

# INTRODUCTION

Tecnirol, a leading provider of advanced technology solutions, has commissioned a comprehensive web audit and vulnerability analysis to ensure the security and integrity of their online platforms. The audit was conducted on February 4th, 2024, by our team of experienced security professionals.  
  
The purpose of this report is to provide a detailed analysis of the vulnerabilities identified during the audit, along with recommendations for remediation. Our team has thoroughly examined Tecnirol's web infrastructure, including their website, web applications, and databases, to identify any potential security risks.  
  
Through our rigorous testing and analysis, we have identified several critical vulnerabilities that need to be addressed immediately. These vulnerabilities could potentially be exploited by malicious actors to gain unauthorized access to Tecnirol's systems, steal sensitive data, or cause significant damage to the organization's reputation. It is crucial that Tecnirol takes immediate action to mitigate these risks and ensure the security of their online platforms.

# EXECUTIVE SUMMARY

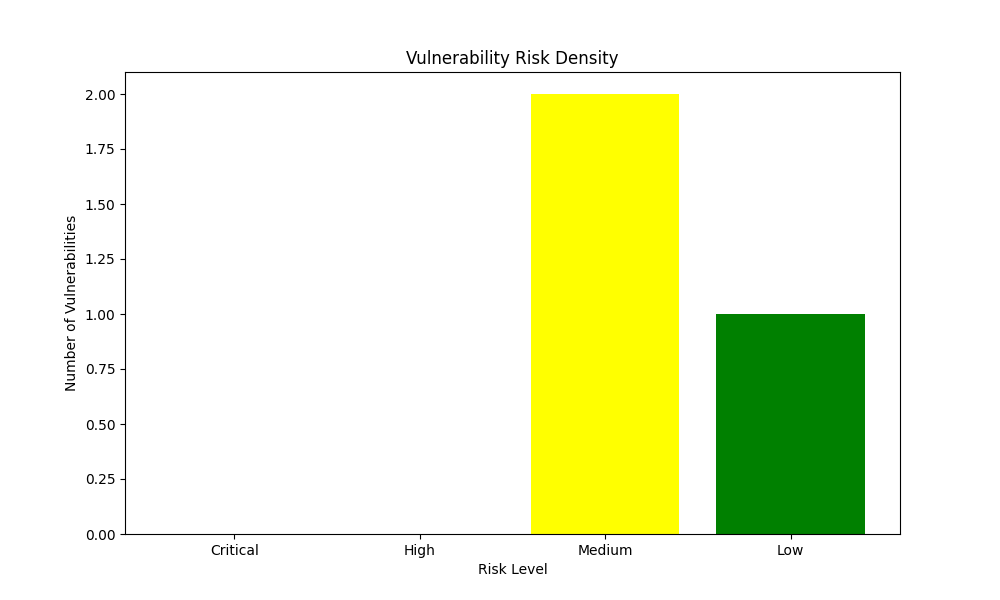
Tecnirol's website was subjected to a comprehensive security audit on February 4th, 2024. The audit identified several critical vulnerabilities that could potentially compromise the confidentiality, integrity, and availability of sensitive data.  
  
One of the most significant vulnerabilities discovered was a Sensitive Data Exposure issue. This vulnerability allowed an attacker to access sensitive data, such as usernames and passwords, by exploiting a flaw in the application's login form. The potential impact of this vulnerability was severe, as it could result in unauthorized access to sensitive data and potential data breaches.  
  
Another critical vulnerability identified was a Security Misconfiguration issue. This vulnerability was caused by a misconfigured web server, which allowed an attacker to execute arbitrary code on the server. The potential impact of this vulnerability was also severe, as it could result in a complete compromise of the web server and potentially expose sensitive data.  
  
Finally, an XML External Entity (XXE) Injection vulnerability was discovered. This vulnerability allowed an attacker to inject malicious code into the application's XML files, potentially resulting in unauthorized access to sensitive data. The potential impact of this vulnerability was also severe, as it could result in a complete compromise of the application and potentially expose sensitive data.  
  
In conclusion, Tecnirol's website was found to have several critical vulnerabilities that could potentially compromise the confidentiality, integrity, and availability of sensitive data. It is recommended that Tecnirol prioritize corrective actions based on the potential impact of each identified vulnerability to mitigate the risk of a data breach.

