Security Assessment of Opportunitree Employment Service

December 10, 2016

Nina Rajic (32093130), Steven Wallace (12228145), Nicholas Handaja (54206131), Bojan Stefanovic (14842124)

Team #2

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

nina.rajic@alumni.ubc.ca, steven@thewallaces.ca, nikoniko562@gmail.com, bojan.stefanovic@alumni.ubc.ca

Opportunitree is a website that helps match job-seekers to jobs. Poor security practices to comparable websites, can lead to serious compromises as shown in the LinkedIn break of 2012. We analysed the system as a black box because this is what the owner of the application allowed us to do. We found a cross site scripting vulnerability in the system, but did not modify any data of other users or the system itself.

Ⅰ. Introduction

Opportunitree is a website that serves as a medium to connect job-seekers and employers. Job-seekers are defined as users looking for a job and employers are defined as companies or organizations looking for new employees. Through this analysis we aimed to find vulnerabilities related to the accuracy, privacy and security of user account information.

Finding employment is an important milestone in a person's life, made easier by services provided by Opportunitree. However, if saboteurs change account information or impersonate other users it may tarnish the reputation of the attacked job-seeker or employer.

Similarly to Opportunitree, LinkedIn is a social networking website that helps match potential employers and employees based on relevant skills and work experience. Recently, in 2012, LinkedIn's system was compromised leading to the distribution of millions of user passwords [5].

For our analysis we started with simple analysis techniques, such as information gathering, and later used more sophisticated analysis methods. We used port scanners, and analyzed client-side source code to better gauge possible Opportunitree vulnerabilities. Finally, we used a tool called OWASP ZAP that allowed us to modify data being sent to the server to de-sanitize inputs and perform cross-site scripting ("XSS") injections on Opportunitree.

Using OWASP ZAP revealed that XSS attacks were able to be injected in many text based fields on Opportunitree, such as a user's messages or qualifications in his/her resume. XSS attacks have the ability to alter the appearance of Opportunitree, propagate to other users and lure system user's into phishing attacks. Therefore, we find them to be a serious threat to Opportunitree. To rectify this, sanitation of data prior to rendering is highly recommended.

In the following sections Opportunitree will be described in further detail, followed by an exploration of past similar analyses. Finally, a detailed guide to our analysis methodology, results, interpretation of results and recommendations will be detailed.

Ⅱ. Analyzed system

Opportunitree is a professional development web application that is intended to help both companies/employers and individuals/job-seekers. Opportunitree primarily helps both parties by providing a medium for both to find a symbiotic relationship, in the form of employment or a new employee. Through our black box analysis, we were able to roughly gauge the following components and technologies used by Opportunitree.

To host user information in addition to providing extra security measures, Opportunitree uses Cloudflare [1].

Cloudflare is a content delivery network provider that also provides some additional security measures to customers, such as DDoS protection [2]. As Cloudflare offers several different plans with different security measures offered [3], we were unable to conclusively determine all security measures taken by Opportunitree. Further, Opportunitree clearly cites the use of SSL as a security measure [4] and uses JavaScript with the jQuery framework to serve content to their users.

We believe that the 3 possible user types that interact with Opportunitree are a job-seeker, employer and an administrator. A job-seeker is able to search for jobs, apply for jobs, view the status of his/her applications for jobs and edit or upload his/her resume. Additionally, to add personality to his/her account, he/she is able to upload a video or photo. Job-seekers are also able to set their privacy setting for each item, such as photos or videos, limiting who sees that information. An employer is able to post jobs, look through applications and search publicly posted job-seeker resumes. Both job-seekers and employers are able to see and reply to received messages and initiate conversations with public account users. Applying for a job is considered to be a initialization of communication between that specific job-seeker and employer. Finally, administrators are users that are in charge of developing and maintaining the system.

Ⅲ. Related work

In 2012 LinkedIn's security was compromised resulting in the distribution of 6.5 million hashed passwords followed by the release of another 177.5 million in 2016 [5]. LinkedIn's use of unsalted passwords and SHA-1 allowed some security analysts to translate 90% of the hashed passwords into plaintext passwords within two weeks [6]. These results were achieved through a black box unauthorized security analysis, exploiting a vulnerability to gain access to LinkedIn's database.

Monster experienced a similar compromise in 2009, having information, such as user ID's, emails and passwords, from 4.5 million users stolen [7]. Because of the unauthorized nature of this analysis we again believe a vulnerability was found through black box testing and used to access Monster's database containing sensitive user information.

