Initial Post

Display replies in nested form

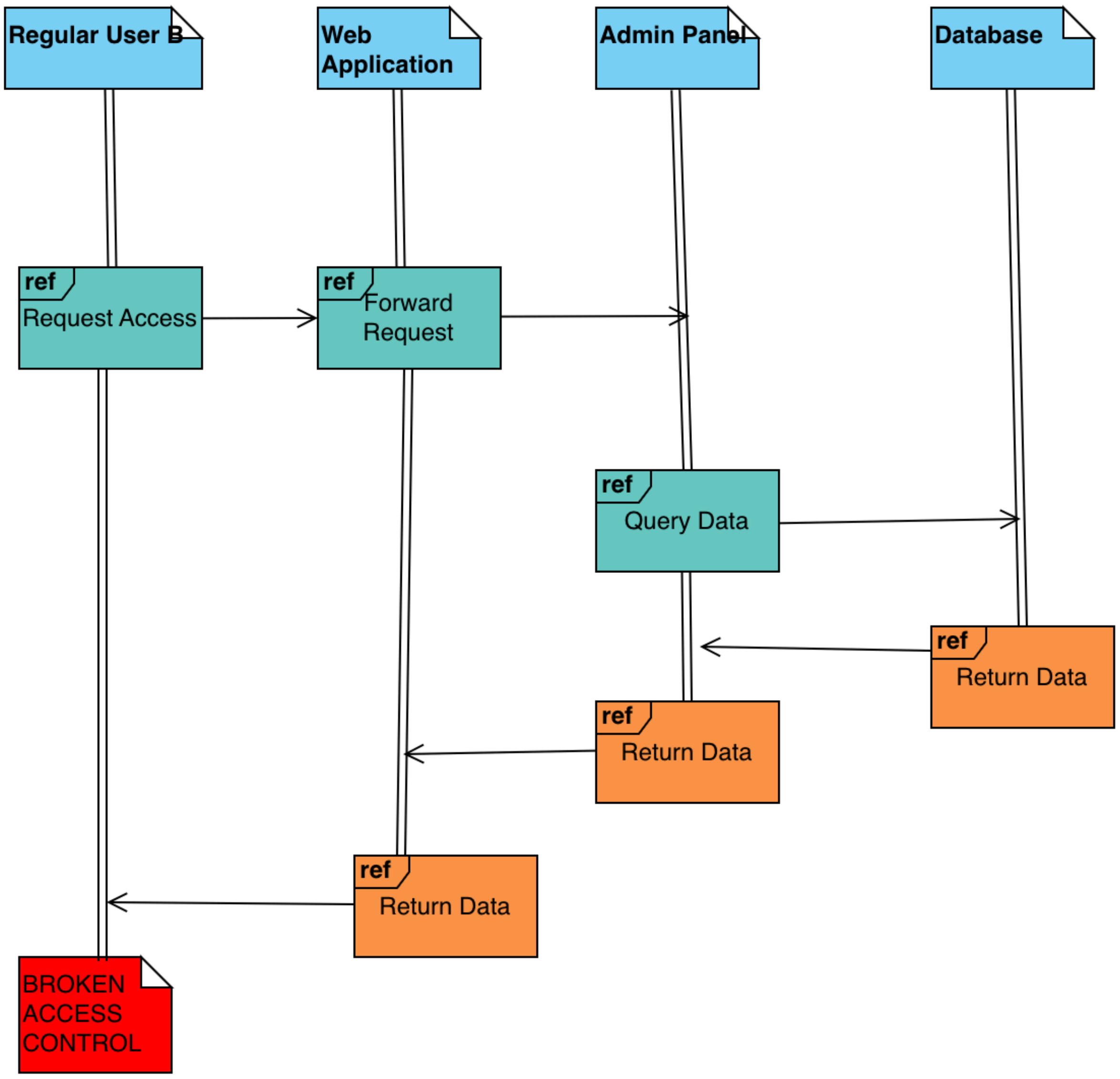
Settings ⌄

Initial Post

by Lauren Pechey - Monday, 5 August 2024, 8:02 PM

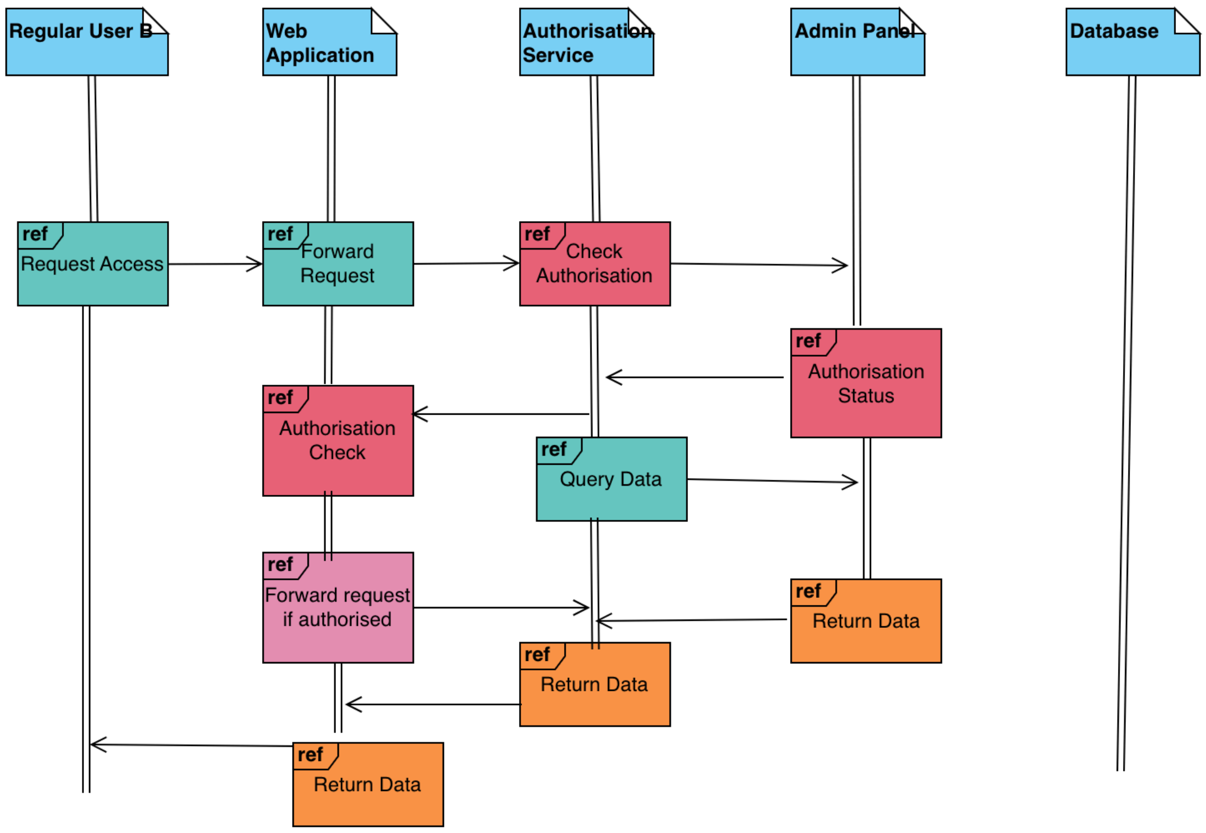
In this discussion, I have chosen to explore Broken Access Control (BAC), which has become the number one vulnerability exploited by cybercriminals today (OWASP, 2021). BAC occurs when restrictions on authenticated users are not implemented correctly, allowing attackers to access unauthorised data and cause serious security breaches (MITRE, 2021). This can allow cybercriminals to take control of user's accounts, access sensitive information and compromise an entire website by changing admin logins.

