Preface

Online games, including World of Warcraft, EverQuest, Second Life, and online poker, have taken the computer world by storm. Gaming has always been (and remains) among the prime drivers of PC technology, with deep penetration into the consumer market. In the last ten years, computer games have grown just as quickly as the Internet and can now be found in tens of millions of homes.

The Internet is experiencing plenty of adolescent growing pains along with its phenomenal growth. These pains are experienced mostly in terms of problematic and pervasive computer security issues. Online games, especially massively multiplayer online role-playing games (or MMORPGs for short), suffer from these security problems directly.

MMORPGs are made of very sophisticated software built around a massively distributed client-server architecture. Because these games push the limits of software technology, especially when it comes to state and time (not to mention the real-time interaction of hundreds of thousands of users), they are particularly interesting as a case study in software security. In fact, MMORPGs are a harbinger of technical software security issues to come.

Modern software of all kinds (not just game software) is evolving to be massively distributed, with servers interacting with and providing services for thousands of users at once. The move to Web Services and Service Oriented Architectures built using technologies like AJAX and Ruby follows hard on the heels of online games. What we learn here today is bound to be widely applicable tomorrow in every kind of software.

Adding to the urgency of the security problem is the fact that online games are big business. The most popular MMORPG in the world, World of Warcraft by Blizzard Entertainment, has over 8 million users, each of whom pay $14 per month for the privilege of playing. Analysts estimate the gaming market will reach $12 billion by 2009.

Inside the virtual worlds created by MMORPGs, simple data structures come to have value, mostly a reflection of the time gamers spend playing the game. Players accumulate and trade virtual wealth (or play money). Many of these virtual economies have per capita GDPs greater than most small nations. Not surprisingly, direct connections between the virtual economies of games and the real economy exist all over the place. Until recently, it was possible to buy in-game play money with real dollars on eBay; now many other well-developed middle markets exist. And the reverse is possible, too. This has led to the emergence of a class of players more interested in wringing virtual wealth out of the game than playing the game itself.

Wherever money is at stake, criminals gather and linger. Cheating happens. In the case of MMORPGs, cheaters have real economic incentive to break the security of the game in order to accumulate virtual items and experience points for their characters. Many of these items and even the characters themselves are then sold off to the highest bidder.

Sophisticated hackers have been working the fertile fields of MMORPGs for years, some of them making a living directly from gaming (or cheating at gaming). This book describes explicitly and in a technical way the kinds of attacks and techniques used by hackers who target games.

Why Are We Doing This?

As you can imagine, game companies take a dim view of cheating in their games. If cheating becomes rampant in a game, unsatisfied noncheating players will simply move on to another. Game developers have taken a number of steps to improve security in their games, some of them controversial (monitoring game players' PCs behind the scenes), others legalistic (imposing strict software license agreements and terms of use), and some of them trivial to break (using symmetric cryptography but including the secret key in the game client code). Our hope is that by understanding the kinds of attacks and hacking techniques described in this book, game developers will do a better job with online game security.

We think our topic is important for several reasons: First, real money is at stake; second, many players are completely unaware of what is going on; and third, online game software security has many critical lessons that we can directly apply to other, more important software. Plus, it's fun and controversial.

For example, some game companies have been known to use stealthy techniques most often seen in rootkits to monitor gamers' PCs. They have also been known to resort to strong-arm tactics to suppress hackers, even those not attempting in any way to be malicious or to make money. Will manufacturers of other software or digital content adopt these techniques for themselves?

Not only are the technical issues captivating, the legal issues surrounding online games and their creative software license terms are also a harbinger of things to come. The legal battles between game companies, academics, and users are by no means over—in fact, they have just begun.

In the end, the topic of online game security poses a number of interesting questions, the most pressing one being this: How do you balance gamers' privacy rights against game developers' desires to prevent their games from being hacked?

Where Do We Draw the Line?

For the record, we do not condone cheating, malicious hacking, or any other game-related shenanigans. We are most interested in deeply understanding and discussing what's going on in online game security. As practical security experts, we believe that only by gaining direct technical understanding of what happens when games are exploited can we begin to build systems that can withstand real attacks. Because in this situation money is at stake, you can be sure that attacks and exploits today are both concerted and organized.

We think it is acceptable and necessary to understand both how games really work and how they fail. The only way to do this is to study them carefully. We pull no punches technically in this book, showing you how online game clients fail from a security perspective in living detail. We also explicitly describe techniques that can be used to exploit online games. We don't do this to create an army of online game hackers—that army is already brimming in numbers, and those already enlisted in it are unlikely to learn much from this book. We do this so that the good guys will know what they are really up against. Our main objective is to describe the kinds of weapons the existing active army of game attackers has.

In our research for this book, we have broken no laws. We expect our readers likewise not to break the law using the techniques we describe.

What's in the Book?

Like most books, this book starts out at a high level and becomes progressively more technical as it goes on.

Chapter 1, Why Games?, poses and answers some simple questions. How big are online games? How many people play? Why would anyone want to exploit them? What motivation is there to cheat in an online game? The answers to these questions will likely surprise you. Believe it or not, 10 million people play online games, billions of dollars are at stake, and some people even cheat for a living. We also provide a gentle introduction to game architecture in Chapter 1, describing the classic client-server model that most games use.

