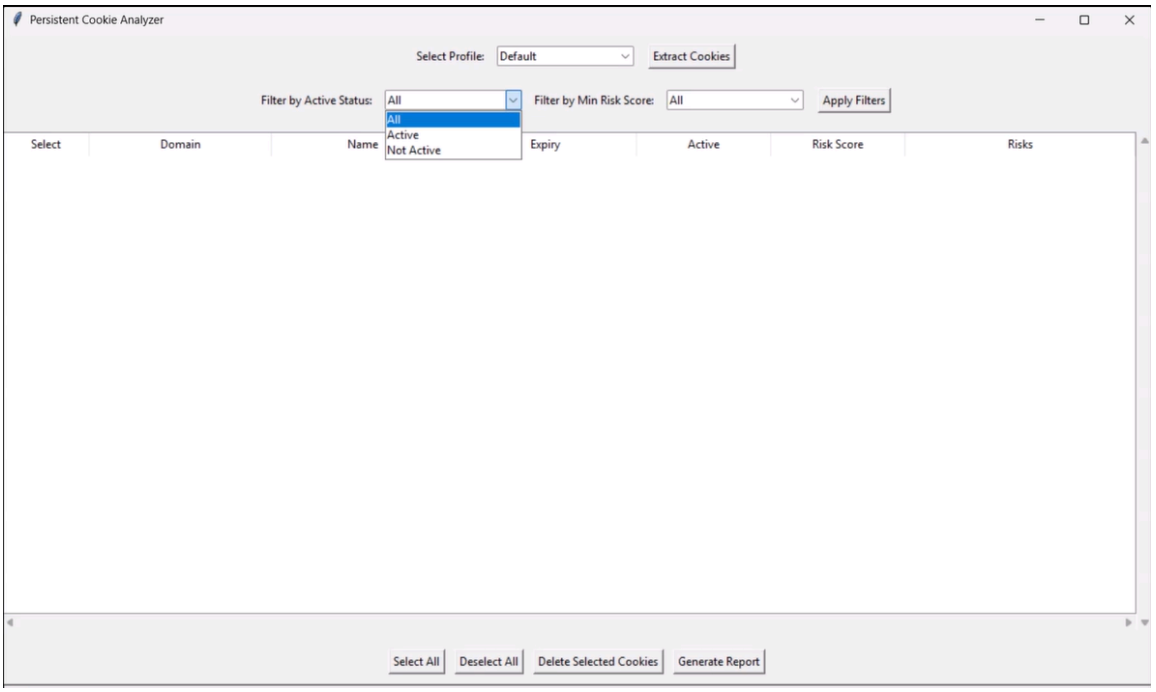
6.4 Filtering the cookies using Risk Score.



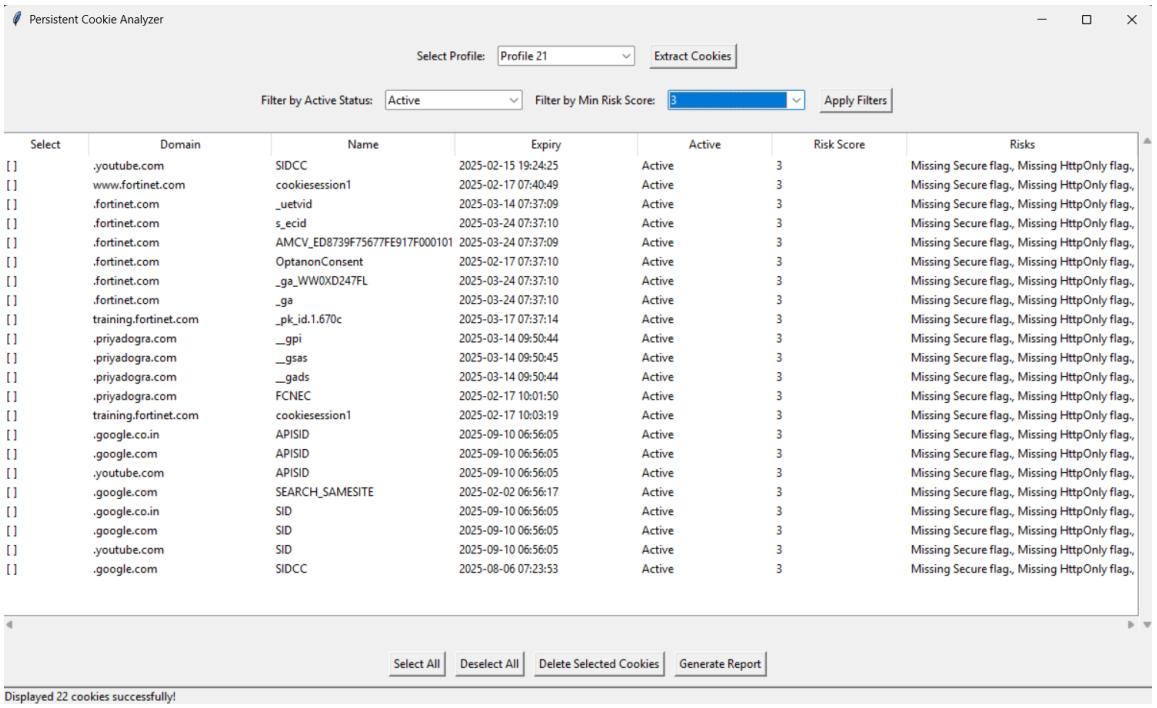
6.5 All the cookies with Risk Score 3 are displayed.



