Final Project

Ms. Janani Tharmaseelan

**Executive Summary**

SQA Workout Inc. is a small startup company that has 10 employees, they are in the stage of gaining profit now from their web application which is connecting customers and trainers. The application provides workout routines, workout tracking, and health tracking for the subscribers and they can also view the videos and contents given by trainers. In this way, trainers could able to earn money. As a platform for joining the customers and trainers, SQA company is earning profit. Their whole business depends on the website. So maintaining a secured system is important for them.

The most possible vulnerabilities that could be identified from the infrastructure could be SQL injection, Missing data encryption, missing authentication, Cross-site scripting, URL redirection to untrusted sites, Weak passwords, password field masked, directory listing, and visibility of documents. 6 of these vulnerabilities, how they will be used against the company, and mitigation plan has been discussed further in the technical report. Not managing the website securely results in information theft, damaged client relationships, revoked licenses, and legal proceedings.

All the vulnerabilities mostly can be prevented by reviewing the application and scanning the application regularly and following organizational level security policy. According to the DREAD analysis, SQL injection has a high threat to their website. So having a proper mitigation plan for that should be a priority task for the SQA company.

Enforcing security best practices will improve the security of the web application, since the company has only 10 employees, making sure they all follow them and monitoring them would be easy. They need to ensure that they use strong passwords, force re-authentication when accessing sensitive information, provide better user privileges, use encryption and salting for passwords and confidential information. So following best practices from the development stage to the monitoring stage will provide a secured web application.

**Technical Report**

**Vulnerability 1: SQL Injection**

**describe in your own words**

SQL injection will allow the attackers to modify the SQL queries that the application has and will try to give access to data for unauthorized users. These data could be other users' sensitive data and also they should only be accessible to admin privileged users. The attacker might use this way to modify or delete those important data, destroy applications' stability, and also can attack a person.

**How might it impact the given company and situation?**

SQA Workout Inc has just started to get profit, and there are different types of users in the system. They are subscribers and fitness trainers. These users will be having different user credentials to log in to the website. An attacker might try to input illegal characters throughout the input fields and will try to modify the queries and will get access to the system.

An attacker may not have subscribed to the web application and may try to view the training without paying. So he may try to illegally log in to the application. This will be a loss of profit for the company if the attacker downloaded all the videos and sell them on a different platform. Where trainers' services will be misused by the attacker to earn profit.

An attacker may also try to log in as a trainer without giving correct login credentials, when he joined as a trainer, he may be able to modify the payment information, and also he can able to delete or modify the videos. That would result in a bad impact on SQA workout company. Their reputation and profit can be damaged as a growing company.

If the attacker could able to access the subscribers' he could be able to view their credit card / Debit Card / Payment gateway information. He can use that information for his usage.

So this would be a danger to company employees as well as for the subscribers.

**How might an attacker use this vulnerability to attack this company?**

As I mentioned above attackers can gain access to the company application as a subscriber to access the videos, view subscribers’ payment options, and information, Access their workout tracking information, routines, and health tracking information. He might threaten the company later saying their customers' information will be posted publicly or will be sold out to a third party. The attacker might threaten the company to pay them or shut down the company.

If an attacker has got employees access, He can modify or delete the videos, and also publish their videos on a different platform so he can earn profit.

The way the attacker tries to attack using this vulnerability will vary from each type of database and each type of query. The attacker will use this to read all the data in the database. He can also able to execute arbitrary commands

**Why would an attacker use it?**

By gaining the access to database and source code, the attacker could be able to access all the inside information, username, passwords, payment details, health tracking information, subscribers' workout plans, trainers’ video, and database connection details.

The attacker would use it to steal information and gain unauthorized access to the application which he can use to spoil the company’s reputation, earn a profit, or as an act of revenge.

The attacker can also use this attack to find more detail, which could be used to attack through different attacks.

