Utilization Of Privilege Escalation Vulnerability In Manipulating Administrator Access Of PT XYZ

Jody Jeremi Hadrian Ritonga[1*], Joy Idoan Sihotang[2]
[1]Dept. of Informatics Engineering, Universitas Advent Indonesia
[2]Dept. of System Information, Universitas Advent Indonesia
1st 2181008@unai.edu. 2ndjay.sihotangg@unai.edu

Abstract. PT.XYZ is a CRM solutions provider that helps businesses manage their interactions with customers. This research was conducted using a qualitative method, focusing on an in-depth exploration of the platform's security aspects. The approach used was manual penetration testing, which allowed for detailed interaction with the CRM application to uncover potential vulnerabilities. The research process included five main stages: application analysis, exploitation of security flaws, impact evaluation, solution development, and reporting. Through this process, a critical vulnerability was discovered in the user management mechanism, which allowed a standard user account to escalate privileges to an administrator level. This posed serious risks such as potential misuse of customer data, operational disruptions, and financial losses. The manual testing approach, combined with qualitative analysis, enabled the identification of this logic-based flaw that automated tools might have missed. Based on the findings, PT.XYZ has implemented several improvements to strengthen its security posture. This research highlights the importance of regular, hands-on security assessments to maintain the integrity of enterprise information systems against evolving cyber threats.

Keywords: Devtools, Penetration testing, Privilege escalation, Website

Received March 2025 / **Revised** May 2025 / **Accepted** June 2025

INTRODUCTION

In today's digital era, information technology has become a vital element in human daily life and various industrial sectors. Every year, technological innovations continue to emerge, bringing significant changes in various aspects of life. Information technology not only functions as a tool but also as a key driver in the digital transformation that enables organizations to identify new business opportunities, improve operational efficiency, and enhance competitiveness. Companies, educational institutions, and governments are examples of entities that can leverage information technology to support strategic and operational decision-making [1]. The appropriate application of technology allows organizations to adapt more quickly to market changes, optimize resources, and provide better services to customers or the public. Thus, understanding and applying information technology has become a necessity for long-term success in various fields.

One technology that drives and enhances existing business industries is the website [2]. In the era of Industry 4.0, the presence of web-based technology has transformed the way people access information and services. Websites offer limitless accessibility, allowing users to meet their needs from various locations, using different devices, and without time constraints. This flexibility has significantly increased productivity and efficiency in various sectors of modern life. The ease of access and 24/7 availability offered by web-based platforms has been a key catalyst in driving innovation and optimizing both business processes and daily activities. Consequently, the integration of web technology has become a crucial aspect in efforts to improve performance and competitiveness in this digital era [3].

As users become accustomed to using websites as tools to assist their daily activities, people's trust in websites increases. As a result, important data such as addresses, company financial reports, and other sensitive files are starting to be stored on websites. Therefore, cybersecurity becomes an important factor in building and developing websites and cannot be overlooked [4].

The security of applications is based on three fundamental principles known as the CIA triad: Confidentiality, Integrity, and Availability. These principles are the main foundations for building and maintaining secure and reliable information systems. Confidentiality ensures that data can only be accessed by authorized parties. Integrity ensures the accuracy and consistency of data throughout its lifecycle, protecting it from unauthorized modification. Availability ensures that information and resources are always accessible to authorized users when needed. To maintain the effectiveness of these three pillars, proactive and comprehensive security strategies are required. Regular and systematic security testing is key to identifying potential vulnerabilities, evaluating the effectiveness of existing security controls [5] [6], and ensuring that systems remain resilient against evolving threats.

Through active penetration testing, organizations can identify and strengthen security against internal and external threats. Such testing provides structured guidance and allows organizations to proactively address attack risks and enhance resilience across the three main pillars [7] [8]. One vulnerability that compromises the integrity of a website is privilege escalation. This vulnerability allows attackers to elevate their access rights, enabling them to alter or view data that should not be accessible. As a result, the integrity of the application can be compromised, and sensitive data may be corrupted or lost. This vulnerability is quite dangerous because access for administrators and users should be different, and therefore, steps must be taken to secure this vulnerability. Privilege escalation can be divided into two main categories: vertical and horizontal.

