Security Analysis of phpBB3 Bulletin Board Software

December 6, 2010

Matthew Fong, Herman Lee, Chih-Hao Lin, and David Yue

Department of Electrical and Computer Engineering
University of British Columbia
Vancouver, Canada

mfidunno@interchange.ubc.ca, hechlee@interchange.ubc.ca, water313@interchange.ubc.ca, davidyue@interchange.ubc.ca

***Abstract*—One of the most popular bulletin board software is phpBB3, a free and open source project built on PHP. The system is a multi-user forum environment used to facilitate non-real time conversations between users. This analysis tests many vulnerabilities found on dynamic websites. A discussion on the risks and assets involved with a website of this nature is presented, along with a discussion on possible fixes and methods to mitigate the risks and vulnerabilities.**

I. INTRODUCTION

THE popularity of Internet forums or message boards have risen considerably in recent years as Internet access is becoming more and more prevalent. Their main attraction is the ability for people of similar interests to hold conversations on a bulletin board style interface. We decided to analyze one of the most widely used forum software, phpBB. phpBB is open source and free, and also includes a large number of features for both users and administrators. Because of its popularity, any potential threat towards this software could affect many online forums. The objective is to make several different types of attacks to investigate the possible exploits and vulnerabilities phpBB possesses, and then suggest improvements to mitigate the threats.

In an attempt to attack the forum, we tried the following methods of attack: SQL injection, session hijacking, XSS scripting, denial of service, packet sniffing, session hijacking, phishing and brute forcing passwords. Lastly, we analyzed the outcomes found through the attacks and make proper suggestions to eliminate the vulnerabilities and improving the overall security of phpBB.

The content of the remaining analysis will continue with related work, risk analysis of forums, explanation on the successful and unsuccessful attacks, counter measures or suggestions for the vulnerabilities, and conclusion.

II. RELATED WORK

Security analysis has always played an important role in the development of phpBB. In phpBB 3.0, the source code was examined by open source communities as well as private corporations. Secunia, a software vulnerability analyst company, publishes phpBB 3.x vulnerability advisories to all known security flaws. Secunia relies on vulnerability submissions from the public for its vulnerability sources. Secunia then verifies, validates these submissions and publishes them as their vulnerability reports. Solutions to the advisories are also offered shortly after the advisories are published. The number of advisories depends on the amount of valid submissions by the public. Public users mostly report vulnerabilities on cross-site scripting and security bypasses [1]. Although Secunia reports security vulnerabilities, the methodology in which they did their tests was not published, and is therefore unknown.

III. ANALYZED SYSTEM

phpBB is one of the largest open source forum software projects. Its popularity is due to its large development team and its frequent updates.

The first thing that was done was setting up a phpBB forum on a private server. The forum is installed on an Apache server running PHP5, with a MySQL database keeping track of the data. Team members signed up for a personal regular user account and have access to a shared administrator account. When creating an account, the user needs to enter a username, password and email. Once the account is created, users are required to confirm their sign up request by clicking onto a link sent to their email. The major functions of the board are posting messages, posting images and sending private messages to other members. All attempts to hack, crack and disrupt the system were performed on this dummy forum.

IV. THREAT ANALYSIS

A. Risk Analysis

The assets at risk in phpBB are generally the information stored in the accounts, such as emails, birthdays, instant message contacts and private messages, should the account possess any of that information. Should the account of an administrator be compromised, everything on the forum, including its posts, user accounts, and configuration is revealed to the attacker.

The threat for the loss of personal information and sensitive information is an example of a disclosure attack. Secondly the threats for the account itself and its functions is an example of deception. In terms of deception, an attacker has full privilege to act in the place of the account in question. Thirdly, a disruption attack could be made when the attacker deletes the victim's account, leading to a denial of service and modification of account. Finally, an there is an usurpation threat, accomplished by modifying the password to take permanent control of the account and creating a denial of service for the victim.

The potential threat agents that would want to attain the assets of phpBB would be identity thieves, spammers, marketers, hackers, and pranksters

All the CIA proprieties, confidentiality, integrity, and availability, are violated by these threats. Confidentiality is threatened by snooping of hidden information on the account, and possibly by spoofing and private messaging with other users to gain sensitive information. Spoofing as the victim and making posts or private messages or modifications to the profile affects both origin and data integrity. Finally through deletion or complete usurpation of the account, a denial of service is performed and therefore violates the availability property.

