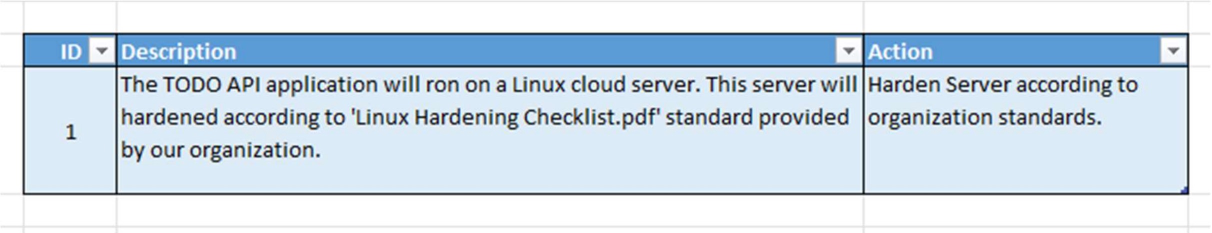
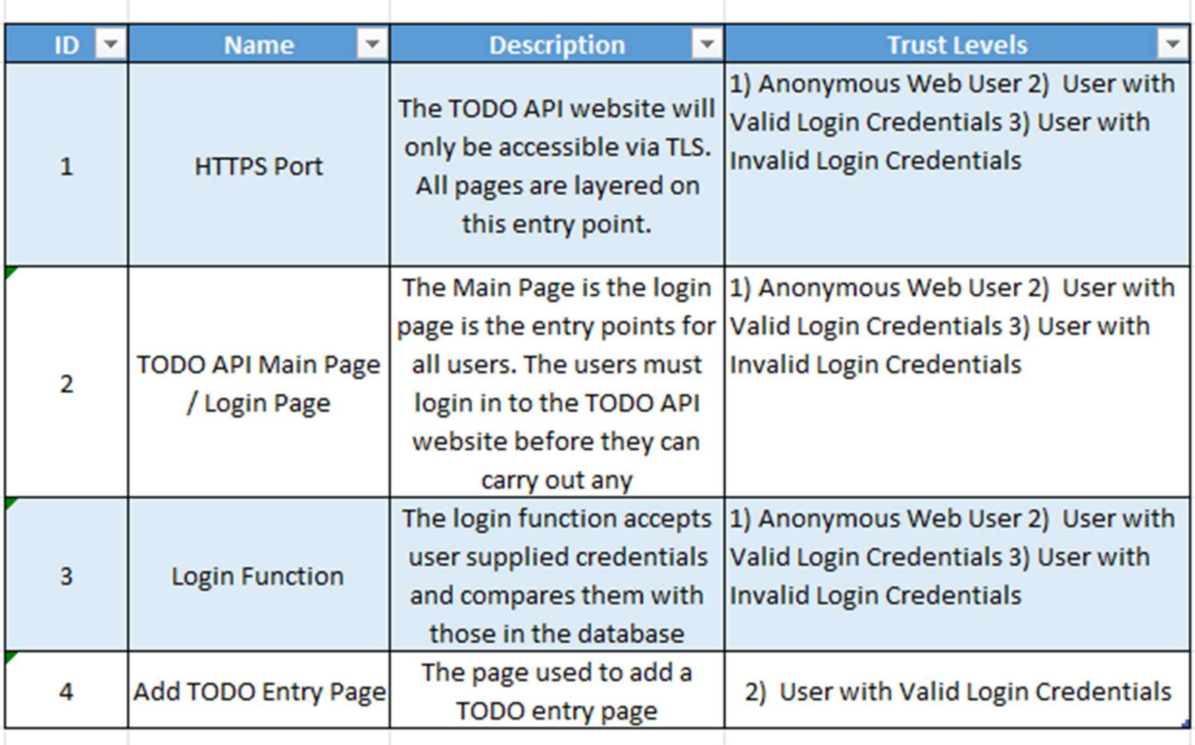
External Dependencies



