Ensuring the security of a digital government

Josh Nelbach

**Introduction**

The purpose of e-Government is to increase the overall efficiency and streamline how citizens interact with its government. Section 207 of the E-Government Act requires that all federal agencies make government information available and accessible to the public through various websites (Johnson, 2004). However, there is a genuine concern over how secure e-Government is given the recent data breach that exposed the PII of 21.5 million people. This paper will focus on one particular website and review what it offers, what potential vulnerability it carries, and how to mitigate potential risks by implementing best practices to reduce its risk exposure.

* **SQL Injection**- A SQL injection attack consists of insertion or "injection" of a SQL query via the input data from the client to the application(OWASP, n.d.).
* **Buffer Overflow-**A buffer overflow condition exists when a program attempts to put more data in a buffer than it can hold or when a program attempts to put data in a memory area past a buffer(OWASP, n.d.).
* **Cross-site Scripting-**Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted web sites (OWASP, n.d.)
* **Error Message**- An error message is text that is displayed to describe a problem that has occurred that is preventing the user or the system from completing a task. The problem could result in data corruption or loss (Microsoft, n.d.)

**Review of USAJOBS.GOV**

            All federal agencies are required to list job opportunities, and while some use their own site, most post to USAJOBS.GOV (Collamer, 2015). USAJOBS.GOV is the government’s official website that is used to list vacant job opportunities to the public. Additionally, jobs are open to anyone with the qualifying skillsets. This means that there are opening for those who are in the cleaning and food industry, to more specialized jobs like network architecture.

            USAJOBS.GOV also has a resource center that is geared towards assisting veterans and individual with disabilities the site includes links and additional forms to make finding and applying to federal jobs easier. Furthermore, the resource center also offers information to students and recent graduates; the portal displays links for available internships for students, and jobs for graduates. There are additional links located in the resource center under the students and graduates tab that lead users to more information about additional internship and job programs that are held by the government.

USAJOBS.GOV uses web applications for users to create an account, log in, and upload resumes. This makes the site vulnerable to SQL injections, buffer-overflow, and cross-site scripting if proper validations checks are not in place.  The National Institute of Standards and Technology (NIST) was tasked by the Federal Information Security Management Act (FISMA) to develop guidelines to help agencies categorize the sensitivity level of information and information systems of its potential security impact (Stine, Kissel, Barker, Fahlsing, & Gulick, 2008). For USAJOBS.GOV, the sensitivity level should be moderate given that the loss of confidentially would have “serious adverse effect on organizational operations, organizational assets, [and] individuals” (NIST, 2004, p. 2).

**Web Application Architecture and Security Issues**

A three-tier client-server architecture consists of the presentation tier that interacts with the user by receiving input, and displaying output, the application layer that processes the requests, and the database tier that stores and manages all of the data (IBM, n.d.) . While this is more secure than a two-tier architecture, it is not without its own vulnerabilities. For example, spoofing can occur, which is giving a false identity to gain access to unauthorized data. Information disclosure for instance from weak exception handing can reveal private data. Denial of services which makes the system and data unviable. And escalation of privileges which grants administration access to unauthorized users. Gaining administrative access also grants a malicious user to modify data. All of the mentioned attacks threatens the confidentiality, integrity, and availability of government data.

**Best practices for securing USAJOBS.GOV**

To help mitigate a potential data breach of USAJOBS.GOV, there are proven best practices available to minimize attack vectors. Information exposure through error messaging is one way a malicious user can gain exploitable information. To prevent this from occurring, it is good to write “custom error handling [messages] for situations that are prone to error, such as database access” (Microsoft, n.d.). Furthermore, to protect against SQL injections, developers must use parameterized queries. Using this type of coding will make the database able to decipher what is code and what is data, so the attacker will not be able to inject any malicious code (OWASP, n.d.). Buffer overflow is another vulnerability USAJOBS.GOV has to consider defending against. In order to stop a buffer overflows occurring, enforcing buffer size limitations and stack validation prevents the buffer from accepting more data than it can hold, and the passing of malicious code (Deckard, 2004, p. 12).

The previously mentioned practices to prevent attacks can be backed up by a second layer of defense. A Web Application Firewall (WAF) can secure existing web apps without having to rewrite code. A WAF can prevent privilege escalation, cross-site scripting, SQL injection, and information exposure. Using defense in depth will helps secure the network and maintain the confidentiality, integrity, and availability of data held by the Federal Government.

References

Collamer, N. (2015, November 14). *5 Tips For Getting A Government Job*. Retrieved from forbs,com: http://www.forbes.com/sites/nextavenue/2012/11/14/5-tips-for-getting-a-government-job/#4fb4211e7ca8

Deckard, J. (2004, April 14). *Defeating Overflow Attacks*. Retrieved from sans.org: https://www.sans.org/reading-room/whitepapers/securecode/defeating-overflow-attacks-1403

IBM. (n.d.). *Three-tiered client/server architecture*. Retrieved from ibm.com: http://www.ibm.com/support/knowledgecenter/SSAL2T\_6.2.0/com.ibm.cics.tx.doc/conc/erziaz0105.html

Johnson, C. (2004, December 17). *MEMORANDUM FOR THE HEADS OF EXECUTIVE DEPARTMENTS AND AGENCIES*. Retrieved from www.whitehouse.gov: https://www.whitehouse.gov/sites/default/files/omb/memoranda/fy2005/m05-04.pdf

Microsoft. (n.d.). *Basic Security Practices for Web Applications*. Retrieved from msdn.microsoft.com: https://msdn.microsoft.com/en-us/library/zdh19h94.aspx#cpconbestsecuritypracticesforwebapplicationsanchor6

Microsoft. (n.d.). *Error Message Guidelines*. Retrieved from msdn.microsoft.com: https://msdn.microsoft.com/en-us/library/windows/desktop/ms679325(v=vs.85).aspx

NIST. (2004). *FIPS PUB 199*. Retrieved from http://csrc.nist.gov/: http://csrc.nist.gov/publications/fips/fips199/FIPS-PUB-199-final.pdf

OWASP. (n.d.). *Buffer Overflow*. Retrieved from owasp.org: https://www.owasp.org/index.php/Buffer\_Overflow

OWASP. (n.d.). *Cross-site Scripting (XSS)*. Retrieved from owasp.org: https://www.owasp.org/index.php/Cross-site\_Scripting\_(XSS)

OWASP. (n.d.). *SQL Injection*. Retrieved from https://www.owasp.org/index.php/SQL\_Injection

OWASP. (n.d.). *SQL Injection Prevention Cheat Sheet*. Retrieved from owasp.org: https://www.owasp.org/index.php/SQL\_Injection\_Prevention\_Cheat\_Sheet

Stine, K., Kissel, R., Barker, W. C., Fahlsing, J., & Gulick, J. (2008, August). *Volume I: Guide for Mapping Types of Information and Information Systems to Security Categories*. Retrieved from csrc.nist.gov: http://csrc.nist.gov/publications/nistpubs/800-60-rev1/SP800-60\_Vol1-Rev1.pdf