Dead Man's Switch: Forensic Autopsy of the Nintendo Switch

Frederick Barr-Smith[a,b], Thomas Farrant[a], Benjamin Leonard-Lagarde[a], Danny Rigby[a,b], Sash Rigby[a], Frederick Sibley-Calder[a]

[a] *Modux Limited, Bristol, UK*
[b] *University of Oxford, Oxford, UK*

Abstract

The Nintendo Switch is a popular handheld gaming console that is used for a variety of purposes. The most common is that of gaming, however, there are supporting activities such as social networking, media consumption and internet connectivity. In this paper, we have detailed the processes that must be conducted in order to extract forensic evidence from Nintendo Switch devices. We extracted a number of different forensic artefacts from a NAND dump of several Nintendo Switch devices. We discovered several key artefacts, notably personally identifiable information, network connection history and displays that have been connected. We also assessed the forensic value of each artefact extracted from the device. We developed software to automate the process of dumping and extracting the content of the NAND memory. Additionally, we developed modules for the forensic software Autopsy and released these as open source software to automate the process of ingestion and analysis.

1. Introduction

Video game consoles are used extensively for entertainment, socialising and also web browsing. The Nintendo Switch was released after Nintendo's previous portable gaming consoles, the pioneering Game Boy and subsequent Game Boy DS series. The Nintendo Switch is a portable gaming console with 32 gigabytes of internal storage, a NVIDIA Custom Tegra processor, and various interfaces for internet connectivity and the connection of peripheral devices.

One reason that the Nintendo Switch is a device worthy of forensic analysis, is the fact that the console itself is portable. This portability and the identification through certain forensic artefacts of location, such as SSIDs, credentials, MAC addresses and connected displays means that we can identify the previous locations of the device and identify the device owner. Another advantage of portability is that there is also a greater volume of forensic data available, due to the fact that the owner of the device will have it in their possession for a longer amount of time and will be engaging in activities that generate data for a greater amount of time.

A motivating factor in any forensic investigation is to be able to establish a timeline of activities relevant to an investigation. In addition to acquiring temporal data to establish a timeline, the acquisition of geolocation data or data that ties a person to a particular address is also useful. We identified forensic artefacts that contained such data within the Nintendo Switch and assessed their potential value in a forensic investigation. The Switch has internet connectivity enabled and keeps a record of all access points it has connected to, meaning that a forensic timeline can be established for the owner of the device.

Forensic acquisition capabilities for mobile devices and traditional laptop and desktops are well known to cybercriminals. Conversely, the ability for law enforcement and other investigating bodies to be able to extract forensic evidence from videogame consoles may not be as well known to criminals.

Furthermore, several homicide cases have used rudimentary evidence of data from a Nintendo Switch console, the evidence in question was messages sent and network connections made, respectively (Murphy et al., 2019; Doolan, 2019). As such, our provision of the ability to extract forensic data from a common device that is present at many crime scenes is a valuable scientific and practical contribution to forensic science.

As with other portable games consoles and the 3DS series that preceded the Nintendo Switch, system configuration and other data is contained within non-volatile NAND memory. Unlike these other consoles, a large amount of user and system data is contained within this NAND memory, due to the increased storage capability and wide variety of uses that this console has. We deployed an exploit and several other steps to enable the extraction and ingestion of data from the NAND memory.

The specific contributions of this research paper are as follows:

1. We provided replicable instructions in addition to developing and releasing automation tools to enable the extraction of the data on the NAND memory from the device.
2. We conducted evidentiary analysis of forensic artefacts on the Nintendo Switch. This enabled us to assess the forensic value of artefacts contained within the NAND memory.
3. We developed Autopsy modules to automate forensic analysis of the Nintendo Switch and provide the returned forensic artefacts to an analyst.

The contributions of this research are clear, in that it is the first academic paper and research that covers the systematic forensic analysis of the Nintendo Switch console. This contribution is coupled with our automation of all possible steps, to allow replication and extension of this research.

2. Related Work

Hosani et al. (2020) state that "having the ability to properly investigate such devices while reducing the potential of corrupting data is crucial", referring specifically to the Nintendo Switch. Similarly, Rogers (2020) refers to the prospect of being able to gather forensic data from a Switch as "an essential source of data and potential evidence".

Existing research papers cover console forensics as a method and process. Research by Read et al. (2019) extracted forensic artefacts from a NAND dump of the Nintendo 3DS, during the presentation of this research the necessity of conducting future research on the Nintendo Switch was stated. This follows from earlier exploratory work by Read et al. (2018), that proposed a methodology for the recovery of forensic artefacts from 3DS devices. The Wii system, another Nintendo product, has been subject to academic forensic analysis by Turnbull (2008).

Work has been conducted in the field of analysis of NAND memory by Fukami et al. (2017). Rabaiotti and Hargreaves (2010) described extraction of RAM memory following the deployment of an exploit, for the purpose of obtaining forensic evidence. Our method is similar, but applied to a different console.

