Local File Disclosure Vulnerability: A Case Study of Public-Sector Web Applications

M. Imran Ahmed\textsuperscript{a}, Md Maruf Hassan\textsuperscript{b}, Touhid Bhuyian\textsuperscript{c}

Robi Axiata Limited\textsuperscript{a}, Daffodil International University\textsuperscript{b,c}
Dhaka, Bangladesh

Email: maruf.swe@diu.edu.bd

Abstract. Almost all public-sector organisations in Bangladesh now offer online services through web applications, along with the existing channels, in their endeavour to realise the dream of a ‘Digital Bangladesh’. Nations across the world have joined the online environment thanks to training and awareness initiatives by their government. File sharing and downloading activities using web applications have now become very common, not only ensuring the easy distribution of different types of files and documents but also enormously reducing the time and effort of users. Although the online services that are being used frequently have made users’ life easier, it has increased the risk of exploitation of local file disclosure (LFD) vulnerability in the web applications of different public-sector organisations due to unsecure design and careless coding. This paper analyses the root cause of LFD vulnerability, its exploitation techniques, and its impact on 129 public-sector websites in Bangladesh by examining the use of manual black box testing approach.

Keywords—Web application vulnerability, Local File Disclosure (LFD), public sector.

1. Introduction
Web applications have received huge acceptance and popularity due to rise in the number of internet users. Online features and facilities are not only inspiring users to ask questions and provide support but also creating new opportunities in every business sector. Thus, companies in Bangladesh are increasingly rearranging their work processes and data distribution through web applications for better performance. Digital Bangladesh is one of the major components of Vision 2021. The government of Bangladesh has undertaken several initiatives to implement numerous projects related to digital technology, and a number of those projects are in currently progress such as e-government, e-business, e-learning, e-health, e-employment, e-environment, e-agriculture, e-science, e-citizen services, online banking, online admission, online registration, online ticketing and mobile ticketing [1]. The government had also made it a regulatory requirement for all educational institutions to set up websites within July 2015 [2].

We have observed that a good number web application in the public sector do not meet fully the security requirements in design and coding, which makes these applications very risky in a number of ways. Data/file sharing through the download facility is the most common feature of the website of a public-sector organisation, as it ensures easy distribution of information. Owing to a lack of security of the download process, an intruder can easily exploit the weakness and download files without authorisation, leading to the disclosure of confidential information.
This paper presents an assessment and analysis of the LFD vulnerability with its four major exploitation techniques. We have also tested these four techniques on the websites of different public-sector organisations in Bangladesh. The paper is structured into five sections. Sections 1 and 2 present an introduction to the study and a literature review respectively. Section 3 gives an overview of LFD and a review of its code. After performing a data analysis, results and statistics are presented in Section 4. Section 5 concludes the paper.

2. Literature Review

Many studies have focused on web application vulnerabilities, and its exploitation and prevention techniques. S. Gupta and L. Sharma (2012) used the XAMPP server for experimenting XSS exploitation techniques and implemented the same in blogs. In their review, they found that the sandbox environment of web browsers would be a novel technique for mitigating XSS vulnerability [3]. S. Chavan and D. Meshram (2013), who identified the root cause of different web application vulnerabilities, segregated the problems from the phase of SDLC. They also proposed countermeasures for those vulnerabilities [4]. R. Johari and P. Sharma (2012) presented a detailed review of SQLi and XSS vulnerability attacks, and the techniques to prevent them [5]. O. B. Al-Khurafi et al. (2015) conducted a survey on SQLi, XSS, broken authentication and session management, and described different attack techniques of the above-mentioned vulnerabilities [6].

Several researches also studied different web application vulnerabilities such as Structured Query Language Injection (SQLi), Cross Site Scripting (XSS), broken authentication, Cross Site Request Forgery (CSRF), Local File Inclusion (LFI), and Remote File Inclusion (RFI), along with their exploitation of Bangladeshi domains. D. Alam et al. (2015) investigated the user-input-based SQLi technique implemented on the web applications of .bd domain [7]. They also studied three major SQLi techniques implemented on Bangladeshi educational websites and analysed the impact after exploitation [8]. T. Farah et al. (2015) conducted a similar study on financial web applications in Bangladesh and provided the security status. T. Farah et al. (2016) also studied XSS and CSRF vulnerabilities of Bangladeshi web applications. In their article, the researchers drew the attention to coding flaws in the web applications and prescribed some solutions [9]. A. Begum et al. (2016) provided a detailed description of LFI vulnerability exploitation techniques based on RFI and SQLi, and showed the impact after conducting the given exploitation on 153 web applications of Bangladesh [10]. M. Hassan et al. (2016) conducted a case study of LFD exploitation on educational websites in Bangladesh [11].