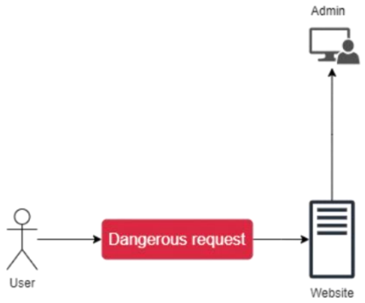


Figure 1. Vertical Privilege escalation

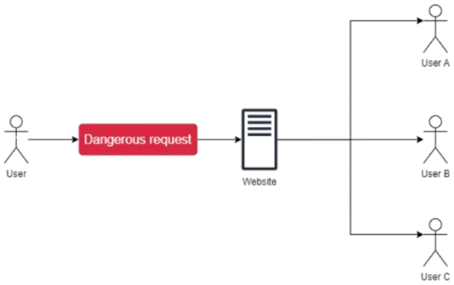


Figure 2. Horizontal Privilege Escalation

Vertical privilege escalation occurs when a regular user successfully elevates their access rights to a higher level, such as an admin. This process typically involves exploiting security vulnerabilities that allow an attacker to gain access they should not have. In this context, an attacker who initially has limited access can carry out potentially harmful actions, such as altering important settings or accessing sensitive data that they should not be able to see. On the other hand, horizontal privilege escalation happens when a user attempts to access data or functions owned by another user with the same level of access. In this case, the attacker does not try to elevate their access rights but instead focuses on gaining access to another user's account with equal access rights [9] [10].

An example of vertical privilege escalation vulnerability can be found on the XYZ website, a fairly large CRM application. The author has identified a security flaw on the website that allows members of an organization, who are invited by the admin to the company dashboard, to unlawfully elevate their access rights from regular user status to admin. By obtaining admin access, they can view sensitive information that should only be accessible to admins and potentially remove other members from the organization. This vulnerability could be exploited by attackers to manipulate the organizational structure and cause significant harm to other entities.

Through this research, the author aims to raise awareness of the importance of thoroughly testing every feature, even seemingly minor ones, and to address the vulnerabilities discovered. This research is expected to serve as a valuable reference for security testers, researchers, and institutions in conducting their security tests, thereby encouraging joint efforts in maintaining data integrity and security. Thus, the synergy between proactive security testing and increased awareness of such vulnerabilities can contribute to strengthening cybersecurity defenses across organizations.

METHODS

This research adopts a manual penetration testing methodology inspired by established frameworks such as the OWASP Testing Guide v4 and PTES (Penetration Testing Execution Standard). These frameworks provide structured phases for identifying and analyzing security vulnerabilities in web applications. As illustrated in the figure, the research process includes five main stages: Application Analysis, Exploitation of the Application, Impact Evaluation, Solution Finding, and Reporting. Each phase aligns with best practices in ethical hacking and vulnerability assessment. By leveraging these references, the study ensures a comprehensive and systematic approach to uncover and address security flaws effectively.

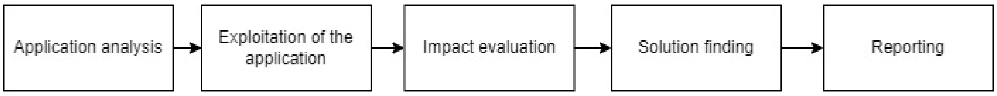


Figure 3. Methods diagram

A. Application analysis

This analysis aims to identify potential attack vectors by examining components vulnerable to exploitation through an in-depth inspection of local storage and HTTP responses [11]. In this analysis, the author performs checks using several well-known application debugging tools, specifically DevTools. DevTools, or Developer Tools, is a highly useful tool for conducting in-depth analysis of web applications, particularly in the context of security testing. By using DevTools, the author can examine key elements such as HTML, CSS, and JavaScript, which are the main components of the application's interface and interactions. One of the most important features is the Application tab. In a security context, the Application tab allows the author to deeply inspect data stored on the client side, such as local storage and cookies. This feature is crucial because it stores various information that may be sensitive, such as authentication tokens, user preferences, or other data that could be exploited if not properly managed.

B. Exploitation of the application

After successfully identifying weak points vulnerable to attacks, the author discovered a vulnerability that allows privilege escalation. To demonstrate the potential for this exploitation, the author conducted a simulated attack using a dummy account as a proof of concept. With the help of debugging tools such as DevTools, the author performed an in-depth analysis of the application's responses and the behavior of client-side stored data. One technique used was manipulating data in Local Storage, where the author stored and modified sensitive information, such as authentication tokens or user parameters, which then enabled the escalation of access rights from a regular user to an admin user.

