James Dermezis, Liam Malone, Will Lentini
CSS1006
Assignment 2

Each year, the Open Web Application Security Project (OWASP), a nonprofit organization aiming to improve software and application security, releases their ten most critical web application security risks. Using this list of critical web application security risks as well as online resources, tools, and training, OWASP strives to expand community outreach, knowledge, and understanding of web application security. This year, OWASP's list of critical web application security risks featured existing risks, such as Injection and Vulnerable and Outdated Components, as well as new risks, such as Insecure Design and Software and Data Integrity Failures. We feel that utilizing OWASP's critical risks for web application security as a standard can be beneficial for Facebooks's security as a whole. After careful consideration and analysis, broken access control is the most critical risk and can be applied to Facebook's web application security standard with some added details and policies.

According to OWASP, out of all critical web application risks, broken access control had the largest incidence rate of all applications tested, revealing the prevalence and severity of broken or vulnerable access control. In short, access control refers to the proper security requirements or authorization needed to access, modify, or update a resource. Access control can also be broken up into three different aspects: authentication, authorization, and accountability. The authentication aspect of access control incorporates validating the identity of the user, often using a username and password or key. This aspect is necessary because the user's given permissions and privileges depend on the specific user. For administrators, the authorization aspect of access control incorporates creating and managing roles given to users and granting/restricting privileges within roles. Overall, creating roles in an access control policy is beneficial because it allows for generalization, organizational structure, and dynamic management of permissions, such as those on basic and administrative levels. For example, a properly designed access control policy would have a default user role with very minimal necessary privileges and an administrative role with elevated privileges, allowing for proper access. Lastly, the accountability aspect of access control incorporates administrators or management monitoring and tracking activity, regardless of the user or action performed. In addition to helping monitor potential brute force attacks or others seeking to identify potential vulnerabilities, the accountability aspect allows administrators to verify that each user has appropriate access to each resource.

An organization's access control policy becoming vulnerable, broken, or failing, poses great risk to the organization, including day-to-day operations and potential confidential or sensitive information stored. If an organization's access control policy is vulnerable or broken, it will be possible for unauthorized elevation of privilege, administrative access, data manipulation,

tampering, or deletion. As a result of this significant threat of vulnerable or broken access control, we are recommending for Facebook to be proactive and take steps to enforce proper access controls. One way Facebook can easily enforce proper access controls is by designating web application/access control administrators. The role of these administrators will be to remain proactive in monitoring potential vulnerabilities in the access control and to readily monitor and update the policy as needed. These administrators will carry out their role and force all requests to be verified through access control checks. This fulfills the first part of access controls, which emphasizes authentication. In addition to this, administrators will properly manage, and update privileges or user roles as needed. This fulfills the second part of access controls, which emphasizes authorization and ensuring proper access. Lastly, administrators will monitor all access control events, regardless of user or action. This fulfills the third part of access controls, which emphasizes accountability and tracking user activity. This will allow for proper monitoring of potential brute force attacks or unauthorized investigation for vulnerabilities.

An organization as large as Facebook, it is very important to have good security to protect users. Broken access control can be extremely harmful to a business, and needs to be taken with great caution. Facebook is a massive company and it is so important for their websites to be secure. As mentioned, one major issue with broken access control is an attack gaining the rights to certain data, they should not have access to. If an attacker is able to compromise the website, and gain access to say someone's account, they could very well gain credit card details, addresses, phone numbers, and much more. All of these things are extremely valuable to both Facebook and the Facebook user. Along with that scenario there are many more. If an attacker is able to gain admin access it can be much worse. Depending on who the person they attacked, they might be able to gain access to private data. Having an administrative position compromised is one of the scariest parts about broken access control. Facebook has a massive database that has had problems with infiltrators before. Knowing this the company should be ten times more cautious with its data. With these admin permissions there can be a lot more vulnerabilities. There is a possibility the attack could have access to either add or delete things from the website. They might be capable of taking down websites causing mass amounts of revenue loss. Facebook is such a large company and website that having things deleted could be detrimental. When attacking a company, the attackers tend to try and gain access to as many accounts as possible. Since they know the website has the flaw they can easily gain access to multiple peoples accounts. Attackers will most likely be able to gain access to multiple administrative accounts which can be very bad news. By having access to so many accounts, they can start a denial of service attack. They can send bots from these accounts causing so much traffic within the Facebook servers. This might cause the website to crash, or possibly stop working for some time depending on the size of the attack. Facebook should be extremely well educated on how to identify an attack of this sort and be able to counter attack it. If the wrong people gain admin control of a website of this size, the users and company will be in trouble for a long time. They can cause lost revenue, leak massive amounts of data, and even knock the servers offline. Broken access control can be very harmful to companies and need to be taken very seriously.