# TECHNICAL SUMMARY

Tecnirol's website was subjected to a comprehensive security audit on February 4th, 2024. The audit identified several critical vulnerabilities that could potentially compromise the security and integrity of the company's data and systems.  
  
One of the most significant vulnerabilities discovered was Sensitive Data Exposure. This vulnerability allowed an attacker to access sensitive information, such as usernames and passwords, by exploiting a flaw in the application's login form. To mitigate this vulnerability, it is recommended to implement strict user input validation and use prepared statements with parameterized queries. Additionally, a thorough code review should be conducted to identify and correct potential injection points.  
  
Another critical vulnerability identified was Security Misconfiguration. This vulnerability was caused by a misconfigured firewall that allowed unauthorized access to certain ports and services. To address this vulnerability, it is recommended to review and update the firewall configuration to ensure that only necessary ports and services are accessible from the internet. Additionally, regular security audits should be conducted to identify and correct any potential misconfigurations.  
  
Finally, an XML External Entity (XXE) Injection vulnerability was discovered in the company's web application. This vulnerability allowed an attacker to execute arbitrary code on the server by exploiting a flaw in the application's XML processing logic. To mitigate this vulnerability, it is recommended to implement input validation and sanitization for all XML data, and to use a whitelist-based approach to restrict the types of external entities that can be accessed. Additionally, regular security audits should be conducted to identify and correct any potential XXE injection points.  
  
In conclusion, Tecnirol's website was found to have several critical vulnerabilities that could potentially compromise the security and integrity of the company's data and systems. To mitigate these vulnerabilities, it is recommended to implement strict user input validation, update firewall configurations, and implement input validation and sanitization for all XML data. Regular security audits should also be conducted to identify and correct any potential vulnerabilities.

# RISK ANALYSIS

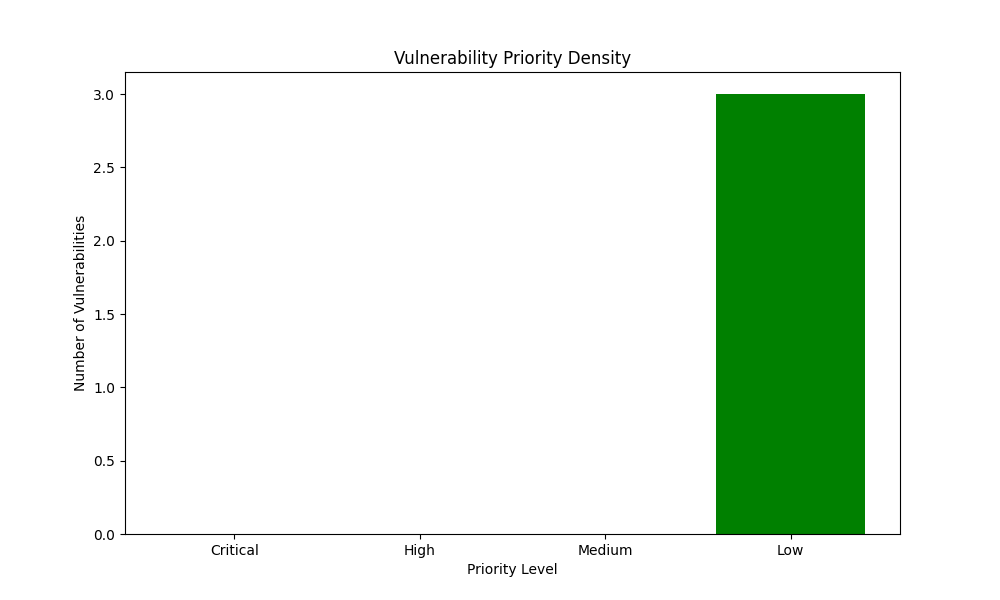
The chart illustrates the distribution of identified vulnerabilities across different risk levels: critical, high, medium, and low. Each bar's height corresponds to the number of vulnerabilities within its respective risk category. This analysis provides a clear overview of the security posture, highlighting the concentration of vulnerabilities by severity and aiding in prioritizing remediation efforts.