LinkedIn and Monster provide great examples of black box analyses of a job recruitment platform, but the limited information concerning each respective analysis approach make it difficult to analyze Opportunitree in a similar manner. Further, the targeting of sensitive user information in the above analyses combined with Opportunitree's instruction to perform a non-data centered analysis contribute to our decision to not analyze Opportunitree in a similar manner.

An analysis of Facebook revealed that it is possible to send an executable file in a message [8]. Although Facebook blocks users from sending executable files by checking file names, it was found that by adding a space at the end of a file name an executable file could be sent [8]. Similarly to this analysis we tried to see if we were able to upload or send code through Opportunitree using the resume and messaging services.

In 2005, Samy Kamkar deployed an XSS worm, called Samy, on the popular social media website MySpace. This worm utilized an XSS exploit that sent a friend request to Samy Kamkar and modified the user's profile to include the XSS worm. Within 20 hours, over 1 million users were affected by the worm [9].

Ⅳ. Analysis methodology

A. System Analysis:

To analyse the web application we followed methodologies put out by the computer security community. We started with information gathering. This included manual exploration of the site to gain a better understanding of the layout and how it worked together. Initial findings were compared against symptoms of common vulnerabilities as documented by OWASP in their Top 10 List [10].

We also used a tool called a port scanner that automatically tested all ports of the site and logged any responses for us to review. Port scanners are helpful to find other services running on the same server as the web application that may have vulnerabilities.

Next we reviewed the source of the web page to see if we could find any evidence of frameworks or libraries being used. If older versions of specific frameworks are in use, there may be publicly available exploits known.

We then used a tool called OWASP ZAP. This tool created a proxy server, which we set our browser to send all data through. It allowed us to intercept packets being sent between the browser and the server. This provided us with information on how data is stored in the database and how we can send HTTP requests to obtain this data. OWASP ZAP also allowed us to resend and modify the packets being sent.

B. Ethical Considerations:

As the web application is currently live and in use by real users, we had to consider the impacts of our analysis on both the system owners and all users of the website. In particular, we refrained from using any analysis methods that may affect the integrity of information found on the web application, as

well as its availability. This was done by minimizing our footprint on the site through the use of private accounts that did not engage in any activity with real users. Furthermore, before we conducted a test that we were unsure of we sought advice from experts in the community and the system owners themselves.

C. Risk Management:

The owners of Opportunitree gave us authorization to perform a black box analysis of their system. To reduce possible risks associated with this analysis we did not attempt to analyze any third party systems, such as Cloudflare, used by Opportunitree. This was largely to reduce legal risks as much as possible. Further, as mentioned in our authorization form we agreed to not limit service to customers. Not only did we abide by this rule because of the legal connotation, but also to ensure our analysis was not limited by unhappy stakeholders.

V. Results

Opportunitree was found to have an XSS vulnerability through this analysis. It was found that it is possible to insert a script in text based resume fields, such as a job-seeker's qualifications or past work experiences. These fields can be seen in Fig. 1 and Fig. 2 respectively.

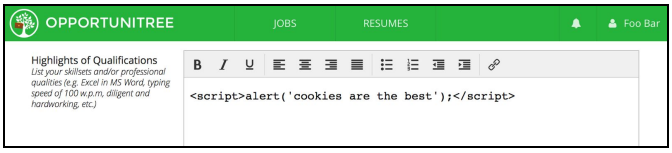

Fig. 1 XSS attack placed in the resume qualification field

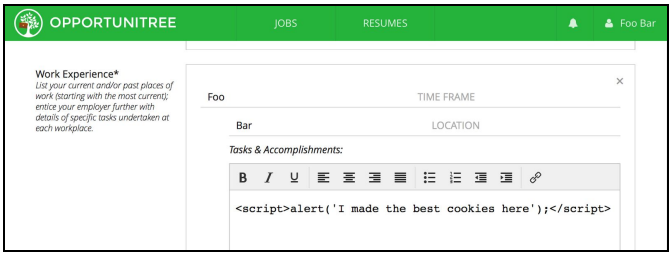

Fig.2 XSS attack placed in the work experience field in the resume

Upon saving the resume, we used OWASP ZAP to view the data being sent to the server. We could see that the JavaScript code running in the browser sanitized the data modifying any angle brackets (<) to an encoded version (<). We used OWASP ZAP to then modify the packet to de-sanitize this data so it would be stored as an HTML element. Upon re-sending the modified packet and refreshing the page, the corresponding script is executed. In the script placed in Fig. 1, an alert will be shown to say "cookies are the best", this can be seen in Fig. 3. The results of the alert script placed in Fig. 2, can be seen in Fig. 4.