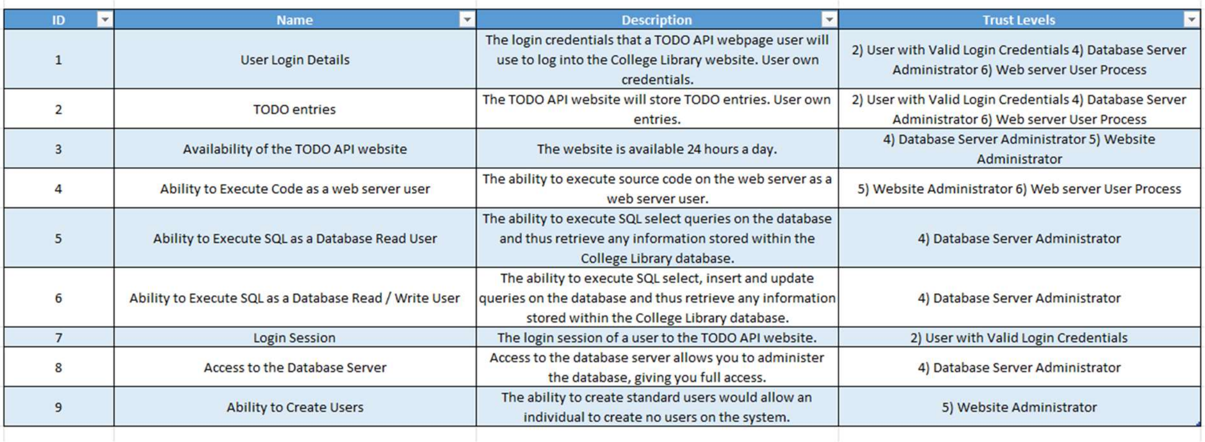
ID	Description	Action
1	The TODO API application will ron on a Linux cloud server. This server will hardened according to 'Linux Hardening Checklist.pdf' standard provided by our organization.	Harden Server according to organization standards.

Entry Points



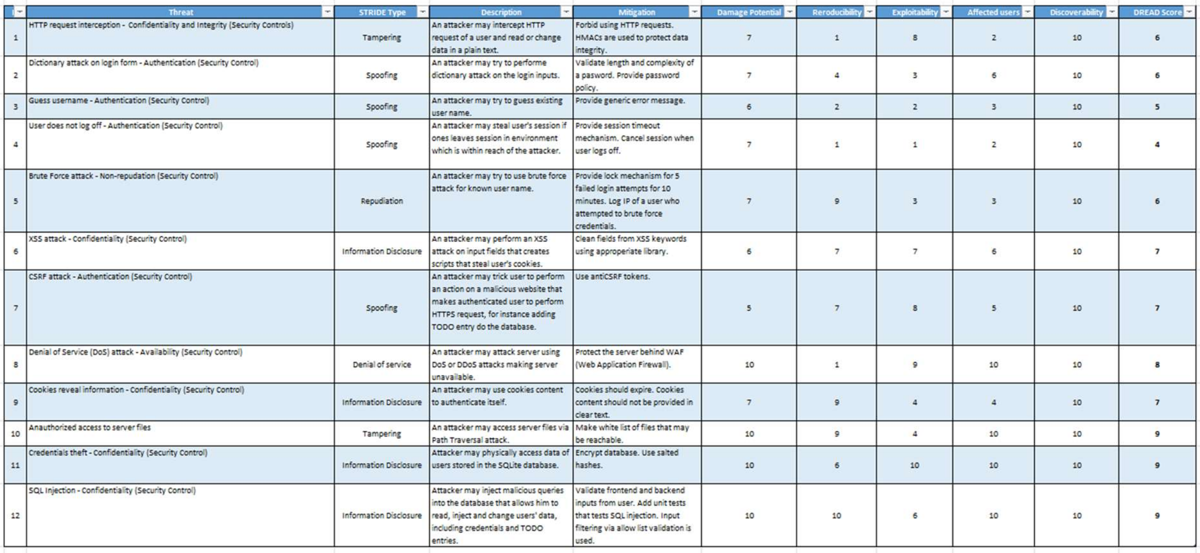
ID	Name	Description	Trust Levels
1	HTTPS Port	The TODO API website will only be accessible via TLS. All pages are layered on this entry point.	1) Anonymous Web User 2) User with Valid Login Credentials 3) User with Invalid Login Credentials
2	TODO API Main Page / Login Page	The Main Page is the login page is the entry points for all users. The users must login in to the TODO API website before they can carry out any	1) Anonymous Web User 2) User with Valid Login Credentials 3) User with Invalid Login Credentials
3	Login Function	The login function accepts user supplied credentials and compares them with those in the database	1) Anonymous Web User 2) User with Valid Login Credentials 3) User with Invalid Login Credentials
4	Add TODO Entry Page	The page used to add a TODO entry page	2) User with Valid Login Credentials