The above literature review shows that there has been no examination of LFD exploitation on public-sector web applications in Bangladesh. In this paper, we will investigate the public-sector web applications using four types of LFD exploitation techniques.

3. Overview of LFD and Its Code Review

LFD is a kind of vulnerability that allows users to download sensitive files/information from a website in an unauthorised manner by misusing download features of the application. For example, on a web application which is built on PHP and where LFD vulnerability is present, users may access several restricted files, e.g. config.php/, boot.ini/, and index.php, using directory traversal techniques, by changing the file path of infected parameters. By downloading those secret files, users may get their hands on very confidential information like usernames, passwords, database username, and port numbers.
A web application comprises two parts, namely the server side and the client side. As soon as a user of the application makes a request for a service (e.g. downloading a file, request for information, etc.) from the web browser, i.e. client side, it refers the request to the main web application server. From the server side, the application will process the requested or retrieved information from the database and return it to the client-side browser’s interface. Figure 1 shows how a hacker can exploit the LFD vulnerability [11].

3.1. Cause of LFD Vulnerability

An inappropriate use of some built-in functions of popular PHP programming language, e.g. `read_file()`, `file()`, and `file_get_contents()`, causes LFD vulnerability.

```php
Line 01: <?php
Line 02: $vulnerable_file=readfile($_GET['file_name']);
Line 03: print_r($vulnerable_file);
Line 04: ?>
```

Figure 2: Example of LFD-vulnerable coding pattern using $_GET method

Figure 2 above gives an example of PHP code with LFD vulnerability. The code highlights the areas of mistake that might occur during the development period leading to LFD vulnerability. The function `readfile()` is used to read a specific file from a particular location on the host server (Line 02, Figure 2). However, the use of the function results in the LFD vulnerability due to the function not being closed, causing serious loopholes in the code. In this case, a user will be able to access files illegally by changing the specific filename/location.

3.2. Exploitation Techniques

There are four types of LFD exploitation techniques:

3.2.1 General LFD Exploitation
3.2.2 Base 64/ URL/Hex encoding exploitation
3.2.3 Long directory transversal exploitation
3.2.4 HTTP POST request base exploitation

3.2.1 General LFD Exploitation. LFD vulnerability is normally found in the document sharing sections of web applications. By simply clicking on the file download button, a user can view/download the specific shared file (e.g. document.pdf). After that, the `file_get_contents()` function
connects to the server’s database and allows the user to send his query to the database for to access the selected file. In this case, the download operation is conducted using the $_GET method, in which the file parameter/name is displayed on the URL bar. If a user changes the file name, the given file will be downloaded straight way in case LFD vulnerability exists in the application. Consider a web application having the following URL with anyfile.pdf to download:

http://www.demoname.com/files/shared.php?file=anyfile.pdf

The above URL shows a page with a file, anyfile.pdf, ready to be downloaded. If the file name is modified by the user, the downloading process will continue in case the said filename is present in the specific directory. For example, if the filename of the above URL is changed as follows, it is clear that config.php will be downloaded.

http://www.demoname.com/files/shared.php?file=config.php

Once the user can download the configuration file, the following web application server information ($hostname = ‘specific server’; $username = ‘root’; $password = ‘dbpass’; $database = ‘dbname’; $connector = mysql_connect ($hostname, $username, $password); mysql_select_db) can be disclosed as well. Using the same technique, one can even download the source code of the index.php file, which is very alarming for the host of the web application.

3.2.2. Base 64/ URL/Hex Encoding Exploitation. To hide the filename from the web application’s URL, developers use encoding methods, e.g. base64, hex, and URL encoding, for security reasons.

http://www.demoname.com/files/shared.php?file=YW55ZmlsZS5wZGY=

The above URL is an example where the filename is encoded using base64. To decode the URL, the attacker first has to identify the encoding methods and reveal the filename using online decoding tools, in this case base64. The attacker then sends the request to the server with the URL, along with the decoded filename. Using this technique, an unauthorised user can download confidential files.

