PROLOGUE

README.1ST

The Xbox™ video game console from Microsoft® is an exciting piece of hardware, and not just because it can play the latest video games. The powerful and cheap Xbox has the potential to be used as a PC, an all-in-one media player, or even a web server. Unfortunately, there is a dearth of books that can teach a reader how to explore and modify modern electronic hardware such as the Xbox. Most electronics textbooks are theory-oriented and very focused, whereas real hacking requires a broad set of practical skills and knowledge. Also, the few practical books on hardware hacking that I had as inspiration as a child have long been outdated by the fast pace of technology. This book is intended to fill the need for a practical guide to understanding and reverse engineering modern computers: a handbook for a new generation of hackers.

The ultimate benefit of hacking the Xbox is its educational value, or as the saying goes, "Given a fish, eat for a day; learn to fish, eat for a lifetime." Hence, this book focuses on introducing basic hacking techniques — soldering, reverse engineering, debugging — to novice hackers, while providing hardware references and insight that may be useful to more seasoned hackers. The Xbox has served to educate both the security community and the hacking community: not because it is an outstanding example of security, but because it is a high profile, high volume product made by a large company whose focus was recently defined to be security by its chairman.[1] The Xbox experience shows that building trustable clients in a hostile user environment is hard, even for a large, well-funded company. One observation is that this risk and difficulty of building cheap, trustable hardware clients places an upper bound on the importance of the secret that can be trusted to such client hardware. In addition, the Xbox provides a consistent teaching example, with almost 10 million nearly identical units out there at the time of writing. The similarity of the Xbox's architecture to a vanilla PC adds even more educational value to Xbox hacking, since much of the discussion in this book also applies directly to the much broader subject of PCs.

[1] "Trustworthy Computing" by Bill Gates, http://www.microsoft.com/mscorp/execmail/2002/07-18twc.asp

Another interesting aspect of Xbox hacking is the underground society of hardware hackers following the Xbox. The people who hacked the Xbox and the expertise they attained will be relevant long after the Xbox has become a dusty yard sale piece. Hence, there is a conscious social focus to this book. I have included profiles of a sampling of Xbox hacking personalities. The hope is to inspire people, through role models, to pick up a screwdriver and a soldering iron and to start hacking. Instilling this sort of exploratory spirit in the younger generations will be important in the long run for preserving the pool of talented engineers that drove the technology revolution to where it is today. Many of today's engineers got their start hacking and tinkering with ham radios, telephones and computers which, back in that day, shipped with a complete set of schematics and source code. This pool of engineering talent is essential for maintaining a healthy economy and for maintaining strong national security in the computer age.

The Video Game Console Market

2002 was a year marked by turmoil, not only abroad, but also in the technology marketplace; PC sales flattened, the server business shrank, and the telecommunications market, with a few exceptions, looked dismal. Despite the bear market for technology, the video game hardware, software and accessories market had a landmark year, hitting a total dollar sales of $10.3 billion — a 10% increase over 2001.[2] This is comparable to the recording industry's sales of $13 billion in the US in 2001.

Even though the market for video games is large, running a profitable console business is a daunting challenge. Video game customers are picky, trendy, and frugal. They demand high-performance, sexy console hardware at the price of a fancy family dinner or a visit to the doctor. This combination of frugality with an expectation for high performance game hardware forces console vendors to sell their hardware at a loss. As a result, a "closed-console" business strategy is used by console vendors: the console is sold as a loss leader, and profits come from future sales of video game titles. This business strategy requires a large amount of up-front investment in console hardware and in advertising. It is the console manufacturer's responsibility to create a market for their hardware so that game developers feel comfortable investing their time and money in the platform.