This process began by examining the data stored in Local Storage and evaluating whether there were any vulnerabilities to modify certain values associated with user access rights. After modifying this data, the author retested the application to confirm that the changes successfully escalated the access level without going through a valid authentication process. This simulation demonstrated that the application's vulnerability could be exploited to gain higher access, potentially compromising the overall security of the system.

C. Impact evaluation

After conducting the attack simulation, the next step is to thoroughly evaluate the impact caused by the discovered vulnerability. This evaluation is essential to understand the extent to which the vulnerability could affect the system and its data. The author evaluated the impact of the privilege escalation vulnerability found on the XYZ website. This evaluation not only focused on the potential for unauthorized access takeover but also examined the effect on the system's dependence on data integrity and how this vulnerability could be exploited in long-term attacks [12]. By understanding the system's reliance on data integrity, the author could assess the potential damage that could be caused by exploiting this vulnerability.

D. Solution finding

After the evaluation process is completed, the author is required to compile a detailed report on the security vulnerabilities found. This report must include comprehensive information about the type of vulnerability discovered, its impact on the system, and the recommended mitigation steps. The report aims to provide clear guidance for web developers so they can understand the issues faced and the proposed solutions [13]. Even though the website developers may not have specialized expertise in cybersecurity, the presentation of the information should be simple yet technical, ensuring it can be effectively implemented.

E. Reporting

After the solutions are thoroughly documented in the report, the document is sent for immediate action by implementing the recommended fixes on the affected website. This stage involves communication and collaboration with the Head of IT responsible for the website's technology, ensuring that the suggested improvements are effectively applied.

he security evaluation process on the website must be conducted with official permission and in compliance with applicable regulations. The author has adhered to ethical principles in handling vulnerabilities, including reporting findings directly to the authorities responsible for managing the website. Additionally, the author has provided relevant recommendations to address the vulnerabilities found, ensuring that the website's security is maintained according to expected standards.

RESULT AND DISCUSSION

The privilege escalation vulnerability discovered in this study aligns with findings from several previous research efforts. Similar to the work of Firdaus and Voutama, who identified broken access control vulnerabilities that allowed identity impersonation, our research demonstrates how client-side storage manipulation can lead to unauthorized access escalation. The following outlines the method the author used in the analysis and testing of the XYZ website, consisting of several crucial stages, from application analysis, application exploitation, impact evaluation, to solution finding and reporting. Each stage was conducted with a systematic approach to ensure that every security vulnerability could be thoroughly identified and evaluated. In this analysis, the author used a manual method, allowing for an in-depth evaluation of each web application component. This approach provides flexibility in identifying vulnerabilities that may not be detected by automated tools and enables a more comprehensive understanding of the risks faced by the website. The exploitation results provide a clear picture of the level of risk involved, which then serves as a basis for giving effective recommendations for improvement.

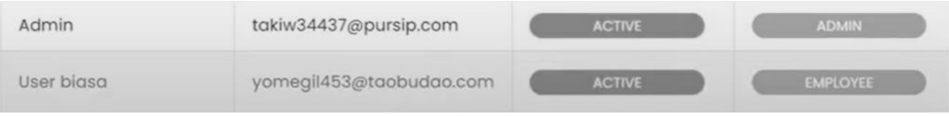
A. Application Analysis

In the initial stage of this research, the author conducted a series of tests on the application to understand the logic and process flow implemented. The main focus of these tests was to analyze the user registration steps leading to account activation, which enables users to manage users within their company. In the initial registration process, there are two different flows: one for users who will act as admins and another for invited employees.

The differences in the registration flow can be explained as follows: For admins, the process starts by filling out a registration form, followed by email verification and an OTP (One-Time Password) verification, before gaining access to the dashboard. Meanwhile, for users, they only need to click the link sent by the admin, set their password, and directly log into the company's dashboard created by the admin. Below are the two types of accounts the author created for testing.

In the initial stage of this research, the author conducted a series of tests on the application to understand the logic and process flow implemented. The main focus of these tests was to analyze the user registration steps leading to account activation, which enables users to manage users within their company. In the initial registration process, there are two different flows: one for users who will act as admins and another for invited employees.

The differences in the registration flow can be explained as follows: For admins, the process starts by filling out a registration form, followed by email verification and an OTP (One-Time Password) verification, before gaining access to the dashboard. Meanwhile, for users, they only need to click the link sent by the admin, set their password, and directly log into the company's dashboard created by the admin. Below are the two types of accounts the author created for testing.



