Matthew Boeck
1/9/2012
SE4930
Dr. Schilling

Article Review #2 – Cryptography in the Web: The Case of Cryptographic Design Flaws in ASP.NET

The article that I read this time was about Cryptographic Design Flaws in ASP.NET and can be found at http://www.ieee-security.org/TC/SP2011/PAPERS/2011/paper030.pdf. This article was written in 2011 by Thai Duong and Juliano Rizzo. It was pages 481-489. This article begins by talking about ASP.NET. ASP.NET is essentially the language (and framework) that was developed and managed by Microsoft to be used on the web. The article mentioned that 25% of the websites online are written in and backed by ASP.NET. The article then goes in to talk about HTTP protocol, and builds a baseline so the reader can understand what exactly the design flaws in cryptographic functions of ASP.NET are.

After laying down an understanding of ASP.NET and web functions, the paper talks about several attacks against the cryptographic functions for which are the fundamental backbone of this language and framework. This paper goes into great detail discussing the Padding Oracle Attack. Prior to reading about this, I had no idea this type of attack even existed. As a developer, I have developed commercial applications for a company using ASP.NET. I had to be self-conscious about cookies (and stealing cookies) as well as session management and object (ex: JSON) protection. Until reading this paper, I thought I knew all attack vectors against ASP.NET and the web.

The paper then goes into detail about how the attack works, and the math behind it. Admittedly, a lot of this math went over my head. It did help that I had studied RC4 encryption in Algorithms last year. This paper then talks about cost, and impact, as well as preventative measures. It was interesting to read that because of the researchers work in writing this article, Microsoft immediately released a patch and prevented all of their found attacks. It is important to note that these attacks are still exploitable if the system is not patched!

After reading this article, I thought about how this new knowledge would make me a more secure coder. I think one of the most important realizations to take away from this is that all software can be easily broken. It is important to realize this fact, and quickly acknowledge that ignorance is no excuse. If your software is broken, you must fix it immediately or run the

risk of severe damage. I plan on staying current with trends and making sure to watch for attacks and exploits like this as a future developer.

One of the questions I had from this article is how easily can the concept of The Padding Oracle Attack be applied to other languages/frameworks? What's to stop a similar approach from being applied to say Ruby on Rails? I don't think I have the knowledge (or time) to investigate this theory, but I would like to see it applied elsewhere!