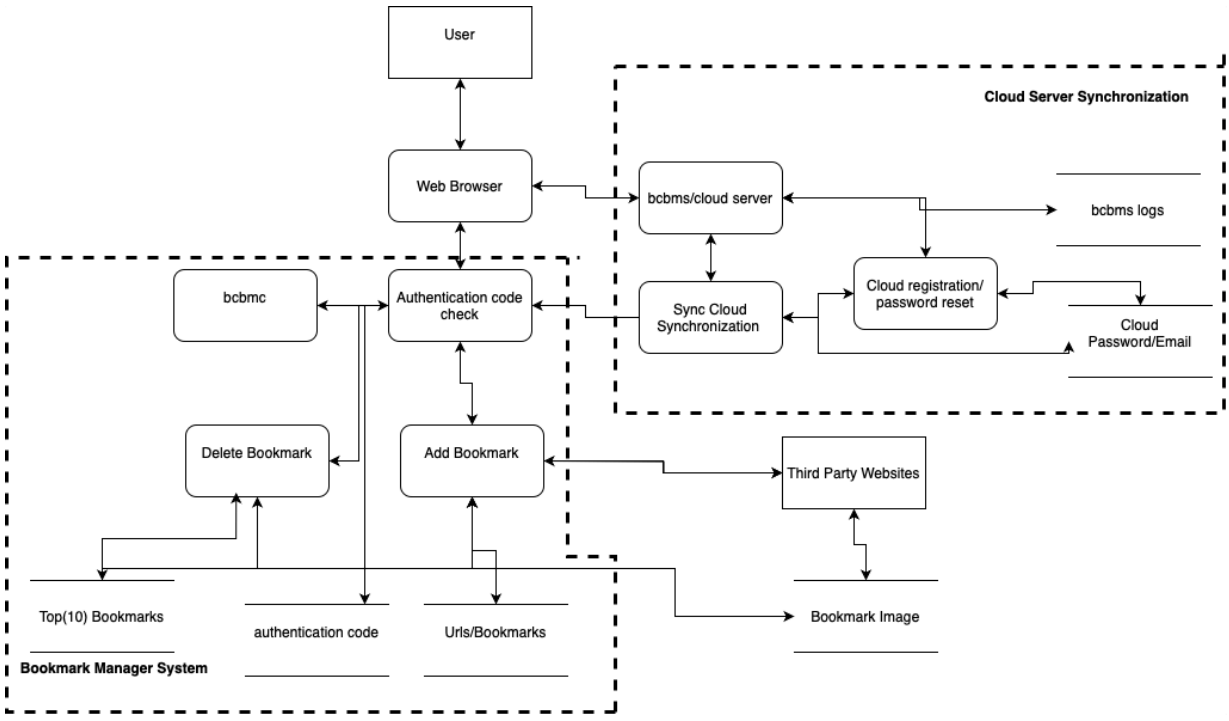
CSCI 4271 Project Report

Jennifer De La Rosa

System Design

The Badly Coded Bookmark Manager (bcbm) is a program that a user can run on a web browser. The bcbm works by running a client's web browser bcbmc, which is generated with its own unique authentication code. The authentication code is generated when the program is called. This is done by making bcbmc call gen_auth_code(), which is a function that generates a random six digit number in the range of (000000, 999999) . The purpose of the authentication code is for more security for each user's bcbm, each user will have its own browser with its own unique authentication code. There is an authentication code that is 999999, and it is set for when the browser doesn't need authentication, the browser is set with no-auth. The bcbmc may add and delete bookmarks of urls that the user specified. This is done by the program decoding urls, the urls are checked by being decoded, but there is no way for the system to verify the urls security. While the authentication code is used for security, bcbmc also provides an option for a user to Sync to Cloud and Register for cloud. Syncing to Cloud requires an email and password. Registering and resetting your password is accessed through https://bcbm.badlycoded.net which allows users from any device to reset their passwords with their email or set up a cloud account with their email. Then an email confirmation link is sent to the user. The purpose of allowing a user to sync to Cloud is to allow the user to be able to access bcbmc from multiple devices so the user won't lose track of their bookmarks. In this implementation of bcbmc the user is able to

login to their Cloud, when the system detects that the user had previously used their email for bcbmc they ask if the user want synchronize with the current device or to replace all of their old bookmarks with the current device's bookmarks. When adding bookmarks the user has the option of ranking the currently added url/bookmark to their top 10 favorites.



