1. What is the paper about? What is/are the vulnerability? What causes the vulnerability?

The paper discusses a new defense strategy against cross-site scripting (XSS) attacks called Blueprint. According to the paper, the number one security threats for the Internet at large are XSS attacks. The attacks rely upon unauthorized script code injected into a web page to extract confidential and sensitive user data. The malicious script code is embedded in untrusted HTML and is executed on the victim’s web browser within the context of the vessel website. Enforcing a no-script policy for untrusted HMTL to prevent unauthorized script execution is the aim of defenses against XSS attacks. The strategy of the defense outlined in this paper and the goal for Blueprint is to minimize the burden on the browser in determining untrusted content.

1. What is/are the contributions of the paper? How was the vulnerability or insecurity discovered?

The paper discusses two main defense approaches: content filtering and browser collaboration. In content filtering application attempt to detect and remove all scripts from untrusted HTML while browser collaboration gets information from the browser as to which scripts are authorized.

Content filtering, also widely known as sanitization, uses specially designed filter functions on user input to protect against/remove any potentially malicious data or instruction. Simple sanitization, such as disallowing HTML syntax control characters, because every control character that can be used in a malicious attack code/script also has a legitimate use in many legitimate non-script contexts. Advanced content filters look at untrusted content and try to predict how the client web browser’s parser will interpret it. This not only makes the web application’s job of protection depend in part on the different web browser parsers, but also leaves the content filters open to exploit via any anomalous behavior by any of the browser parsers.

Browser collaboration involved tools to help browsers distinguishing authorized scripts from unauthorized scripts. Protocols are used to communicate a set of authorized/unauthorized scripts, which the browsers use to enforce policies denying execution of the unauthorized scripts. This unfortunately requires web application to use custom browsers to be able to communicate with the server containing the set of scripts; therefore, it lends itself more to a long-term solution, where all browsers already contain this communication by default, rather than a near-term one. Also, the standards by which all browser use for communication would have to be agreed upon, which in itself is a long arduous process.

1. The detailed techniques to solve the problem.

Blueprint claims to satisfy the three main objectives of preventing XSS attacks: 1) it is robust, protecting even with browser quirks, 2) it supports structure, benign HTML derived from untrusted user input, and 3) it is compatible with existing browsers current in use. Since the parsing behavior of browser can be unreliable, Blueprint effectively takes over and controls the parsing decisions instead of the browser. The Blueprint application uses information about the flow of untrusted HTML in a browser to create a “blueprint” or structural representation of untrusted web content, with XSS attacks removed. The generated tree is fed to the browser’s document generator bypassing the browser parser and avoiding its analogous and/or unreliable behavior. The implementation does not require knowledge of how the browser parses the data and instead enforces the application’s understanding of the web content on the browser. In doing this, the effect of the browser parser’s analogous behavior does not come into play.

1. What are the strength/weaknesses of the paper?

The first strength of the program is that it was resilient against even subtle XSS attacks. The platforms used in testing embedded attack strings in a variety of different contexts for a template web page. The program was able to defend against all 94 different XSS attacks testing and there were no successful attacks due to exploits of browser parsing quirks. Since it was testing on eight of the most popular web browsers, which make up over 96% of the browser market, this goes a long way to show how effective it is against for the vast majority of users.

Another strength is they assessed different tools for the prevention/protection not only for how well they did their intended job, but whether or not base code needed to be modified, whether detection/prevention were automated, and identified additional infrastructure that might be necessary.

The first weakness I could see is for any application not written in PHP, an alternate version of Blueprint needs to be run, which has a separate process to communicate with the web application over TCP. The program already consists of a server-side component as well as a client side script library; therefore, adding the separate process makes things even more complicated and increases the possible points of failure. A second weakness that I saw is they ran their tests on popular web application using the most popular web browsers. While that will give the public at large an idea how well it works overall, many company’s application will be unique in the types of data and content that they deal with and may in some cases be using old/outdated browsers. The type of attacks that would work in those cases may never be exercised in the testing on with the most popular applications/browsers.

1. What can you do better?

*The authors of the paper compiled a lot of very useful information about SQL injection attacks and the current techniques for detecting and preventing them. They were also able to discover trends in how many existing products/techniques fail to detect/prevent attacks. However, the authors fail to provide their own way to combat these trends or suggestions as to how to prevent against these pitfalls when creating a product to detect and prevent SQL injection attacks. In addition, the authors also fail to address any possible future attacks. For example, their research could have shown areas which were vulnerable to attacks, but that have not yet been exploited or do not yet have a technique to provide protection against them. This would not only help them in creating a new product/technique for the industry, but also helping those who wish to improve their own products/techniques to try to get ahead of the curve.*