Admin	takiw34437@pursip.com	ACTIVE	ADMIN
User biasa	yomegil453@taobudao.com	ACTIVE	EMPLOYEE

Figure 4. Two dummy account

To explain the context of the testing, it is important to understand the differences between admin accounts and employee or user accounts. Admin accounts have access to view the user list and other important features, including the ability to manage members and access a dashboard that displays complete information about members. Meanwhile, employee or user accounts have limited access and cannot open the dashboard or view the member list, as well as other important features that are only available to admins.

With this understanding, the attacker then used two dummy accounts, one acting as an admin, to begin testing features such as inviting employees or other users through the "Invite Employee" feature. However, at this stage, no significant vulnerabilities were found in that feature. The attacker then proceeded with testing using tools like DevTools for deeper inspection. They examined server responses, modified requests, and attempted to exploit other potential security gaps that may have been missed in the initial testing.

Next, the attacker utilized the employee/user account and opened DevTools. In this process, they used the "Inspect" feature to explore the website's elements. After that, the attacker navigated to the "Application" tab and then to "Local Storage" to check how data was stored by the application. At this stage, the attacker searched for various pieces of information stored in Local Storage, including cookies and other data, to look for potential security vulnerabilities. During this process, the attacker discovered interesting information stored on the website that could potentially reveal vulnerabilities that need to be addressed.

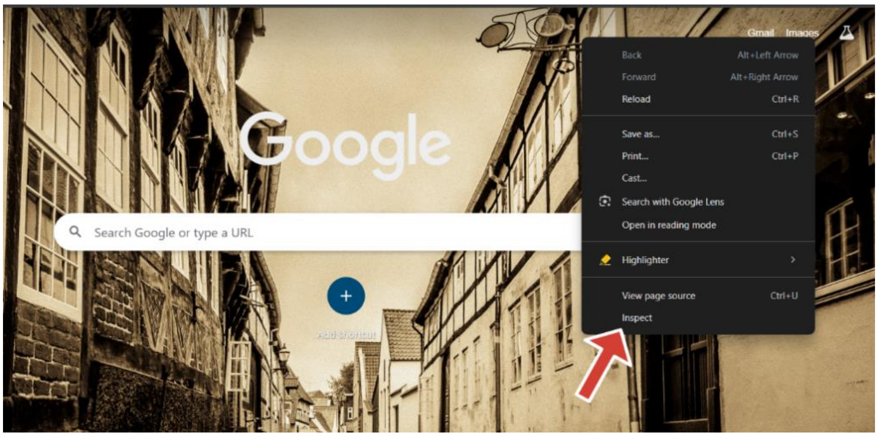


Figure 5. The first step to inspect Local Storage is to open DevTools.

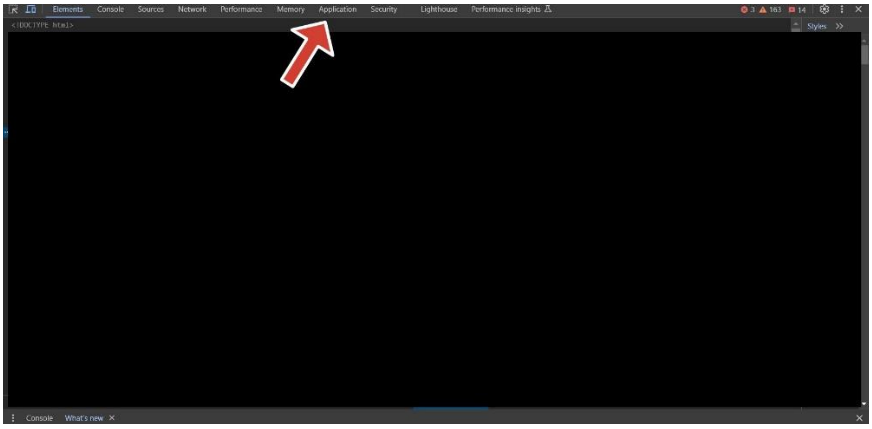
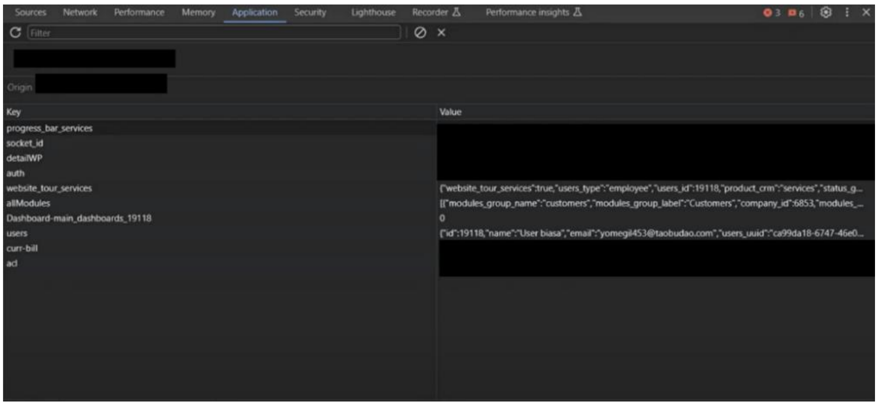