The DFD shows that a user can access bcbmc then they get an authentication code which allows the user to access the web browser that had their bookmarks saved. In the web browser the user is able to add, delete bookmarks and sync to the cloud if they have a valid password and email. The DFD also displays what data is stored in the cloud server and bcbmc. The bcbmc stores the authentication code, urls/Bookmarks, top(10) bookmarks and the cloud password and email. Which the Cloud server will also have access to since the system is designed to allow users to access the same bcbmc from a cloud server when they are using a different device.

Threat Model

The security goals for bcbmc are based on the STRIDE threat model. STRIDE is a threat model that is broken into six cases of what could go wrong with the security of a system. The six cases are split into spoofing, tampering, repudiation, information disclosure, denial of service and elevation of privilege. Spoofing is when an entity pretends to be something or someone that they are not. Tampering is when data is modified or deleted without authorization. Repudiation is when someone does something they claim they didn't. Information disclosure is when private data is leaked without authorization. Denial of service is when an entity should have authorization to access certain resources but are denied because of system faults. Elevation of privilege is when an entity is allowed to execute code or access data without proper authorization.

S*poofing:*

The potential of spoofing happening in bcbmc occurs when an attacker pretends to be a different user and access their Cloud server account by finding their email and password. Another example of spoofing that is similar is an attacker using someone else's authentication code. Since each browser is generated with a unique six digit authentication code an attacker could potentially hardcode this code and gain access to someone else's bookmarks. There should be more measures taken to make sure the person accessing the web server is the correct person. Ways to mitigate spoofing for the cloud server would be having a two-factor authentication for the Cloudserver making sure the person logging in really is the owner of the account possibly by sending them a code in their email address. Another mitigation would be just having more than just an authentication code that grants a user access to the web server. The web server should

also have a password, or user name that is associated with the code as well, to ensure it's the right person using the web server.

Tampering:

For bcbmc tampering may happen when an attacker has access to the data files, like the bookmarks and their urls, cloud server password and email. So if somehow an attacker gains access to these files they might tamper with the files to gain easier access to the cloud server by reading the cloud server's passwords and emails. Also the attacker might be able to alter the bookmarks urls by replacing the urls with a bad website that might cause harm to the user. Another potential issue is when the user adds a bookmark the image is retrieved through a third party website, this could lead to possible cases of malicious code being passed through in the image. In result this causes tampering because the potential damage of the code from the image can lead to either corrupting or altering the data on bcbmc. The ways to mitigate any tampering would be to make sure there isn't any way for an outsider to access these files.

Repudiation:

A situation where repudiation is possible would be if a user bookmarked a bad website knowing the website wasn't a good website then claiming they didn't insert a bad website. Another example of repudiation would be if the user changed their password for their cloud server account and then they claim that they never changed their password. A different situation of repudiation occurring is if the user syncs to their cloud on a different device and replaces the bookmarks they might claim that they didn't. Mitigations for this would be maybe knowing when a user changed their password and when the user added urls. To know that a user is adding their own urls and not an outside entity.

Information - Disclosure:

Something in bcbmc that may be information disclosure is having an outside entity leaking information about the authentication codes and possibly the cloud server passwords and emails. This would be an information disclosure since that information is sensitive and private and should only allow the user to be aware of. Another example of information disclosure is the fact that a user might save a private url and change the title, but the url is still visible when the user hovers over the bookmark. Another potential case of information disclosure is when the user adds a bookmark the image added might not be something they want visible on the bcbmc but the user has no option to remove the image. The way to mitigate these would be to ensure the security of these files, don't let the urls be visible on the homepage at all, and allow the user the choice to have an image for their bookmarks.

