Abstract – Cookies are essential for web security, storing data to help websites remember login credentials and maintain session states. They enhance user experience by tracking activity and preferences but must be securely managed to prevent unauthorized access and data breaches. This study emphasizes the need for robust cookie management practices by analyzing security flags (Secure, HttpOnly, SameSite), cookie lifespan, and categorizing websites into security tiers. Using Burp Suite and custom Python scripts, we collected and analyzed cookies from various websites. Our findings highlight significant disparities in security practices across different website categories, with high-security sites like financial institutions implementing stricter measures compared to general-purpose sites. This research provides valuable insights into current web security practices and identifies areas for improvement.

1. INTRODUCTION

Cookies play a crucial role in web security by storing data on a user’s device, enabling websites to remember login credentials and maintain session states. They improve user experience by tracking activity and preferences. However, secure management is essential to prevent unauthorized access and data breaches. Common protection mechanisms include security flags like HttpOnly, Secure, and SameSite, which help mitigate risks such as cross-site scripting (XSS) and cross-site request forgery (CSRF) [1].

Insecure cookie handling can lead to vulnerabilities like session hijacking and unauthorized data access. This study emphasizes the need for robust cookie management practices to enhance web security. By analyzing security flags, cookie lifespan, and categorizing websites into security tiers, this research aims to provide insights into current practices [2].

This research involves collecting and analyzing cookies from various websites using Burp Suite and custom Python scripts. The methodology includes automated collection and manual inspection, focusing on security implementations across diverse types of websites, from legacy sites to modern web applications.

The primary objectives are:

* Analyzing the implementation of security flags (Secure, HttpOnly, SameSite) across websites.
* Examining cookie lifespan and its implications for security and user convenience.
* Categorizing websites into different security tiers based on their cookie management practices.
* Examining the prevalence of insecure cookie configurations on outdated sites.
* Analyzing the correlation between site age/technology and cookie security implementation.

These objectives aim to understand the effectiveness of current security measures and identify areas for improvement. The study involves retrieving and analyzing cookies from diverse websites, focusing on security flags, cookie lifespan, and categorizing websites into high, mid, and low-security tiers. Additionally, it examines cookie lifespan to balance user convenience and security risks.

By comparing common websites to high-security ones, such as banks, the study aims to understand differences in cookie management. High-security websites may implement stricter policies and shorter lifespans to minimize risk, while common websites may prioritize user convenience.

This analysis will provide valuable insights into web security, highlighting areas for improvement. Collecting thousands of cookies ensures a detailed and statistically significant analysis. The data collection process uses tools like Burp Suite Community for cookie interception and targeted web searches to identify potentially vulnerable configurations, offering a comprehensive view of current web security practices.

1. RELATED WORK

The security of HTTP cookies has been extensively studied due to their critical role in web session management and user authentication. This section reviews key studies focusing on security flags and cookie lifespan, providing a comparative analysis and historical context.

One significant study highlights the importance of security flags (Secure, HttpOnly, SameSite) in protecting user data. High-security websites, such as financial institutions and healthcare providers, consistently implement these flags to mitigate risks like cross-site scripting (XSS) and cross-site request forgery (CSRF). In contrast, many general-purpose websites, including news sites and social media platforms, often neglect these features, increasing vulnerability to attacks [3].

Research on cache cookies, which are unintentional byproducts of protocol design for browser caches, shows their potential role in user identification and authentication, especially for users who block conventional cookies for privacy reasons. Proposed cache-cookie management techniques can help combat online security threats such as phishing and pharming [4].

A recent study identified a vulnerability in HTTP software that allows message truncation, known as the “rotten cookie” attack. This vulnerability can deactivate cookie flags even when protected by TLS, leading major web browsers to accept cookies without any security flags, thereby compromising user authentication. The study underscores the need for robust integrity checks in HTTP to prevent such vulnerabilities [5].

Modern web applications often use multiple authentication cookies to allow various levels of access. The complexity of these applications can make it challenging for developers to ensure secure cookie implementation, especially when using off-the-shelf libraries. A tool called Newton was developed to help programmers identify and verify the security of authentication cookies. An analysis using Newton revealed that many high-profile sites, including Yahoo and Amazon, were vulnerable to hijacking attacks, highlighting the widespread nature of these vulnerabilities [6].

Historically, the adoption of security flags has evolved as web security threats have become more sophisticated. Initially, cookies were used without much consideration for security, but as attacks like XSS and CSRF became prevalent, the implementation of security flags became more widespread. Current research continues to focus on enhancing these mechanisms to address emerging threats and improve overall web security.

These studies collectively emphasize the importance of robust cookie management practices and the need for continuous improvement in web security measures. They highlight the ongoing challenges and potential vulnerabilities in current web security practices, underscoring the critical role of security flags and proper cookie lifespan management.