Figure 6. The second step to inspect Local Storage is to open the Application tab.



[redacted]

Origin [redacted]

Key	Value
progress_bar_services	[redacted]
socket_id	[redacted]
detailWP	[redacted]
auth	[redacted]
website_tour_services	{"website_tour_services":true,"users_type":"employee","users_id":19118,"product_crm":"services","status_g...
allModules	[{"modules_group_name":"customers","modules_group_label":"Customers","company_id":6853,"modules_...
Dashboard-main_dashboards_19118	0
users	{"id":19118,"name":"User biasa","email":"yomegil453@taobudao.com","users_uuid":"ca99da18-6747-46e0...
curr-bill	[redacted]
acl	[redacted]

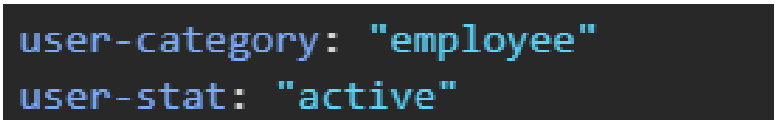
Figure 7. Information on how the application's local storage saves data.

Local Storage is a web storage feature that allows web applications to store data locally in the user's browser. The data stored in Local Storage is persistent, meaning that the information remains available even after the user closes the browser or refreshes the page. Local Storage offers a larger storage capacity compared to cookies and does not send data to the server with every HTTP request, making it suitable for storing information that does not require immediate transmission to the server. Local Storage uses key-value pairs, which makes it easy for developers to quickly store and access data. However, because the data is stored on the client side, there is a security risk if sensitive information is not managed properly.

During the testing process, the author discovered a specific key in Local Storage that stores user information, which caught their attention. This key holds important data such as names, email addresses, UUIDs, and roles assigned by the admin. This information becomes a target because the key provides direct access to various user details, which attackers can exploit for further exploitation. Therefore, the key that stores user information is considered a high-value target.

B. Exploitation of the application

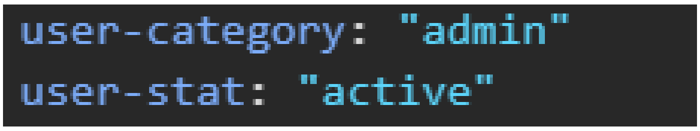
After conducting an in-depth analysis of the data storage mechanisms in local storage, the author discovered an intriguing parameter. The key "Users" stores various details about users, including their categories and active status. This finding provides new insights into how local storage can be used to efficiently store important information in web applications.



```
user-category: "employee"
user-stat: "active"
```

Figure 8. Vulnerability parameter

The attacker performed data tampering for privilege escalation by changing the user-category value from "employee" to "admin" (see Figure 9). With this change, the attacker successfully elevated their access level, allowing them to control features that are restricted to admins. After modifying and entering the data into the values of the user key, the page was refreshed, and the employee/user account was successfully converted into an admin account.



```
user-category: "admin"
user-stat: "active"
```