3.2.3. Long Directory Traversal Exploitation. For security reasons, developers sometimes store files in different directories, thereby making it difficult for the attacker to find the desired file. In this case, the attacker uses directory traversal query for accessing the desired file to bypass this security trick. By adding command (../), the user can get access to the previous directory.

http://www.vulnsite.com/downloads.php?file=../../../../../../../../config.php

If the above config.php does not appear in the given directory where the downloading file is located, every time the ‘../’ command is used, it will go to the next directory. It will explore the root and check whether the desired file exists or not. The file in question will automatically download if and when the intruder selects the right directory.

3.2.4. Long Directory Traversal Exploitation. Web developers usually design data processing techniques through HTTP POST methods for enhanced security. A general user will not be able to view the transformation data easily as it is stored in the cookie. An attacker can bypass this restriction if the code contains a design bug as follows:
/* code start
Line 01: $var_namee = readfile($_POST['anyfile']);
/vulnerable coding pattern ($_POST method)
Line 02: Print_r($vulnerable file);
Line 03: ?>
*/ code end

Figure 3: Example of LFD-vulnerable coding pattern using $_POST method

In the above code in Figure 3, the data parameter would not be seen on the URL bar as it uses the POST method. Using tools like live HTTP headers, Burp Suite, temper data, and the regular HackBar on Firefox, a user can retrieve cookies where the download file information can be visible and the filename can also be modifiable.

3.3. Test Case

The test cases stated below are being followed to conduct our study for exploiting LFD vulnerable public sector web applications of Bangladesh.

i. Check whether the file name is visible in the URL or not after clicking the downloading link of the web application.

ii. Check whether the cookies can be retrieved or not in case the file name is not visible in the URL.

iii. Check whether the filename is in plaintext or in encoded format.

iv. Check whether the encoded filename can be converted into plaintext or not using tools.

v. Check whether the filename can be modified from the URL or not.

vi. Check whether the desired file can be discovered through directory traversal technique or not in case the file is not in public download folder of the web application.

vii. Check whether the concerned file is downloaded or not after modifying the desired file name in URL.

viii. Check whether the database connectivity can be established using third party software or not after getting the respective information from the downloaded confidential file.

4. Result Analysis

In this research, we have articulated the sample size mechanism using the universal calculator provided by G*Power 3.1.9.2. We studied 129 websites of public-sector organisations in Bangladesh using the random sampling method. Of these, 29.45% websites were found to have LFD vulnerability. The existence of four types LFD vulnerability was found on those websites. We used the manual black box testing approach to collect data for this study. We analysed this data set according to the LFD exploitation type, the institution type, and the level of access to the host system. The analysis is discussed below:

4.1. Result Based on the Access Level

Figure 4 represents the level of access to the host system achieved after exploiting the LFD vulnerability of 38 public-sector web applications.
In this examination, full access to the host system was obtained in the case of 44.74% websites, whereas only database access could be established on 31.58% websites through remote access using third-party software. In our case, we used HeidiSQL for connecting with the host’s database system remotely. On 23.68% of the sites, system parameters such as port number, username, password, and IP address could be accessed; however, we were unable to access such information due to a port forwarding problem.

4.2. Result Based on the Institution Type
Figure 5 shows institution-wise LFD exploitable web applications of Bangladeshi public-sector organisations. The figure shows that LFD vulnerability exists mostly on the web applications of government organisations and public universities in Bangladesh, their respective percentages being 21.05% and 28.95%.

The same weakness was found on the websites of 15.79% of public colleges and 18.42% of state-run schools. The web applications of 10.53% of government institutions have this vulnerability. Noticeably, only 5.26% of ministry websites suffer from the above weakness.

4.3. Result Based on LFD Exploitation Types
Figure 6 gives the percentages of Bangladeshi public-sector web applications that are exploited by LFD vulnerability exploitation techniques.
Figure 6: LFD Exploitation Types

| LFD Type                                      | Percentage |
|-----------------------------------------------|------------|
| General LFD Exploitation                     | 55.26%     |
| Base 64/ URL/Hex Encoding Exploitation        | 15.79%     |
| Long Directory Transversal Exploitation       | 23.68%     |
| HTTP Post Request Base Exploitation           | 5.26%      |

Figure 6 gives the percentages of Bangladeshi public-sector web applications that are exploited by LFD vulnerability exploitation techniques. It can be seen that more than half the LFD-vulnerable sites in our sample are exploitable through General LFD Exploitation. HTTP Post Request Base Exploitation is the least-used technique to get unauthorised information/access. Long Directory Transversal Exploitation and Base 64/URL/Hex Encoding Exploitation are used 23.68% and 15.79% respectively.