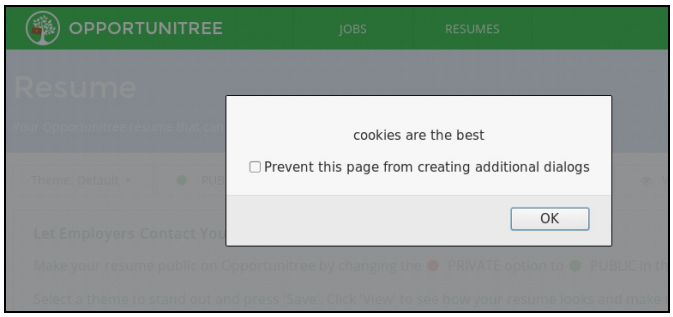

Fig. 3 Result of qualification field injection

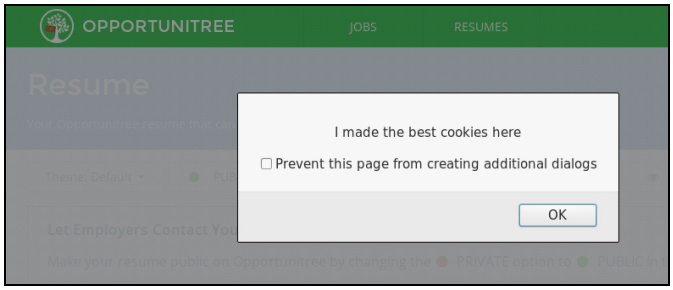

Fig. 4 Result of work experience field injection

Further, in both job-seeker and employer accounts it is possible to insert scripts in messages. An example of a script inserted into a message can be seen in Fig. 5. Similarly to XSS injections in resume fields, using OWASP ZAP to account for security measures and resending the packet allowed us to execute our injected scripts. In both situations because injections are saved within fields on Opportunitree they make up a persistent XSS attack, which would execute again after a page refresh.

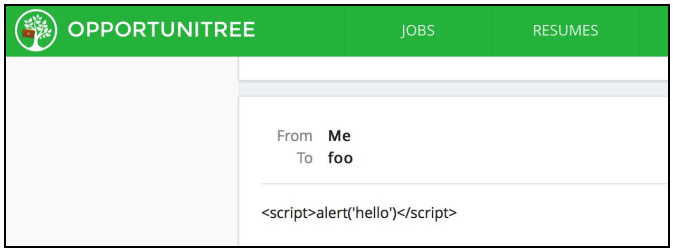

Fig. 5 XSS injection placed in messages

VI. Discussion

A. Interpretation of Results:

Extrapolating from our XSS trials shown in section V, XSS is very dangerous with potentially devastating impacts. Through XSS injections any user with nefarious intent is able to seriously harm Opportunitree and it's users. XSS can be used for phishing and data mining attacks, in addition to the

ability to temporarily change the functionality of a given Opportunitree page.

Using XSS to redirect a user to a different page, could be used to redirect a user to a phishing website where the user is asked to verify his/her account through entering his/her username and password. Upon credential collection the attackers have gained access into the user's Opportunitree account and are able to view sensitive user information, such as job applications. Not only does this allow attackers access to the victimized user's Opportunitree account, but if the user uses the same password for his/her email the attacker has gained access to that email account as well. This access would allow the attackers to gain more personal information further harming the victimized user.

Further, XSS can be used to alter the design of Opportunitree. For example, imagine a job-seeker inserts a script in his/her resume that upon clicking only displays his/her resume in the sidebar, as shown in Fig. 6. Alternatively, the user could have elected to add a beautiful wallpaper to Opportunitree as seen in Fig. 7. While both of these attacks are not grave, they exemplify the capability for large alterations to be made to Opportunitree. Other such large alterations may tarnish Opportunitree's reputation in addition to frustrating user's causing potential user distrust and reluctance to use the system.