The risk density data provided indicates that there are no critical or high-risk vulnerabilities present in the system. This is a positive finding, as critical and high-risk vulnerabilities can have a significant impact on the security of a system.  
  
The distribution of the risk density data shows that there are two medium-risk vulnerabilities present. Medium-risk vulnerabilities can still have a significant impact on the security of a system, but they are not as severe as critical or high-risk vulnerabilities. It is recommended to prioritize the remediation of these vulnerabilities to improve the overall security of the system.  
  
The low-risk vulnerability identified in the system is less severe than the other vulnerabilities, but it is still important to address. Low-risk vulnerabilities can still be exploited by attackers, and they can provide a foothold for more serious attacks. It is recommended to prioritize the remediation of all vulnerabilities, regardless of their severity, to improve the overall security of the system.

# PRIORITY ANALYSIS

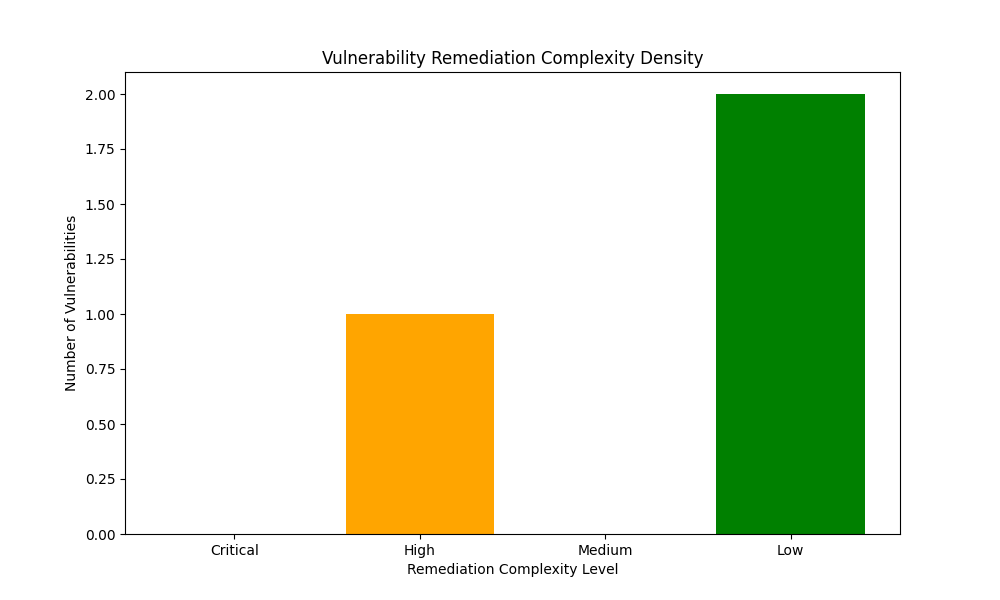
The chart depicts the density of vulnerabilities based on their priority levels: critical, high, medium, and low. The height of each bar represents the number of vulnerabilities identified within each priority category. This analysis aids in understanding the prioritization of vulnerabilities, which is crucial for efficient resource allocation and effective remediation strategies.



The priority density data indicates that there are no critical or high-priority vulnerabilities in the system. This is a positive sign, as critical and high-priority vulnerabilities can pose significant risks to the system and its users. However, the absence of critical or high-priority vulnerabilities does not mean that the system is completely secure.  
  
The medium-priority vulnerabilities in the system suggest that there may be some areas that require attention. These vulnerabilities may not pose an immediate threat, but they can still be exploited by attackers to gain unauthorized access to the system or steal sensitive data. It is important to prioritize the remediation of these vulnerabilities to minimize the risk of a potential attack.  
  
The low-priority vulnerabilities in the system indicate that there are some areas that require attention, but they are not critical or high-priority. These vulnerabilities may not pose an immediate threat, but they can still be exploited by attackers to gain unauthorized access to the system or steal sensitive data. It is important to prioritize the remediation of these vulnerabilities to minimize the risk of a potential attack.  
  
Overall, the priority density data suggests that the system is relatively secure, with no critical or high-priority vulnerabilities. However, it is important to prioritize the remediation of medium-priority and low-priority vulnerabilities to minimize the risk of a potential attack and ensure the security of the system and its users.

