We Test Pens Incorporated

COMP90074 - Web Security Assignment 3

<Lihua Wang>
<1164051>

THREAT MODELLING REPORT FOR Bank of UniMelb Pty. Ltd. - WEB APPLICATION

Report delivered: 04/06/2021

1. Threats identified using STRIDE

S: Spoofing

Threat 1: <Pretend to be an ordinary user to login>

An attacker might obtain a login account credential by cheating and then pretend to be a legal user to access the web application. For example, due to improper logout from the devices, attackers can get access to the last web application session.

Threat 2: <Pretend to be a developer to access its resources>

On the Developer Login page, assume the web application has provided an identity authentication scheme. The attacker might abuse a poorly managed password policy or obtain the password from developers illegally. In case of password compromise, an adversary will be able to use the stolen password and gain access to the developer resources.

Threat 3: <Pretend to be an administrator to gain their privilege>

On the Developer Login page, assume the web application has provided an identity authentication scheme. The attacker might obtain an administrator account illegally and gain the administrator privilege.

T: Tampering

Threat 1: <Tampering users' profile>

Assume an attacker gained a credential illegally to login. They might tamper the User Profile which result in integrity of a data compromise in database.

Threat 2: <Tampering the password>

Assume an attacker gained a credential illegally to login which allows them to change the password on Settings page causing the users account taken over.

Threat 3: <Tampering the database>

An attacker who might be an administrator or gain a credential of admin privilege, might tampering the database without authentication.

R: Repudiation

Threat 1: <Deny activities on the web application by editing records>

Assume there existed record files about activities on the web application. An attacker who has the priority to operate the database might deny performing a destructive action on the application (e.g., deleting all records) by erasing or truncating log files for hiding their tracks. For example, the attackers might deny changing passwords, promoting user privilege, accessing sensitive files, and logging into developers or admin accounts.

Threat 2: <Deny activities on the web application by fulling of logs>

Assume the database will keep an access log which saves to storage space. An attacker might initial many trivial actions to make the log storge full, so that the attacker could conduct illegal actions without any records.

I: Information Disclosure

Threat 1: < Sensitive directories accessible >

Attackers might access the development directories and sensitive files by auto-scanner if the website backend data were not hidden, delete or restrict the accessing privilege. This will reveal the implementation details of the website, as well as the confidential data disclosed causing property damage of the bank.

Threat 2: < Users private information leakage >

Attackers might take over the user accounts by exploiting the vulnerabilities in changing password functionality on Settings page. This will also lead to user private information leakage.

Threat 3: < Confidential data only for admin/developers were disclosed>

Assume attackers have an account for ordinary users. Attackers might promote their accounts to administrators or developers by exploiting vulnerabilities on Admin / Developer Login page, which allows attackers access some resources only available for administrators or developers.

D: Denial of Service

Threat 1: < Denial of service due to large number of requests>

The attacker might command a bot army to flood the web application server with traffic. This may result in the web server denial of service. The huge amount of traffic hits the network, causing all available network resources to be exhausted, and finally causing legitimate user requests to fail. The attacker can prevent the server from providing service to a legitimate user. Besides, continuously send requests including accessing, or other

actions, forcing the number of IIS connections to exceed the limit. When the CPU resources or bandwidth resources are exhausted, the website is also crashed.

Threat 2: <Denial of service due to server taken over by attackers>

Assume an attacker have a normal user account. By exploiting the vulnerabilities on Admin, Developer Login pages, attackers could promote their privileges to administrator or developers, so that they could get control of the web application and make it crash (such as closing the database, destroy the configuration files).

E: Elevation of Privilege

Threat 1: <Promote ordinary user to administrator - horizontal>

Assume an attacker gain an account with ordinary user privilege. They might promote their account to admin privilege by exploiting vulnerability on the Admin page, so that attackers could access the admin resources and get control of the web application.

Threat 2: <Promote ordinary user to developer - horizontal >

Assume an attacker gain an account with ordinary user privilege. They might promote their account to developer privilege by exploiting vulnerability, for example, authenticate the developer password illegally on the Developer Login page, so that attackers could obtain the privilege of developer and control the web application.

Threat 3: <Access other users' profile without authentication - vertical>

Assume an attacker has an account with ordinary user privilege. Attackers might read other users' private information illegally by exploiting vulnerability on the User Profile page. Access the resources that not allowed to access is also a kind of elevation of privilege threat even if the obtained privilege is not admin.

2. Threat actors

S: Spoofing

Threat 1: <Attackers who do not have website login credentials>

Cyber-criminals / The adversary of the Bank of UniMleb

Threat 2: < Attackers who do not have developer password>

Cyber-criminals / The adversary of the Bank of UniMleb/ the ordinary user (client/staff) of the bank who doesn't have authority of accessing developer resources.

Threat 3: < Attackers who do not have admin credentials >

Cyber-criminals / The adversary of the Bank of UniMleb/ the ordinary user (client/staff) of the bank who does not have authority of administrator privilege.

T: Tampering

Threat 1: <Attackers who have login credentials>

Cyber-criminals / The adversary of the Bank of UniMleb/ the ordinary user (client) of the bank

Threat 2: <Attackers who have login credentials>

Cyber-criminals / The adversary of the Bank of UniMleb/ the ordinary user (client) of the bank

Threat 3: < Attackers who have higher privilege credentials >

Cyber-criminals / The high privilege users of the Bank, like administrator or developer.

R: Repudiation

Threat 1: < Attackers who have higher privilege credentials >

Cyber-criminals / The high privilege users of the Bank, like administrator or developer.

