FIRM VENDOR

Database

Application

Internet

Web Browser

* Application is used for storing and making purchases
* .net Application, Oracle database
* Username and password authentication
* Passwords are hashed with SHA1
* Username and password for database are stored as environment variables in application
* Chrome browser, frontend is built on Angular JS
* Communicates over HTTP

The diagram above provides an illustration of the client-server model along with various system components that shows the architecture of web-based tool that is used for storing and making purchases. Application architecture defines the various components and their interactions in context of a whole system. Architecture is based on 3-tier model in which client program makes requests to a server that executes the applications. The firm is using chrome browser and Angular JavaScript based front-end that is interacting with external application over the internet. The Application is based on .net which is using Oracle database to store information and to get access to database username and password are stored as environment variables in an application. Application is using one factor authentication that is username and password, but passwords are encrypted using SHA-1 algorithm. All the communication takes place over HTTP. In the current architecture, there are some risks to the system and it can be mitigated easily using correct implementation. The very first risk to this architecture is Man in the middle(MiTM) attack that occurs when a malicious user compromise confidentiality or integrity of data in transit. While the username and passwords can easily be intercepted while transmitted over internet using MiTM attack and it can mitigate if HTTPS is implemented which transmit data in an encrypted form rather than sending plain data over HTTP. Since risk is associated with Angular JavaScript, which is vulnerable to Cross Site Request Forgery (CSRF) which is an attack that forces an end user to execute unwanted actions on a web application in which they're currently authenticated. To mitigate such attacks CSRFGuard technique is used in which synchronized token checks has been done. Angular JavaScript has another vulnerability called JSON Hijacking or JavaScript Hijacking, which is the ability of hidden script that is loaded from one domain to read data transported from another domain. It can be easily mitigated if JSON request is sent along with specific string, that string is auto stripped while processing JSON requests. Another risk is the way username and password is stored as environment variables which is bad practice because data is stored in plain text (unencrypted) which can easily be read if the system gets compromised. To mitigate such risk credential vault technique should be used that makes it possible to store, manage and retrieve username and passwords for such database systems to properly authenticated users. Secure Credential Vault (SCV) uses Kerberos authentication. Kerberos is a network authentication protocol that allows individuals communicating over an unsecured network to prove their identity to one another in a secure manner. Some other precautions that can be done is upgrading hashing technique from SHA-1 to SHA-2 or SHA-3 since SHA-1 is no longer considered secure. One factor authentication (username and password) can be upgraded to two-factor authentication also known as two-step verification that requires not only username and password, but something that only, and only, that user has on them, i.e. physical token, secure soft token, biometrics (such as fingerprint, eye iris, voice). Another most common attack that is in action is Cross-Site Scripting (XSS) which is a type of vulnerability typically found in web applications. It enables an attacker to inject client-side script into legitimate, but variable Web pages, making it appear that the script has come from a trusted source. Due to such attacks, confidential information can be accessed and page content can be changed.