Despite the popularity of the console, at the time of our investigation, no other published forensic research had been made into the device and its supporting firmware, regarding the information it is possible to extract. This presents a potentially important source of information which is currently being unused due to a lack of research and tooling. A master's thesis by Lagerholm et al. (2020) directly cites the modules we developed and investigates some of the forensic artefacts that can be recovered using these modules. This thesis also identifies an artefact which we do not, the 'Cache.dat' file, which contains user browsing history on the Nintendo Shop.

In this work, we have leveraged a number of open-source tools to facilitate the exploitation, extraction of NAND dump and subsequent forensic analysis of the extracted dump. These tools are from the security research community. The modding and homebrew community for videogames consoles means that there are often a wide variety of exploits available for most videogame consoles within a short time after their release. Some companies such as Microsoft and Nintendo have pursued aggressive litigation against researchers or businesses that have reverse engineered and exploited their devices, including a notable recent lawsuit against 'Team Xecuter', who were selling exploitation tools for the Nintendo switch (Huang, 2003; BBC News, 2020). These tools provide a useful starting point for taking initial steps such as obtaining memory dumps of the onboard NAND storage. An example of the accumulated knowledge of the homebrew community assisting our research, is the SwitchBrew Wiki (2020a) that offers a lot of information regarding the flash file system structure.

Partition Name	Details
USER	FAT32 Filesystem
SYSTEM	FAT32 Filesystem
SAFE	FAT32 Filesystem
PRODINFO	Used for System Calibration
PRODINFOF	Used for System Calibration

Table 1: NAND Structure.

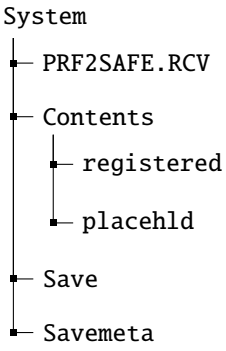

Figure 1: System partition.

3. Background

The Nintendo Switch was at the time of writing, the most popular portable video games console in the UK market, with 61.44 million units sold worldwide (Nintendo, 2020). This device is very popular and preceding our research there were no precise and reliable instructions for forensic analysis; meaning that providing the capability for automated extraction of forensic artefacts from a Nintendo Switch device may help a large number of investigations.

The demographics for the user base of the Nintendo Switch are people typically under the age of 35 and have a slight majority of male users (EEDAR, 2018). Both of these demographics indicate that they are the target demographic for cybercrime offenders and people veering on the edge of cybercrime (United Nations Office on Drugs and Crime, 2013). There can be a crossover between potential cybercrime offenders and people who play videogames, as shown by research from the National Crime Agency (2017).

3.1. NAND Structure

The Nintendo Switch stores its data mostly in one of two locations, the onboard NAND flash memory and the removable SD card. The majority of data that is useful for forensic analysis is located on the NAND chip, as can

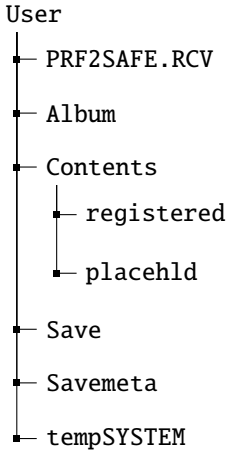

Figure 2: User Partition.

be seen in SwitchBrew Wiki (2020a). We list the five key partitions in the dump in Table 1. The partitioning scheme used for the SD card is MBR, the partioning scheme used for the NAND is GPT.

Of these partitions it was only the System and User partitions that we assessed to contain information of forensic relevance. Other partitions contain things such as the boot partition, keyblob and other low level system elements, therefore being comparatively unlikely to contain forensically relevant information SwitchBrew Wiki (2020a). The directory structure of the System partition is shown in Figure 1 and the directory structure of the User partition is shown in Figure 2.

Information locations within files in the *System* partition, appeared to be the same across devices. However within the *User* partition, filenames and data offsets varied, which required us to use pattern matching to extract data.

Within the save directory of each partition we identified several savedata files. Each of these files was structured in a proprietary format. As a result of this, it was necessary to carve the artefacts of interest out of the file.

3.2. Security of Device

Encryption is used in various places on the Switch device. Particularly relevant to our research is the use of encryption of the NAND, where some of the partitions (including the SYSTEM and USER partitions we analyse) are encrypted using AES. To decrypt these partitions, we extracted keys using the tool 'Biskeydump' (Stojadinovic, 2020), which requires exploitation of the Switch before it is able to extract these keys.

Steps have been taken by Nintendo to prevent exploitation of the device. Unlike the 3DS, all userland processes on the device use ASLR, making certain avenues of exploitation more difficult. There are other protections against exploitation such as the use of a microkernel, sandboxing of userland applications and the provision of ARM's TrustZone (Tanguy et al., 2019).

4. Method

The methods we used to obtain forensic evidence from the Nintendo Switch device took a multi-staged approach. Several methods can be used to extract the NAND dump. The methods we chose were taken from the homebrew Nintendo Switch community, that attempts to circumvent copy protection methods and hardware security mechanisms of the Switch to run arbitrary code and other games.

