Doug Pham collaborated with Jack Huffman

Spoofing:
- An attacker could try to make a similar webpage that looks like our Tapirs Unlimited one that uses a very similar URL. If a user then goes to this page mistakingly, the attacker could get access to any sensitive information that user put in, for example, if there was a login popup and the user put their information in there, the attacker would then have their credentials. A corresponding mitigation for this scenario and the one below could be to require two-factor authentication.
- Closely related, website spoofing could start with phishing attacks. An attacker could create in a phishing email for example, that imitates the legit Tapirs Unlimited and provide a link to a spoofed website. Corresponding mitigation again could be using two-factor authentication

Tampering:
- Since Tapirs Unlimited will have features such as creating posts by users that get sent to the web server, attackers may take advantage of this by intercepting this communication from the client to the web server and being able to edit the original post and put something unwanted by the original user. Corresponding mitigation could be to use HTTPS for all the interactions.
- Building off the spoofing attack, if an attacker has the login credentials of a user, they then would also have to ability to delete or create posts maliciously. Corresponding mitigation could be to use two-factor authentication.

Repudiation:
- An attacker could make a malicious or harmful post to Tapirs Unlimited, something that does not follow the guidelines, and then claim that they did not make that post and says some other user must have made that post. A corresponding mitigation for this would be having a feature that keeps track of all interactions of logged in users, between the user and web server so that posts can be retraced back to the user.

Information Disclosure:
- An attacker could use a SQL injection within the search feature of Tapirs Unlimited that would show data that the user should not have access to. Corresponding mitigation for this would be to have the application code not use the input directly and remove any potential malicious code elements.
- Using methods in the spoofing section, an attacker could get the login credentials for an admin and could have access to very sensitive information of users and the company. Corresponding mitigation for this would be two factor authentication.

Denial of Service:
- An attacker could use a DDoS attack which would prevent users access from the web server. The attacker would flood the open ports of the web page with requests/packets and the user

would be denied access. Corresponding mitigations for this would be to limit the amount of traffic that can be sent to the web server, though you would have to be wary to not block legitimate traffic.

Elevation of Privilege:

- An attacker could try to impersonate a legit user and gain unauthorized access to sensitive data by trying to exploit APIs if it lacks proper authentication controls. Corresponding mitigation for this would be to implement secure API authentication and make sure the endpoints
enforce the proper authorization checks. Using HTTPS can also help protect the data in communication.
- An attacker could try to exploit a web server's PHP code which would allow them to execute arbitrary code and they can gain access to data that is not permitted. Corresponding mitigation for this would be to employ a firewall to help with attacks like these as well as doing regular code reviews and patches to minimize vulnerabilities.