**What opportunities might it present to the attacker?**

* The attacker can able to access the system as an authorized user.
* An attacker could also able to gain access to the source code and Database of the application
* The attacker will be able to determine the operating system and the system architecture.
* The attacker could use this as a way of doing another attack
* The attacker can use this to find more vulnerabilities

**Assess vulnerabilities with the formula of your choosing**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Damage Potential | Reproducibility | Exploitability | Affected Users | Discoverability |
| Value | 10 | 8 | 10 | 10 | 5 |
| Explanation | We can able to see that by doing this attacker could be able to gain access / full control of the system, source code, and Database. So the damage from this attack is high, so this should be considered as a critical vulnerability. | The way of doing the attack is different for types of databases and different queries. So this could not be done with one or two steps, or this can not be done with a web browser. So this can be assigned with 8 | Doing this attack needs more knowledge, Each database is different, and also attacker should be able to have knowledge about different queries, modify them, and also he needs knowledge about attacking the underlying architecture. Attacker should have able to think clearly and perform by reading the exception messages that appear in the application. So required technical knowledge and experience. | All the users will get affected when the application is controlled by an attacker. When the database got controlled by the attacker, the entire information will be at risk. | This can be discoverable by monitoring the system data and progress. |
| Rating : (10+8+10+10+5)/5 = 8.6 | | | | | |

**Provide mitigations and suggestions for addressing discovered issues. These should consider the company, its situation, and its resources.**

This needs to be avoided from the development stage when the team is writing code, they should adopt good programming techniques, User’s input data need to be sanitized, those data should have properly defined data types, Improper characteristics should not be accepted from the user’s site, When error messages are shown to the users It should not reveal any internal information such as table name or attribute names. Define the data types of the columns and make sure they are maintained at the code level. If a user input field is accepting only the whole numbers, make sure it doesn’t accept any floating values or strings. Define the input length that could be accepted by the user.

Other than the code level prevention Disable the system giving direct Database error messages, which will allow the users to see the database information like table names. So don’t include any detailed database error messages. Error messages will be given to the user not give direct information. Don’t include messages as “Access Denied”. So the attacker knows that if he is looking for a file, then the file is available only the thing he needs to fix, he needs to try as another user. Error messages, should not include any path information, file names, or table information as I mentioned before.

Use stored procedure. Because procedures will be written to accept specific parameters. So attackers trying to match the parameters would be hard. So it little hard for the attackers.

Introduce least Database Privileges, In this company there are 10 employees, So it might be easy that give all the privileges to one user, or all the access to every user. But this will result in huge risk. So need to make sure who needs to be given with which access. If a user needs to have read access only for a table, then need to provide only the read access. Because if all the access is given to the table for that user, then the attacker might modify the data, logged in as that specific user.

The company should think of investing in a firewall, to filter out malicious data. So these could be used to prevent their workout application from SQL injection.

**Vulnerability 2: Password Field Masked**

**Describe in your own words**

Normally in any type of application, when the users' type their passwords, the password fields should be masked. The user will not be able to see the characters they have typed in the field. So if anyone looking at the screen when the screen is being shared, will not be able to see what the user is typing. The passwords should not be read from the server-side as well.

This has been introduced to increase security. So unmasking the password will be a potential risk for the website.

**How might it impact the given company and situation?**

If an attacker sees the password of a subscribed user he might share the information with many people, and they all can see the trainers' videos and personal information about that user. When people try to access the trainers' videos without subscribing, the startup company will lose their profit and growth.

If an attacker sees the password of a trainer, he may get access to subscribers' health data, and workout data. This will be used against the trainer and subscribers. The attacker can able to modify the workout content, which cannot be easily identified by the trainer. That will result in losing the trainer’s career and the company’s reputation.

**How might an attacker use this vulnerability to attack this company?**

Not masking password fields would show the passwords of the user, to attackers who are physically around him, or virtually when the user shared his screen in a meeting/call. If an attacker sees the password of the trainer, then he can able to stop the money-back which the trainer gets, He also can modify the workout content, delete any videos of the trainer. This will affect the subscribers and will make them follow the wrong workout. This could be a danger to subscribers' health.