We deployed a known exploit (Fusée Gelée) against the NVIDIA Tegra processor within the device, to gain remote code execution. The aim of this was to enable the Switch system to be able to run unverified code. Once unverified code was running on the device, we were then able to use open-source tooling to extract the contents of the NAND memory. Following the extraction of NAND memory, we then conducted analysis on the extracted contents. At first our analysis was conducted manually, but we developed automated tools to facilitate analysis of the extracted NAND memory.

4.1. Creation of Test Data

We performed forensic investigations on three used Nintendo Switch devices to discover the forensic artefacts which are available on the device and provide test data for tool development and evaluation. One bought on launch by one of the researchers and two used devices purchased second-hand, both with firmware version 7.0.1.

We then put all the devices through additional usage, similar to how they would be used in day-to-day life. We put the devices through a multitude of use-cases, including powering off and on, connecting to wireless networks, social media access and gameplay across several games. We conducted additional usage until we evaluated that we had created sufficient test data relating to geolocation, wireless network connection logs and other artefacts we would later be looking for evidence of. We recorded all of these actions, such as applications opened, networks connected to and webpages visited, in a log file. We noted these actions to be able to correlate the validity of any future artefacts which were found.

We took care to store all forensic data on an encrypted hard drive and disposed of it after concluding our experiments. The hex dumps and screenshots contained within the paper are from our manually created test data. We do not share parts of this dataset, to avoid contravening ethics and data protection law.

We took copies of the flash memory using the process we enumerate in our methods section. We searched for keywords and associated strings within the filesystem. Some of these keywords are visible in the source code for our Autopsy modules. Following identification of a string match, we recorded the filename and memory offset or surrounding bytes, to aid in future identification. We verified these steps on a second console to ensure files and offsets remained consistent, if they were not discovered on a second console we would discard them from our analysis. Our

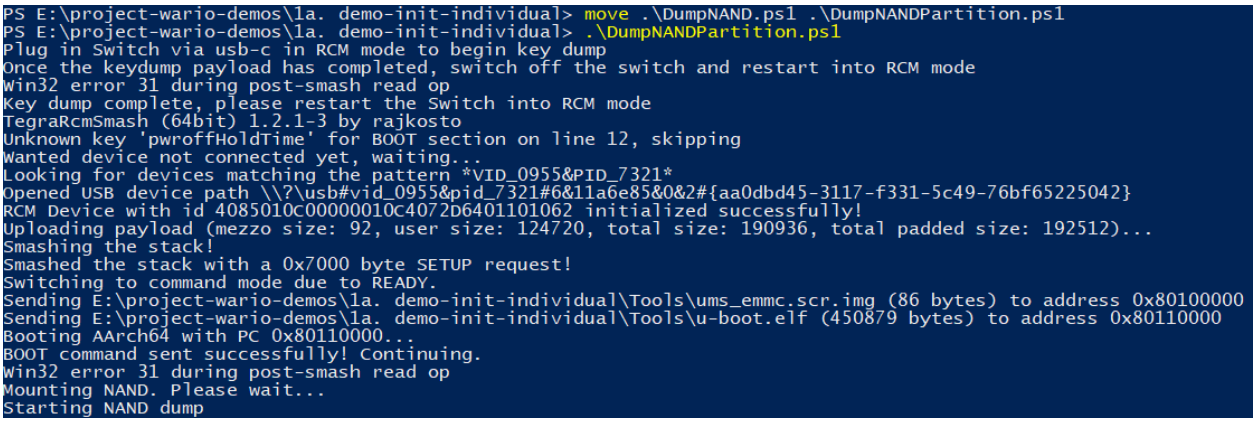

Figure 3: Execution of DumpNANDPartition script.

validation of our artefact identification on more than one device ensures that our results are reproducible.

4.2. Exploitation

To extract the flash memory dump, we used an NVIDIA Tegra exploit (Nvidia Support, 2018). By using this exploit chain, we were able to bypass secure boot and have the ability to run unverified code. Some NVIDIA Tegra processors support a Recovery Mode which allows a user with physical access to the affected processor's USB connection to bypass the secure boot and run unverified code. The particular exploit we made use of was the Fusée Gelée exploit chain, developed by Temkin (2018). Whilst deployment of this exploit chain enables execution of unverified code, it does not corrupt or alter the data on the Switch device in the partitions we used for analysis. We can see this in the aforementioned vulnerability disclosure report, in addition to the fact that during this exploitation, the data on the partitions we examine remains encrypted.

To prepare the system to be able to load external code, we needed to boot the system into recovery mode by using a physical 'Jig' (Nintendo Homebrew Discord Server, 2020). After we had placed the device into recovery mode, we connected the device to a computer and verified its recovery mode status by using the tool TegraRcmGUI (Eliboa, 2020).

4.3. Flash Memory Dumping

We made use of the Tegra RCM exploit to gain code execution. After we had gained code execution, we were then able to execute code that extracted the contents of the NAND memory. As part of this research we developed automation scripts (Modux Labs, 2020), one of which is the *DumpNANDParitition.ps1* script, the execution of which is visible in Figure 3. This script automated the extraction of the device's encryption keys using Biskeydump (Stojadinovic, 2020) and subsequently automated the NAND dump extraction. Once arbitrary code execution was achieved, we then extracted the NAND dump.