The Catch-22 is that nobody wants to buy a console that has few game titles. Thus, the risk of building and deploying millions of units of hardware, and the hundreds of millions of dollars of up-front losses taken on the hardware, is shouldered almost entirely by the console manufacturer. As a result, there are currently only three players in the game console business today: Sony, Nintendo, and Microsoft. Of these three, Sony has a head-and-shoulders lead in the console market, while Nintendo has cornered the handheld market with its Gameboy line of

[2] source: NPDFunworld

products. Microsoft is the new player in the game console market. The race for second place is yet undecided. In early 2003, Gamecube sales were leading Xbox sales in Japan and Europe, while the Xbox maintained a sales lead over the Gamecube in the huge North American market.

Crucial to the success of the closed-console business model is the idea of locking consumers into buying only approved, royalty-bearing game titles. In other words, piracy and unapproved game titles can destroy the profitability of the business. Hence, a console must employ security mechanisms that hamper game copying and unapproved game development and distribution. The failure of the Sega Dreamcast is a salient example of what happens when security mechanisms fail.

The Dreamcast was launched in Japan on November 1998. Production problems with the NEC PowerVR2 DC chip, the graphics accelerator used by the Dreamcast, limited initial shipments. The following three years were a rollercoaster ride for the Dreamcast. Popular games such as Soul Caliber, Dead or Alive 2, Resident Evil, Crazy Taxi and Shen Mue buoyed the Dreamcast's popularity, while Sony's Playstation2 launch ate away at the Dreamcast's sales and ultimately the confidence of software developers. Ironically, the quality of the Dreamcast graphics was equivalent or superior to quality to early Playstation2 titles, such as Dead or Alive 2, despite the extra horsepower packed by the Playstation2. (The Playstation2 is difficult to program, and it took a couple of years for developers to realize its full potential.)

The final nail in the Dreamcast's coffin was hammered in the spring and summer of 2000. A German hacker group, Team Utopia, discovered a back door inside the Dreamcast's mask-ROM BIOS that allowed the Dreamcast to boot from a standard CD-ROM. Nominally, the Dreamcast uses a proprietary format called the "GD-ROM" for game distribution. The GD-ROM format cannot be copied using standard CD or DVD burners. However, the back door in the Dreamcast's ROM BIOS enabled pirates to eventually create monolithic CD-ROM images of video games that were bootable without any need for hardware modification. Who was going to pay for a game when it could be downloaded for free on the internet? The resulting rampant piracy diminished game sales, discouraging game developers from developing for the console and damaging Sega's business. Six million units sold, and about three years after its launch, the Dreamcast was pulled from the market. Now, Sega is exclusively in the game development business, and even makes games for their former competitors Sony and Nintendo as well as Microsoft.

While there are many lessons to be learned from the Dreamcast experience, this message is clear: the ability to run code from near-free sources such as CD-Rs, DVD-Rs, or the network, without significant hardware modifications, is the kiss of death for any console business based on the closed-console model. This is a brutal problem for the Microsoft Xbox, since it is built from standard PC hardware originally designed to be open and to run code loaded from numerous sources. Hence, Microsoft's fate in the console market is intimately linked to the success and robustness of the

Xbox security system. The security system has held up fairly well so far: all of the weaknesses found require at least a solderless, warranty-voiding modification to be installed. The need for hardware modifications limits the practical impact of these weaknesses, since most users are afraid to take the cover off their appliances. However, there is an intense desire from multiple groups, legitimate and illegitimate, to get the Xbox to run code from arbitrary sources without hardware modifications.

The Xbox is a victim of its own design: the choice to use standard PC hardware vastly increases the value of an "opened" Xbox to hackers and pirates alike. The Xbox is a rather satisfying target for weekend hackers and hobbyists for the same reason Microsoft adopted the PC architecture for the Xbox: existing PC programs are easily ported to the Xbox. In addition, there is a wide and deep knowledge base about PC hardware, so the learning curve for hacking the Xbox is not as steep as for other consoles. On the other hand, the Playstation2 and the Gamecube have a steep learning curve and they also have architectural limitations that hamper the porting of most PC applications. The Xbox is also a popular target for pirates because of the ease of porting legacy game emulators, and because of its high profile and ease of obtaining compatible debugging and testing hardware.