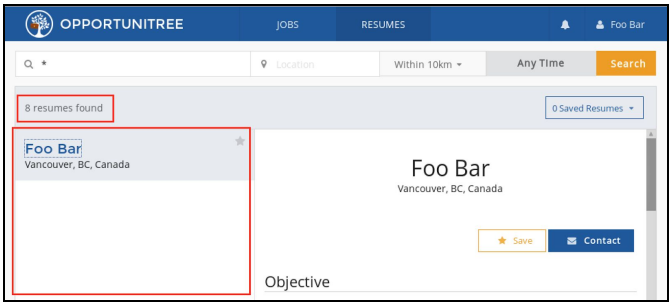

Fig. 6. Only Foo Bar's resume is displayed in sidebar

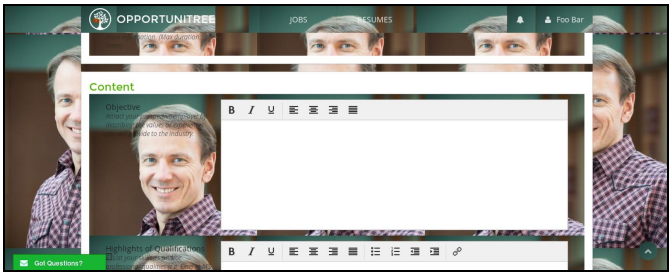

Fig 7. Changing the background of Opportunitree using XSS

An XSS attack like this allows for any JavaScript code the attacker chooses to run on the client's computer. The attacker can access session keys and cookies, send HTTP requests to the site or other webpages with the privileges of the client. On behalf of the user, the code could send messages to other users, change passwords, or modify the user's profile to even include the XSS code itself.

As seen in the Samy worm, discussed in section Ⅲ, XSS injections have the capability of producing a rapidly reproducing worm. Samy's effects were noticeable, with over 1 million users being affected in less than 20 hours [9].

B. Adversary Model:

Our objective was to run JavaScript code on Opportunitree users' computers. We did this through XSS injections that allowed us to perform various acts ranging from simple alerts, to page modification and phishing.

Our initial capability was the ability to partially control the system, by logging into our respective account. We also had some access to the server as we were able to save specific data that was inputted through various fields in resumes and messages.

During the attack we were capable to modify the sanitized data allowing us to complete our XSS attack and the system to execute our scripts. Setting our resume status to public also gave us the ability to run our attack on other Opportunitree users' computers, given that a user clicks on the resume containing our injected script.

C. Principles of Design:

To further secure Opportunitree, emphasis can be placed on adding depth and layers to the design, maintaining an open design, adding fail-safe measure in addition to continuously questioning assumptions.

Currently a user's data is only sanitized once when the packet is first sent from Opportunitree, but not before the input is rendered to a page. The data needs to be checked and sanitized on both the client and server to ensure that no malicious code can be executed. Adding multiple sanitation steps would build the design depth of the system.

Opportunitree never implicitly states what security measures they use, only stating that they use SSL and the latest security measures [4]. This secrecy challenges the open principle of design.

Through using a blacklist filter and basing access decisions based on permissions instead of exclusions the fail-safe principle of design has been neglected.

Finally, the need to constantly question assumptions has been highlighted by the assumption made by designers that all data on the server had been sanitized. As shown by our analysis it is possible to add unsanitized data to the Opportunitree server.

VII. Recommendations

To help alleviate the possibility of XSS attacks we recommend to sanitize data before rendering and filter output text after input text has been converted to HTML.

We currently believe that Opportunitree only sanitizes data from client to server and assumes that all data on the server is valid. Our hypothesized data flow between application components is shown in Fig 8. This assumption allows attackers to step in and use a tool like OWASP ZAP and modify the data after sanitation, but before it gets stored on Opportunitree's servers. Thus, to alleviate this problem sanitizing the data before rendering it would allow XSS injections like ours to be caught prior to execution.

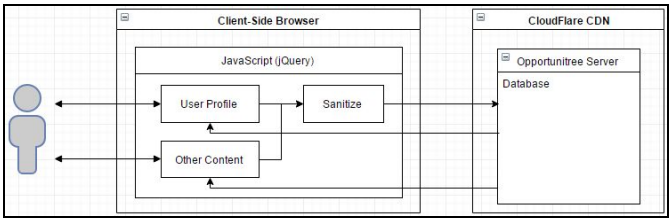

Fig 8. Data Flow Between Components

VIII. Conclusion

Opportunitree is a professional development web application that serves as a medium for job seekers to connect with employers. Some similar analyses include the 2012 LinkedIn data breach [5], Monster's 2009 data breach [7], the ability to send an executable file on facebook [8], and Samy, the XSS worm that infected more than 1 million users in under 20 hours [9].