The process that we followed, precisely, is:

1. We connected the Switch to the PC via USB-C.
2. We entered the Switch into Recovery Mode (RCM). We accomplished this via powering off the Switch then using a specially constructed jig to short pins, then holding down volume up button and tap the power on button. The Switch would then boot without the Nintendo logo.
3. We confirmed the device had successfully entered into RCM mode by opening TegraRcmGUI. The GUI displayed a green screened Switch in the lower left-hand corner.
4. We then executed the DumpNANDPartition.ps1 script from the Windows machine with elevated permissions. This automation made use of the open source software 'Biskeydump' (Stojadinovic, 2020) to extract encryption keys to the local machine, for the decryption of data.
5. We turned the Switch off and turned it on again, causing the script to continue to the next stage and check whether the Switch is in RCM mode such as in step 2.
6. The script then injected the memloader payload to the Switch.
7. We then opened the tool HacDiskMount, requiring us to open the mounted memory as a physical drive (UMS disk 0) and double click the partition in order to dump it.
8. Using the keys dumped to '.\prod.keys' on the local machine by 'Biskeydump' in step 4. We entered these keys — BIS KEY 2 (Crypt) and BIS KEY 2 (Tweak) into the Crypto (Upper) and Tweak (Lower) fields respectively.
9. Within the directory selected as the 'Dump to File' destination, there was our extracted binary. The device could now be switched off and disconnected.

4.4. Manual and Automated Analysis

Following on from our extraction of the memory dump, we then searched for and found artefacts on the system that were of forensic interest. At a high level, the data

types we were looking for were location data, usage times, wireless network data and social connections. We chose these artefact types as our criteria were to find artefacts that may be of interest to an investigator or forensic analyst.

Our process for discovery of forensic artefacts was initially a manual keyword search within the NAND dump ingested into Autopsy. We set out to identify several items of forensic relevance and discover their traces in the dump. Once we discovered key words and offsets that identified given artefacts, we then automated the process of artefact extraction, via development of Autopsy ingest modules.

4.5. Autopsy Modules

We initially identified data of interest via manual keyword searches within the NAND dumps. When we discovered relevant data in a reliable way and were able to repeat its discovery on a second device, we then automated their extraction via the development of Autopsy modules. Autopsy is open source forensics software, developed by the author of File System Forensic Analysis, Brian Carrier (2005), that allows the development of additional modules and plugins to augment the capabilities of the software.

These modular plugins augment the typical use case of Autopsy which is the forensic analysis of hard drives. These modules come in a number of different subtypes (Basis Technology, 2020). We chose to develop ingest modules to automate the extraction of forensic artefacts. These modules are run when a new data source is ingested and can be re-run manually against already ingested data sources.

We created these modules using version 2.7 of the Python programming language. These modules work on extracted NAND dumps from any Nintendo Switch for which it is possible to extract the NAND dump.

4.6. Autopsy Modules Structure

Autopsy modules are built in a standard format which keeps to a similar structure:

- **Ingest Module Factory:** Creates an instance of the File Ingest Module.
- **Ingest Module:** Performs the business logic of the ingest module. It is made up of a number of important standard functions:
  - **Ingest Module: Startup():** Initialisation of the ingest module class.
  - **Ingest Module: Process():** Where all the work of the data source ingest module is performed.
  - **Ingest Module: Shutdown():** Performs all the termination actions to destroy the ingest module instance.

These functions form the basis of the Autopsy ingest modules that we created during this research. To import the Autopsy Ingest modules, the Python files need to be placed in the *%APPDATA%_modules* directory.

4.7. Example Module Development: Connected Display Logs

To exemplify how we developed these Autopsy modules, we show an example of one of the simplest artefact extractions, the 'Connected Display Logs' and identify how we automated this process.

To discover the initial location of artefact we used the known string 'Benq' and conducted a string search for this string. This then revealed the model was 'BenQ GL2460s'. This led to our discovery of this string in 2 separate locations, namely:

1. Following the string 'EdidBlock' and preceding the string 'EdidExtensionBlock'.
2. In other seemingly random places.

Both of these locations were contained within the partition */SYSTEM.bin/save/80000000000000d1*.

```
names = names + re.findall("EdidBlock
.*?\\\\xfc\\\\x00(.*?)\\\.*?EdidExten
sionBlock", repr(currentBuffer))
```

Listing 1: Python regular expressions to find display data.

In order to repeatedly find this data, we chose to investigate the first of these locations. We used regular expressions to find any occurrences of the strings 'EdidBlock' and 'EdidExtensionBlock', as shown in Listing 1.

After the Autopsy plugin finds all occurrences of these strings, they are returned to the Autopsy interface wherein they are displayed to the analyst.