Denial of Service:

An example of denial of service in bcbmc is when the client's web browser is loaded and the user wants access to their cloud to have access to their other bookmarks. But they forgot their password. When the user clicks the button that says “Register to cloud/ forgot password” nothing happens. Then that user doesn’t even have a chance to get their bookmarks back. A way to mitigate this would be for the user to actually be able to load the link that helps them reset their password. Maybe instead of a link to register and reset password just have the reset password link in the cloud sync page. This is a denial of service because the user doesn’t have access to the bookmarks they had saved on a different device.

Elevation of Privilege:

Elevation of privilege would be having someone have access to the authentication codes and running them without any permission. Examples of this include outside entities deleting

bookmarks. Attackers can also add malicious websites that the user otherwise might assume they were safe if the title were something trustworthy or familiar. Another possibility is if an attacker is able to successfully get a user's email and password for the cloud from that they can overwrite all of the users bookmarks by replacing them with malicious websites. A mitigation would be making sure the authentication codes are placed securely and not easily accessible by any outside entities, also ensuring that the cloud sync has two-factor authorization.

Summary of Findings

1. The first vulnerability demonstrates tampering and elevation of privilege and is demonstrated by running the provided proof of concept(PoC) script titled *vulnerability1.py*. This vulnerability is found in bcbmc and it allows an attacker to be able to delete a user's bookmarks with only the users authentication code and the bmids. This is possible by either having the attacker guess the authentication code or for the attacker to be on path and see when the user gets the authentication code. From that the attacker can have access to all the information they need on the users bcbmc, including the bmids. The user can get the bmids and then proceed to delete every bookmark the user had saved. A Mitigation for this problem could include having two-factor authentication to who has access to bcbmc and another possible mitigation would be having the user backup their bookmarks, have somewhere else that the user could save versions of their bcbmc. If the user had bookmarks deleted by an outside entity they should be able to recover these bookmarks by either being notified they were deleted or having access to previous versions of their bcbmc.

2. The second vulnerability found also shows elevation of privilege and tampering, this is demonstrated by running the PoC script titled *vulnerability2.py*. In this case an attacker can insert a malicious website, with just the user's authentication code. The attacker can do this by making the title of the bookmark seem like something secure and trustworthy, even familiar if they are able to figure out what titles the user already had in their bookmark list. Doing this increases the likelihood that a user will select the attacker's malicious website. The user won't have any idea that their bookmarks have been tampered with, until they click on the malicious one . This shows its an elevation of privilege since the attacker is able to get the authentication code and add their own malicious website. Potential ways to mitigate this would be to ensure two-factor authentication to who has access to bcbmc and by making bcbmc ensure the security of the websites being added.

3. The third vulnerability found shows denial of service. This is demonstrated by running the PoC script titled *vulnerability3.py*. When this script runs it shows that the server crashes after having a user add a bookmark with a fav ranking greater than ten. For example in the script it's assumed the user themselves accidentally added a bookmark with their fav ranking as eleven, not realizing that the limit is ten. This shows denial of service because when the user goes back to their website and clicks on their top ten fav bookmarks the server crashes. It's possible that someone could do this by accident either thinking they can rank more than ten or thinking they had to rank every book mark they add. Mitigations for this would include having an input check that ensures the user only inputs a number between one and ten when they add the bookmark. Then if the user puts

a number greater than ten simply don't add the bookmark and inform the user that it has to be a number between one and ten.

4. The fourth vulnerability found shows spoofing and information disclosure. This is demonstrated by running the PoC script titled *vulnerability4.py*. This vulnerability shows that if an attacker knows a victim has a cloud account, and they know the victim's email address. They can gain access to the victim's cloud account by resetting the password. Since bcbms sends a Token and email link confirmation, an attacker can easily mimic this url by just having the correct token and email. This is possible since the attacker has access to the bcbms journal logs therefore they can read what Token is sent to the email when the password is reset. After mimicking the url link the attacker now has their own password also set to the victim's email. The worst part about this is the fact that the attacker doing this doesn't overwrite the old password, so the victim might not even know someone else has access to their account. A Mitigation for this would be to not show the Token in the logs, or at least put a hash over the token, just like how passwords are hashed. Another thing that would help mitigate this is to inform the user that someone else logged into their cloud account from a different device just to alert the user of suspicious behavior.