Figure 9. Tampering variables for exploitation

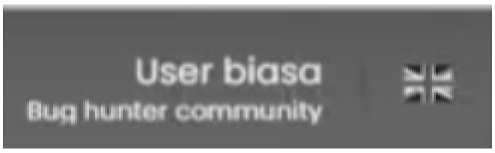


Figure 10. Before becoming Admin

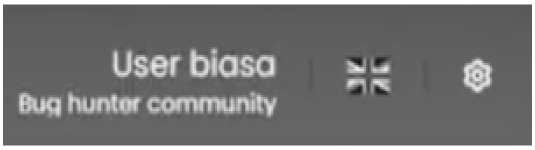
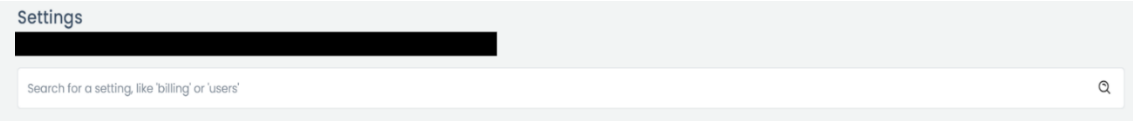


Figure 11. After becoming Admin

There is a fundamental difference between the access rights of regular users and admins, particularly in managing features within the application. One of the differences can be seen in the admin's ability to access additional settings, as shown in the image below, where the admin has full control to view billing, modify, and search for all users in the organization. This access rights allow the admin to modify the user structure and manipulate features that are not available to regular users.



Settings

[redacted]

Search for a setting, like 'billing' or 'users'

Figure 12. Features that are only accessible by the admin

In this situation, the existing vulnerability can be exploited by a bad actor to change their access rights from a regular user to an admin. This would allow the attacker to activate and access features that should only be available to admins, including the ability to manage other users, manipulate customer data, and modify critical system settings. Given the potential for privilege escalation, an effective solution is required to prevent unauthorized access escalation.

C. Impact evaluation

The vulnerability discovered on the website has significant potential to damage user trust in the platform. Attackers can exploit this weakness to perform various harmful actions, such as data manipulation, information theft, or service abuse. If this vulnerability is not addressed promptly, users may lose confidence in the platform's security, which could lead to a decrease in the number of users, a tarnished reputation, and potential financial losses for the company. Moreover, a successful exploitation could also open the door for further attacks, such as identity theft targeting other users.

This vulnerability could also cause long-term damage to relationships with business partners and other stakeholders, who may question the security and integrity of the platform. If sensitive or personal data is exposed, it could trigger legal and regulatory liabilities related to data protection, as well as increase costs for litigation and compensation to affected parties . With these wide-ranging impacts, it is crucial for the company to immediately address the vulnerability and implement corrective measures to restore user trust and protect their valuable assets.

D. Solution finding

By identifying the appropriate technical solutions, strengthening authorization mechanisms, and restricting access based on the principle of least privilege, the website will be better prepared to face various security threats, including privilege escalation attacks and other vulnerability exploitations. Additionally, implementing strict input validation and utilizing proactive monitoring tools can enhance early threat detection, enabling effective remediation before an attack causes significant damage to the system.The implementation of strategic mitigation measures will help reduce risks and prevent the exploitation of vulnerabilities. Moreover, regular updates and security training for the team can ensure that the platform remains resilient against evolving threats. With this comprehensive approach, website XYZ can minimize potential damage and protect its integrity and user trust.

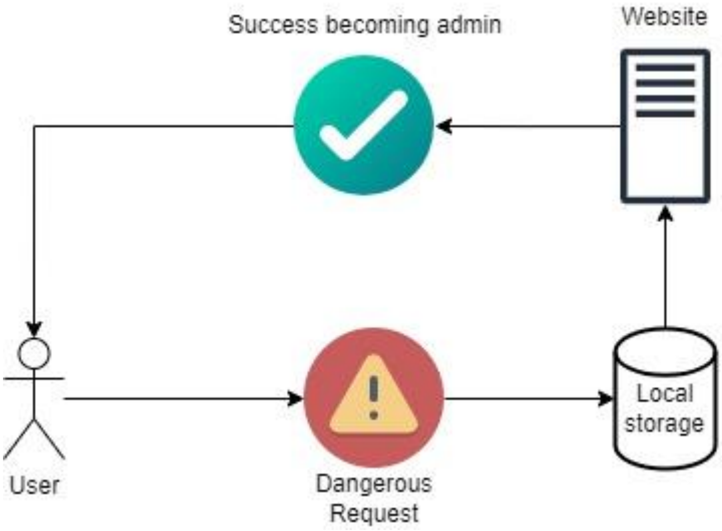


Figure 13. The website flow before fix were made.

In the image above, the website flow before improvements is shown, where the parameters received from local storage are processed directly without further validation. This approach is risky because user data is accepted as-is, without undergoing a verification process on the server, opening the potential for exploitation by malicious actors.