**Why would an attacker use it?**

The attacker might use this to gain access to the system and to view the contents which should not be seen from outsiders. So workout routines, workout tracking and health tracking features of users, workout content, and videos can be accessed and used in an unethical way. The attacker may try to use this way because it is easy, the attacker doesn’t need technical knowledge. Hidden passwords which will be returned from the server also can be readable easily through HTML source code. So attacker may try to use this way to access the website.

**What opportunities might it present to the attacker?**

* The attacker can easily access the system even without any technical knowledge
* It will provide access to the contents on the website
* It will allow the attacker to find more vulnerabilities.
* The victim might use the same password for his/ her other applications too, so not masking the password field for this health application will be a threat to other applications as well

**Assess vulnerabilities with the formula of your choosing**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Damage Potential | Reproducibility | Exploitability | Affected Users | Discoverability |
| Value | 5 | 10 | 10 | 5 | 0 |
| Explanation | This will damage only that specific victim. It depends on the user’s access level even though, but I have assumed that this will be a normal user and gave 5 | This doesn’t need any technical knowledge. Only the skill need is a good visualization.  Reading the HTML code which would be also easy to find and read. Just a web browser is enough | Just a web browser is enough. Not many technical skills are needed. | If the victim is a subscribed user then it will affect only that user. If the victim is a trainer it will affect the trainer as well as his subscribers only. So considering that value 5 is given | Very hard to find that someone has used or accessed some else account. Since this will give website-level application unless the huge change in the web content is changed, no one will get to know about it. |
| Rating : (5+10+10+5+0)/5 = 6 | | | | | |

**Provide mitigations and suggestions for addressing discovered issues. These should consider the company, its situation, and its resources.**

SQA Company should make sure the development team keeps the password field of the web application to be masked so that others will not be able to see them. Also when information is retrieved to the user from the server application should not return any values which have been taken from the masked input field.

These will not cost a lot of money or resources so as a startup growing company, this would be an affordable way of prevention.

**Vulnerability 3: Cross-site Scripting**

**describe in your own words**

This is a way of executing code / small applications on a different web application. This is an injection that will inject malicious scripts to websites and will be sent to other users. It will be in the form of a browser-side script. The end-user or end user’s web browser will not have any idea that this is sent from an attacker, and this is not a trusted script. So this could able to easily access cookies, session tokens, and other important information which were saved In the browser

**How might it impact the given company and situation?**

The attacker might get the session details and will try to showcase different content in their health web application, such as a video ad playing to eat unhealthy foods. The attacker may also provide a different section to the user to enter payment details again as the company is asking for it. The victim might think that is generated from the company and will provide his payment credentials, which will end up in victims losing money, and the company losing customers.

**How might an attacker use this vulnerability to attack this company?**

When a subscriber is checking his/ her workout plan, attacked can show a script generated such as your trainer want to eat this item with your lunch. Subscribers might not know that this is not generated from the actual website. So he might click on that link and will try to purchase the item. That could direct the customer’s to a different link and ask him to log in again giving the credentials, so in the first point, the customer is going to lose his login credentials to the attacker, and after the attacker might ask the user to buy the item, and the user will be then giving his credit / Debit card details. There, now the customer is going to lose his sensitive card details. When the customer loses his money and information the company will not get any new customers and the system will not be used by anyone.

The attacker can access the user’s session before the session expires and use that to access the system. The attacker can also change the users' settings using this attack.

**Why would an attacker use it?**

In this way users will not be able to identify that they have been attacked, they will be thinking that they are still on the actual system, The script will not be identified by the end users’ web browser. Using the malicious scripts, the attacker can get access to users’ cookies, session tokens, or other sensitive information retained by the browser and used with that site. . These scripts can even rewrite the content of the HTML page. So using cross-site scripting attackers can do a lot.

