System Description and Risk Analysis

Mustapha Malik Bekkouche Oscar Felipe Toro
Steffen Mogensen Yumer Adem Yumer

April 12, 2016

Contents

1 System Characterization

1.1 System Overview

The mission for the server is to host a web application, where users can upload their images, and share them with each other. The users have the power to choose who they wish to share their images with, and also if they would like to unshare an image with another user. When a user uploads an image, we own that image, therefore, the user cannot delete or remove the image from the web application.

The server is a Unix-based system, build on Ubuntu 14.04. It has a bare minimum of users on the server since there should only be one for maintaining the services running on the system.

This web application is set to start-up as soon as the server is booted. This is expected to be the best case, since the main purpose of the service, is to host this application. So in case of failure where the system reboots (E.g. power out, or alike), it will be available again as soon as possible, without having personal to start up the application manually.

The server is also set-up so that it can be accessed remotely. This is for maintenance reasons, and will require the proper use of login and password for gaining access. For sending files to and from the server, this should happen within the encrypted connection services running on the server, and not by additional software.

1.2 System Functionality

As originally stated in the project description, the system implements the main requirements, namely:

1. the system should allow the user to upload pictures
2. the user can share his own pictures with other named users on a picture-by-picture basis
3. the user can view his own pictures and pictures other has shared with him
4. the user can comment on any picture he can view
5. and the user can view comments on any picture he can view

1.3 Components and Subsystems

We recognize mainly two elements of the system, the web application and the database. Both sitting on top of the Ubuntu Server. In order to connect to the server we are using the OpenSHH suite.

- Platform: the virtual machine is running Ubuntu Server 14.04.01, known as the Long Term Support version of the server which offers updates for five years.

- Web Application: The web application is build in Python 2.7, together with the micro-framework Flask. Hence, the web server currently in use is SimpleHTTPServer, that is a part of the Python standard library. The framework offers inbuilt security features, such as sanitisation of fields and it only includes the minimum building blocks of a web system, which gives us the possibility to decide which parts of the Python community library we want to use to construct the web application.

- Database: SQLite3 is used as storage of users for the web application, and all the dynamic content such as comments, pictures paths, and relationships between users. This means that there is no daemon running, since everything is stored in the same file. To access and manipulate the database, it is needed to have access to this file.

- SSH: Open-SSH is installed on the system, this is so it is possible for a system administrator to connect to the system and do any configuration that is necessary, without having to be in front of the machine. This also means that scp is available for uploading and downloading files to the server.

1.4 Interfaces

The main interface of the application is the web site, where the user interacts with the application, therefore a series of test have been performed to ensure that the information coming from the user is free of potential html tags that can compromise the security of the system.

There is a second interface between the web application and the database, in this case, the application communicates with the database using SQL. To avoid SQL injections we prepare the statements before sending them to the database.

Administrators and web developers can access the system using SSH, a special attention has been put in the selection of a long and unique password plus the creation of a very strong SSH key that allows the superusers access the server.

1.5 Backdoors

Easy to find:
Netcat is running on port 60606. This makes it possible to make remote access to the service through this port.

The reason for putting the port number in the higher numbers, is first of all

that the first 1024 ports are "well-known" ports. Meaning that they have been defined to serve a purpose. E.g. 80 is HTTP, 443 is HTTPS, 666 is Doom. If the user makes a nmap scan on the system, these are the port the scanner will look for by default. This means that if someone wish to find this backdoor, the attacker will need to improve the search a bit more.

In order to accomplish that, an adversary, could intent a couple of strategies :

- Nmap: To find this with nmap:
  Command: nmap 192.168.1.10 -p 1-65535
- Netstat: With netstat, it is possiable to see all the activity:
  Command: sudo netstat -plunt

Hard to find:

- Ubuntu 14.04 has an overlays vulnerability (CVE-2015-8660). The code to use for this exploit can be found at: https://www.exploit-db.com/exploits/39166/ - This will give local root access.

To perform the exploit, proceed as following:
We have compiled the exploit, and placed it under /bin/pwn. This means that if someone on the system calls the command "pwn" from anywhere on the system, the user will gain root access.

For the fun of it, we have made a user with a low amount of privileges on the system. The user name is admin, which have been given a password from the top 300 most used passwords according to SecLists[1]. It should therefore, be fairly easy to brute force.

So if/when this user has been cracked, someone can login to the server with SSH, call the command pwn – And bingo! Root access...

Something to note: With the web application, it is possible to upload files, as long as the file extension is of an image sort. This means that it is possible to upload any kind of data to the server since it's only the naming of the file that makes a difference, and not the content of the file.

2 Risk Analysis and Security Measures

2.1 Assets

2.1.1 *Physical assets*

Server - The web server is located on a virtual machine and is up to date. The administrator is responsible of installing all the patches guarantying the proper

[1] github.com/danielmiessler/SecLists

function of the server. Physical access to it may allow an adversary to gain control of the system. A server can, for example, be booted with a different operating system.