5. Forensic Artefacts

After extracting the NAND dump, we forensically analysed the data. The key objectives of this analysis were to recover as much data as was possible from various elements of the filesystem. We recovered these forensic artefacts from the extracted NAND dump. We list the artefacts that we recovered in Table 2. In this table we also list the specific data types discovered and which memory partition they were extracted from. This list is not exhaustive and there are other artefacts discovered on the dump that we did not prioritise investigation of, such as paired 'Joy-Cons'. We developed a number of Autopsy modules that facilitated the automated extraction of data from the NAND dumps and subsequent forensic analysis of this data (Modux Labs, 2020). We list these modules in Table 3 and describe them in more detail in the following section.

Effect of Factory Reset. We observed that data persisted across factory resets for the Nintendo Switch device, meaning that attempts by a malfeasor to erase any data pertaining to previous owners will be ineffective. This can also prevent the success of criminals utilising Nintendo Switch devices and attempting to obfuscate their behaviour by factory resetting the device. In cases of a Nintendo Switch device being stolen, the original owner's details would still persist on the device.

Artefact	Location	Available Data
Connected Displays	/SYSTEM.bin/save/80000000000000d1	Connected Displays
Error Logs	/SYSTEM.bin/save/80000000000000d1	Last 50 Error Codes
Game Saves	/USER.bin/save/	Save Time & Game Title
Gameplay Logs	/SYSTEM.bin/save/80000000000000a2	Access Time & Game Title
Multiplayer User History	/SYSTEM.bin/save/8000000000000001	300 users last played with
Power On/Off Logs	/SYSTEM.bin/save/80000000000000a1	Last Boot Time & Power State Changes
Screenshots & Videos	/SYSTEM.bin/	.JPG, .MP4 & .PNG with Timestamps
User Accounts	/SYSTEM.bin/save/8000000000000010	Birth Date, Email, Gender & Location
Wifi Networks	/SYSTEM.bin/save/8000000000000050	MAC Addresses, NAT, Passwords & SSIDs

Table 2: Forensic artefacts of interest.

Autopsy Module	Returned Artefacts
ingest_connected _displays	All recently connected displays
ingest_game_history	Recent game history
ingest_crash_dumps	Last 50 crash dumps
ingest_device_accounts	All user accounts saved to device
ingest_gamesaves	All saved games
ingest_last_boot	Last boot time of device
ingest_mp_user_history	Last 300 users played with and game played
ingest_power_states	Device power history
ingest_screenshots	All saved recordings and screenshots
ingest_wifi	Details of all WiFi networks recently connected to

Table 3: Autopsy modules list.

5.1. User Accounts

We found several items of personally identifiable information. This included what is defined as sensitive personal data under the GDPR (Information Commissioners Office, 2020b). We discovered this data contained within one offset location, containing several items relevant to user accounts, including sensitive information such as gender, email address and date of birth. We show these details, with comparison to the raw hex in Hex Dump 1, with the relevant bytes and information highlighted in blue. We have amended 16 bytes at the offset of *0007805A* to null bytes, to maintain the privacy of the subject.

We developed an Autopsy ingest module that facilitates extraction of all the Nintendo Switch's user accounts that have ever existed on the device. This module extracted data from the */SYSTEM.bin/save/8000000000000010* file.

5.2. Wifi Networks

A history of all wireless networks that the Switch device has interacted with is available within the NAND dump. The bytes and information showing the SSID and PSK are displayed as blue text, visible in Hex Dump 2, in the following format:

⟨16 bytes of unknown data ⟩[arbitrary null bytes] ⟨SSID ⟩[arbitrary null bytes] ⟨PSK ⟩

We also discovered additional details within the NAND dump, that allowed for other details pertaining to the network to be discovered. These details are the NAT gateway, MAC addresses and SSIDs of relevant routers.

The forensic value of network access point history in addition to other networking information is significant. It allows location to be determined and also provides evidence of prior knowledge of a Private Shared Key (PSK), if the network is password protected.

We automated the capability to extract details of wireless networks that have been recently connected to, in addition to other network metadata. We also automated the extraction of passwords for wireless networks. To accomplish this, we created an Autopsy ingest module which discovers the network SSIDs, network PSKs and other networking details from the target Nintendo Switch device. This module accesses file */SYSTEM.bin/save/8000000000000050*.

5.3. Error Logs

When errors occurred during the operation of the Nintendo Switch, they were saved within the device error history log for future reference. We were also able to access the 10 latest error logs via the support menu option. Within the NAND flash memory, we could view the latest 50 error logs. We observed that Error Logs were stored and serialised in MessagePack format (Sadayuki, 2012). We used Hactoolnet (Barney, 2020) to extract this information from the MessagePack format.

We developed an Autopsy ingest module to extract a log of the last 50 errors registered by the device. The errors were extracted from the */SYSTEM.bin/save/80000000000000d1* file. The external dependency 'hactoolnet' (Barney, 2020) was used to process much of the data.