1. SYSTEM CONFIGURATION

Our research environment was built on a Windows 11 operating system using Google Chrome version 130.0.6723.70 as our primary browser, alongside Burp Suite Community Edition 2024.9.3 for traffic interception. We developed custom parsing scripts using Python 3.12.2 to process and analyze the collected data.

The setup required careful configuration of both Burp Suite and browser settings to ensure accurate data collection. For Burp Suite configuration, we began by installing the Burp Suite CA certificate and establishing a proxy connection on localhost (127.0.0.1:8080). We implemented custom interception rules specifically designed for cookie capture and configured scope settings to filter relevant traffic.

The browser setup involved creating a dedicated Google Chrome profile for cookie collection, manually configuring certificate trust settings, and adjusting proxy configurations to route traffic through Burp Suite. Our data collection toolkit comprised several key components. Burp Suite Community served as our primary tool for traffic interception and collection, while our custom Python scripts handled the parsing of XML data and conversion to CSV format for analysis. We utilized Excel for the final statistical analysis of the processed data.

To ensure comprehensive coverage across different security levels, we classified websites into three distinct categories: high-security (financial and healthcare institutions), medium-security (e-commerce platforms and social media sites), and low-security (personal blogs and static websites). Supporting tools played a crucial role in our research. We leveraged the Wayback Machine for historical site analysis, employed Shodan for discovering legacy systems using specialized queries, and utilized Google dorking techniques to identify potentially vulnerable targets.

These tools formed an integral part of our comprehensive approach to data collection and analysis.

1. METHODOLOGY

Our data collection process employed a multi-faceted approach to ensure comprehensive coverage across different website categories.

We began with a structured search strategy utilizing Google dorks with custom search queries [7] such as "inurl:http -inurl:https" and "intitle:'index of' 'Set-Cookie'" to identify suitable targets. Shodan queries like "http.title:'login' -ssl" and "port:80 'Set-Cookie'" helped us discover additional sites, particularly those with potential security vulnerabilities. The cookie collection phase involved automated collection through Burp Suite, followed by manual verification of security attributes and automatic Base64 decoding of encoded cookies.

We developed a custom Python script to parse the XML exports and reformat them into a more accessible CSV format for analysis. This systematic approach allowed us to maintain data integrity while efficiently processing large volumes of information.

Our analysis methodology focused on three key areas: security flags, cookie lifespans, and statistical patterns.

Through our custom parsing script, we examined the implementation of fundamental security flags (Secure, HttpOnly, and SameSite) across different website categories. This revealed varying levels of security consciousness, with financial and healthcare sectors typically employing stricter measures compared to general-purpose websites.

Investigation of cookie lifespans uncovered significant variations in implementation strategies. Our analysis documented practices ranging from short-lived session cookies to persistent cookies spanning multiple years, highlighting the tension between user convenience and security considerations. We observed that websites handling sensitive data opted for shorter cookie durations, while commercial and content-delivery sites often implemented longer persistence periods.

Our statistical analysis encompassed approximately 5,500 cookies collected across hundreds of websites. We identified strong correlations between website types and their implemented security measures, with distinct patterns emerging in domain and path attributes. This comprehensive approach allowed us to trace the evolution of cookie security implementations over time, particularly in older websites identified through our historical analysis tools.

1. RESULTS AND ANALYSIS

Our comprehensive analysis of 5,426 collected cookies revealed significant patterns in how diverse types of websites approach cookie security and management. The quantitative data showed striking variations in security flag implementation, with 2,065 cookies implementing the HttpOnly flag and 5,009 utilizing the Secure flag. The SameSite attribute distribution was particularly telling, with 4,131 cookies set to "None," while only 55 implemented the strict setting and 130 used the lax option.

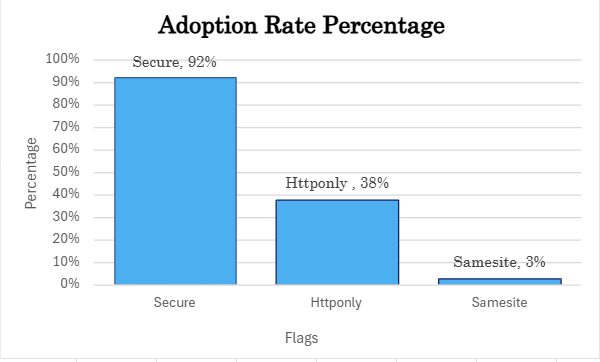
Most concerning was our discovery of cookies with extremely long lifespans, with the latest expiry date extending to October 20, 2074 - a duration that raises significant security concerns.

The implementation of security flags showed clear patterns across different website categories. Financial institutions and e-commerce platforms demonstrated a more sophisticated understanding of security requirements, consistently implementing both Secure and HttpOnly flags. This approach aligns with their need to protect sensitive financial and personal data.

