System Description and Risk Analysis

Albert LLebaria Alberto Martin Sabin Rimniceanu

Rafa Markiewicz

November 15, 2018

Contents

1 System Characterization

1.1 System Overview

The system is a web-based photo-sharing service. It allows users to upload, share and comment images. This service runs on a linux virtual machine. The service consists in php files used as views or model, a CSS file for styles, folders to store images and a MySQL databade.

1.2 System Functionality

Anyone must be able to create an account for the system. Once the account is created, that person becomes a user. A user must be able to:

- Upload pictures.
- Share their own pictures with other named users on a picture-by-picture basis.
- View their own pictures and pictures other users have shared with them.
- Comment on any picture they can view.
- View comments on any picture they can view.

Regarding security, the system must satisfy the following requirements:

1. Confidentiality: Only a user authorised for a picture can view, comment or read comments on that picture.
2. Integrity: No unauthorised user can modify any picture or comment.
3. Availability: No unauthorised user can prevent an image or a comment from being shown to authorised users.

1.3 Components and Subsystems

Files that display pages - Views:

footer.php	Contains the footer view of the page.
header.php	Contains the header view of the page.
index.php	Contains the view for one image - The image itself and the comments of the users.
login.php	Contais the welcome page to the service if no user is loged in. Otherwise, it shows the user images (own and shared images).
logout.php	Logs out the user and redirects to index.
register.php	Contains the register form. If the user is created correctly, it redirects to the index. Otherwise it displays an error message.
upload.php	Contains the form to upload images.

Database tables:

User	Stores users data
Image	Stores images data
Shared_image	Stores image sharing information
Post	Stores comments information

Files related with the database:

ssas.php	Creates the connection to the MySQL database and contains all the queries to extract and add information to the database. It is the model of the web service.
datamodel.sql	Contains the script that creates the database.

Other:

style.css	Contains the styling of the service.
images/	Folder that contains the welcome page images.
uploads/	Folder that contains all the images uploaded by the users.
vendor/composer/	Folder that contains Composer (tool for dependency management in PHP).
vendor/firebase/	Folder that contains Firebase (SDK to interact with Google Firebase from the PHP application).

1.4 Interfaces

Index (before login): Information flow: when a user logs in or registers is exchanging information with the database. When logging in, the model (ssas.php) checks that the user exists, and when registering the model checks that the username and password are valid.

Index (after login): Information flows: once a user is logged in, he/she can see the pictures that has uploaded or the pictures shared with him/her. The model gets the pictures for this user from the database. The user can also upload a new picture, and the information for this picture is stored in the database.

Image: Information flows: the image comments are shown to the user. When a user comments, this comment is sent to the database, also trough the model. Finally, a user can share the image with another user, which is another information flow.

1.5 Backdoors

Backdoor 1:
This backdoor affects confidentiality. Because our server does not have a certificate, a user can be vulnerable to a man-in-the-middle attack where an attacker can just copy the front-view in order to steal user's cerdentials. On the other hand, HTTP POST data is not encrypted also because we don't have an SSL certificate. This means that the users send login POST requests with their credentials as plain text instead of encrypted text provided by HTTPS.

Backdoor 2:
This backdoor allows to execute commands by passing the cmd parameter in a HTTP GET request to the backdoor.php url. The executed result of the commands are displayed as plain text in the browser. However, the user who

is executing the commands is the www-data, which is the provided user for the apache2 process providing the webservice. In order to gain root access, the user has to decrypt a text in topsecret directory, which has the password for the ssas default user. Once it has the decrypted the password, the ssas user can be logged in via the backdoor.php command line and do root operations.

Backdoor 3:
This backdoor consists on a SSH backdoor. By finding that the 22 port is open on the server, trying to login and getting the message "Hi, I'm Jim from IT and I like 8 digit hex numbers". From here the attacker is supposed to guess that his username is jimssh and bruteforce his password as an 8 digit hex.

2 Risk Analysis and Security Measures

2.1 Assets

Hosting computer: the server is running in a virtual machine inside a computer. Only the system administrator should have access to this computer. The required security for this should be locking the computer somewhere and restrict the access. Another security requirement would be to always update to the last versions all the software used by the host.

User information: our system stores user information such as usernames and passwords. Having this information one can access to the system and log into the account to view the pictures and comments. The required security for this information is to avoid any kind of SQL attacks such as SQL injection and also to store the passwords hashed in the database instead of plain text, for instance.