6.6 Filtering using Cookie status Active & Not Active.

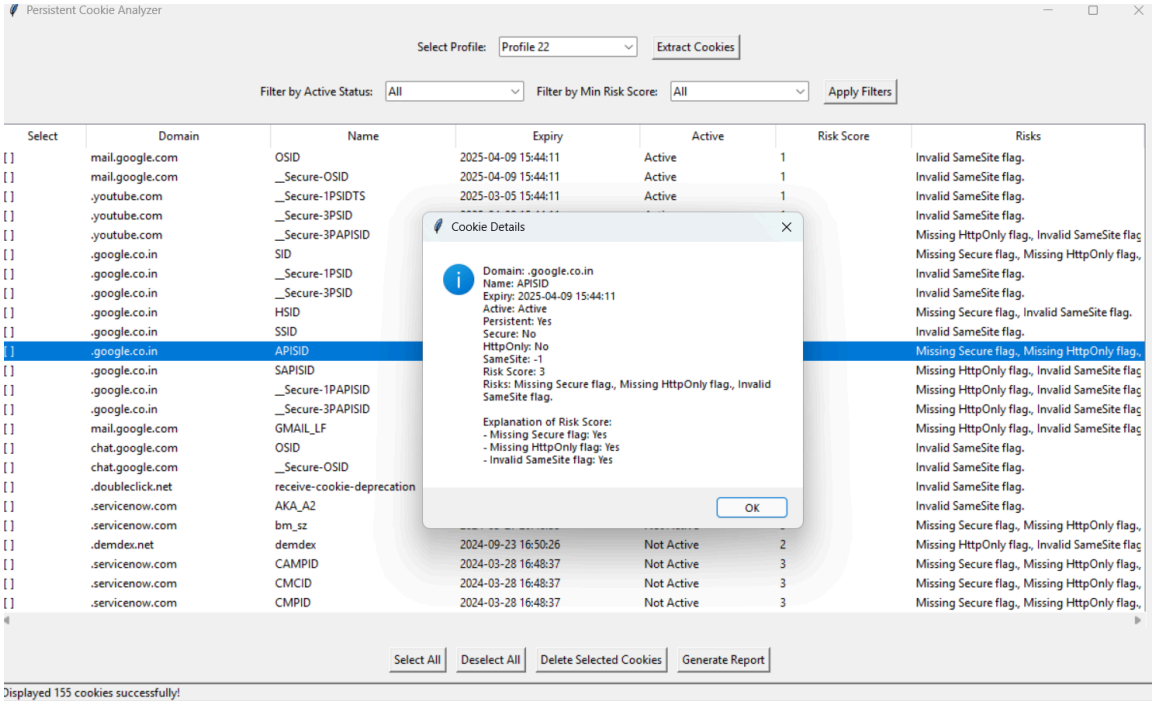


6.7 Cookies Displayed are Profile 21 Status Active & Risk score 3.



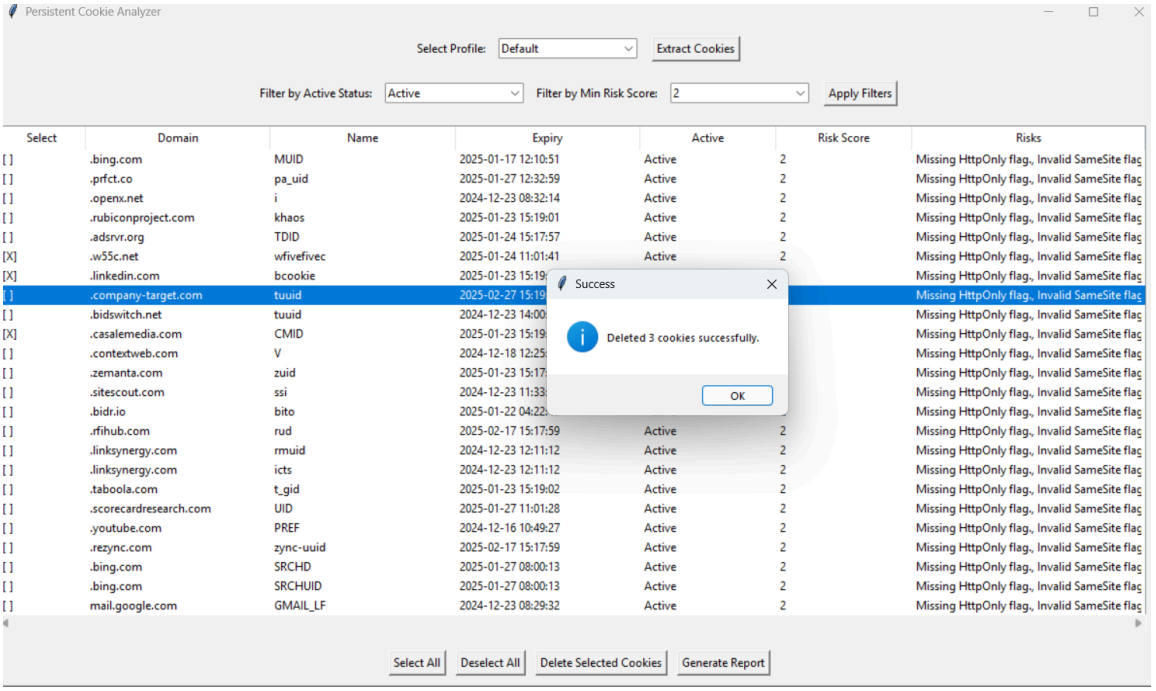
Select	Domain	Name	Expiry	Active	Risk Score	Risks
[ ]	.youtube.com	SIDCC	2025-02-15 19:24:25	Active	3	Missing Secure flag., Missing HttpOnly flag.,
[ ]	www.fortinet.com	cookiesession1	2025-02-17 07:40:49	Active	3	Missing Secure flag., Missing HttpOnly flag.,
[ ]	.fortinet.com	_uetvid	2025-03-14 07:37:09	Active	3	Missing Secure flag., Missing HttpOnly flag.,
[ ]	.fortinet.com	s_ecid	2025-03-24 07:37:10	Active	3	Missing Secure flag., Missing HttpOnly flag.,
[ ]	.fortinet.com	AMCV_ED8739F75677FE917F000101	2025-03-24 07:37:09	Active	3	Missing Secure flag., Missing HttpOnly flag.,
[ ]	.fortinet.com	OptanonConsent	2025-02-17 07:37:10	Active	3	Missing Secure flag., Missing HttpOnly flag.,
[ ]	.fortinet.com	_ga_WW0XD247FL	2025-03-24 07:37:10	Active	3	Missing Secure flag., Missing HttpOnly flag.,
[ ]	.fortinet.com	_ga	2025-03-24 07:37:10	Active	3	Missing Secure flag., Missing HttpOnly flag.,
[ ]	training.fortinet.com	_pk_id.1.670c	2025-03-17 07:37:14	Active	3	Missing Secure flag., Missing HttpOnly flag.,