5.4. Screenshots and Video Captures

We discovered screenshots and video captures taken by users, within the NAND dump, that were in .jpg, .mp4 and .png formats. The list of game ID hashes to which this 'Game-ID-hash' corresponded could identify the game to

Hex Dump 1 - User Account Details:

```
00077FE2 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |..............................|
00078000 7B 22 67 65 6E 64 65 72 22 3A 32 6D 61 6C 65 22 2C 22 69 64 22 3A 22 64 66 61 66 39 |."gender":"male","id":"dfaf9|
0007801E 31 35 64 33 37 34 61 66 63 22 2C 22 72 65 67 69 6F 6E 22 3A 22 22 2C 42 6C 69 6E 69 |15d374afc","region":"",Blini|
0007803C 64 42 3A 44 42 2C 22 74 69 6D 65 7A 6F 6E 65 22 3A 22 45 75 72 6F 70 65 4F 4C 6F 6E |dB:DB,"timezone":"Europe-Lon|
0007805A 6E 22 2C 22 65 6D 61 69 6C 22 3A 22 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |n","email":"----------------|
00078078 68 40 6D 6F 64 75 78 2E 63 6F 2E 75 6B 22 2C 22 61 6E 61 6C 79 74 69 63 73 46 6F 72 |-@modux.co.uk","analyticsFor|
00078096 74 65 74 6E 61 6C 41 6E 61 6C 79 73 69 73 50 65 72 6D 69 74 74 65 64 22 3A 74 72 75 |tetnalAnalysisPermitted":tru|
000780B4 A2 61 6B 61 6C 79 74 69 63 73 46 6F 72 54 61 72 67 65 74 4D 61 72 6B 65 74 69 6E 67 |.akalyticsForTargetMarketing|
000780D2 72 6D 69 74 74 65 64 22 3A 74 72 75 65 2C 42 6E 69 63 6B 6E 61 6D 65 32 3A 22 77 61 |rmitted":true,Bnickname2:"wa|
000780F0 69 67 69 22 2C 22 73 63 72 65 65 6E 4E 61 6D 65 22 3A 22 74 6F E2 80 A2 E2 80 A2 E2 |igi","screenName":"to.......|
0007810E A0 6D B2 80 A2 E2 80 A2 E2 80 A2 E2 80 A2 22 2C 42 69 73 43 68 69 6C 64 22 3A 66 61 |.m.....",BisChild":fa|
0007812C 65 2C 22 6C 61 6E 67 75 61 67 65 22 3A 42 65 6E 2D 47 42 22 2C 42 62 69 72 74 68 64 |e,"language":"en-GB","birthd|
0007814A A2 3A 44 34 30 30 2D 30 31 2D 30 31 22 2C 22 63 6F 75 6E 74 72 79 22 3A 22 47 42 |.:"2000-01-01","country":"GB|
00078168 A4 69 73 4B 6B 4C 69 6E 6B 65 64 22 3A 66 61 6C 73 65 2C 22 69 73 54 77 69 74 74 65 |.isKkLinked":false,"isTwitte|
00078186 69 6B 6B 65 64 22 3A 66 61 6C 73 65 2C 22 69 73 46 61 63 65 62 6F 6F 6B 4C 69 6E 6B |ikked":false,"isFacebookLink|
000781A4 22 3A 66 61 6C 73 65 2C 22 69 73 47 6E 6F 67 6C 65 4C 69 6E 6B 65 64 22 3A 66 73 65 |i:false,"isGoogleLinked":fal|
000781C2 7D 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |}...........................|
000781E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |..............................|
```

Hex Dump 2 - Wireless Details (SSID and PSK):

```
0005C850 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |..............................|
0005C86E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |..............................|
0005C88C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0D 49 8E 5C DA E6 41 2C 8E B6 |.....................I.\..A,....|
0005C8AA ED 93 A7 8E F3 6A 03 00 00 00 00 00 00 00 00 00 00 42 65 6E 27 73 20 69 50 68 6F 6E 65 |.....j..........Ben's iPhone|
0005C8C8 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0C 00 00 03 00 00 00 02 00 |..............................|
0005C8E6 00 00 61 73 64 66 67 6A 6B 6C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |..asdfgjkl....................|
0005C904 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |..............................|
0005C922 00 00 00 00 00 00 09 00 00 00 04 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |..............................|
```

which a screenshot belonged. These game IDs are listed in the Switch-Screenshots repository (Greca, 2020).

We observed the format of these screenshots to be:

> *⟨Timestamp in %Y%m%d %H%M%S%f⟩- ⟨Game-ID-hash ⟩.jpg*

We developed an Autopsy ingest module to process each of the screenshots which were taken on the device. This module parsed the title of the game being played at the time of the screenshot, and the timestamp of the screenshot itself.

5.5. Gameplay Logs

Applications and games for the Nintendo Switch were observed to have unique identifiers, such as those referenced by save game files (SwitchBrew Wiki, 2020a). It was possible to view the save times and details of applications and games that were opened on the device.

We developed an Autopsy ingest module that extracted the timestamps of when individual game titles were played. We discovered this data within the System partition at */SYSTEM.bin/save/80000000000000a2*.

An example application can be seen in Hex Dump 3. The unique identifier for the YouTube application *01003 A400C3DA000* is highlighted for clarity, this unique identifier can be verified on SwitchBrew Wiki.