Additionally, the similarity of the Xbox architecture to the PC architecture makes the Xbox a good educational vehicle. The knowledge gained from this book applies to more than than just embedded hardware or game consoles; you should be able to apply most of the knowledge in this book directly to PCs. Too, vast documentation resources applicable to the Xbox, inherited from the PC world, are conveniently indexed by web search engines. The ready availability of documentation will assist motivated readers to build upon the knowledge contained in this book.

The Xbox is also a more appealing educational example than the run-of-the-mill PC. There is too much variation between the hardware details of PC implementations to make useful step-by-step hacking guides for the PC, whereas step-by-step guides for the Xbox are guaranteed to be accurate across millions of units that are conveniently available for purchase in almost any mall or electronics retailer.

About Hackers and Hacking

This is a book about hacking in the traditional sense: about the process and methods of exploration. Some may be surprised that this book doesn't have chapters devoted to ripping games and patching specific security checks — after all, isn't that what hacking is all about? In reality, the term "hacker" has evolved quite dramatically over the years as the public's awareness of technology has increased and as a sensationalist mass media continues to color the public's opinion of hackers.

In the beginning, a hacker was someone who worked passionately for the sake of curiosity and exploration. There were hardware hackers who took it upon themselves to remove the covers from computers to

optimize their design (early computers were built out of discrete components, so they could be modified in meaningful ways with simple tools), and there were software hackers who labored to make the most compact and elegant code, since computational resources were scarce and slow. There were hackers who explored the ins and outs of the phone system, and those who explored the roofs and tunnels of buildings of university campuses. Quite often, early hackers engaged in all of these activities. Hackers would share their findings or results (hacks) with each other freely, as their rewards were not financial, but came from satisfying their intellectual curiosity and from the enthusiasm of their peers. As a result, hackers tended to form into meritocratic groups where membership and advancement were based entirely upon a person's ability to hack.

As technology evolved and computers became faster and more integrated, hackers found that the effort involved in hardware hacking was not worth the benefits. The interesting pieces of computers were quickly becoming buried deep within hermetically sealed ceramic packages, etched into silicon structures that were difficult to see even with a good microscope. A difficult hardware hack that might double the performance of a computer was made moot within months by Moore's Law.

On the other hand, software hacking was beginning to focus more on applications and less on algorithms or optimization. The compactness or elegance of a program was no longer directly important as memory and processor power became cheap and plentiful. Besides, compiler technology had also improved to the point where compiled code ran almost as fast as hand assembly. By the late 80's, the term "hacker" had grown to imply someone who could write volumes of C code in their sleep and create brilliant applications overnight. The old hardware hackers were either converting to software hackers, or retreating to university labs and corporations that could afford to support their expensive hobbies.[3]

The term "hacker" at that time was increasingly associated with people who cracked passwords and programs to gain access to machines and software that was otherwise off limits. Hollywood was partly responsible for this stereotype, with a slew of movies that portrayed teenagers bringing the world to the brink of nuclear annihilation with a few keystrokes, or closet geniuses creating artificially intelligent cybermonsters in their basement.[4] Unfortunately, the hyberbole of these movie

[3] The good news is that hardware hacking technology has been catching up with Moore's Law lately, leading to a hardware hacking renaissance. Affordable circuit board fabrication services have spring up, and the birth of the Internet has simplified the process of acquiring components. In addition, services such as the Mosis chip foundry service and FIB (focused ion beam) services have started to bring integrated circuit hacking into the realm of financial possibility for individual hardware enthusiasts.

[4] Rodney Brooks, the Director of the Artificial Intelligence lab at MIT, once said that the Hollywood idea of a crackpot inventor making an artificially intelligent being in their basement was about equivalent to someone building a 747 jumbo jet in their backyard.