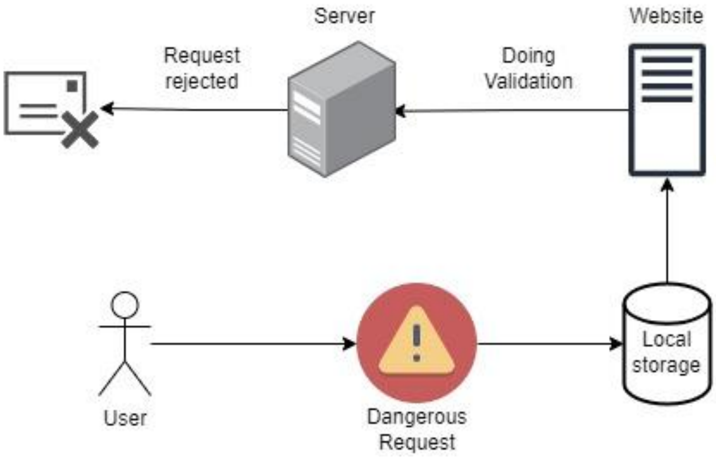


Figure 14. The website flow before fix were made.

After the improvements, significant changes were made by implementing server-side validation first. Every parameter received from local storage must now undergo verification and sanitization on the server before it can be used. This step ensures that every input received is truly safe and meets the established security standards, thereby minimizing the potential for misuse.

E. Reporting

In January 12, 2024, via WhatsApp, an IT employee at XYZ received a report from the Author regarding a vulnerability found on the website. This vulnerability allowed an employee or user to escalate their access rights to the admin level, opening features that should only be accessible to admins, as well as granting the ability to add or remove other employees or users within the company.

After evaluating and verifying the reported attack, on January 22, 2024, PT XYZ confirmed that the vulnerability was valid and expressed appreciation to the Author for the report. This validation indicated that the vulnerability report had a significant impact on the security of the company's ecosystem. PT XYZ reaffirmed its commitment to maintaining the security of its website and ensuring that user comfort and safety are top priorities. This step also serves as proof that the company is serious about addressing system weaknesses and preventing potential threats in the future.

The use of DevTools was sufficient to identify and exploit the vulnerability, allowing developers to debug from there and implement the necessary fixes. Each vulnerability has varying consequences and fixing methods. One of the steps taken by the developers to address this issue was to ensure that the website does not accept requests from users directly without verification. This verification is conducted on the server side to prevent further exploitation and enhance the overall security of the system.

On January 22, 2024, a retest was conducted, and the results showed that the vulnerability had been fully resolved and could no longer be exploited. This was achieved by implementing server-side verification to ensure that user input was not accepted directly without checking. This fix successfully eliminated the previous security gap, preventing the bug from being exploited for attacks or misuse.

CONCLUSION

The testing conducted on the XYZ website revealed a high-risk vulnerability that allows attackers to escalate access rights from regular users to admin, posing serious consequences such as loss of user trust, financial losses, and exposure of sensitive data. To mitigate this issue, it is essential to shift authentication to the server side, ensuring that any client-side changes, particularly those involving local storage, undergo validation by the server. Recommendations for developers include implementing server-side authentication, encrypting and validating local storage data, and conducting rigorous input validation and sanitization to prevent malicious data injection. Additionally, proactive security monitoring should be employed to detect threats early. Future research should focus on more in-depth manual penetration testing and explore aspects like availability and confidentiality, as well as the latest security technologies such as multi-factor authentication (MFA), to enhance overall website security.

REFERENCES

[1] A. R. S. Firdaus and A. Voutama, “Memanfaatkan Kerentanan Broken Access Control pada Website Orami untuk Membatalkan Pesanan dan Meniru Identitas Pengguna,” TeIKa, vol. 13, no. 02, Art. no. 02, Oct. 2023, doi: 10.36342/teika.v13i02.3113.

[2] F. Fachri, A. Fadlil, and I. Riadi, “Analisis Keamanan Webserver menggunakan Penetration Test,” J. Inform., vol. 8, no. 2, pp. 183–190, Aug. 2021, doi: 10.31294/ji.v8i2.10854.