2.1.2 *Logical assets*

It includes the operating system, the website, the database, the information related to the users and customer confidence.

Firewall - The IP table (firewall) of the server is properly configured and restricts access to the server. It keeps track of each connection passing through it and filters all the attempts to connect to the server except through allowed ports. The firewall is kept up-to-date and the administrator installs all security-relevant updates.

Website - The website provides the following functionality - uploading images, sharing them with others user, and posting comments. Only a user authorised for a picture can view, comment or read comments on that picture. No unauthorised user can prevent an image or a comment from being shown to authorised users. The website runs on the web-server which is kept up-to-date. The web developer is responsible for updating the functionality of the website.

Database - it keeps all the information related to the users, using the website. The username, passwords, pictures are stored in the database. It runs on the server and the access to it is restricted. The administrator is responsible for its maintenance.

Information - all the informations related the users are valuable and proper measures are taken to guarantee their confidentiality. The informations include especially the username, password, pictures uploaded by users.

Pictures - all the uploaded images are kept on the server and are visible to the owner and other users allowed by the owner. There is no restrictions on the size of the images.

Username and passwords - They identify the owner of the pictures. All password are saved, using the proper hash functions guarantying the security.

Customer confidence - since the user can upload private pictures, which should be hidden for the world, user confidence is important for a successful business relationship.

2.2 Threat Sources

- *Employees* : the employees in our tiny company consist of a system administrator who has access to the server and a web developer, they could possibly leak sensitive information (intentionally or unintentionally) or weaken the system security,

- *Hackers* : since the system is connected to internet,it is exposed to various attacks,the attackers vary from highly motivated people with good

skills actively trying to penetrate the system to script kidies just messing around.

- *Malware* : as any it system, malware could possibly be a problem, it could be directed malware (unlikely) or undirected malware (more likely).

2.3 Risks and Countermeasures

Impact	
Impact	Description
High	Complete shutdown of the system, user data is compromised,apocalypse and a major loss in asset value, complete loss of the customers confidence
Medium	System slow down, loss in asset value
Low	Relatively affect the credibility of the company,lower the customers confidence and a relative loss in asset value

Likelihood	
Likelihood	Description
High	The threat source has the power to exploit vulnerabilities in the system, the countermeasures are inexistant or ineffective.
Medium	The threat source is motivated,some countermeasures are implemented which may prevent him to do harm
Low	The countermeasures are completely effective, (almost) nothing to worry about

Risk Level			
Likelihood	**Impact**		
	Low	Medium	High
High	Low	Medium	High
Medium	Low	Medium	Medium
Low	Low	Low	Low

2.3.1 *Evaluation Asset Firewall*

Table 1: Evaluation Asset Firewall

No.	Threat	Implemented/planned countermeasure(s)	L	I	R
1	Malware: Virus/worm spreads over the Internet possibly affects system files and change firewall settings	Proper maintenance of the server, security patches installed, restricted user rights	Low	Medium	Low

2.3.2 *Evaluation Asset Website*

Table 2: Evaluation Asset Website

No.	Threat	Implemented/planned countermeasure(s)	L	I	R
1	Skilled hacker gains control over the website, steals confidential data, modifies website settings because of vularability either in the server or on the website	The server is hardened and regularly updated. System administrator is trained to notice irregularities on the server.	Low	Medium	Low
2	Script kiddies makes modifications on the website as a result of an attack	The server is properly maintained, the website is hardened, all input is sanitized	Low	Medium	Low

2.3.3 *Evaluation Asset Database*

Table 3: Evaluation Asset Database

No.	Threat	Implemented/planned countermeasure(s)	L	I	R
1	Skilled hacker gains control,over the database, steals confidential data, make,changes on the database like deleting tables, editing records	Hardened server and kept up-to-date, use of hashed and salted passwords	Low	High	Low

2.3.4 *Evaluation Asset Images*

Table 4: Evaluation Asset Images

No.	Threat	Implemented/planned countermeasure(s)	L	I	R
1	Web developer unintentionally breaks confidentiality during the update of the website	Well trained web developer	Low	High	Low

2.3.5 *Evaluation Asset Username and Password*

Table 5: Evaluation Asset Username and Password

No.	Threat	Implemented/planned countermeasure(s)	L	I	R
1	Script kiddies try to guess the username and the password	Advising users not to choose simple usernames and passwords	Medium	Medium	Medium
2	Skilled hacker attacks with special software to break passwords	Encryption passwords with strong hash functions choosing arbitrary salts	Medium	High	Medium

2.3.6 *Evaluation Asset Customer confidence*

Table 6: Evaluation Asset Customer confidence

No.	Threat	Implemented/planned countermeasure(s)	L	I	R
1	Theft of confidential data	State-of-the-art security measures, hashing and salting all passwords not keeping any sensitive data	Low	High	Low

2.3.7 Risk Acceptance

No. of threat	Proposed countermeasure including expected impact
2.3.5	Allowing users only 3 login attempts
2.3.5	using one time password sent by sms
2.3.5	forcing users to create long passwords (including capital letters, numbers and special characters)