Nintendo® Entertainment Security:
A Review of Nintendo Console Hacking

Chris Phifer

Abstract

The ubiquity and popularity of video gaming cannot be overstated; there are about 2.5 billion video gamers [1] from all over the world, playing on various PC configurations and proprietary consoles produced by giants such as Microsoft, Sony, and Nintendo. Nintendo in particular shines in the industry, having released a plethora of revolutionary consoles throughout the past 36 years. Like all computational devices, though, these consoles were immediately a target for exploitation, both for fun and for profit. In this paper, we provide a brief account of Nintendo console hacking, focusing on a selection of the non-handheld consoles released to the date of this paper's construction. We will loook at the Nintendo Entertainment System (NES), the Wii, and the Switch. For each of these devices, we'll look at some examples of exploits that have been discovered, both in terms of Internet connectivity for the later consoles and local system security. We'll conclude by summarizing the ways in which these historical exploits can be used both for future system hacking and for understanding the ways in which practitioners go about reverse engineering special-purpose hardware and the software written for it.

1 Introduction

Nintendo is, to date, the top seller of video game hardware [3], trumping both Sony (the manufacturer of the PlayStation consoles) and Microsoft (who manufactures the XBox.) There was a point at which the flagship console, the Nintendo Entertainment System (which, according to some, was solely responsible for saving the video game industry), was in 33% of homes in the United States [2]. This type of market penetration is no laughing matter: Nintendo has made and continues to make its mark on the world of video gaming with no obvious end-date on the momentum.

With great success comes great responsibility, however. This level of success entails a need for strong security, since popular things seem to always make their way into the hands of those who would seek to break them, be it for fun or otherwise.

In order to explore the security of Nintendo consoles, we present here a brief review of the ways in which three of the best-selling home consoles have been broken: The Nintendo Entertainment System (NES), the Wii, and the Switch. For each console, we'll provide a collection of examples of exploits both at the hardware and software level, identifying techniques utilized and common patterns when possible. Before that, though, we provide a little bit of history of these three consoles.

The Nintendo Entertainment System was released in 1983 in Japan, and nearly 62 million units

have been sold during its lifetime [5]. The NES featured an 8-bit CPU, with a clock rate of 1.79MHz. There were 2kB of RAM to work with, and an equal amount of video RAM [6]. It outsold both the Sega Master System and Atari 7800, both of which were released later. The console had a wide range of games available, as well as many accessories. It is also the console with the most released pirated games and clone consoles [6]. We'll see both hardware and software exploitation of this classic console.

The Wii, one of the first consoles to popularize the concept of motion controls, was released in 2006 in North America, and is the best-selling Nintendo home console at nearly 102 million units sold since its inception [5]. While less is known about the technical specifications of the Wii, it reportedly features a processor clocking at around 729MHz, a discrete GPU clocking around 243MHz [7], 88MB of main memory [8], 3MB of texture / framebuffer memory on the GPU, a host of ports for peripheral support, built-in flash memory, support for GameCube controllers and games, and the ability to connect to the Internet for online play and browsing. Despite being a much less powerful console than the others released around the same time, it brought together many revolutionary ideas about how we play games. For the Wii, we'll also see a combination of hardware and software hacking.

Finally, Nintendo's most recent home console: The Switch. The Nintendo Switch was released in 2017, and has sold about 42 million units since release [5]. The details of the CPU and GPU have not been revealed, but laboratory tests suggest a range of 1 - 2 GHz for the CPU and a range of clock rates for the GPU depending on whether the console is docked or not. There are a reported 4GB of system RAM, wireless Internet connectivity, and wireless controllers inspired by the success of the Wii. With the Switch, Nintendo once again released a console which had significantly less 'horesepower' than its generational kin. This was made up for by the novelty of blurring the lines between home and handheld console. Though it has only been around for a couple of years, the Switch has already been a target for hackers; we'll see a small number of software exploits that have been discovered in the two years since the console's release.