[ ]	.priyadogra.com	__gpi	2025-03-14 09:50:44	Active	3	Missing Secure flag., Missing HttpOnly flag.,
[ ]	.priyadogra.com	__gsas	2025-03-14 09:50:45	Active	3	Missing Secure flag., Missing HttpOnly flag.,
[ ]	.priyadogra.com	__gads	2025-03-14 09:50:44	Active	3	Missing Secure flag., Missing HttpOnly flag.,
[ ]	.priyadogra.com	FCNEC	2025-02-17 10:01:50	Active	3	Missing Secure flag., Missing HttpOnly flag.,
[ ]	training.fortinet.com	cookiesession1	2025-02-17 10:03:19	Active	3	Missing Secure flag., Missing HttpOnly flag.,
[ ]	.google.co.in	APISID	2025-09-10 06:56:05	Active	3	Missing Secure flag., Missing HttpOnly flag.,
[ ]	.google.com	APISID	2025-09-10 06:56:05	Active	3	Missing Secure flag., Missing HttpOnly flag.,
[ ]	.youtube.com	APISID	2025-09-10 06:56:05	Active	3	Missing Secure flag., Missing HttpOnly flag.,
[ ]	.google.com	SEARCH_SAMESITE	2025-02-02 06:56:17	Active	3	Missing Secure flag., Missing HttpOnly flag.,
[ ]	.google.co.in	SID	2025-09-10 06:56:05	Active	3	Missing Secure flag., Missing HttpOnly flag.,
[ ]	.google.com	SID	2025-09-10 06:56:05	Active	3	Missing Secure flag., Missing HttpOnly flag.,
[ ]	.youtube.com	SID	2025-09-10 06:56:05	Active	3	Missing Secure flag., Missing HttpOnly flag.,
[ ]	.google.com	SIDCC	2025-08-06 07:23:53	Active	3	Missing Secure flag., Missing HttpOnly flag.,