Things get more technical beginning in Chapter 2, Game Hacking 101, where we describe the very basics of game hacking. The chapter is organized around describing six basic techniques: (1) building a bot, (2) using the user interface, (3) operating a proxy, (4) manipulating memory, (5) drawing on a debugger, and (6) finding the future. We pay special attention to the topic of bots since most game exploits exist to create and operate them. Late in the chapter, we even show a very simple bot that we built so you can see exactly what bot software looks like. We then describe controversial moves taken by one game maker to thwart cheating—installing rootkit-like spyware on a gamer's PC to keep track of what's going on. We hold this approach in low opinion and have written a program to help you know what's going on with these monitoring programs on your own machine. We believe game makers would be better off spending their resources to build games that were less broken than to build monitoring technology.

The next two chapters take a break from technical material to cover money and the law. In particular, Chapter 3, Money, helps us understand why some players might want to cheat. The recent book *Play Money* by Julian Dibbell (Basic Books, 2006) describes one (pathetic) man's foray into professional game farming, something that a number of people actively pursue. There is enough money in play here that entire enterprises have grown up around providing middleman services for gamers, buying and selling virtual items in a marketplace. The biggest and most interesting company, Internet Gaming Entertainment, known as IGE to most people, deserves and gets a treatment of its own in this chapter.

Chapter 4, Enter the Lawyers, is about the law. Game companies (and indeed a whole host of other software makers) have created a licensing jungle in the form of end user license agreements (EULAs) and terms of use

(TOU) documents. Though we are not lawyers, and by no means should you rely on our advice, we provide a brief description of U.S. copyright law and the Digital Millennium Copyright Act (DMCA). Then we go through an entertaining (and somewhat scary) parade of EULAs gone bad—from Sony's rootkit debacle to viruses protected by EULAs. We end up with a discussion of your rights as a software user and gamer.

Technical aspects of online game security begin to pick back up in Chapter 5, Infested with Bugs. We spend this chapter talking about the kinds of vulnerabilities found in many games, explaining how attackers use them to build working exploits. We pay particular attention to bugs involving time and state, which, as we alluded to earlier, are the kinds of bugs we can expect to see much more of as other software evolves to become more like game software.

Chapter 6, Hacking Game Clients, really digs in and gets technical. As we move more deeply into games, we are forced to use a game or two as a particular target. We don't do this to single out any one game; instead, we do this to make our examples salient and technical. We have chosen to concentrate on World of Warcraft (WoW), a game produced by Blizzard Entertainment, mostly because it is the number one online game in the world. The kinds of techniques we demonstrate using WoW as an example can be applied by analogy to almost any online game (and to modern Web 2.0 software, for that matter). Chapter 6 begins with a discussion of the attacker's toolkit (many of the tools we describe are standard software testing tools). We then organize our discussion of game hacking techniques into four equally important areas: (1) getting over the game by using its user interface directly, (2) getting inside the game by manipulating memory, (3) getting under the game by interposing on services like video drivers, and (4) getting way outside the game by intercepting and manipulating network traffic. Chapter 6 is in some sense the heart of the book, introducing a large number of techniques commonly used by game attackers.

Chapter 6 has lots of data in it, probably too much. Sometimes it is easier to understand these kinds of techniques by seeing how they are put into practice. Toward that end, in Chapter 7, Building a Bot, we put together all of the lessons of Chapter 6. Chapter 7 is technical, with plenty of example code showing how the ideas in Chapter 6 work in concert to exploit a game.

Chapter 8, Reversing, is even more technical, focusing its attention on reverse engineering techniques that many attackers use to exploit software. Though the techniques we describe in this chapter are intense, they are by

no means new. As we describe in our book *Exploiting Software* (Addison-Wesley, 2004), disassemblers and decompilers are used in computer security every day, both by good guys and by bad guys.

The final technical topic in our book is presented in Chapter 9, Advanced Game Hacking Fu. This chapter discusses what is known in the trade as total conversions and game mods. We describe the process by which some people take apart game data files in order to build their own games or combine different aspects of games in interesting ways. Though some game companies try hard to quash all discussion of total conversion, it happens anyway. We describe how.

Of course, our purpose in this book is to help those who build games understand how to do a better job with security. Chapter 10, Software Security Über Alles, provides a flyover of the new field of software security. Game developers would do well to adopt some of the best practices in common use in the financial vertical today. We also describe a set of questions that everyday gamers can ask their game companies about security. Our fervent hope is that this book will lead to more secure software—both in the game community and beyond.

The Software Security Series

This book is part of the Addison-Wesley Software Security Series of software security books for professional software developers. The series includes the following titles:

- *Exploiting Online Games: Cheating Massively Distributed Systems*
- *Secure Programming with Static Analysis*
- *Software Security: Building Security In*
- *Rootkits: Subverting the Windows Kernel*
- *Exploiting Software: How to Break Code*
- *Building Secure Software: How to Avoid Security Problems the Right Way*

More books in this series are planned for the future. Contact Addison-Wesley or Gary McGraw for more information. Also see the Web site <http://buildingsecurityin.com>.

Contacting the Authors

The authors welcome e-mail contact from readers with comments, suggestions, bug fixes, and questions. Contact us through the book's Web site: <http://www.exploitingonlinegames.com>.