5. Conclusion

Owing to their versatility and easy accessibility, the web applications of public-sector organisations in Bangladesh are growing in importance. In our evaluation of 129 such websites, we found that web applications with LFD vulnerability would concede their host’s administrative access to unauthorised intruders in most cases. This vulnerability would give an attacker the ability to manipulate sophisticated and confidential documents such as student selection lists for admission and academic results. Many websites of public universities in Bangladesh contain LFD vulnerability, which makes their system very unsafe due to the risk of the disclosure of sensitive data. This may have adverse implications not only for the future of the students, but also for the reputation of the university. Our research in this area is a continuous process. Our observation reveals that the presence of LFD vulnerability in our domain of study is created only because of the carelessness of application developers. Errors on the developers’ part include, for example, not closing the readfile() function, not using the SWITCH function, etc. We believe that careful coding and regular monitoring during the development of web applications will mitigate the risk of such oversight that causes serious vulnerability in the applications.

5.1. Limitation & Future Work

We conducted our study using manual black box testing and did a review of public-sector web applications. In the future, we will design an LFD detection model and implement it using automated detection tools.
References

[1] “Digital Bangladesh: Dreams and reality”, 2017. [Online]. Available: http://www.thedailystar.net/supplements/24th-anniversary-the-daily-star-part-1/digital-bangladesh-dreams-and-reality-73118

[2] “School & college compulsory website of Bangladesh”, 2017. [Online]. Available: http://allnewsview.com/school-college-compulsory-website-of-bangladesh [Accessed: 11 September 2016]

[3] Gupta, S., & Sharma, L. (2012). Exploitation of cross-site scripting (XSS) vulnerability on real-world web applications and its defense. International Journal of Computer Applications, 60(14).

[4] Chavan, S. B., & Meshram, B. B. (2013). Classification of web application vulnerabilities. International Journal of Engineering Science and Innovative Technology (IJESIT), 2, pp. 226–34.

[5] Johari, R., & Sharma, P. (2012, May). A survey on web application vulnerabilities (SQLIA, XSS) exploitation and security engine for SQL injection. In Communication Systems and Network Technologies (CSNT), 2012 International Conference on (pp. 453–58). IEEE.

[6] Al-Khurafi, O. B., & Al-Ahmad, M. A. (2015, December). Survey of web application vulnerability attacks. In Advanced Computer Science Applications and Technologies (ACSAT), 2015 4th International Conference on (pp. 154–158). IEEE.

[7] Alam, D., Kabir, M. A., Bhuiyan, T., & Farah, T. (2015, October). A case study of SQL injection vulnerabilities assessment of .bd domain web applications. In Cyber Security, Cyber Warfare, and Digital Forensic (CyberSec), 2015 Fourth International Conference (pp. 73–77). IEEE.

[8] Alam, D., Bhuiyan, T., Kabir, M. A., & Farah, T. (2015, November). SQLi vulnerability in education sector websites of Bangladesh. In Information Security and Cyber Forensics (InfoSec), 2015 Second International Conference on (pp. 152–57). IEEE.

[9] Farah, T., Shojol, M., Hassan, M., & Alam, D. (2016, July). Assessment of vulnerabilities of web applications of Bangladesh: A case study of XSS & CSRF. In Digital Information and Communication Technology and its Applications (DICTAP), 2016 Sixth International Conference (pp. 74–78). IEEE.

[10] Begum, A., Hassan, M. M., Bhuiyan, T., & Sharif, M. H. (2016, December). RFI and SQLi based local file inclusion vulnerabilities in web applications of Bangladesh. In Computational Intelligence (IWCI), International Workshop on (pp. 21–25). IEEE.

[11] M. M. Hassan, T. Bhuiyan and S. Biswas, “An Investigation of Educational Web Applications in Bangladesh: A Case Study on Local File Disclosure Vulnerability”, 2016 4th International Conference on “Engineering & Technology, Computer, Basic & Applied Sciences” (ECBA-2016), Sydney, Australia, (pp-11-16).

Acknowledgments

We thank the authorities of the organisations who have given us permission to examine their websites.