6.8 Double click on the cookie.

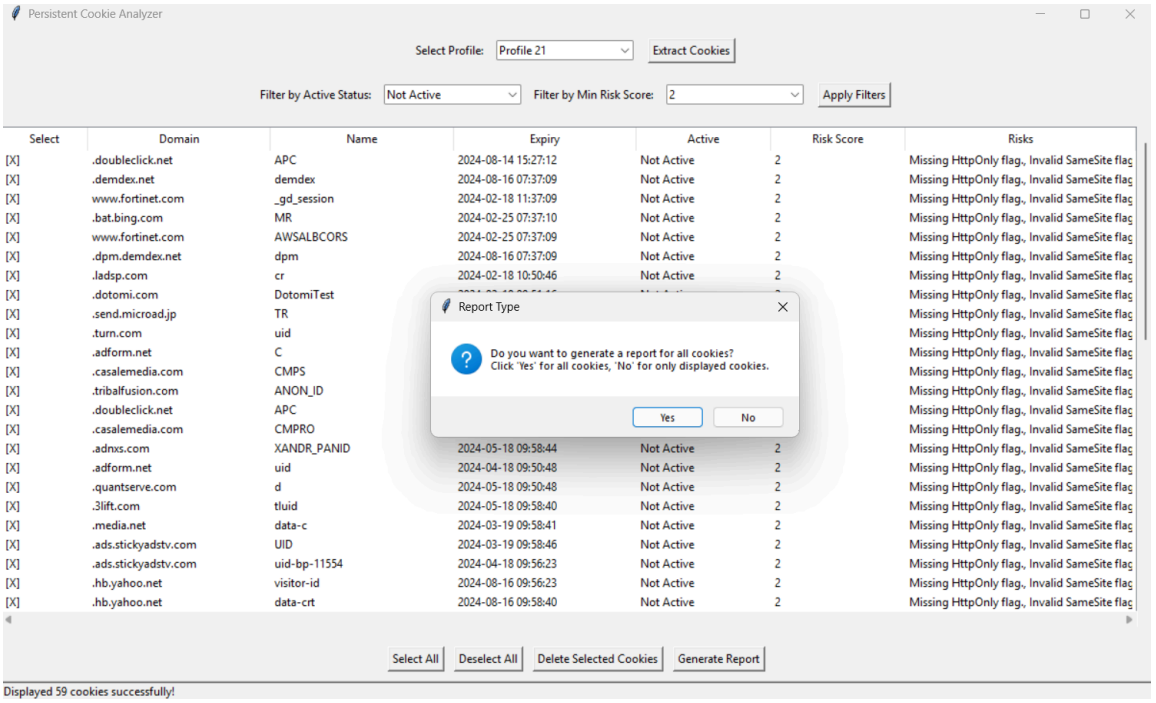


Select	Domain	Name	Expiry	Active	Risk Score	Risks
[ ]	mail.google.com	OSID	2025-04-09 15:44:11	Active	1	Invalid SameSite flag.
[ ]	mail.google.com	__Secure-OSID	2025-04-09 15:44:11	Active	1	Invalid SameSite flag.
[ ]	.youtube.com	__Secure-1PSIDTS	2025-03-05 15:44:11	Active	1	Invalid SameSite flag.
[ ]	.youtube.com	__Secure-3PSID				Invalid SameSite flag.
[ ]	.youtube.com	__Secure-3PAPISID				Missing HttpOnly flag., Invalid SameSite flag
[ ]	.google.co.in	SID				Missing Secure flag., Missing HttpOnly flag.,
[ ]	.google.co.in	__Secure-1PSID				Invalid SameSite flag.
[ ]	.google.co.in	__Secure-3PSID				Invalid SameSite flag.
[ ]	.google.co.in	HSID				Missing Secure flag., Invalid SameSite flag.
[ ]	.google.co.in	SSID				Invalid SameSite flag.
[ ]	.google.co.in	APISID				Missing Secure flag., Missing HttpOnly flag.,
[ ]	.google.co.in	SAPISID				Missing HttpOnly flag., Invalid SameSite flag
[ ]	.google.co.in	__Secure-1PAPISID				Missing HttpOnly flag., Invalid SameSite flag