2 To the Community

It is tempting to call the study of video gaming platforms inane since they are typically limited-access special-purpose hardware designed for the particular use of running specially-made software, potentially stored on media only usable by a very particular device. This is fallacious, however, since the video gaming industry has become ubiquitous and offers yet another point of connection to the Internet for a good percentage of the world's population. Even with the earlier consoles (as in, those with no connectivity properties), though, hobbyists and professionals alike have targeted the hardware and the software it runs.

In addition to the importance of studying the security of ubiquitous Internet-connected devices (and their predecessors), it is important that records be maintained of work done in this domain, because historical analysis can be used to anticipate problems, hone in on reasonable solutions, and prevent future devastating security vulnerabilities. We first expand on the importance of keeping a historical record, followed by a quick look at the ways in which console hacking is and has been used.

2.1 Motivation: Historical Record

An oft-repeated qoute from Winston Churchill says "Those that fail to learn from history are doomed to repeat it." This idea applies to security as much as anything else, and many of the problems we face in the field today are because history has not been learned from; still, the most common general vulnerabilities as reported in MITRE's Common Weakness Enumeration (CWE) are fundamental security issues such as buffer overruns, cross-site scripting, and SQL injection [9]. With this in mind, it is imperative to recognize the value of documenting the history of security work, even as it pertains to devices that are not general-purpose computers or "Internet of Things" gadgets.

2.2 Motivation: How It's Used

From the perspective of Nintendo, it is important to understand the security of their devices so that consoles are appropriately safeguarded. Of course, physical access to a device is almost always equivalent to complete theoretical control of what it's doing, so there isn't too much to be done in that regard; but this isn't to say the physical exploits done on older devices aren't worth exploring, in particular because they can suggest ways of obfuscating the physical structure or, perhaps more surprisingly, because they can give insight into what capabilities users want. Anecdotally speaking, this happened with devices such as the iPhone, which sported a strong homebrew community dedicated to gaining root access in order to add features Apple had not yet implemented: After a few years of 'jailbroken' devices sporting certain features, Apple added many of those features to their officially released iOS (e.g. the control bar.) It isn't hard to imagine this same thing happening with video game consoles, where new console releases can be tailored to even better meet the desires of the masses based on how they tinker with older devices.

From the user perspective, there are a number of reasons one might want to exploit a video gaming device. Some users simply want full control of their device; they wish to bypass the restrictions built in to the operating systems / firmware running their consoles in order to be a true administrator. Others may be looking for ways to play games unavailable for devices in their region of purchase. Some tinkerers seek to do weird things simply because they can (we'll see an example of this with the NES.) A more nefarious reason to desire fuller control of a device is to steal digitally distributed games, or use Internet connectivity to exploit vulnerabilities at the level of the Web for a variety of information-stealing reasons. A study of the history of console exploitation can be used to both understand how these needs are met, and in the ethically questionable cases, what can be done to prevent them via future hardware and software choices.

3 Console Hacking Review

We now provide a collection of samples of console hacking that has been done for the consoles mentioned above. By no means is this a complete historical account; rather, it is a sampling presented as a loose chronology in order to highlight some of the techniques used and successes found in trying to exploit flaws in the design of both the hardware and software for these consoles. We'll work from Nintendo's first home console to their most recent, taking one stop in between to explore the best-selling Nintendo home console of all time.

3.1 Nintendo Entertainment System (NES)

We start by looking at the Game Genie, a third-party passthrough device that sat between an NES game cartridge and the console itself. This device was released in 1990, and met with immediately backlash from Nintendo who attempted to sue the creators for creating derivative works [11]. The device worked by accepting references to the game cartridge ROM, allowing modified values to be read by the system granting benefits such as inifinite health, infinite ammunition, and access to otherwise unreachable areas or states of the game.

This exploitation works simply by putting a physical bridge between an intended data pathway, and giving the user freedom to manually specify how to use the intercepted data. This was done via "patch codes" which were distributed via subscription service throughout the year [11]. According to executives at the company responsible for the creation of the Game Genie, the controversy and conflict with Nintendo was an issue of "personal freedom", that owners of the console and its games should have the freedom to use the game as they see fit.