**What opportunities might it present to the attacker?**

1. Attackers can access user’s sessions
2. Attackers can autonomously inject users into pages, and easily re-inject the same victim with more XSS.
3. Attackers can access session tokens.
4. The attacker can access sensitive information retained by the browser.
5. The attacker can connect users to a different malicious server

**Assess vulnerabilities with the formula of your choosing**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Damage Potential | Reproducibility | Exploitability | Affected Users | Discoverability |
| Value | 10 | 10 | 0 | 5 | 10 |
| Explanation | Even though it depends on functions done by the external script executed. Attacking the user from the cross-site scripting going to give access to all his web browser’s sessions and information So it should be categorized as high damage. | This can be easily identified by a web browser or many free vulnerability scanning tools in the market can be able to find it. | Advanced scripting knowledge, session knowledge is needed. Programming knowledge is important. The attacker should have the knowledge to do stored and reflected attacks from the log. | Some users might already have the idea, or can aware of these external scripts, they can able to see from the URL and identify if the malicious script is redirecting to a different target server. So affected users will be medium | Can be easily discoverable by the employees looking at the web browser. |
| Rating : (10+10+0+5+10)/5 = 7 | | | | | |

**Provide mitigations and suggestions for addressing discovered issues. These should consider the company, its situation, and its resources.**

Even though cross-site scripting is a user’s side problem, Attackers can use mislead the users by showing unrelated content and redirecting the users to different sites. Attackers also can use this to change the contents of the website. It is possible to identify whether the website is vulnerable to cross-site by running a vulnerability scanner.

To prevent the XSS, the website should not output data that is directly given in the input text field without checking for untrusted code. Any user input which is part of HTML can be a risk. So appropriate encoding techniques can be used to prevent this. If the website should accept user input in HTML format tags, then valid trustworthy libraries can be used to sanitize HTML code.

If the HttpOnly flag is set for cookies, then client-side java scripts will not be able to access the cookies. Doing regular scanning (eg – Acunetix, NetSparker) will help to prevent cross-site scripting, Data validation, Filtering, and escaping can prevent these attacks. Many free tools can be used to do this scanning which could be affordable for SQA company as a startup.

Remove conflicting characters, brackets, and single and double quotes from user input by encoding user-supplied data. This will prevent inserted scripts from being sent to end-users in a form that can be executed.

**Vulnerability 4: Directory listing**

**describe in your own word**

In a web application, showing the directory listing is a common most vulnerability. , the reason for this is the server misconfiguration. This will lead to showing sensitive information. The attacker can also check all the files such as configuration files, backup files, Temporary files, and hidden files. The risks of this depend on which files were accessible through the visible directory. This could be classified as information disclosure.

**How might it impact the given company and situation?**

The company is a startup with fewer employees, so there might be a chance that the server was not configured properly because of inexperienced developers. If the directory list is showing the trainers' information, such as names, and qualifications that are not a risk. Because these could be found anywhere on the internet, But if the directory list has detail about the source code path, Database information, this would result in high damage.

**How might an attacker use this vulnerability to attack this company?**

The attacker can use this to access the directories which are hidden directly from the website but are visible through the attack. If an attacker gets access to the money details/ payment details he can use it for personal. The attacker might try to find mainly the trainers' videos, and sell them to third-party competitive sites. So if an attacker can able to find the directory path for the videos files, it would be a high risk for the company.

**Why would an attacker use it?**

The attacker would use this attack to find hidden files. From these listings, the attacker can view sensitive files such as password files, database files, FTP logs, and scripts. This information was not intended for public view. These details could be used against the company.

**What opportunities might it present to the attacker?**

When a web server reveals a directory's contents to the attacker , the listing could contain information not intended for public viewing.

1. Configuration file contents

2. Backup files - with extensions such as .bak, .old or .orig

3. Temporary files - these are files that are normally purged from the server but for some reason are still available

4. Hidden files - with filenames that start with a "." period.

5. Naming conventions - an attacker may be able to identify the composition scheme used by the website to name directories or files. Example: Admin vs. admin, backup vs. back-up, etc...

6. Enumerate User Accounts - personal user accounts on a web server often have home directories named after their user account.