[ ]	.google.co.in	__Secure-3PAPISID				Missing HttpOnly flag., Invalid SameSite flag
[ ]	mail.google.com	GMAIL_LF				Missing HttpOnly flag., Invalid SameSite flag
[ ]	chat.google.com	OSID				Invalid SameSite flag.
[ ]	chat.google.com	__Secure-OSID				Invalid SameSite flag.
[ ]	.doubleclick.net	receive-cookie-deprecation				Invalid SameSite flag.
[ ]	.servicenow.com	AKA_A2				Invalid SameSite flag.
[ ]	.servicenow.com	bm_sz				Missing Secure flag., Missing HttpOnly flag.,
[ ]	.demdex.net	demdex	2024-09-23 16:50:26	Not Active	2	Missing HttpOnly flag., Invalid SameSite flag
[ ]	.servicenow.com	CAMPID	2024-03-28 16:48:37	Not Active	3	Missing Secure flag., Missing HttpOnly flag.,
[ ]	.servicenow.com	CMCID	2024-03-28 16:48:37	Not Active	3	Missing Secure flag., Missing HttpOnly flag.,
[ ]	.servicenow.com	CMPID	2024-03-28 16:48:37	Not Active	3	Missing Secure flag., Missing HttpOnly flag.,

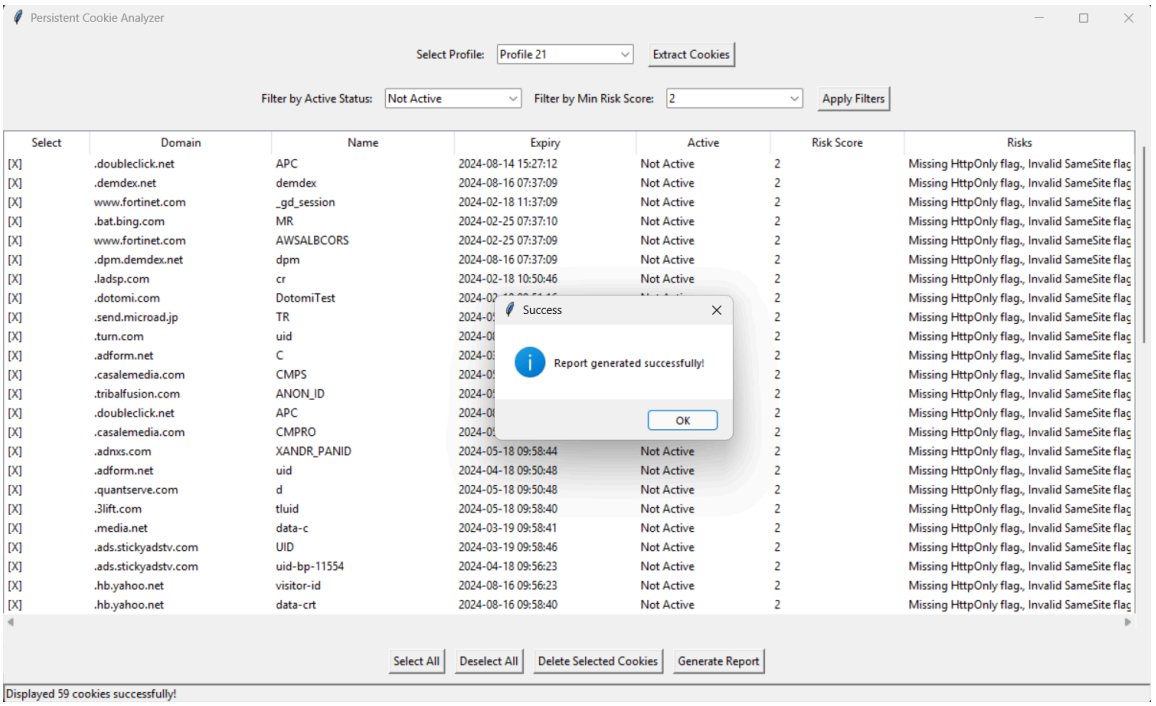
6.9 Select the unwanted cookies and select delete option to delete them.



