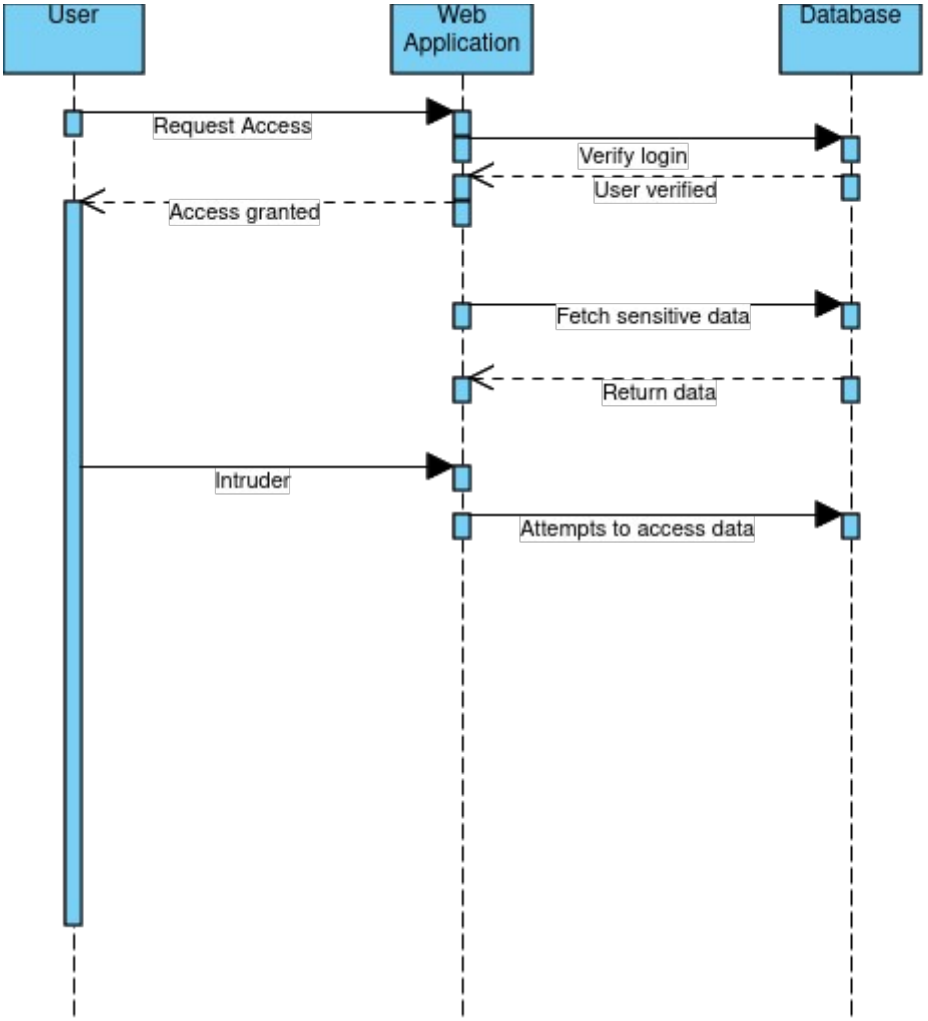
OWASP's Open Access Control

UML tool - Sequence diagram

OWASP - Open Web Application Security Project is an international organisation dedicated to website application security by initiating community-led open-source software projects aimed at improving the state of web security in the industry.

One of the initiatives is the Top 10 project, which tests multiple web applications for vulnerabilities and in turn, produce an awareness document revealing the most critical security risks to web applications.



According to the 2021 report, Broken access control ranks as the top security vulnerability seen in web applications, as 94% of applications were tested for some form of broken access control, having more occurrences in applications than any other vulnerability category (OWASP, 2021).

Access control mechanisms ensure that access to web resources, such as web pages and database tables, is restricted to authorized users only. These features are critical for preventing unauthorized access and protecting sensitive information from potential intruders. Hence, broken access control is an important vulnerability that cannot be overlooked as it exposes organisations to cyber espionage. It has been found in high profile applications such as IIS and Wordpress (Sharma, 2023).

One of the techniques often employed by intruders to exploit broken access control is broken session management. In this, intruders can carry out a number of session hijack through network sniffing or man-in-the-middle attacks, to intercept a user's session ID so as to impersonate them. Intruders could also inject malicious scripts into web pages to steal session cookies from users and use these stolen data, especially, on sessions that have a very long expiration time.

To illustrate the steps that may lead to a broken access control, a sequence diagram or a class diagram can be used:

SEQUENCE DIAGRAM:

A sequence diagram provides a dynamic view of the system, showing how the objects interact and this is helpful in illustrating the flow of data, including where they can be intercepted or altered. It can also be used to reveal points where requests are sent over the system.

Since sequence diagrams reveal interactions in a sequence, it is easy to observe the timing of events and detect potential vulnerabilities. It can also be used to analyse when and how sessions are initiated and terminated, which is crucial for identifying session hijacking vulnerabilities.

CLASS DIAGRAM:

Class diagrams provide a static view of the system by depicting the system objects (actors) and their interactions with the system.

A class diagram helps identify how different classes interact, the attributes and methods of each class, and what kind of access they possess. It also shows inheritance chains, which reveal points of potential vulnerabilities.

Although, this helps give a structural representation of the system interactions and reveals where sensitive data is stored, it is not dynamic and does not show the timing information of these interactions as revealed by the sequence diagram.

In their book on Penetration testing and network defense, A. Whitaker and DP Newman explained how session hijacking works; where the intruder takes the control of a valid session in which a valid user has successfully logged into the webserver and created a session between himself and the server. Upon successful hijack of the session, the intruder can then replay packets to the server, pretending to be the real user (Whitaker, A., & Newman, D., 2006). Some of the circumstances that can grant the intruder some access to data include:

1. Session ID predictability: If the session IDs generated by the server are predictable, an invader can guess possible session IDs and upon identifying a valid one, can hijack sessions. Session IDs can be stolen through sniffing, Brute force, Misdirected Trust using HTML injection or Cross Site Scripting (Kamal, P., 2016).
2. Denial of service attack: Intruders can take over active sessions by launching DoS attacks. First, they put themselves between the connection, sniffing data using packet capturing tools like Wireshark, then the intruder launches the DoS attack which could put the active user out of the connection while the intruder takes over the session.

Gaps in the system that can permit this include, but not limited to:

1. Weak session management, as illustrated in the predictable session IDs or using sequential session IDs discussed above.

2. Lack of session expiration or long session expiration time; this allows sessions to remain valid for longer periods of time, giving intruders enough time to hijack sessions.
3. Inadequate monitoring and logging.
4. Insecure storage of session data: Although, this is being worked on, it still remains an area of research (Alexei, LA, 2021). Storing session data in easily accessible URLs and unencrypted cookies makes a web system prone to session hijacking. In their report on cybersecurity in higher education institutions, Alexei et al showed that there are significant security threats and loopholes in cloud computing systems and Learning Management Systems through wireless access granted for distance learning.

References

Alexei, L.A. and Alexei, A., 2021. Cyber security threat analysis in higher education institutions as a result of distance learning. International Journal of Scientific and Technology Research, (3), pp.128-133.

Kamal, P., 2016. State of the art survey on session hijacking. Global Journal of Computer Science and Technology, 16(1), pp.39-49.

OWASP Top 10, 2021; https://owasp.org/www-project-top-ten/

Sharma, P., 2023. A Deep Dive into OWASP Top 3 Security Risks.

Whitaker, A. and Newman, D.P., 2005. Penetration testing and network defense. Cisco Press.