B. Failed Attacks

1) SQL Injection

One of the most basic attacks on an application that operates on user generated content is the use of SQL injection. In its most basic form SQL injection involves prematurely terminating an input string and appending extra parameters into the SQL statement. Generally, the attacker achieves this with an ending quote '. Built into PHP version 5 and higher is a method called *mysqli_escape_string()* which replaces all instances of quotes and other terminating characters with characters that are SQL friendly [2]. This method is used extensively by phpBB3, with each POST and GET variable processed through this method, thus invalidating all attempts to compromise the system through this attack. One thing to note is that *mysqli_escape_string()* does not escape "%" and "_", which may alter results in *LIKE* clauses in a SQL statement, but these flaws do not appear to pose any security risks.

2) Cross-site Scripting

A XSS scripting attack involves injecting either an inline frame or a link to a JavaScript program into the page. Depending on how often a thread or a post is viewed, this could be potentially a very widespread attack on a forum and its users. With this potential vulnerability in mind, we attempted to inject Javascript into private messages, posts, post titles, and user account fields:

```
<script>alert("attack successful");</script>
```

The attack failed, as it did not successful activate the script from the personal message, thread or post. The result was only the text being printed out with no adverse effects. In another attempt, we tried to inject an inline frame into similar places, but only to find the same result. Similar to SQL injection, the input is

sanitized by escaping necessary characters for scripting. This time, instead of using *mysqli_escape_string()*, phpBB uses a regular expression replacement to replace HTML tags with their HTML friendly equivalents, such as "<" to "<".

3) Denial of Service

Denial of service refers to taking down a service for either a user or multiple users. This attack can be either performed either by overworking the server the forum is running on by sending it a lot of traffic or specially targeting a user and locking up their account. Attacking the server will work on virtually any website as long as one has a way of sending a very large amount of data all at once. Instead, it would be more applicable to phpBB's resistance against denial of service if individual accounts were locked out instead. To do this, an attempt was made to break the log in form, by inputting many incorrect passwords in hopes that the system will recognise an attempt to hack into the account. The only consequence was that after the fifth attempt, which was the default setting upon installation of the forum, subsequent login requests will require a CAPTCHA authentication before the login will be initiated. The system did not disallow any continuous attempts to log in. It is apparent that phpBB does not have any form of flood control for user authentication, and will not deny users from attempting to log in.

C. Successful Attacks

1) Packet Sniffing

By default, the traffic between both client and server is unencrypted. While not every forum has a need for an SSL certificate, the lack of encryption is still a security flaw. For example, upon login, the username and password of the user is sent in plaintext and is clearly marked in the HTTP POST headers. Using a packet sniffer, it is possible to intercept this message and use it to gain access to a user's account. In the event that an administrator's account is compromised this way, the whole forum's integrity is at risk. Unfortunately, encrypting data is the only method of preventing this sort of attack, which may not be feasible given the price of SSL certificates. This breaks one of the principles of secure design, question assumptions. It is assumed that the system is secure because phpBB checks for a username and password upon each login.

2) Session Hijacking

Web applications keep track of logged in users for the convenience of the user, which is implemented through the use of sessions. With phpBB, users are tracked using three cookies: one serves as an autologin flag, with suffix k, another as the user identifier, with suffix u, and the last one as a session identifier, with suffix sid as shown in Figure 1.

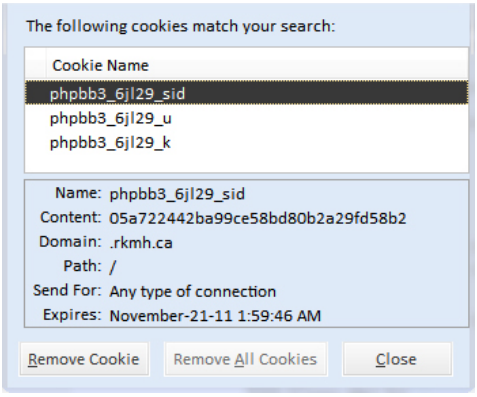

Figure 1. Session identifier cookie.

By default, phpBB checks these three cookies, the user's user-agent, as well as the first three bytes of a user's IP address to authenticate the user. If the attacker happens to be on the same network, spoofing these three things is quite simple, as traffic sent between client and server is unencrypted. Gaining access to the user's account is simple because the cookie is exchanged quite often. Once again, the only method of preventing this attack is through encryption of the session cookies. This again breaks the "question assumptions" principle of secure design. The system only checks for identification that can be very easily stolen. It compromises security for convenience, but should perform more checks to prevent such an attack.