The following UML Sequence Diagram illustrates a serious example where BAC allows an attacker to access the admin panel and take control of the website:



In this scenario, the admin panel URL is protected by a simple URL path check, not proper authentication and authorization mechanisms. Therefore, a regular user, User B, discovers the URL of the admin panel through either simple guesswork or more advanced software engineering skills. For example, it could be as simple as adding "/admin" to the end of the URL: **https://example.com/admin**.

The application does not have robust access controls in place to verify that only users with admin privileges can access this admin URL. As a result, User B gains unauthorised access to the admin functionality, allowing them to view all user accounts, change passwords and permissions of other users, delete user accounts or even change administrator passwords. This clearly leads to a severe breach of system security and data privacy (Pillai, 2017). Therefore, as a solution, it is advisable for the website to implement strong authentication mechanisms, ensuring only users with valid admin credentials can access the admin panel (Gupta et al., 2022).



In the above diagram, the application employs authorisation checks that verify whether the authenticated user has the required administrative role before allowing access to admin functionality. Any attempts by non-admin users to access the admin panel are logged, and appropriate security measures are taken (e.g., alerts to system administrators), as suggested by Kellezi et al., (2021). In the resolved scenario, the application ensures that only properly authenticated and authorised users can access critical administrative features, significantly reducing the risk of unauthorised access and potential system compromise.

References:

Gupta, C., Singh, R.K. and Mohapatra, A.K. (2022) An approach for verification of secure access control using security pattern. *Wireless Communications and Mobile Computing*, 2022(1): 1-11. DOI: **https://doi.org/10.1155/2022/1657627**

Kellezi, D., Boegelund, C. and Meng, W. (2021) Securing open banking with model-view-controller architecture and OWASP. *Wireless Communications and Mobile Computing*, 2021(1): 1-13. DOI: **https://doi.org/10.1155/2021/8028073**

MITRE. (2021) *CWE Common Weakness Enumeration: A community-developed list of SW & HW weaknesses that can become vulnerabilities.* Available at: **https://cwe.mitre.org/data/definitions/1028.html** [Accessed 5 Aug. 2024].

OWASP. (2021) *OWASP Top Ten*. Available at: **https://owasp.org/www-project-top-ten/** (Accessed: 5 August 2024).

Pillai, A.B. (2017) *Software architecture with Python: Design and architect highly scalable, robust, clean, and high performance applications in Python*. 1st ed. Birmingham: Packt Publishing.

Maximum rating: -

Permalink **Reply**

Peer Response

by Hristo Todorov - Saturday, 10 August 2024, 1:37 PM

Dear Lauren,

Thank you for choosing to cover Broken Access Control.

The first UML diagram showcases an example of an access control system with lacking security very well. The diagram is very easy to read and understand, and the explanations cover all the necessary details and clarify why this is a poor approach to access control.

The second UML diagram is also very informative in showcasing a more secure access control system. The sequence is a bit more difficult to track because of the added element of the authorization system, perhaps it would help to number the steps accordingly. Despite this, the description is clear and concise and presents the solution well.

It is important to design authentication systems in order to prevent the former scenario presented, as that could lead to theft or leaking of data and hijacking of the entire system. You have correctly identified that putting authentication measures into place, coupled with logging attack attempts in order to prepare administrators to take the necessary defensive actions.

Best,

Hristo Todorov

Permalink Show parent Reply

Re: Initial Post

by Todd Edge - Monday, 12 August 2024, 1:32 PM

Peer response

Hi Lauren,

Your choice of security weakness is a matter close to my heart, working as an IT technician! Your explanation of the weakness is comprehensive and helps me to make sense of your flowcharts. I like how you have demonstrated both broken and complete access control here.