In a similar manner to power-on and power-off logs, logs were not written to with every gameplay event. As such, data may have been missing.

5.6. Saved Games

Within the User partition's save directory there were files which corresponded to seperate game titles that have been played on the device. From the last write time of these files it was possible to determine the time at which each game had last been saved.

To extract the game title a given save data file corresponded to, we extracted the game ID. We accomplished this via looking at an offset of 1752 bytes, to see an 8 byte buffer offset, which was in little endian format. These 8 bytes are highlighted in blue in Hex Dump 4. We then correlated these 8 bytes to the game ID to identify the game in question.

We developed an Autopsy ingest module which extracts the state of the save data for each of the game titles which have been saved on the device. Each gamesave is stored in an individual file in */USER.bin/save/*. The module processes and extracts each game that has been played and the last time it was saved. It should be noted that some games do not provide timestamps of the last time they were saved.

5.7. Multiplayer User History

A list of the previous 300 players that have been played with can be found within the NAND dump. Similar to the Saved Games, we identified the game ID from its representation. The location of this data was within the System partition at */SYSTEM.bin/save/0000000000000001* and we show it in Hex Dump 5. These bytes corresponding to the game ID are highlighted as red text, shown in little endian format. By performing a lookup of the game ID, we

Hex Dump 3 - Youtube Application Opened:

```
00058E90 95 71 FF 10 14 E3 D8 8E C1 67 2D 42 0E AF 45 9E 3D C3 DC 86 F5 27 53 2E |.q.......g-B..E.=....'S.|
00058EA8 2F 60 73 52 60 7C FB 4E 2F F3 AF AE 10 06 79 DE 6D E9 7F C6 C9 F0 F6 C9 |/.sR.|.N.......y.m......|
00058EC0 39 62 EF F0 7E 6F 04 F3 EE 40 B1 69 8E 36 B6 7F 85 D2 35 A9 74 69 63 6B |9b...o....@.i.6...5.tick|
00058ED8 65 74 5F 69 64 D7 00 01 00 7D 66 0A FD D5 8F A6 61 70 70 5F 69 64 D7 00 |et_id.....f.....app_id..|
00058EF0 01 00 3A 40 0C 3D A0 00 A4 74 79 70 65 A7 64 69 67 69 74 61 6C A5 65 76 |..:@.=...type.digital.ev|
00058F08 65 6E 74 A6 6C 61 75 6E 63 68 A8 73 65 71 75 65 6E 63 65 29 A7 73 74 6F |ent.launch.sequence).sto|
00058F20 72 61 67 65 A7 73 64 5F 63 61 72 64 82 A8 73 79 73 5F 69 6E 66 6F 89 AE |rage.sd_card..sys_info..|
00058F38 61 70 70 6C 69 63 61 74 69 6F 6E 5F 69 64 D7 00 01 00 00 00 00 00 10 25 |application_id.........%|
00058F50 A8 65 76 65 6E 74 5F 69 64 BD 61 70 70 6C 69 63 61 74 69 6F 6E 5F 65 78 |.event_id.application_ex|
00058F68 65 63 75 74 69 6F 6E 5F 68 69 73 74 6F 72 79 AE 6F 70 65 72 61 74 69 6F |ecution_history.operatio|
00058F80 6E 5F 6D 6F 64 65 00 AE 6C 63 5F 72 65 63 6F 72 64 65 64 5F 61 74 B3 32 |n_mode..lc_recorded_at.2|
00058F98 30 31 39 2D 30 33 2D 32 35 20 31 35 3A 32 32 3A 33 36 AE 6E 63 5F 72 65 |019-03-25 15:22:36.nc_re|
00058FB0 63 6F 72 64 65 64 5F 61 74 B3 32 30 31 39 2D 30 33 2D 32 35 20 31 35 3A |corded_at.2019-03-25 15:|
00058FC8 32 32 3A 33 36 A6 6E 73 61 5F 69 64 C0 AA 6F 73 5F 76 65 72 73 69 6F 6E |22:36.nsa_id..os_version|
00058FE0 A5 37 2E 30 2E 31 B1 72 65 70 6F 72 74 5F 69 64 65 6E 74 69 66 69 65 72 |.7.0.1.report_identifier|
00058FF8 D9 24 37 35 38 39 38 66 33 66 2D 31 36 33 65 2D 34 34 33 30 2D 39 30 32 |.$75898f3f-163e-4430-902|
00059010 35 2D 39 36 62 32 62 35 36 39 66 34 33 34 D9 26 72 65 62 6F 6F 74 6C 65 |5-96b2b569f434.&rebootle|
00059028 73 73 5F 73 79 73 74 65 6D 5F 75 70 64 61 74 65 5F 76 65 72 73 69 6F 6E |ss_system_update_version|
```