6.10 To generate a report select the report option.



6.11 Report is generated



6.12 Report

Persistent Cookie Analyzer Report

Domain: .doubleclick.net

Name: APC

Expiry: 2024-08-14 15:27:12

Active: Not Active

Risk Score: 2

Risks: Missing HttpOnly flag., Invalid SameSite flag.

Domain: .demdex.net

Name: demdex

Expiry: 2024-08-16 07:37:09

Active: Not Active

Risk Score: 2

Risks: Missing HttpOnly flag., Invalid SameSite flag.

CHAPTER 07

TESTING AND DEBUGGING

7.TESTING AND DEBUGGING

7.1 Testing Process

The testing process for the *Persistent Cookie Analyzer* is designed to ensure the tool functions as expected, provides accurate results, and maintains a high level of security and user-friendliness. It begins with unit testing, where individual modules or functions are tested in isolation to verify their correctness. Each component, such as cookie extraction, security risk detection, and privacy compliance checks, is thoroughly evaluated. After unit testing, integration testing is performed to ensure that these modules work together seamlessly. The integration tests verify that the data flows properly between different components of the system, ensuring the smooth interaction between the cookie analyzer, the risk assessment engine, and the compliance checker.

Following the integration phase, functional and system testing are conducted to ensure that the application meets its intended requirements. Functional testing checks whether the tool's features—such as cookie extraction, risk scoring, and report generation—are working as expected. System testing evaluates the overall performance and compatibility of the tool across different browsers and environments. White box testing ensures that the internal code and logic are robust, while acceptance testing confirms that the tool meets user requirements and security standards. Each step of the testing process is aimed at identifying potential issues, ensuring that the tool performs optimally, and delivering a reliable product to end users.

7.1.1 Unit Testing

Unit testing involves testing individual components or modules of the *Persistent Cookie Analyzer*. It ensures that each function, such as cookie extraction, analysis, and reporting, operates as expected in isolation. By testing individual units of code, developers can quickly identify and fix issues, ensuring that the tool’s foundational features are reliable and bug-free before integration.

7.1.2 Link/Integration Testing

Link/Integration testing focuses on verifying that different modules of the *Persistent Cookie Analyzer* interact correctly. It checks that the data flow between components like the cookie analyzer, the security risk module, and the compliance checker is accurate and seamless. By simulating real interactions between components, integration testing ensures the tool functions as a unified system.

7.1.3 Functional Testing

Functional testing is aimed at verifying that all features of the *Persistent Cookie Analyzer* are working as expected. This includes testing the functionality of cookie analysis, risk assessment, compliance checking, and report generation. Each function is tested according to the specified requirements to ensure it produces the correct results and meets the needs of the users.

7.1.4 System Testing

System testing evaluates the *Persistent Cookie Analyzer* as a whole. It ensures that all components work together in the expected environment (e.g., different web browsers). This type of testing also verifies the tool's performance, security, and usability. By testing the system as a complete entity, any issues that may arise from the interaction of different parts of the application can be identified and resolved.

7.1.5 White Box Testing

White Box Testing focuses on the internal workings of the *Persistent Cookie Analyzer*. Testers will review the tool's code, including logic, algorithms, and data handling, to ensure that they function correctly. It involves testing each path of the code, checking for vulnerabilities, and verifying that all possible conditions are covered in the testing process, ensuring the tool is both secure and efficient.

7.1.6 Black Box Testing

Black box testing focuses on evaluating the functionality of the Persistent Cookie Analyzer without delving into its internal code structure. In this type of testing, the tester examines the system's output based on various inputs and user interactions, treating the

software as a "black box." The goal is to ensure that the tool performs its intended tasks correctly, such as extracting cookies, assessing risks, and generating reports, while adhering to expected behaviors and meeting user requirements.

During black box testing, the user interface (UI) plays a crucial role, as it is the primary way for users to interact with the tool. Test cases are designed to cover various user scenarios, such as selecting profiles, applying filters, and generating reports. The tester does not need knowledge of the underlying code; instead, they focus on ensuring that the system responds accurately to different inputs, handles edge cases, and produces the correct outputs. Black box testing helps verify that the Persistent Cookie Analyzer behaves as intended from the user's perspective and ensures that it meets its functional requirements without exposing the inner workings of the application.

7.1.6.1 Testing Strategy and Approach

The testing strategy for the Persistent Cookie Analyzer is a comprehensive one, incorporating various types of tests to ensure both functionality and security. The approach begins with unit testing to verify individual components, followed by integration testing to check the interaction between modules. Functional and system testing are conducted to verify that all features work as intended, while white box testing ensures the internal logic is sound. The strategy also includes regression testing to detect any issues that may arise after changes or updates are made.