[3] I. Riadi, Herman, and A. Z. Ifani, “Optimasi Keamanan Web Server terhadap Serangan Broken Authentication Menggunakan Teknologi Blockchain,” JISKA J. Inform. Sunan Kalijaga, vol. 6, no. 3, pp. 139–148, Sep. 2021, doi: 10.14421/jiska.2021.6.3.139-148.

[4] I. M. Edy Listartha, I. M. A. Premana Mitha, M. W. Aditya Arta, and I. Km. W. Yuda Arimika, “Analisis Kerentanan Website SMA Negeri 2 Amlapura Menggunakan Metode OWASP (Open Web Application Security Project),” SIMKOM, vol. 7, no. 1, pp. 23–27, Jan. 2022, doi: 10.51717/simkom.v7i1.63.

[5] E. I. Alwi, H. Herdianti, and F. Umar, “Analisis Keamanan Website Menggunakan Teknik Footprinting dan Vulnerability Scanning,” INFORMAL Inform. J., vol. 5, no. 2, p. 43, Aug. 2020, doi: 10.19184/isj.v5i2.18941.

[6] I. G. A. S. Sanjaya, G. M. A. Sasmita, and D. M. S. Arsa, “Evaluasi Keamanan Website Lembaga X Melalui Penetration Testing Menggunakan Framework ISSAF,” J. Ilm. Merpati Menara Penelit. Akad. Teknol. Inf., p. 113, Jul. 2020, doi: 10.24843/JIM.2020.v08.i02.p05.

[7] S. Utoro, B. A. Nugroho, M. Meinawati, and S. R. Widianto, “Analisis Keamanan Website E-Learning SMKN 1 Cibatu Menggunakan Metode Penetration Testing Execution Standard,” MULTINETICS, vol. 6, no. 2, pp. 169–178, Dec. 2020, doi: 10.32722/multinetics.v6i2.3432.

[8] A. Budiman, S. Ahdan, and M. Aziz, “ANALISIS CELAH KEAMANAN APLIKASI WEB E-LEARNING UNIVERSITAS ABC DENGAN VULNERABILITY ASSESMENT,” vol. 9, no. 2, 2021.

[9] Y. Arta, M. Ilhan, and A. Hanafiah, “Analisis Keamanan Informasi Aplikasi HRIS Dengan Metode SQUARE Pada PT. XYZ,” vol. 7, 2021.

[10] R. V. Aditama and E. S. Negara, “Pemindai Kerentanan Terhadap Website Jago Masak Dengan Metode Pengujian Penetrasi OWASP ZAP,” no. 03.

[11] Mira Orisa and M. Ardita, “VULNERABILITY ASSESMENT UNTUK MENINGKATKAN KUALITAS KEMANAN WEB,” J. Mnemon., vol. 4, no. 1, pp. 16–19, Feb. 2021, doi: 10.36040/mnemonic.v4i1.3213.

[12] M. D. P. Khairani, “Audit Web E-Government Dengan Acunetix Web Vulnerability Guna Menganalisis Dan Perbaikan Celah Keamanan Website,” vol. 9, 2024.

[13] D. Ariyana, S. Ningtyas, A. Fauzi, and R. Ramadhan, “Implementasi Metode Pemindai Online Untuk Menemukan Kerentanan di Server Website: Studi Kasus: website gramedia.com,” vol. 1, pp. 16–25, Jun. 2023, doi: 10.56855/jeep.v1i1.304.

[14] K. Isnaini, G. J. Nofita Sari, and A. P. Kuncoro, “Analisis Risiko Keamanan Informasi Menggunakan ISO 27005:2019 pada Aplikasi Sistem Pelayanan Desa,” J. Eksplora Inform., vol. 13, no. 1, pp. 37–45, Sep. 2023, doi: 10.30864/eksplora.v13i1.696.

[15] R. Yulia Andarini, P. Hendradi, and S. Nugroho, “MENINGKATKAN KEAMANAN TERHADAP SQL INJECTION STUDI KASUS SISTEM KEPEGAWAIAN BNN,” Indones. J. Bus. Intell. IJUBI, vol. 6, no. 1, Jun. 2023, doi: 10.21927/ijubi.v6i1.3161.

[16] W. Wahyudin, H. Kuswara, R. Resti, and S. Dalis, “Metode Vulnerability Assesment Dalam Pengujian Kinerja Sistem Keamanan Website Points of Sales,” Comput. Sci. CO-Sci., vol. 4, no. 1, pp. 44–52, Jan. 2024, doi: 10.31294/coscience.v4i1.2978.