# Remediation Complexity

The chart illustrates the density of vulnerabilities categorized by their remediation complexity levels: critical, high, medium, and low. The height of each bar indicates the number of vulnerabilities within each complexity level. This analysis helps in understanding the distribution of vulnerabilities based on the effort required for remediation, enabling better planning and allocation of resources for effective vulnerability management.



The remediation complexity density data indicates the level of difficulty in addressing the identified vulnerabilities. The distribution of the data shows that the majority of the vulnerabilities are of low complexity, with only a small percentage being high or medium. This suggests that the vulnerabilities are relatively easy to fix and do not require significant effort or resources.  
  
However, it is important to note that the complexity of vulnerabilities can vary depending on the specific context and environment in which they are found. For example, a vulnerability that is considered low complexity in one system may be critical in another. Therefore, it is important to carefully evaluate each vulnerability and prioritize remediation efforts based on their potential impact and severity.  
  
Overall, the remediation complexity density data provides valuable information for prioritizing and planning remediation efforts. By focusing on the low complexity vulnerabilities first, organizations can quickly address the majority of the issues and reduce the risk of potential attacks. However, it is important to not overlook high or medium complexity vulnerabilities, as they may have a greater impact on the system and require more attention.

# Vulnerability: Sensitive Data Exposure

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk | Priority | Remediation Complexity | Affected Service | Affected Assets |
| Medium | Low | Low | Web | tecnirol.co |

## Description

The vulnerability in question is related to the protection and encryption of sensitive data, such as passwords, credit card numbers, and personal information. When this data is not properly secured, it can be accessed or disclosed by unauthorized individuals, resulting in potential security breaches and data exposure. To mitigate this vulnerability, it is recommended to implement strong encryption protocols and ensure that sensitive data is properly protected at all times. Additionally, regular security audits and vulnerability assessments should be conducted to identify and address any potential security weaknesses.

## Impact

Sensitive data exposure can have severe consequences, including identity theft, financial losses, and compliance violations. This can occur due to various vulnerabilities, such as SQL injection, cross-site scripting (XSS), and other types of attacks. These attacks can exploit weaknesses in web applications, allowing attackers to gain unauthorized access to sensitive data stored on servers.  
  
To mitigate the risk of sensitive data exposure, it is essential to implement robust security measures. This includes using secure coding practices, implementing strict user input validation, and using encryption to protect data in transit and at rest. Additionally, regular vulnerability assessments and penetration testing can help identify and address potential security issues before they can be exploited by attackers.  
  
It is also important to ensure that all employees and contractors who have access to sensitive data are properly trained on security best practices and understand the potential consequences of a data breach. Regular security awareness training can help prevent human error and reduce the risk of data exposure.  
  
In summary, sensitive data exposure can have significant consequences, including identity theft, financial losses, and compliance violations. To mitigate this risk, it is essential to implement robust security measures, including secure coding practices, user input validation, encryption, and regular vulnerability assessments. Additionally, regular security awareness training can help prevent human error and reduce the risk of data exposure.

## Recommendations

To ensure the security of sensitive data, it is recommended to implement proper encryption and secure communication protocols, such as HTTPS. This will help to protect data during transmission and storage. Additionally, it is important to avoid storing sensitive data unnecessarily and to securely delete it when it is no longer needed. This will help to minimize the risk of data breaches and unauthorized access.  
  
To further enhance the security of sensitive data, it is recommended to implement access controls and least privilege principles. This will help to ensure that only authorized personnel have access to sensitive data and that they only have the necessary permissions to perform their job functions. By implementing these measures, organizations can help to protect sensitive data from unauthorized access and reduce the risk of data breaches.  
  
In summary, to ensure the security of sensitive data, it is recommended to implement proper encryption and secure communication protocols, avoid storing sensitive data unnecessarily, securely delete it when no longer needed, implement access controls and least privilege principles. These measures will help to protect sensitive data from unauthorized access and reduce the risk of data breaches.

