Preface

Fire. The wheel. Electricity. All of these pale next to the monumental achievement that is Ajax. From the moment man first walked upright, he dreamed of, nay, lusted for the day that he would be able to make partial page refreshes in a Web application. Surely Jesse James Garrett was touched by the hand of God Himself the morning he stood in his shower and contemplated the word Ajax.

But like Cortés to the Aztecs, or the Star Wars prequels, what was at first received as a savior was later revealed to be an agent of ultimate destruction. As the staggering security vulnerabilities of Ajax reared their sinister heads, chaos erupted in the streets. Civilizations crumbled. Only two men could dare to confront the overwhelming horror of Ajax. To protect the innocent. To smite the wicked. To stave off the end of all life in the universe.

And we're glad you've paid $49.99 for our book.

Preface (The Real One)

Ajax has completely changed the way we architect and deploy Web applications. Gone are the days of the Web browser as a simple dumb terminal for powerful applications running on Web servers. Today's Ajax applications implement functionality inside a user's Web browser to create responsive desktop-like applications that exist on both the client and the server. We are seeing excellent work from developers at companies like Google and Yahoo! as well the open source community pushing the bounds of what Ajax can do with new features like client-side storage, offline applications, and rich Web APIs.

As Web programmers and security researchers, we rushed out and learned as much as we could about these cool new applications and technologies. While we were excited by all the possibilities Ajax seemed to offer, we were left with a nagging feeling: No one was talking about the security repercussions of this new application architecture. We saw prominent resources and experts in the Ajax field giving poor advice and code samples riddled with dangerous security vulnerabilities such as SQL Injection or Cross-Site Scripting.

Digging deeper, we found that not only were these traditional Web vulnerabilities ignored or relegated to passing mention in an appendix, but there were also larger security concerns with developing Ajax applications: overly granular Web services, application control flow tampering, insecure practices for developing mashups, and easily bypassed authentication mechanisms. Ajax may have the inherent usability strengths of both desktop and Web applications, but it also has both of their inherent security weaknesses. Still, security seems to be an afterthought for most developers.

We hope to change that perspective.

We wrote this book for the Ajax developer who wants to implement the latest and greatest Ajax features in their applications, while still developing them securely to avoid falling prey to evil hackers looking to exploit the applications for personal and financial gain. Throughout the book, we focus not just on presenting you with potential security problems in your Ajax applications, but also on providing guidance on how you can overcome these problems and deliver tighter, more secure code. We also analyze common Ajax frameworks like Prototype, DWR, and Microsoft's ASP.NET AJAX to find out what security protections frameworks have built-in and what you, as a developer, are responsible to add.

We also wrote this book for the quality assurance engineer and the professional penetration tester. We have tried to provide information about common weaknesses and security defects found in Ajax applications. The book discusses the testing challenges you will face in auditing an Ajax application, such as discovering the application's footprint and detecting defects. We review a few tools that aid you in completing these challenging tasks. Finally, we give details on new Ajax attack techniques such as JavaScript hijacking, persistent storage theft, and attacking mashups. We also provide fresh takes on familiar attacks, such as a simplified Ajax-based SQL Injection method, which requires only two requests to extract the entire backend database.

This is not a book for learning Ajax or Web programming—we expect you to have a pretty good handle on that already. Instead, we will focus on the mistakes and problems with the design and creation of Ajax applications that create security vulnerabilities and provide advice on how to develop Ajax applications securely. This book is not program language specific and does not force you to write the server-side of your application in any specific language. There are common components to all Ajax applications, including HTTP, HTML, CSS, and JavaScript. We focus our analysis on these components. When we do provide security advice with respect to your Web server code, we do so using techniques such as regular expressions or string operations that can be implemented using any language.

This book also contains a great deal of material that should benefit both the developer and the tester. Case studies of real-world Ajax applications and how they were hacked, such as MySpace's Samy worm and Yahoo!'s Yamanner worm, are discussed. Sample applications and examples, such as an online travel booking site, provide guidance on how to secure an Ajax application for testers and developers alike.

While we do mean for the book to be read cover-to-cover, front-to-back, each chapter stands on its own. If there's a particular topic you can't wait to discover, such as the analysis of specific Ajax frameworks for security issues (which can be found in Chapter 15, "Analysis of Ajax Frameworks"), feel free to skip ahead or read out of order.

Ajax provides an exciting new philosophy for creating Web applications. This book is by no means an attempt to dismiss Ajax as silly or infeasible from a security perspective. Instead, we hope to provide a resource to help you develop powerful, feature-rich Ajax applications that are extremely useful, while at the same time robust and secure against malicious attackers.

Enjoy,
Billy and Bryan