

**Finding Name: Use of Password Hash With Insufficient Computational Effort**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Team** | **Role** | **Project** | **Quality Assurance** | **Is this a re-tested Finding?** |
| Payas Paul | SCR | Senior Lead | Ontrack | Jaspriya k , Natalia K |  |
|  |  |  |  |  |  |

|  |
| --- |
| **Was this Finding Successful?** |
| Yes |

**Finding Description- Hashing**

: MD5 hash (used in ts-md5/dist/md5.Md5.hashStr) is insecure. Consider changing it to a secure hashing algorithm.

**Risk Rating**  
Impact: minor  
Likelihood: moderate

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Impact values** | | | | |
| **Very Minor** | **Minor** | **Significant** | **Major** | **Severe** |
| Risk that holds little to no impact. Will not cause damage and regular activity can continue. | Risk that holds minor form of impact, but not significant enough to be of threat. Can cause some damage but not enough to impede regular activity. | Risk that holds enough impact to be somewhat of a threat. Will cause damage that can impede regular activity but will be able to run normally. | Risk that holds major impact to be of threat. Will cause damage that will impede regular activity and will not be able to run normally. | Risk that holds severe impact and is a threat. Will cause critical damage that can cease activity to be run. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Likelihood** | | | | |
| **Rare** | **Unlikely** | **Moderate** | **High** | **Certain** |
| Event may occur and/or if it did, it happens in specific circumstances. | Event could occur occasionally and/or could happen (at some point) | Event may occur and/or happens. | Event occurs at times and/or probably happens a lot. | Event is occurring now and/or happens frequently. |

**Business Impact**

**Impact on the Company**

Risk of Password Cracking Increased: MD5 is regarded as unsafe and is vulnerable to hash collisions and brute-force attacks due to documented flaws. This implies that user accounts could be accessed and passwords hashed by an attacker.

Loss of Confidentiality: The confidentiality of sensitive user data may be compromised if an attacker manages to decipher hashed passwords and obtains unauthorized access to user accounts.

Reputational Damage: If unsafe hashing algorithms are used, there could be a security breach that harms your company's standing. Consumers might stop believing in your offerings, which would result in a decline in sales and clientele.

Problems with Regulatory Compliance: Insecure hashing algorithms may be used in violation of industry and regional regulations safeguarding user data (e.g., GDPR, HIPAA).

**Affected Assets**

User Accounts: Using MD5 for password hashing has a direct effect on user account security. Attackers may be able to access user accounts without authorization if passwords are cracked.

User Data: If hackers manage to access user accounts, they may be able to access user data contained in the program, which may include sensitive and/or personally identifiable information.

Application Integrity: If attackers are successful in using the insecure hashing algorithm to obtain unauthorized access or carry out other malicious actions, the application's overall integrity may be jeopardized.

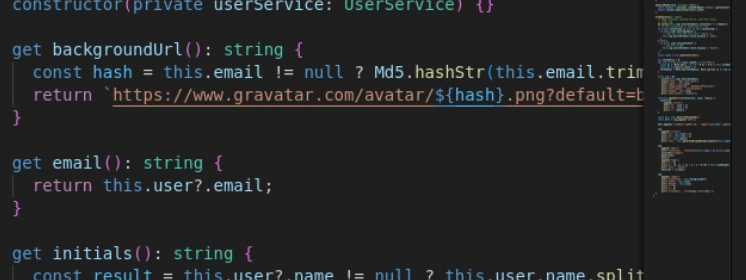
Reputation: If consumers believe that their data is not being sufficiently protected, confidence may be lost and there may be possible negative effects on the organization's company.

**Evidence**

**Step 1.**

In this case, the vulnerability is Path: src/app/common/user-icon/user-icon.component.ts, line 39

**Step 2.**

****

**Step 3.**

**Remediation Advice**

**Take these actions to rectify the password hashing using MD5:**

Change to a Secure Hashing Algorithm: Rather than using MD5, use a secure hashing algorithm made especially for hashing passwords, like bcrypt, Argon2, or PBKDF2. These algorithms offer a better level of security and are impervious to brute-force attacks.

Put Salting into Practice: Before hashing, use a different salt for every password to protect yourself from rainbow table attacks. Even when the passwords are the same, each hash is distinct thanks to the addition of a salt—a random value—to the password before hashing.

Upgrade Current Passwords: If your current passwords are hashed with MD5, you should think about coming up with a plan to switch them to the new, secure hashing method. This can entail offering users the option to reset their passwords or putting in place a one-time.

Update Code and Libraries: Make sure the new secure hashing method is used in lieu of every MD5 hashing instance in your codebase. Update any third-party libraries or dependencies that might be hashing passwords with MD5 as well.   
  
Inform Developers: Provide information to developers regarding safe password storage techniques and the significance of utilizing secure hashing methods. Educate people on the correct use of salting and hashing passwords.  
Security Testing: To make sure the new password hashing implementation is safe from assaults, perform security testing, including penetration testing and vulnerability assessments.   
  
Keep an eye out for Security Breach: Put monitoring and logs in place to find out about any strange activity connected to hashing passwords or possible security breaches. Investigate and address security incidents as soon as possible**.**

**References**

1. **OWASP Password Storage Cheat Sheet: The Open Web Application Security Project (OWASP) provides guidance on securely storing passwords, including recommendations for using bcrypt, Argon2, or PBKDF2 instead of MD5. OWASP Password Storage Cheat Sheet**
2. **NIST Special Publication 800-63B: The National Institute of Standards and Technology (NIST) provides guidelines for digital identity, including recommendations for secure password hashing algorithms.** [**NIST SP 800-63B**](https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63b.pdf)
3. **Password Hashing Competition: The Password Hashing Competition website provides information about modern password hashing algorithms, including bcrypt, Argon2, and others.** [**Password Hashing Competition**](https://password-hashing.net/)
4. **Mozilla Web Security Guidelines: Mozilla's Web Security Guidelines include best practices for secure password storage and recommendations for using secure hashing algorithms.** [**Mozilla Web Security Guidelines**](https://infosec.mozilla.org/guidelines/web_security)

**Contact Details**

Payas Paul (Senor Lead)

[612payaspaul@gmail.com](mailto:612payaspaul@gmail.com)

**Pentest Leader Feedback.**

The lead will provide feedback to enact on.