Threat Modeling Report on LinkedIn – User Data Security [Web Application]

1. Introduction

Possible security problems with LinkedIn's website were looked into, particularly how well it protects user information, handles logins, and defends against common web attacks. A standard method for categorizing threats was used and ideas to fix them came up.

2. Scope

Potential security problems in these key areas were looked into:

- How well user's personal information is protected, and the risk of data being stolen or accidentally shared.
- Weaknesses in the website's code that could be used for attacks, like cross-site scripting, SQL injection.
- Who might try to attack the website, from outside hackers to people within the company, automated bots, and those who exploit the connections between the site and other services.

3. Potential Threat

The different kinds of people or programs that could cause problems were also considered:

- Hackers: People trying to break into the system and steal user information.
- Users with bad intentions: Individuals who might use LinkedIn in harmful ways, like setting up fake profiles or trying to trick people into giving away their information.
- People working at LinkedIn: Employees who might misuse the data they have access to.
- Automated programs: Computer programs designed to gather large amounts of user information from the site.

4. Possible Attack Options

Potential attack methods were considered:

1. Cross-Site Scripting (XSS): This is where someone might try to inject malicious code into the website, which could let them steal login details or take over a session.

2. SQL Injection (SQLi): This kind of attack could allow someone to get into the website's database and steal information.

3. Phishing and Social Engineering: This involves tricks like posting fake job offers or pretending to be someone else to get users to give up their personal information.

6. STRIDE Threat Analysis

Threat Category	**Violates**	**Examples**
Spoofing	Authenticity	An attacker steals the authentication token of a legitimate user and uses it to impersonate the user.
Tampering	Integrity	An attacker abuses the application to perform unintended updates to a database.
Repudiation	Non-repudiability	An attacker manipulates logs to cover their actions.
Information Disclosure	Confidentiality	An attacker extract data from a database containing user account info.
Denial of Service	Availability	An attacker locks a legitimate user out of their account by performing many failed authentication attempts.
Elevation of Privileges	Authorization	An attacker tampers with a JWT to change their role.

7. Impact Analysis

1.User data being leaked, which would violate people's privacy.

2.People's LinkedIn accounts being stolen, along with their login information.

3.LinkedIn suffering financial losses and damage to its reputation.

4.An increased risk of people being tricked by phishing scams and targeted cyberattacks."

8. Mitigation Strategies

1.Carefully check all user inputs: This helps prevent attacks like cross-site scripting and SQL injection.

2.Make logins more secure: This could involve things like multi-factor authentication (where you need more than just a password) and using secure login methods like OAuth.

3.Limit how often the API can be used: This helps stop people from abusing the API to gather data or cause other problems.

4.Regularly check for security weaknesses: This means constantly monitoring the system and hiring security experts to try and find vulnerabilities.

9. Conclusion

LinkedIn must continuously monitor and improve its security to protect user data and maintain trust. Implementing the suggested mitigations can significantly reduce threats.

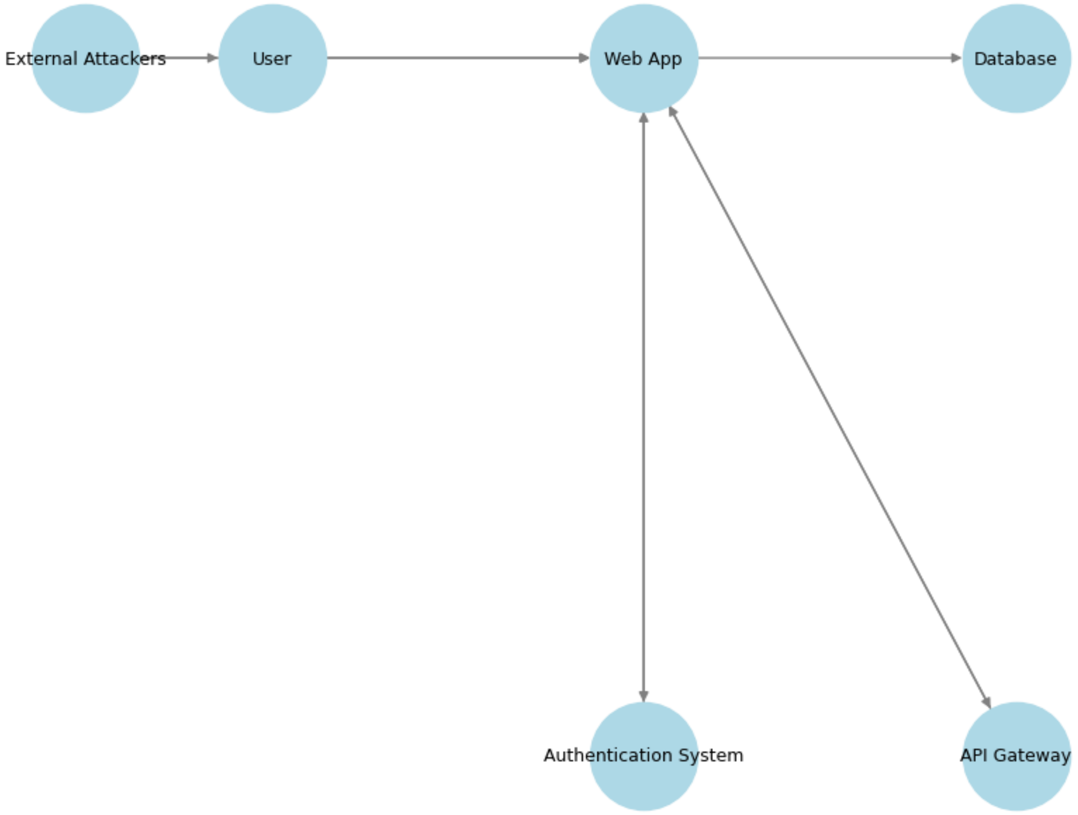
10. Reference

1. https://owasp.org/www-project-top-ten/

2. https://learn.microsoft.com/en-us/previous-versions/commerce-server/ee823878(v=cs.20)?redirectedfrom=MSDN

3. https://cheatsheetseries.owasp.org/cheatsheets/Threat_Modeling_Cheat_Sheet.html

An DFD Images for illustration of thread modelling of linkdIn.


External Attackers
User
Web App
Database
Authentication System
API Gateway