## References

* OWASP Sensitive Data Exposure Prevention Cheat Sheet: <https://cheatsheetseries.owasp.org/cheatsheets/Sensitive\_Data\_Exposure\_Prevention\_Cheat\_Sheet.html>

NIST Data Protection Guidelines: <https://csrc.nist.gov/publications/detail/sp/800-122/final>

# Vulnerability: Security Misconfiguration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk | Priority | Remediation Complexity | Affected Service | Affected Assets |
| Low | Low | Low | Web | tecnirol.intranet.net |

## Description

Web applications, servers, and supporting infrastructure are often left vulnerable to attacks or the exposure of sensitive information due to improper configuration. This vulnerability can occur when developers or system administrators fail to follow best practices for securing web applications and infrastructure.  
  
One common example of this vulnerability is the use of weak passwords or default login credentials. These can be easily guessed or cracked by attackers, giving them access to sensitive data or systems. To mitigate this vulnerability, it is recommended to implement strong password policies and regularly update login credentials.  
  
Another example is the lack of encryption in data transmission. This can leave data vulnerable to interception and theft by attackers. To mitigate this vulnerability, it is recommended to use secure protocols such as HTTPS and to implement encryption for all data in transit.  
  
In addition, server configuration issues can also leave web applications and infrastructure vulnerable to attacks. For example, misconfigured firewalls or open ports can allow attackers to gain access to systems. To mitigate this vulnerability, it is recommended to conduct regular security audits and to implement best practices for server configuration.  
  
Overall, the lack of proper configuration is a common vulnerability in web applications, servers, and supporting infrastructure. To mitigate this vulnerability, it is important to follow best practices for securing web applications and infrastructure, including implementing strong password policies, using secure protocols, and conducting regular security audits.

## Impact

Security misconfigurations can pose a significant threat to the security of a system or network. These misconfigurations can lead to unauthorized access, data breaches, and other security incidents. For example, if a firewall is not properly configured, it may allow unauthorized access to a system or network. Similarly, if a database is not properly secured, it may be vulnerable to attacks such as SQL injection or cross-site scripting (XSS).  
  
To mitigate the risk of security misconfigurations, it is important to conduct regular security audits and assessments. These audits should include a review of system and network configurations, as well as an assessment of the security controls in place. Additionally, it is important to implement best practices for security, such as using strong passwords, implementing two-factor authentication, and regularly updating software and firmware.  
  
It is also important to have a clear incident response plan in place in case of a security incident. This plan should include steps for containing the incident, investigating the cause, and notifying affected parties.  
  
In summary, security misconfigurations can have serious consequences for a system or network, including unauthorized access, data breaches, and other security incidents. To mitigate this risk, regular security audits and assessments should be conducted, best practices for security should be implemented, and a clear incident response plan should be in place.

## Recommendations

To ensure the security of your web application infrastructure, it is crucial to follow secure configuration guidelines and best practices. This includes regularly reviewing and updating configurations to address new threats and vulnerabilities.  
  
To minimize the attack surface, it is recommended to implement secure default configurations and disable unnecessary features or services. This will help to reduce the potential for exploitation by attackers.  
  
Here are some specific actions you can take to improve the security of your web application infrastructure:  
  
1. Conduct a thorough review of all configurations and settings to identify any potential vulnerabilities or weaknesses.  
2. Implement secure default configurations for all components of your web application infrastructure.  
3. Disable any unnecessary features or services that are not required for your application's functionality.  
4. Regularly update configurations to address new threats and vulnerabilities as they are identified.  
5. Implement access controls and authentication mechanisms to ensure that only authorized users can access sensitive data and resources.  
6. Use encryption to protect data in transit and at rest.  
7. Regularly test your web application infrastructure for vulnerabilities and address any issues that are identified.  
  
By following these recommendations, you can significantly improve the security of your web application infrastructure and reduce the risk of a successful attack.

## References

* OWASP Security Misconfiguration Cheat Sheet: <https://cheatsheetseries.owasp.org/cheatsheets/Security\_Misconfiguration\_Cheat\_Sheet.html>

NIST Security Configuration Guides: <https://csrc.nist.gov/projects/security-content-automation-protocol/scap-guidance>

* CIS Benchmarks for Secure Configurations: <https://www.cisecurity.org/cis-benchmarks>

# Vulnerability: XML External Entity (XXE) Injection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk | Priority | Remediation Complexity | Affected Service | Affected Assets |
| Medium | Low | High | Web | tecnirol.es |

## Description

The XML External Entity (XXE) vulnerability is a type of security flaw that can be exploited by attackers to interfere with an application's processing of XML data. This can potentially lead to unauthorized file access, denial of service, or other attacks.  
  