3) Phishing

Security is only as strong as the weakest link in the design. Once the weak link has been breached, the system is vulnerable to attack. Often, the weak link does is located within the software itself, but the users themselves [3]. Phishing requires the use of social engineering techniques to deceit the victims into providing their own personal information, such as passwords, to the attacker without their knowledge.

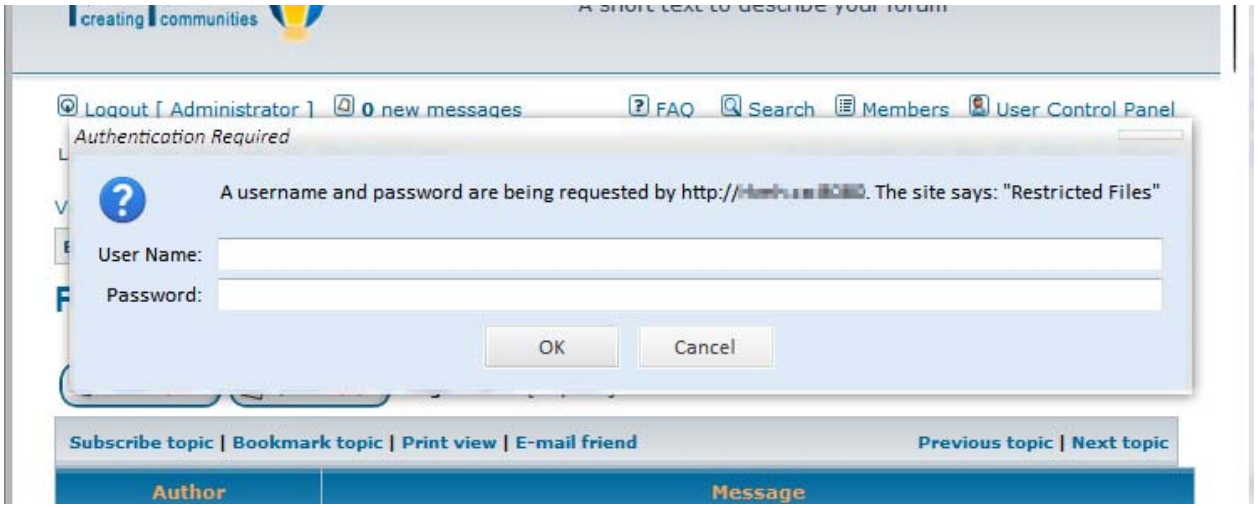

Figure 2. Phishing attempt for username and password.

Phishers can launch a phishing attack from a forum thread. Unlike the spam email phishing method, where the attacker must recreate a counterfeit website, this attack is setup on the official website. It avoids any flaws to the website which may cause the victims to doubt the authenticity of the program. Within any thread, the phisher embeds an image which is stored in a protected folder in the phisher's server. When the Internet browser loads the thread to the embedded image, the browser will prompt the victim for username and password such as in Figure 2.

Unknowingly, the victim may assume the login screen is legitimate and enter his or her account information. The information is then be logged in the phisher's server and the account is then compromised. This method breaches one of the principles of secure design, complete mediation. Since phpBB does not verify the accessibility of images by default, attackers can force the browser to prompt for username and password when loading the image.

4) Exhaust Search

A simple method of attack is to gain access of an account through exhaustive search. This can be time consuming as there is no fixed number for password lengths and users can choose a variety of characters for their passwords. However, phpBB does not implement exponential delay or account disabling for multiple login attempts. In theory, an attacker can conduct an online exhaustive search on a specific account until password is revealed. The only preventive tool available is CAPTCHA. However, CAPTCHA can be defeated by using an algorithm to calculate the CAPTCHA message [4]. Therefore, in theory, an attacker can bypass the CAPTCHA anti-automation feature and conducts the password search with nothing to stop the attacker from attempting such attack.

V. SECURITY PRINCIPLES

Generally, there are 10 security principles a system must follow in order it to be secure. Four of these principles were violated in this system.

A. Complete Mediation

The principle of complete mediation is to ensure that every object must be checked to prevent actions from being performed without the system's consent. This is evident in the system allowing a user to embed a protected image into the page and thus pop up a deceiving login prompt to the user. Mitigating this deception and thus following the security principle simply requires issuing a check for each image embedded into the post before publishing.