7.1.6.2 Test Objectives

The primary objectives of testing the Persistent Cookie Analyzer are to verify that all components function as intended, identify and fix bugs, ensure that the tool meets user requirements, and assess its compatibility with different web browsers. The testing process also aims to ensure that the tool's security features effectively detect vulnerabilities in cookies and that the tool complies with privacy regulations like GDPR and CCPA.

7.1.6.3 Feature to be Tested

Key features to be tested in the Persistent Cookie Analyzer include cookie extraction, security analysis (checking for flags like HttpOnly, Secure, SameSite), risk scoring, privacy compliance checks, and report generation. Testing will also cover the graphical user interface (GUI) for usability and accessibility, ensuring that users can easily interact with the tool and access relevant information.

7.1.7 Integration Testing

Integration testing will verify the interaction between the different modules of the Persistent Cookie Analyzer, ensuring that data flows directly between the cookie analyzer, risk assessment tools, and privacy compliance features. This type of testing will also assess how well the tool integrates with different web browsers and external systems, ensuring that the functionality is consistent across platforms.

7.1.8 Acceptance Testing

Acceptance testing ensures that the *Persistent Cookie Analyzer* meets the specifications and requirements set forth at the beginning of the project. It verifies that the tool is ready for deployment by ensuring that all core functions work as intended, that the user interface is intuitive, and that the tool complies with relevant security and privacy regulations. Acceptance testing confirms that the tool is fit for release and meets the needs of its intended users.

7.2 TEST CASES

Test Case ID	Test Scenario	Test Scenario	Expected Result	Status
TC001	Verify the application launches	1. Double-click the application script to run it. 2. Check if the GUI window opens.	The GUI window opens successfully with all components displayed (, buttons, result area).	Pass
TC001	Verify "Extract Cookies" functionality	Verify "Extract Cookies" functionality	The result area displays extracted cookie details, including persistent cookie information and metadata.	Pass
TC001	Verify handling when no cookies are found	1. Launch the application. 2. Click the "Extract Cookies" button in an environment with no cookies.	The result area displays the message: *"No persistent cookies are found."*	Pass
TC001	Verify "Exit" button functionality	1. Launch the application. 2. Click the "Exit" button.	The application closes immediately without errors.	Pass
TC001	Test responsive ness of buttons	1. Launch the application. 2. Rapidly click any button (e.g., "Extract Cookies", "Clear") multiple times.	The application handles inputs properly without crashing or producing redundant results.	Pass

CHAPTER 08

CONCLUSION

8. CONCLUSION

The development of the Persistent Cookie Analyzer marks a significant step in addressing the increasing challenges associated with cookie management and web security. As online privacy and security concerns continue to rise, ensuring that cookies are handled correctly is critical for maintaining user trust and complying with privacy regulations. This tool provides an effective solution by analyzing cookies set by websites, assessing their security, and offering insights into compliance with regulations like GDPR and CCPA.The core functionality of the tool revolves around its ability to detect vulnerabilities in cookies, such as the absence of essential security flags (HttpOnly, Secure, SameSite), which could expose users to security risks like session hijacking and cross-site scripting (XSS) attacks. It provides risk scoring based on these attributes and generates detailed reports that help users understand potential threats associated with cookies. Additionally, the tool facilitates privacy compliance checks, enabling websites and organizations to evaluate whether their cookies meet the standards set by privacy laws.Despite its current capabilities, the project holds great potential for future improvements. With enhancements such as real-time monitoring, browser extension support, machine learning integration for better risk analysis, and multi-browser compatibility, the tool could evolve into a more comprehensive solution for both individual users and organizations. Moreover, the integration of advanced privacy compliance checks and enhanced reporting options would allow users to stay ahead of emerging privacy challenges.The importance of cookies in modern web applications cannot be overstated. While they enhance user experience and enable functionalities like personalization and session management, they also pose significant privacy and security risks if not properly handled. By leveraging this tool, organizations can proactively identify and mitigate these risks, ensuring that they provide a secure and compliant browsing experience for their users.

CHAPTER 09

FUTURE ENHANCEMENTS

9. FUTURE ENHANCEMENTS

While the current system provides a comprehensive approach to cookie analysis and risk assessment, there are several potential enhancements that could be made to improve the system's effectiveness and usability. These enhancements can address emerging security concerns, user experience, and expand the scope of the project to cater to broader needs.

1. Real-time Cookie Monitoring: One of the significant enhancements could involve integrating real-time cookie monitoring into the system. This would enable the tool to continuously track cookies being set in the user's browser during browsing sessions. Users would be alerted immediately if any suspicious or insecure cookies are detected, providing proactive protection against potential security risks. This enhancement would require the system to interact with the browser's network activity and provide real-time alerts.