7. Configuration file contents - these files may contain access control data and have extentions such as .conf, .cfg or .config

**Assess vulnerabilities with the formula of your choosing**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Damage Potential | Reproducibility | Exploitability | Affected Users | Discoverability |
| Value | 1 | 10 | 2 | 2 | 10 |
| Explanation | It will only become an issue when sensitive data is exposed as part of the listing. That doesn’t mean there is no damage, so I didn’t give 0, and I gave 1. | This can be identified by many tools easily. So it has high reproducibility. | This depends on the data that is visible in the list, It doesn’t need expert-level knowledge expert, even not moderate level knowledge should be advanced than that, So I gave a score of 2 | I cannot decide, how it will affect the users because it depends on the information shown in the list, but I can't assume that it will not affect any users, so I gave a score of 2 | This is easy for anyone to find out about this issue. Most web scanning tools will directly find out whether directory listing is available or not. |
| Rating : (1+10+2+2+10)/5 = 5 | | | | | |

**Provide mitigations and suggestions for addressing discovered issues. These should consider the company, its situation, and its resources.**

As a web administrative admin, Restrict files/ directory listings to only the users who need it and ensure that files containing sensitive information are not left publicly accessible, or that comments. The automatic Directory listing feature needs to be turned off. Most of the freely available scanning tools would be able to find this issue. So doing regular proper scanning will make sure to prevent this issue.

**Vulnerability 5: Insufficient Authentication / Access control**

**describe in your own words**

This could happen when the application allows the attacker to access hidden and sensitive information without properly accepting user credentials for the admin. For example, an attacker can simply access the admin site by modifying the URL with /admin in the end. This is possible because most of the simple new web applications have their directories saved in the root directory, which is not shared anywhere on the website. The team will think that no one will access this since it is not linked with the website. The attacker will just visit this page and can perform all the malicious activities.

**How might it impact the given company and situation?**

The attacker can able to access the root directory or any sensitive information. Even though the URLs are not directly linked with the application, they could be identified through a Brute force attack. This could be used for further vulnerabilities exploitable. When the user gets access to the root directory or bypasses the access control, then the attacker can gain access to all the data, that are not available for the public. In the SQA company if the attacker gets access to the root directory which will be accessed only by the system administrator the attacked can take control over the full company and also can get full access to internal infrastructure

Even though the attacker gets access to subscribers' low-privileged accounts, that will also still grant the attacker to access data, such as videos and content. The company is a startup company and their entire business depends on the website. So accessing their internal pages and will allow the attacker to read more about their business information.

**How might an attacker use this vulnerability to attack this company?**

The attacker might use this to attack the company because the attacker can able to get the company’s control by accessing or finding a way to access the root directory. An attacker can use a variety of vectors to exploit this weakness, including brute-force, session fixation, and Man-in-the-Middle (MitM) attacks, bypassing two-factor authentication.

Attacker might try to do this using weak credential recovery and forgotten-password measures.

**Why would an attacker use it?**

Attackers will mostly try this is because doing a brute force attack is a common and well-known thing for all the attackers. Most authentication attacks occur due to the continued use of passwords as a sole factor. Finding a password for an application will allow the attackers to gain the victim’s access to different applications as well.

Most of the application doesn’t have timeouts set, so this will be a clear success for an attacker to try. When the user uses a public computer to access the health application. Instead of selecting “log out” the user simply closes the browser tab and walks away. An attacker uses the same browser an hour later, and the user is still authenticated. So this is a simple task for the attacker to gain access to the system.

**What opportunities might it present to the attacker?**

* Provide a list of username and passwords or entire sensitive information which are available in the root directory
* Exposes sessions IDs in the URL
* Alter the application.
* Execute arbitrary code.