B. Psychological Acceptability

Psychological acceptability is a principle regarding user-friendliness. Specifically, a user should not have to do anything extra, yet still be secure from attacks. The current implementation of CAPTCHA is not only difficult for computers to understand, but for humans as well. It should be possible to create a different form of CAPTCHA easy for people to understand and hard for computers, for example, one involving illustrations instead of text.

C. Layer Your Defenses

In order to get to an Administrator Control Panel (ACP), there should be multiple layers of authentication. In its current state, phpBB only requires the username and password of an administrator. Because the ACP has access to everything from user accounts to forum structure, it should have more layers of security than the login information of an administrator. It is suggested that an extra password and some form of human validation be included to ensure that the administrator in question has the proper authentication.

D. Question Assumptions

The system should frequently check the credentials of the user. Currently, session cookies go unchecked by the system, and it is assumed that the user's session cookies, user agent, and IP address are not compromised. It should, at least for the purposes of this security principle, verify that the user has the credentials to access the particular account in question. A possibility would be to assign the user a new session identifier upon each page load, and logging the user out upon five minutes of inactivity, leaving the attacker very little time to steal the correct session cookies.

VI. PROPOSED COUNTERMEASURE

E. Strict Password Policy

A good password policy is the first line of defense in preventing attackers from infiltrating the software. By default, phpBB does not invoke any password strength requirements to its users. However, administrators can increase the strength of passwords to increase security measures. The software allows administrators several options for password strengths: mixed characters, alphanumeric passwords, or passwords with special characters. Administrators can also set the minimum and maximum lengths for passwords. The National Institute of Standards and Technology recommends the use of passwords with more than 8 characters, containing mixed cases with numbers and special characters, and avoiding the use of words that can be found in a dictionary [5].

F. Exhaustive Search Prevention

Password guessing involves the mass generation of passwords and using them to attempt account access. Generally, this testing is done by a computer. To prevent this computer intervention, phpBB employs CAPTCHA, a character recognition system used to differentiate humans from computers. In its current state, the system uses CAPTCHA when a user fails to enter the correct password five times at a specific physical location. We would suggest that the CAPTCHA be used on a per account basis, which will prevent a distributed attack from multiple IPs.

Another way to prevent this form of attack is to limit the number of login attempts a user has for a set period of time. While this form of prevention can create a denial of service problem, it is a more secure method of preventing unauthorized access to an account. A proposed solution to that this would be to deactivate the account in question and send an activation code to the account holder’s email, notifying the user of a hacking attempt, suggesting the user change passwords.

G. Hacker Warning System Implementation

A solution to the previous section of having an attacker brute force a password is to deactivate the account and send the user an email to reactivate the account. A less intrusive would be to require a security question written and answered by the user during registration. This question should show up after a set number of attempts to login. If the question is also wrong, then the account should be frozen until the user changes passwords.

VII. CONCLUSION

Overall, phpBB3 is quite secure as a software system. From the attempted attacks that were conducted, phpBB3 had found to be safe from attacks such as SQL injection, cross-site scripting, and denial of service. However, attacks such as packet sniffing, session hijacking, phishing, exhaustive search are still methods which can be used to attack phpBB3. Improvements which would patch the vulnerabilities include stricter password policies, extended CAPTCHA features, and forced encryption for login and cookie traffic at the very least.

With an immense support from the phpBB developers, computer security companies, as well as the general public, the amount of vulnerabilities to the software decreases as the life of the software continues. Whenever vulnerability is found, it is shortly reported and developers will initiate the fix. The open-source forum software, phpBB3, will continue to strive to be a safe and secure, user friendly software.

REFERENCE

[1] Secunia. (2010, October) Vulnerability Report: phpBB 3.x. [Online]. http://secunia.com/advisories/product/17998/?task=statistics

[2] The PHP Group. (2010, February) PHP. [Online]. http://php.net/manual/en/mysqli.real-escape-string.php

[3] Gary McGraw and John Viega. (2000, Oct) Software security principles: Part 1. [Online]. http://www.ibm.com/developerworks/linux/library/s-link.html#h3

[4] MessageLabs. (2008, February) MessageLabs Intelligence: Spammers defeat Google CAPTCHA mechanisms. [Online]. http://www.messagelabs.com/resources/press/11657

[5] William Burr, Donna Dodson, and Timothy Polk, "Electronic Authentication Guideline," U.S. Department of Commerce, Gaithersburg, NIST Special Publication 800-63, 2006.