Don’t Worry, Bar Happy

Security Research Paper

According to the Bureau of Justice from 7,818 business “67% detected at least one cybercrime…nearly 60% detected one or more types of cyber attack.” (“Cybercrime”) This statistic should make anyone thinking of creating any kind of application or website extra consideration in their security efforts. For our group project we are creating a website that takes information from a popular review website and populates a database that consumers can query for information. Our most vulnerable part is the form submission we have as part of our database development. In order to get the most accurate information we are allowing users to anonymously submit data to us to help us get the most up to date information. This however presents a very credible risk, but we can take steps to minimize the threat.

The easiest way we could mitigate any risk from form based attacks would be to eliminate the form altogether. After some discussion we see the form as a critical component to our user experience. While we are able to get the majority of our data from scraping Yelp for data, we are missing important things like specials, specific amenities, and events. While our team will make an earnest effort to fill out a lot of that data we don’t have the manpower to complete all bars necessary which is where the consumer will come in by submitting data via web form.

SQL injections can present a danger to a web application in that sensitive data from tables could be exposed with some “smart” inputs by the user. “SQL injection is a technique where malicious users can inject SQL commands into an SQL statement, via web page input.” (“SQL Injection") There are various ways to try and mitigate damage caused by SQL injection techniques. One of them is to disallow certain words or characters. This isn’t considered ideal though since many of the words disallowed are commonly used for non malicious purposes. The recommendation by W3 Schools is do use SQL parameters. “SQL parameters are values that are added to an SQL query at execution time, in a controlled manner.” (“SQL Injection") This means that input is checked for validity and cannot be used as part of the SQL input to add malicious code.

XXS or Cross Site Scripting is a type of an attack that can be especially pervasive since it uses web applications that are seemingly harmless to infect users. A famous example of XSS was Samy which back in 2005 infected over one million users in less than 24 hours. Typically through the use of Javascript, XXS, will be embedded in code that browser needs to execute and once it has then it is on the user’s device. Typically the goal of XSS attacks “include transmitting private data, like cookies or other session information, to the attacker, redirecting the victim to web content controlled by the attacker, or performing other malicious operations on the user’s machine under the guise of a vulnerable site” (“Cross-site-scripting”) We can protect our web application by performing a security audit of the code. This is accomplished by searching for areas where input from HTTP request could affect our HTML output. There are also tools out there that can assist in securing these flaws.

Server and browser side validation is another important part of securing our web application. Since we are using a web form to submit data we have to be careful about what kind of information we accept. On the browser side of things we need to make sure the correct type of data is being inputted in the right cells. For example we only allow numbers in number fields, and text in text fields. Additionally making sure they are the correct syntax and contain only permitted characters will add a safeguard on the browser side in regards to validation. Additionally, validation should occur on the server side as well. Syntax, range, length, and grammar should all be checked on the server side again before executing.

There are a multitude of different ways for any type of application to be attacked. We see today that applications by multi-billion dollar companies are constantly being updated for “security risks” which goes to show that no project is infallible. With how languages, frameworks, and technologies evolve there will be a constant whack-a-mole game where a new security vulnerability pops up and a fix will whack it down only to have a different flaw pop up someplace else. For this reason active management of web applications threats is a must to provide a secure experience to your users.

Sources

"Cybercrime." Bureau of Justice. National Computer Security Survey, n.d. Web. 15 Nov. 2016.

"10 Security Tips to Protect Your Website from Hackers." Creative Bloq. Creative Bloq ART AND DESIGN INSPIRATION, 2014. Web. 15 Nov. 2016.

"SQL Injection." SQL Injection. N.p., n.d. Web. 21 Nov. 2016.

"Cross-site-scripting." Cross-site-scripting - OWASP. N.p., n.d. Web. 15 Nov. 2016.