Pictures: the system also stores all the pictures that the users upload. Only the user that posts the picture and the ones with whom he/she decides to share them are allowed to view such pictures. Therefore, the required security must avoid that other unauthorised users can see pictures. This can be achieved again avoiding SQL attacks and url manipulation. The same applies to comments.

2.2 Threat Sources

Hackers: of course, hackers are a potential threat source. However, there is not a big motivation for them unless the users post sensible data or images. They could use this data to blackmail.

Employees: the employees are the first physical threat source given that they work for the company and may have access to the system. Their motivation could be revenge on the boss for instance.

"Friends": there could be some people that would like to "hack" other known

user's account in order to view pictures or mess with this user.

2.3 Risks and Countermeasures

Impact	
Insignificant	Generally, a result of a minor security breach in a single area. Impact is likely to last less than several days and requires...
Minor	Result of a security breach in one or two areas. Impact is likely to last less than a week but can be dealt with at the segment ...
Moderate	Limited systemic (and possibly ongoing) security breaches. Impact is likely to last up to 2 weeks and will generally require ...
Major	Ongoing systemic security breach. Impact will likely last 48 weeks and require significant management intervention and resources ...
Catastrophic	Major systemic security breach. Impact will last for 3 months or more and senior management will be required to intervene for ...
Doomsday	Multiple instances of major systemic security breaches. Impact dura- tion cannot be determined and senior management ...

Likelihood	
Rare	May occur only in exceptional circumstances and may be deemed as unlucky or very unlikely.
Unlikely	Could occur at some time but not expected given current controls, circumstances, and recent events.
Possible	Might occur at some time, but just as likely as not. It may be difficult to control its occurrence due to external influences.
Likely	Will probably occur in some circumstance and one should not be surprised if it occurred.
Almost Certain	Is expected to occur in most circumstances and certainly sooner or later.

	Consequences					
Likelihood	**Doomsday**	**Catastrophic**	**Major**	**Moderate**	**Minor**	**Insignificant**
Almost certain	E	E	E	E	H	H
Likely	E	E	E	H	H	M
Possible	E	E	E	H	M	L
Unlikely	E	E	H	M	L	L
Rare	E	H	H	M	L	L

2.3.1 *Evaluation Asset: SQL injection*

No.	Threat	Implemented/planned countermeasure(s)	L	I	Risk
1	SQL injection	Usage of prepared statements instead of standard queries (implemented).	*Rare*	*Moderate*	*Medium*

2.3.2 *Evaluation Asset: Manipulate url*

No.	Threat	Implemented/planned countermeasure(s)	L	I	Risk
2	Manipulate url	Change Apache configuration. Deny permissions to the users to navigate trough the project folders (implemented).	*Rare*	*Minor*	*Low*

2.3.3 *Evaluation Asset: Cross-site scripting*

No.	Threat	Implemented/planned countermeasure(s)	L	I	Risk
3	Cross-site scripting	Sanitize user inputs to avoid for example Javascript injection on the comments (implemented).	*Rare*	*Moderate*	*Medium*

2.3.4 *Evaluation Asset: Password storage*

No.	Threat	Implemented/planned countermeasure(s)	L	I	Risk
4	View passwords in DB	Store hashed passwords instead of plain text (implemented).	*Rare*	*Minor*	*Low*

2.3.5 *Evaluation Asset: Bruteforce attack*

No.	Threat	Implemented/planned countermeasure(s)	L	I	Risk
5	Bruteforce attack for passwords	Require more difficult passwords -at least 8 characters with both capital and lowercase letters- and add a one second sleep() in the login method (implemented).	*Rare*	*Minor*	*Low*

2.3.6 *Evaluation Asset: Man-in-the-middle*

No.	Threat	Implemented/planned countermeasure(s)	L	I	Risk
6	Man-in-the-middle attack	Add a SSL certificate (planned).	*Likely*	*Moderate*	*High*

2.3.7 Risk Acceptance

No. of threat	Proposed countermeasure including expected impact
1	We cannot think about more countermeasures. The risk is medium only because the impact would be Moderate, but we use prepared statements on all the queries, so it is not possible to do SQL injection.
3	Same as No 1. We sanitize all the inputs which it's value is stored into the database. For instance, comments and usernames, this way if other users want to see either comments or user names, no scripts can be executed.
6	Add SSL certificate.