A version of the Game Genie would be created for the Super Nintendo, the GameBoy, the Sega Genesis, and the Sega Game Gear. Similar products were created for the Nintendo DS, namely the Action Replay. All such devices have worked very similarly, by capturing and modifying a data pathway.

In addition to devices such as the Game Genie, the NES has been exploited to pull off clever tricks. One such clever trick was the creation of a PDF document that also happens to be a valid NES ROM (that is, a valid NES game that could be loaded onto an NES cartridge), and a ZIP archive [12]. The paper goes into great technical detail about how this feat was achieved: Briefly, the NES ROM format has special sections that can be used to embed other types of data header, such as PDF. This, combined, with clever tricks to make this program a "hash quine", are the meat and potatoes of this clever trick.

Both the Game Genie and this embedded data / quine hacking show a mastery of the NES ecosystem: They depend on the way data is read and written, were developed using standard reverse engineering techniques, and suggest that the exploitability of console hardware and software has great potential.

3.2 Wii

The Nintendo Wii has a vibrant commmunity centered around "homebrew", a term borrowed from the brewing / distillation of liquor in one's home. The goal of homebrewing a device is to gain fuller control over its capacities and install custom software that isn't otherwise allowed given the constraints of the base operating system. This umbrella term can be used for the act of injecting a payload to allow root access through the built-in system, replacing the on-board operating system with something completely different, installing applications that aren't licensed by Nintendo / whoever manufactured the device, and so on.

Nintendo has a history of being averse to homebrew, going back to the aforementioned lawsuit around the Game Genie being used to modify game states in unintended ways. This aversion is due to a combination of factors, including the possibility of bricking the device (that is, damaging the hardware / firmware to the point that the console has the functional capacity of a brick [13]) and the possibility of piracy of licensed video games.

The aversion aside, homebrewing the Wii is done via exploits discovered in certain Wii games / built-in applications, such as the message board. Through the system's SD card support, a user can force a system crash leading to arbitrary payload execution by carefully configuing an SD card with the correct data and performing a sequence of inputs on the exploitable application. This is done to install useful apps that don't exist on the Wii Shop, access system settings unavailable under normal use, and (more nefariously) install unlicensed and potentially pirated versions of official Wii game releases.

3.3 Switch

Like the Wii, the Switch is developing a homebrew community. A number of exploits allowing the injection of homebrew payloads already exist [14], and once more the custom firmware that is loaded via these payloads allows the user to install software and perform tasks not otherwise allowed by the restrictions Nintendo built in to the device.

The interesting observation at this stage is the prevalance of online play, and the shift in the video game market towards digital-only game releases. With this very online / web-focused model, Nintendo has been extremely careful with the Switch, using the connectivity of the Switch to (in some cases) detect that the device has been modified to run homebrew software and subsequently ban the device from participating in official Nintendo activities, such as multiplayer gaming and the use of the digital game store.

A similar crackdown on piracy hit the Nintendo 3DS in 2018 when Nintendo patched their game servers such that it was no longer possible to use applications such as Freeshop (a version of the Nintendo eShop that allowed one with a homebrewed device to download any game on Nintendo's servers for free) [15]. This did not stop anyone from pirating games, but it did make it significantly more difficult for those with little technical background (i.e. "script kiddies") to obtain games outside of the law. What this shows is the other usefulness of console hacking: Discovering the exploits that Nintendo could themselves use to understand where in their product designs the flaws that allow such exploits are.

4 Applications

As already mentioned as a motivating force, there are a number of reasons to care about console hacking from a practical standpoint. Users can gain more control over their hardware, allowing for fuller use of the full capabilities of the device in question. Nintendo can gain insight into what features and capabilities users want, and factor this into the design of future consoles.

As we could see with the Game Genie for the NES, a major application is to gain advantages in difficult games via cheats; these software exploits have become an industry in and of themselves by way of quality assurance testing through the use of tools such as Cheat Engine on PC. The same reverse engineering techniques used in the setting of QA testing can be used by console hackers to discover new ways to exploit software vulnerabilities in games to gain advantages.