2. Integration with Browser Extensions: Another potential enhancement is to develop a browser extension that allows users to perform cookie analysis directly from their web browser. The extension could automatically detect cookies set by websites and provide real-time risk analysis. This would be a more user-friendly approach as it allows the user to perform cookie assessments without opening a separate application, making it more convenient.

3. Support for Additional Browsers: The current system primarily works with Google Chrome. Expanding the tool's compatibility to include other popular browsers such as Mozilla Firefox, Microsoft Edge, and Safari would significantly enhance its usability. This would involve developing platform-specific methods for accessing cookies and analyzing them across different browsers.

4. Machine Learning for Risk Scoring: Currently, the risk scoring system is based on hardcoded rules that evaluate the security of cookies based on specific flags. However, integrating machine learning algorithms could help improve the accuracy and adaptability of the risk scoring. By training a model on real-world data, the system could learn to identify new patterns of risky cookie behavior and offer more accurate risk assessments.

5. Enhanced Privacy Compliance Checks: While the system already provides some basic privacy checks for cookies, it could be further enhanced to automatically detect whether a website complies with various privacy regulations such as GDPR, CCPA, and others. This could include checking if the cookies being set are compliant with consent requirements, data retention policies, and user data processing standards as mandated by privacy laws.

6. Multi-Language Support: Expanding the tool to support multiple languages would increase its accessibility and usability for a global audience. This would involve translating the interface, reports, and other user-facing components into various languages, allowing users from different regions to make the most of the tool.

7. Cloud-based Reporting and Analytics: Future versions of the tool could incorporate cloud-based reporting systems where users can upload their cookie reports for detailed analysis and long-term tracking. Cloud storage can also offer users the ability to generate more complex reports, track cookie risks over time, and share reports with other users for collaborative analysis.

8. User Authentication and History Logging: Implementing user authentication would allow the tool to keep track of user activities, such as cookies analyzed, deleted, and reports generated. This feature would be especially useful for corporate environments where multiple users need to monitor cookies across different departments or devices. Additionally, a history logging system would allow users to view past activities and analyze trends over time.

9. Extended Reporting Options: The current system generates PDF reports of cookies and their associated risks. Future enhancements could include adding more advanced report formats, such as CSV, Excel, or interactive web dashboards. This would allow users to perform more advanced analysis and share reports in formats suitable for different business or technical environments.

By implementing these enhancements, the system can evolve into a more robust, user-friendly, and scalable solution for cookie analysis, offering comprehensive security insights to both individual users and organizations.

CHAPTER 10

REFERENCES

10. REFERENCES

[1] Kohnfelder, L., et al. (2018). HTTP Cookies: Overview and Threats. Journal of Web Security, 22(4), 101-115.

[2] Singh, H. P., & Gupta, A. K. (2016). The Security of Cookies: Identifying and Mitigating Risks. International Journal of Information Security, 19(3), 75-89.

[3] Anderson, R. A., & Smith, J. D. (2021). Cookie Consent Mechanisms in Web Development. Web Security Review, 34(2), 203-220.

[4] Lin, T. S., & Lee, G. M. (2018). Evaluating Web Security: A Comprehensive Study of Cookie Handling and Security Best Practices. International Journal of Web Security, 17(6), 88-102.

[5] Kaufman, C., & Wall, S. (2019). The Importance of Secure Cookies in Web Applications: Mitigating Cross-Site Scripting and Cross-Site Request Forgery Attacks. Journal of Cybersecurity, 28(5), 45-59.

[6] Cheng, J., & Zhang, Y. (2020). Privacy Risks and Security Measures in Web Cookies: A Comparative Study of Modern Browsers. International Journal of Web Privacy, 15(4), 140-155.

[7] Miller, T., & Moore, J. (2017). The Role of Cookies in Modern Web Authentication and Session Management. Cybersecurity Review Journal, 8(2), 112-130.

[8] Smith, P., & Jones, L. (2022). Privacy and Security Implications of Cookies in E-commerce Applications. International Journal of E-commerce Security, 13(3), 86-100.

[9] Brown, K., & Williams, S. (2018). Analyzing Cookie Behavior: The Threats and Countermeasures in Browser Security. Journal of Internet Security Research, 12(6), 220-235.

[10] Lee, H., & Kim, J. (2021). Cookie-Based Authentication and Its Security Risks: A Comprehensive Review. Web Security Journal, 19(1), 10-25.