Through the use of OWASP ZAP we were able to insert and execute XSS attacks on the system. XSS is a large vulnerability with attackers able to modify Opportunitree pages in addition to performing phishing and data mining attacks. XSS attacks allow any JavaScript code chosen by the attacker to be run on the client's computer. The chosen JavaScript can be as simple as inserting an alert to show a message or can collect client session keys and cookies. Even worse, inserted scripts can modify user passwords, information and propagate the script to other users.

To help circumvent XSS, Opportunitree can sanitize data before rendering in addition to their current system of sanitizing when data is sent from the client to the server. The assumption that all data on the server has been properly sanitized showcases the need for better assumption questioning in addition to adding layers of defense in the design. Working on keeping Opportunitree's design open and having additional fail-safe defaults will further help strengthen Opportunitree's security.

As always we would like to extend our sincerest regards to the system owners of Opportunitree, for making this analysis project possible.

References

[1] netcraft.com, "Site report for opportunitree.ca," 2016. [Online]. Available: http://toolbar.netcraft.com/site_report?url=https://opportunitree.ca/htt [Accessed: Nov. 8, 2016].

[2] Cloudflare, "Making the Internet Work the Way It Should for Anything Online," cloudflare.com. [Online]. Available: https://www.cloudflare.com [Accessed: Nov.8, 2016].

[3] Cloudflare, "Cloudflare Pricing," cloudflare.com. [Online]. Available: https://www.cloudflare.com/plans/ [Accessed: Nov.8, 2016].

[4] Opportunitree, "Privacy Policy," opportunitree.ca, 2016. [Online]. Available: https://opportunitree.ca/legal/privacy [Accessed: Nov. 8, 2016].

[5] J. M. Gosney, "How LinkedIn's password sloppiness hurts us all," *Ars Technia*, June 1, 2016. [Online]. Available: http://arstechnica.com/security/2016/06/how-linkedins-password-sloppiness-hurts-us-all/ [Accessed: Nov. 6, 2016].

[6] J.M. Gosney, "The Final Word on the LinkedIn Leak," *Security Nirvana*, 2012. [Online]. Available: http://securitynirvana.blogspot.ca/2012/06/final-word-on-linkedin-leak.html [Accessed: Nov. 6, 2016].

[7] "Millions of jobseeker details stolen in Monster hack," *ITPro*, Jan. 27, 2009. [Online]. Available: http://www.itpro.co.uk/609662/millions-of-jobseeker-details-stolen-in-monster-hack [Accessed: Nov. 6, 2016].

[8] "Facebook security under attack," Computer Fraud and Security, vol 2011, no. 11, pp, 3, Nov. 2011.

[9] "Technical explanation of The MySpace Worm," Applied Hacking. [Online]. Available: http://samy.pl/popular/tech.html [Accessed: Nov. 29 2016].

[10] "OWASP Top 10 2013," OWASP. [Online]. Available: https://www.owasp.org/index.php/Top_10_2013-Top_10 [Accessed: Nov 30 2016].

Appendix A

The code of conduct we have followed through this analysis is:

- Respect individuals' privacy and property
- Avoid harm to others
  - Respect the safety of team members, Opportunitree stakeholders and others
- Act in an honest and trustworthy manner
  - Communicate all vulnerabilities found to Opportunitree
  - All actions are constrained to those agreed upon in authorization form
- All actions taken are intended to ultimately strengthen the security of Opportunitree.

Through this code of conduct we invoked the principles that ensure that people are treated as an end and not a means to an end and acting in a manner that is fair to all parties involved.

APPENDIX B

This report has served as a method to convey the findings of analysis to the Opportunitree's CTO, Mesbah Mowali. As Mesbah is more involved with other stakeholders, such as his partners and system users, we have entrusted him to communicate the findings of our analysis. Nicholas Handaja will further discuss the findings of our analysis with him by Dec. 31, 2016. This date marks the start of the agreed upon six month responsible disclosure period, making the findings available to the public on July 1, 2017. Mesbah can be reached at admin@opportunitree.ca or 604-440-6932.