Threat 2: < Attackers who have login credentials >

Cyber-criminals / The adversary of the Bank of UniMleb/ the ordinary user (client/staff) of the bank

I: Information Disclosure

Threat 1: < Attackers who do not have any login credentials >

Cyber-criminals / The adversary of the Bank of UniMleb

Threat 2: < Attackers who have login credentials >

Cyber-criminals / The adversary of the Bank of UniMleb/ the ordinary user (client/staff) of the bank

Threat 3: < Attackers who have login credentials >

Cyber-criminals / The adversary of the Bank of UniMleb/ the ordinary user (client/staff) of the bank.

D: Denial of Service

Threat 1: < Attackers who do not have higher privileges >

Cyber-criminals / The adversary of the Bank of UniMleb/ the ordinary user (client) of the bank

Threat 2: < Attackers who have admin/developers' credentials >

Cyber-criminals / The high privilege users of the Bank, like administrator or developer.

E: Elevation of Privilege

Threat 1: < Attackers who have login credentials >

Cyber-criminals / the ordinary user (client) of the bank web application.

Threat 2: < Attackers who have login credentials >

Cyber-criminals / the ordinary user (client) of the bank web application.

Threat 3: < Attackers who have login credentials >

Cyber-criminals / the ordinary user (client) of the bank web application.

3. Threats remediations

S: Spoofing

Threat 1: <Additional authentication to confirm user identity>

If possible, make the web application can only be accessed via bank provided devices and ensure only the minimum services/features are enabled on devices to protect against attacker reusing the last session. Otherwise, it is hard to prevent an attacker to gain a credentials and spoof a normal user to login the website. The bank can only do is monitoring the users' activities on the application, and once suspicious actions detected like withdrawing cash or transferring money, requires additional identity authentication, for example, authorised captcha.

Threat 2: <Mitigations for spoofing developers>

Enable Multi-Factor Authentication for developers, not only authorise the password. It would be better that hidden the developer login portal. Assign another independent login portal for bank staff. Ensure that auditing and logging is enforced on the database so that the attacker might be easy to track.

Threat 3: <Mitigations for spoofing admins>

Enable Multi-Factor Authentication for admins. Ensure that do not use access tokens that provide direct access to the server. Ensure that auditing and logging is enforced on the database。

T: Tampering

Threat 1: <Mitigation for tampering User Profile>

Multi-authenticate the identity before allowing users editing the profile, not only use password policy.

Threat 2: <Mitigation for tampering credentials (password)>

Multi-authenticate the identity before allowing users change password, can use security answers or captcha.

Threat 3: <Mitigation for tampering the database>

Backup the database with encryption. Ensure that auditing and logging is enforced on the database to track suspicious activities.

Ensure that proper access database authorization is in place and principle of least privileges is followed. Implement dynamic data masking and encryption to limit sensitive data exposure non privileged users.

R: Repudiation

Threat 1: <Mitigation for deny activities>

Backup the database and ensure that log rotation and separation are in place, which will keep more records and make it updated.

Ensure that login auditing is enabled when accessing the database. And record all the actions of this web application. Collecting and storing audit data regularly and uploading the logs to storage for long term retention. Enable the following monitoring categories: Device identity operations, data modifying operations.

Threat 2: <Mitigation for repudiation due to logs exception>

Keep logs in the cloud storage so that it can extend storage space automatically.

Implement sufficient Audit Failure Handling. Ensure that the configuration can generate an exception when it fails to write to an audit log.

I: Information Disclosure

Threat 1: < Mitigation for sensitive directories leakage >

Realize real-time monitoring of web files. Monitor file integrity, read, write, delete, create, execute, link, rename.
Implement real-time protection of web files. Prevent website directory files from being maliciously tampered with, deleted and created.

Threat 2: < Mitigation for users' information leakage >

The application should review and verify the login host and IP information strictly when they try to access the website.
Using multi-authentication when changing the password.

Threat 3: <Mitigation for admin/developer resources leakage>

Change the token or password regularly, strengthen the strength of encrypted password. Apply the multi-authentication to identify the users.

D: Denial of Service

Threat 1: < Mitigation for tensor requests >

Establish a trust level for all visiting IP. When a DDoS attack occurs, the IP with a high level of trust has priority access, thus solving the identification problem.

Set up rules by installing a firewall, such as allowing or denying specific communication protocols, ports, or IP addresses. When the attack is sent from a few abnormal IP addresses, can simply use the denial rule to block all communications sent from the attack source IP.

Threat 2: < Mitigation for server taken over>

I would recommend restricting the privilege of developers and administrators. For example, any configuration modifying need to be admitted by other 2-3 administrators or get the permission from the super administrator.

E: Elevation of Privilege

Threat 1: <Mitigation for admin privilege escalation>

The best solution is implementing an independent portal for administrator or higher privilege users logging in.

Regularly ask authoritative audit institutions to audit the distribution and use of internal permissions.

Set up a reasonable permission distribution mechanism, such as applying for promoting the privilege requires multiple audits.

Threat 2: <Mitigation for developer escalation>

The similar mitigation from the above. It would be better to use another login portal for developer to reduce the possibility of buffer overflow exploits abusing the privileges of advanced users. Besides, apply multi-authentication methods to identity the user.

Threat 3: <Mitigation for user profile accessible problem>

Adopt the principle of least privilege (such as setting a policy that a user could not access others information without authorising). Have a strict mechanism to authenticate the network.