Broken access control is one of the most common critical risks that occurs in the security field. According to OWASP, There are many vulnerabilities that can occur which make broken access control the web application risk with the highest incident rate. Vulnerabilities such as violation of the principle of least privilege, where access should only be available to certain people, but is instead available to anyone. Bypassing access controls by simply changing the URL, internal application state, or the HTML page, or by using a tool to modify API requests. Permitting the viewing or editing of someone else's account by providing its unique identifier. Accessing API with missing access controls such that you will not be able to submit entities to a specified source (POST), replace the current representation of the target resource with the request payload (PUT), and delete the specified resources (DELETE). Elevation of privilege, which is simply acting as if you have higher permissions than you actually have, such as acting as a user with no login information, or acting as an admin when you are just a user. Metadata manipulation, which is the tampering of JSON web tokens, access control token, cookie, or hidden field in hopes to manipulate privileges or JWT invalidation. CORS misconfiguration which is allowing API access from unauthorized and untrusted users. Last but not least, force browsing to authenticate pages as an unauthenticated user, or to privilege pages as a standard user.

With the many vulnerabilities that come with broken access control, there is prevention for each of them. According to the OWASP and CrashTest Security, there are eight ways to prevent broken access control. Access control is only useful and effective when the attacker cannot modify the access control check or metadata. One exception for this though, is public resources, you want to deny that by default. This is a good rule of thumb because denying by default is using the minimum privilege functions possible, meaning that only the people who are legitimate have permission to view, access, and modify documents. Another way to prevent broken access control is to implement access control mechanisms and reuse these mechanisms throughout the application. This is important because having a standard method for reading the effectiveness of the access control decisions helps us make decisions on what needs to be implemented to help us keep the integrity and security of our system. The third way to prevent broken access control is that access controls should enforce record ownership rather than the users that can edit or delete records. This means that the access control can associate each record with a user ID and the tasks they perform rather than allowing the user to modify any document they want. The fourth way to prevent broken access control is that application business limit requirements should be enforced by domain models. The fifth way to prevent broken access control is to disable web server directory listings and ensure file metadata and backup files are not visible within the web roots. This helps to protect against broken access control because web server directory will not be available, so a URL won't be requested to direct you to a specific website, and the metadata will not be able to be found and breached within the web roots because the metadata will be hidden within the website files. The sixth way to prevent broken access control is to keep logs on access control failures, and alert admins when appropriate, this is important because keeping logs on what failed will further help you fight off risks in the future.

The seventh way to prevent broken access control is to limit API and controller access, this helps to prevent automated attacking tools from causing harm to the system and possibly breaching and stealing information. The eighth and last way to prevent broken access control is to have JWT tokens be rather short lived because that allows for the least amount of time for an attacker to breach, which minimizes the chance of having a breach. In any situation where the JWT tokens are longer lived, it is recommended to follow OAuth standards to revoke access. This helps to prevent broken access control because the JWT tokens which are called access tokens have a certain amount of time until they expire, once they expire, a client application can choose to refresh the token so it can be accessed again. Refreshing the token allows for people to be able to access the sensitive information that is stored on that token, but when it expires, they are no longer able to access the information until a request is put out to refresh the token.

References

Arias, D. & Bellen, S. (2021, October 7). What Are Refresh Tokens and How to Use Them Securely. Retrieved December 2, 2021 from https://auth0.com/blog/refresh-tokens-what-are-they-and-when-to-use-them/#What-Is-a-Refresh-Token-

Common Weakness Enumeration (2021). CWE-284: *Improper Access Control*. Retrieved December 1, 2021 from https://cwe.mitre.org/data/definitions/284.html

Open Web Application Security Project. (2021). *OWASP Top 10- 2021*. Retrieved December 1, 2021 from https://owasp.org/Top10/

Open Web Application Security Project. (2021). *A01:2021 – Broken Access Control*. Retrieved December 1, 2021 from https://owasp.org/Top10/A01_2021-Broken_Access_Control/

Open Web Application Security Project. (2021). *OWASP Application Security Verification Standard*. Retrieved December 1, 2021 from https://owasp.org/www-project-application-security-verification-standard/

Sengupta, S. (2021). Broken Access Control and how to prevent it. Retrieved December 1, 2021 from https://crashtest-security.com/broken-access-control-prevention/