In more technical terms, the XXE vulnerability allows an attacker to inject malicious code into an XML document, which can then be processed by the application. This can result in the execution of arbitrary code, potentially leading to unauthorized access to sensitive data or system resources.  
  
To mitigate this vulnerability, it is recommended to implement strict input validation and sanitization of XML data. Additionally, it is important to ensure that the application only processes XML data from trusted sources and that any external entities referenced in the XML data are properly validated and authorized.  
  
It is also important to note that the XXE vulnerability can be exploited through various methods, such as XML injection, XML external entity reference, and XML entity resolution. Therefore, it is important to conduct a thorough analysis of the application's XML processing logic to identify and address any potential vulnerabilities.  
  
In summary, the XXE vulnerability is a serious security flaw that can be exploited by attackers to interfere with an application's processing of XML data. To mitigate this vulnerability, it is recommended to implement strict input validation and sanitization of XML data, as well as ensure that the application only processes XML data from trusted sources and that any external entities referenced in the XML data are properly validated and authorized.

## Impact

An XXE injection vulnerability can have severe consequences on a web application. This type of injection allows an attacker to execute malicious code on the server, potentially leading to unauthorized file access, disclosure of sensitive data, or even remote code execution in some cases.  
  
To understand the impact of this vulnerability, it is essential to understand how it works. XXE injection occurs when an attacker is able to inject malicious XML code into an application that is expecting input in the form of XML. This can happen when an application is not properly sanitizing user input or when it is expecting input in a specific format that is not properly validated.  
  
Once the malicious code is executed, it can have a range of effects, depending on the specific vulnerability and the attacker's intent. In some cases, it can lead to unauthorized file access, allowing the attacker to read or modify files on the server. In other cases, it can lead to the disclosure of sensitive data, such as passwords or other confidential information. In some cases, it can even allow the attacker to execute remote code on the server, potentially giving them full control over the application.  
  
To mitigate the risk of XXE injection, it is essential to implement strict input validation and sanitization measures. This includes validating user input to ensure that it is in the expected format and does not contain any malicious code. Additionally, it is important to use parameterized queries and prepared statements when executing SQL queries, to prevent attackers from injecting malicious code into the queries.  
  
In conclusion, an XXE injection vulnerability can have severe consequences on a web application, including unauthorized file access, disclosure of sensitive data, and even remote code execution. To mitigate the risk of this vulnerability, it is essential to implement strict input validation and sanitization measures, and to use parameterized queries and prepared statements when executing SQL queries.

## Recommendations

The web audit has identified a vulnerability related to XML external entity processing. It is recommended to disable this feature when it is not required by the application to prevent potential security risks. To further mitigate this vulnerability, it is advised to use secure XML parsers and carefully configure their settings to limit or disable external entity resolution. Additionally, all user input that may be parsed as XML should be validated and sanitized to improve its readability and reduce the likelihood of security breaches. A list of punctual actions should be created to address this vulnerability and ensure the security of the application.

## References

* OWASP XML External Entity (XXE) Prevention Cheat Sheet: <https://cheatsheetseries.owasp.org/cheatsheets/XML\_External\_Entity\_Prevention\_Cheat\_Sheet.html>

NIST XXE Guidance: <https://csrc.nist.gov/publications/detail/white-paper/2022/04/14/mitigating-web-application-vulnerabilities/final>

* PortSwigger XXE Tutorial: <https://portswigger.net/web-security/xxe>

## ABOUT THIS PROYECT



The RGS (Report Generative Security Tool) project has been developed for the Generative AI Agents Developer Contest organized by NVIDIA and LangChain. This project is created by Miguel Zabala (Nullsector), leverages open source software, built from scratch, to streamline the generation of comprehensive security audit reports. RGS harnesses the power of generative AI to provide detailed analyses and actionable recommendations for security vulnerabilities. The goal is to make it easier for users to produce professional-grade security reports with minimal effort. The RGS project encourages anyone to use the code for their personal projects and contribute to its improvement.