Assets



ID	Name	Description	Trust Levels
1	User Login Details	The login credentials that a TODO API webpage user will use to log into the College Library website. User own credentials.	2) User with Valid Login Credentials 4) Database Server Administrator 6) Web server User Process
2	TODO entries	The TODO API website will store TODO entries. User own entries.	2) User with Valid Login Credentials 4) Database Server Administrator 6) Web server User Process
3	Availability of the TODO API website	The website is available 24 hours a day.	4) Database Server Administrator 5) Website Administrator
4	Ability to Execute Code as a web server user	The ability to execute source code on the web server as a web server user.	5) Website Administrator 6) Web server User Process
5	Ability to Execute SQL as a Database Read User	The ability to execute SQL select queries on the database and thus retrieve any information stored within the College Library database.	4) Database Server Administrator
6	Ability to Execute SQL as a Database Read / Write User	The ability to execute SQL select, insert and update queries on the database and thus retrieve any information stored within the College Library database.	4) Database Server Administrator
7	Login Session	The login session of a user to the TODO API website.	2) User with Valid Login Credentials
8	Access to the Database Server	Access to the database server allows you to administer the database, giving you full access.	4) Database Server Administrator
9	Ability to Create Users	The ability to create standard users would allow an individual to create no users on the system.	5) Website Administrator

Threats risk scoring



I	Threat	STRIDE Type	Description	Mitigation	Damage Potential	Reroducibility	Exploitability	Affected users	Discoverability	DREAD Score
1	HTTP request interception - Confidentiality and Integrity (Security Controls)	Tampering	An attacker may intercept HTTP request of a user and read or change data in a plain text.	Forbid using HTTP requests. HMACs are used to protect data integrity.	7	1	8	2	10	6
2	Dictionary attack on login form - Authentication (Security Control)	Spoofing	An attacker may try to performe dictionary attack on the login inputs.	Validate length and complexity of a pasword. Provide password policy.	7	4	3	6	10	6
3	Guess username - Authentication (Security Control)	Spoofing	An attacker may try to guess existing user name.	Provide generic error message.	6	2	2	3	10	5
4	User does not log off - Authentication (Security Control)	Spoofing	An attacker may steal user's session if ones leaves session in environment which is within reach of the attacker.	Provide session timeout mechanism. Cancel session when user logs off.	7	1	1	2	10	4
5	Brute Force attack - Non-repudation (Security Control)	Repudiation	An attacker may try to use brute force attack for known user name.	Provide lock mechanism for 5 failed login attempts for 10 minutes. Log IP of a user who attempted to brute force credentials.	7	9	3	3	10	6
6	XSS attack - Confidentiality (Security Control)	Information Disclosure	An attacker may perform an XSS attack on input fields that creates scripts that steal user's cookies.	Clean fields from XSS keywords using approperiate library.	6	7	7	6	10	7
7	CSRF attack - Authentication (Security Control)	Spoofing	An attacker may trick user to perform an action on a malicious website that makes authenticated user to perform HTTPS request, for instance adding TODO entry do the database.	Use antiCSRF tokens.	5	7	8	5	10	7
8	Denial of Service (DoS) attack - Availability (Security Control)	Denial of service	An attacker may attack server using DoS or DDoS attacks making server unavailable.	Protect the server behind WAF (Web Application Firewall).	10	1	9	10	10	8
9	Cookies reveal information - Confidentiality (Security Control)	Information Disclosure	An attacker may use cookies content to authenticate itself.	Cookies should expire. Cookies content should not be provided in clear text.	7	9	4	4	10	7
10	Anauthorized access to server files	Tampering	An attacker may access server files via Path Traversal attack.	Make white list of files that may be reachable.	10	9	4	10	10	9
11	Credentials theft - Confidentiality (Security Control)	Information Disclosure	Attacker may physically access data of users stored in the SQLite database.	Encrypt database. Use salted hashes.	10	6	10	10	10	9
12	SQL Injection - Confidentiality (Security Control)	Information Disclosure	Attacker may inject malicious queries into the database that allows him to read, inject and change users' data, including credentials and TODO entries.	Validate frontend and backend inputs from user. Add unit tests that tests SQL injection. Input filtering via allow list validation is used.	10	10	6	10	10	9