**Assess vulnerabilities with the formula of your choosing**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Damage Potential | Reproducibility | Exploitability | Affected Users | Discoverability |
| Value | 8 | 5 | 5 | 6 | 5 |
| Explanation | Admin non-sensitive data could be compromised | Few steps and tools and steps required to reproduce | Special tools and knowledge about brute-force, session fixation, and Man-in-the-Middle (MitM) attacks are needed to somehow do this attack properly. With the knowledge, it could be exploitable so I have given the value 5 | This would affect all the users depending on the authentication broken | It is not publicly available or very hard to find. Can figure it out by guessing or monitoring network traces |
| Rating : (8+5+5+6+5)/5 = 5.8 | | | | | |

**Provide mitigations and suggestions for addressing discovered issues. These should consider the company, its situation, and its resources.**

Two-factor authentication (or 2FA) for all logins. This prevents accounts from being brute-forced so easily, Ensuring users have sufficiently-strong passwords by implementing validation on password creation. Requirements such as a minimum password length, mandatory complexity and denying the setting of common passwords/patterns is a good way to increase protection.

Numerous other steps can be taken to further reduce the likelihood of Broken Authentication exploits: Rate-limit login attempts. While this is usually done via tracking cookies or IP addresses which a determined attacker can spoof (or simply acquire an entire VPS for an alarmingly affordable price), it is still another hurdle which will help prevent exploitation. In the same way, a response delay (e.g. using a sleep function) can be implemented to further slow an attacker’s brute-force attempts Deny IP addresses from which suspicious activity has been detected. Cross-reference user passwords against recent/common password leaks from other data breaches, and notify any users which are making use of passwords that are frequently applied in brute-force attempts. Make use of a server-side, built-in session manager which generates a random, high-entropy session ID after a user has logged in. don’t even include this ID in URLs and ensure that it is invalidated after a period of inactivity.

Above mentioned mitigation plans don’t need many resources, So. could be manageable by the SQA company’s available resources.

**Vulnerability 6: Documentation**

**describe in your own words**

This vulnerability determines when one or more documentation files are found. These files will have the information which could be used by the attacker. Find more detail about the application, software used to implement the application, database used, and web application hosting detail.

**How might it impact the given company and situation?**

If the company’s readme file is accessible for the attacker in any directories the attacker may use that to read software application installation, versions being used, attacker must use this information to find more vulnerabilities of the system based on the software version.

The company has fewer employees, so managing all the details by only 10 members might be difficult for them so they may try to keep all the detail in one directory and that’s how the attacker might able to access all the documentation.

**How might an attacker use this vulnerability to attack this company?**

The attacker might read the information about the name of the application, a recent version of the application, version of software used, location to download the software, hosting information, technical support detail, payment gateway information, how employees are getting paid, and reference to technical documents.

This could be used as a base for an attacker to get to know more detail about the website and find more vulnerabilities.

**Why would an attacker use it?**

This could be identified easily to readable because readme.txt will not be encrypted and stored most of the time. The attacker might use this to get more detail about the website.

**What opportunities might it present to the attacker?**

To know more detail about the application

• Name of the application.

• Most recent version information. Note: The last date indicates what version is on the website.

• A location to download the software.

• The company site that holds the software.

The following items are especially good if the software is not freeware, because you can discover the default installation layout, debug configurations, etc.

• A technical support forum or mailing list.

• References to technical documentation.

**Assess vulnerabilities with the formula of your choosing**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Damage Potential | Reproducibility | Exploitability | Affected Users | Discoverability |
| Value | 1 | 10 | 5 | 7 | 5 |
| Explanation | This will not provide more information to exploit, the attacker might need to know more in detail which will not be available in the readme.txt so it is not that much damageable | Trival, this doesn’t need any authentication to produce. Could be done through the web browser | This could be exploitable, not expert knowledge is necessary to exploit but moderate level knowledge is important, so score of 5 is given | This could attack authenticated  Users such as admins, but might not affect all the users | It is not publicly available or very hard to find, it could be found through HTTP requests. |
| Rating : (10+8+10+10+5)/5 = 8.6 | | | | | |

**Provide mitigations and suggestions for addressing discovered issues. These should consider the company, its situation, and its resources.**

Whenever there are documents to be stored, store them outside the directory website, keep only the needed files for the website. Since this company has only 10 employees introducing best practices and monitoring would be easy. So having all the documents organized outside the application will prevent any outsiders to access them.