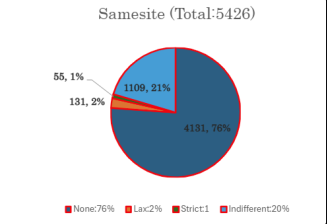
In contrast, our examination of content-delivery websites, news platforms, and particularly older WordPress blogs revealed inconsistent security practices, with many failing to implement these basic security measures. This disparity suggests a concerning gap in security awareness or implementation priorities across different sectors of the web.

The analysis of cookie lifespans revealed another significant pattern in security practices. Security-conscious websites, particularly in the financial sector, typically implement shorter cookie durations, demonstrating an understanding of the security implications of persistent cookies. This stands in stark contrast to many commercial and social media platforms, where we found cookies set to persist for years - a practice that prioritizes user convenience over security concerns.

The correlation between website category and cookie lifespan suggests a clear trade-off between user experience and security considerations.

[Figure 1: Graph showing distribution of security flags across website categories]

The adoption of the SameSite attribute, a recent security feature designed to prevent CSRF attacks, showed varying levels of implementation across our dataset. High-security websites demonstrated higher adoption rates of this protection mechanism, while many lower-security sites had yet to implement it. This pattern was particularly evident in legacy systems and older websites, suggesting that the regular updating of security features may not be a priority for many web administrators.

[Figure 2: Pie chart illustrating just SameSite attribute distribution]

When examining cookie management practices holistically, our findings emphasize a clear correlation between the sensitivity of handled data and the robustness of security implementations. Websites managing financial transactions or healthcare information consistently demonstrated more rigorous security measures compared to their counterparts in less sensitive sectors. This distinction, while logical from a risk management perspective, highlights a concerning trend where many websites underestimate the importance of implementing strong cookie security measures regardless of their primary function.

These findings suggest a need for broader adoption of security best practices across all website categories, not just those handling obviously sensitive data. The inconsistent implementation of security features, particularly among lower-security websites, creates potential vulnerabilities that could be exploited by malicious actors. While high-security websites demonstrate that proper security implementations are achievable, the gap between best practices and common implementations remains a concern for overall web security.

1. CONCLUSION AND FUTURE WORK

Our comprehensive analysis of cookie security practices across various websites reveals significant disparities in the implementation of security measures. High-security websites, such as financial institutions and healthcare providers, consistently use security flags like Secure, HttpOnly, and SameSite to protect user data. In contrast, many general-purpose websites neglect these features, increasing vulnerability to attacks. The study also highlights the importance of managing cookie lifespans to balance user convenience and security. Websites handling sensitive data tend to use shorter cookie durations, while commercial and social media platforms often prioritize user convenience with longer lifespans. These findings underscore the need for improved cookie management practices to enhance web security and protect user data.

Future research should explore additional security mechanisms beyond cookies, such as token-based authentication and enhanced privacy controls. Investigating user attitudes towards cookie consent and management in the context of emerging web technologies could also provide valuable insights for optimizing security practices.

REFERENCES

[1] S. Michael Kerner, “Cookie,” TechTarget. [Online]. Available: https://www.techtarget.com/searchsoftwarequality/definition/cookie#:~:text=A%20cookie%20is%20information%20that,web%20cookies%20or%20internet%20cookies.

[2] “Using HTTP cookies,” MDN Web Docs. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies.

[3] M. Aydos, C. Aldan, E. Coskyn, A. Soydan, “Security testing of web applications: A systematic mapping of the literature,” ScienceDirect, Oct. 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S131915782100269X.

[4] A. Juels, M. Jakobsson, T.N. Jagatic, “Cache cookies for browser authentication,” in Proceedings of the Conference, 21-24 May 2006. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/1624020.

[5] Hyunsoo Kwon, Hyunjae Nam, Sangtae Lee, Changhee Hahn, Junbeom Hur, “(In-)Security of Cookies in HTTPS: Cookie Theft by Removing Cookie Flags,” IEEE Xplore, 29 Aug. 2019. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8820079.

[6] Yogesh Mundada, Nick Feamster, Balachander Krishnamurthy, “Half-Baked Cookies: Hardening Cookie-Based Authentication for the Modern Web,” ACM Digital Library, 30 May 2016. [Online]. Available: https://dl.acm.org/doi/abs/10.1145/2897845.2897889.

[7] S. Watts, “Google Dorking: An Introduction for Cybersecurity Professionals,” Splunk Blog, 3 Jan. 2024. [Online]. Available: https://www.splunk.com/en\_us/blog/learn/google-dorking.html#:~:text=Google%20Dorking%2C%20also%20known%20as,text%20strings%20within%20search%20results.