To improve - and this is something that I have only recently discovered myself - you could adhere more strictly to UML protocol by, for example, detailing the parts of the system (e.g. 'database') in simple rectangular shapes, rather than post-it blocks (which, I have recently discovered, are reserved for comments). I found this section on Wikipedia really helpful for me, and realised a few mistakes I made: https://en.wikipedia.org/wiki/Sequence_diagram

Kind regards,

Todd

Permalink Show parent Reply

Peer Response

by Timothy Brayshaw - Tuesday, 13 August 2024, 12:17 AM

Hi Lauren, I also covered Broken Access Control. This is a large category of vulnerabilities, and you have provided a concise and accurate definition. The scenario where a user discovers an admin panel through simple URL path manipulation is a very realistic example. It demonstrates how easily BAC can be exploited, by attackers, when proper access controls are not implemented.

You have discussed consequences of Broken Access Control and have mentioned unauthorised access and system compromise. Expanding this by mentioning wider issues such as potential financial losses, reputational damage, or legal implications, would strengthen your argument for the need to address this particular security issue.

It is great to see that you included a potential mitigation strategy for this attack. Ensuring that appropriate security settings are in place is crucial and can be reinforced through training as part of an organisation's policy.

Thanks for sharing your thoughts on Broken Access Control.

Tim.

Permalink Show parent Reply

Re: Initial Post

by Hamdan Almheiri - Thursday, 22 August 2024, 11:32 AM

Lauren, your discussion of broken access control is very relevant and well-explained. The example you gave about a user accessing an admin panel unauthorised really brings home the importance of this vulnerability. Strong authentication mechanisms should be in place for preventing this risk ensuring that only users who have logged on with admin credentials can actually access the admin panel. Moreover, integrating MFA can enhance this to a higher level of security by requiring more factors for verification to minimise the chances of unauthorised access.
Also, I would recommend periodic checking of access logs for any abnormal activity and running penetration testing to identify any vulnerability that an attacker could exploit before it happens. In addition, RBAC can be implemented to ensure users have the least number of permissions required for their tasks; this will minimise the potential impact in case one of the accounts gets compromised.
Your UML sequence diagram definitely is an excellent idea to picture the flow of events that leads to BAC. This diagram is not only clarify the problem but also help a little in designing systems more secure against such vulnerabilities.
References

• Gupta, P., Rathore, A., & Sharma, R. (2022). Strengthening Access Controls in Web Applications. International Journal of Information Security.
• OWASP Foundation. (2021). OWASP Top 10. Available at: https://owasp.org/Top10/

Permalink Show parent Reply

Peer response

by Craig Bourne - Wednesday, 16 October 2024, 2:11 PM

Hi Lauren,

This is an excellent analysis of Broken Access Control and shows why it is the top vulnerability in the OWASP Top 10.

I found your take on the admin panel vulnerability particularly insightful. Your use of sequence diagrams to illustrate the vulnerable system and the proposed solution is highly effective.

What I like about the sequence diagrams:

1. Clarity: Both diagrams depict the interaction between users and the system, making it easy to understand the vulnerability and its resolution.
2. Comparison: Presenting both the vulnerable and secure versions allows for easy comparison and highlights the specific changes that address the broken access control issue.
3. Real-world applicability: Your example of simply adding "/admin" to a URL is a practical and relatable scenario, effectively illustrating how easily broken access control issues can occur.

Possible improvements:

1. While implementing proper authentication and authorisation is crucial, you could explore additional security measures for a more robust solution:
- Multi-factor authentication: Implement an extra layer of security for admin access.
- Principle of least privilege: Ensure admin accounts have only the necessary permissions, minimising potential damage if compromised.

2. Error handling: I appreciate that you've mentioned logging attempts by non-admin users to access the admin panel. It might be valuable to visually represent this in your secure version diagram, showing how the system responds to and logs unauthorised access attempts. This could further demonstrate the proper logging and alert mechanisms you've described.

Your discussion on the importance of strong authentication mechanisms and proper access controls is excellent. It would be interesting to see how you might apply these principles to other potential vulnerabilities in web applications, such as API security or session management.

Overall, your post effectively demonstrates the critical nature of access control in secure software development.

Permalink Show parent Reply

◀ Initial Post