A further application is the development of speedrunning techniques, a phenomenon which has taken off in recent years thanks to large charity events such as Awesome Games Done Quick (AGDQ) and Summer Games Done Quick (SGDQ). Through manipulations of game code by way

of tricky exploits, speedrunners have been able to perform incredible feats such as completing Super Mario World in 6 minutes by skipping straight to the credits [10]. Such pagaentry in gaming is incredibly popular, and has grown into a very supportive community that each year raises hundreds of thousands of dollars for charity organizations such as Doctors Without Borders and the Prevent Cancer Foundation.

Finally, and perhaps most importantly, console hacking (and the various motivations behind it) offers high-quality work done mostly by hobbyists and security professionals with training that does not come from a formal background. It is valuable work in terms of security insights that is being done without being tied to the idea of consulting or finding exploits for the sake of making money to find exploits; that is, work motivated by an actual end-user desire to do something beyond what's capable with the device and its software as provided by the manufacturer.

5 Summary

We have discussed motivations for studying console hacking, including the need for keeping good historical records and the various ways in which such a record can be used to both safeguard future work and perform useful tasks in the current setting. We've seen a review of three Nintendo consoles and some exploits of them, including the Game Genie for the NES, a way of embedding files in still-playable game cartridges for the NES, the WiiBrew community, and the budding Switch homebrew community. Finally, we saw a number of applications of these exploitations, including gaining fuller control over owned hardware, identifying user desires, discovering and taking advantage of game bugs to enable game-code level cheats, and developing techniques to speedrun games (valuable due to its popularity and work in charity.) All of this together provides a brief but valuable overview of what exists in the space of console hacking and why such security work is valuable in the first place.

References

[1] (Various Sources and Authors) WePC 2019 Video Game Industry Statistics, Trends & Data, https://www.wepc.com/news/video-game-statistics/

[2] Littrell, Drew. "That Time Atari Cracked the Nintendo Entertainment System", https://hackaday.com/2018/10/22/that-time-atari-cracked-the-nintendo-entertainment-system/

[3] Aclin, Justin and Garcia, Eddie. "Two Mario Games Continue Nintendo Switch and Nintendo 3DS Momentum Into 2019", https://www.businesswire.com/news/home/20190111005064/en/Mario-Games-Continue-Nintendo-Switch-Nintendo-3DS

[4] Welsh, Oli. "A complete history of Nintendo console launches", https://www.eurogamer.net/articles/2017-02-24-a-complete-history-of-nintendo-console-launches

[5] (Company collected statistics) "Dedicated Video Game Sales Units", https://www.nintendo.co.jp/ir/en/finance/hard_soft/index.html

[6] Base System. "Nintendo Entertainment System/Famicom: Console Information", https://www.consoledatabase.com/consoleinfo/nes/

[7] IGN. "IGN: Revolution's Horsepower", http://wii.ign.com/articles/699/699118p1.html

[8] TechOn! "PS3 VS Wii, Comparisons of Core LSI Chip Areas", http://techon.nikkeibp.co.jp/english/NEWS_EN/20061127/124495/

[9] MITRE. "2019 CWE Top 25 Most Dangerous Software Errors", https://cwe.mitre.org/top25/archive/2019/2019_cwe_top25.html

[10] SethBling. "Super Mario World – Credits Warp in 5:59.6 (First Time Ever on Console)", https://www.youtube.com/watch?v=14wqBA5Q1yc

[11] NESWorld. "Game Genie - The Video Game Enhancer", http://www.nesworld.com/gamegenie.php

[12] Sultanik, Evan & Teran, Evan. *PoC —— GTFO*. "This PDF is an NES ROM that prints its own MD5 hash!", https://www.alchemistowl.org/pocorgtfo/pocorgtfo14.pdf

[13] WiiBrew. "Brick", https://wiibrew.org/wiki/Brick

[14] Switch Homebrew Guide. "The Ultimate Noob Guide for Hacking your Nintendo Switch", https://switch.homebrew.guide/

[15] Andy. *TorrentFreak*. "Nintendo Plugs Leak That Provided Free 3DS Game Donwloads", https://torrentfreak.com/nintendo-plugs-leak-that-provided-free-3ds-game-downloads-180823/