Hex Dump 4 - Game Saves Title Identification:

```
00000690 52 4D 41 50 00 00 01 00 08 00 00 00 08 00 00 00 |RMAP............|
000006A0 04 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |................|
000006B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |................|
000006C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |................|
000006D0 00 00 00 00 00 00 00 00 00 E0 01 04 90 9B 00 01 |................|
000006E0 B9 4D F9 46 5B 43 00 10 B2 BA FB 46 BA 24 13 16 |.M.F[C.....F.$..|
000006F0 00 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00 |................|
00000700 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |................|
00000710 00 00 00 00 00 00 00 00 00 E0 01 04 90 9B 00 01 |................|
00000720 F2 0C 29 5C 00 00 00 00 00 00 00 00 00 00 00 00 |..)\............|
```

Hex Dump 5 - Multiplayer User History:

```
0005C6B0 6A 61 00 00 00 00 00 00 31 7E 93 59 00 00 00 00 |ja......1~.Y....|
0005C6C0 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |................|
0005C6D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |................|
0005C6E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |................|
0005C6F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |................|
0005C700 D9 BB 4B 72 A2 28 6F 7C BC BB 63 C0 27 15 47 24 |..Kr.(o|..c.'.G$|
0005C710 9D D6 2D 4D 5F 1C ED C6 00 20 0A 00 F0 F8 00 01 |..-M_.... ......|
0005C720 00 C0 09 00 70 3C 00 01 44 72 61 6E 6B 73 00 00 |....p<..Dranks..|
0005C730 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |................|
0005C740 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |................|
0005C750 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |................|
0005C760 18 70 DB AC 45 00 00 00 6A 61 00 AC 45 00 00 00 |.p..E...ja..E...|
0005C770 54 6F 6D 00 45 00 00 00 80 D1 87 AD 45 00 00 00 |Tom.E.......E...|
0005C780 80 6C B3 4D 42 00 00 00 80 32 18 94 35 00 00 00 |.l.MB....2..5...|
0005C790 84 6C B3 4D 42 00 00 00 50 00 00 00 00 00 00 00 |.l.MB...P.......|
```

determined that this game ID corresponds to Splatoon 2. The player that was played with, 'Dranks', is highlighted as blue text.

The capability of extracting forensic artefacts showing the last players that an individual interacted with, allows the establishment of patterns of association. Identifying these patterns is useful in any forensic investigation.

We developed an Autopsy ingest module to extract the latest 300 Nintendo Switch users which have recently been played with. This module used the savedata file at */SYSTEM.bin/save/8000000000000001* to extract the users that have been played with and the game that was played.

5.8. Power On/ Power Off Logs

This data was contained within the System partition within the binary file */SYSTEM.bin/save/8000000000000 0a1*, in MessagePack format. We display this data in Hex Dump 6. Not all power on and off events were seen to be stored within the system logs. We developed an Autopsy ingest module that extracted the timestamps of when the device was powered on and off. The power state changes were denoted in the following format (denoted as blue text in the hex dump):

nc_started_at⟨timestamp ⟩powerstate_state_start ⟨state ⟩powerstate_state_start ⟨state ⟩

We calculated the last boot time by ascertaining the last write time of a single file in the System partition. We developed an Autopsy ingest module to process the last time the Nintendo Switch was booted. This module made use of the last write time of the */SYSTEM.bin/save/8000 000000000060* file.

5.9. Connected Display Logs

Logs of display devices that have been connected to the Nintendo Switch were stored within the file at */SYST EM.bin/save/80000000000000d1*. In this file, the name of the connected display, followed the string *EdidBlock* and preceded the string *EdidExtensionBlock*. We display the location of the display name in Hex Dump 7, with the relevant bytes highlighted. We developed an Autopsy ingest module that extracts a list of connected displays that have been connected to the Switch at any point from this file.

In a similar manner to the forensic value of wireless network connection details, logs of previously connected devices can prove a person had previous knowledge and familiarity with a given location.

6. Discussion

The ultimate goal of digital forensic analysis is to establish a high level timeline by automated extraction of artefacts that may be forensically relevant (Hargreaves and Patterson, 2012). These timelines can then be used to construct pattern-of-life analyses or to analyse the circumstances surrounding a particular criminal event. Our development of these tools allows a number of different

Hex Dump 6 - Hex Dump of Power State Change:

```
00058098 64 C0 AA 6F 73 5F 76 65 72 73 69 6F 6E A5 37 2E 30 2E 31 B1 72 65 70 6F |d..os_version.7.0.1.repo|
000580B0 72 74 5F 69 64 65 6E 74 69 66 69 65 72 D9 24 37 62 38 39 39 62 31 31 2D |rt_identifier.$7b899b11-|
000580C8 38 37 63 37 2D 34 32 66 66 2D 62 34 38 33 2D 36 32 37 34 65 32 63 34 65 |87c7-42ff-b483-6274e2c4e|
000580E0 35 33 64 D9 20 72 65 62 6F 6F 74 6C 65 73 73 5F 73 79 73 74 65 6D 5F 75 |53d. rebootless_system_u|
000580F8 70 64 61 74 65 5F 76 65 72 73 69 6F 6E A1 35 A4 64 61 74 61 DE 00 05 A8 |pdate_version.5.data....|
00058110 64 75 72 61 74 69 6F 6E 0A AD 6C 63 5F 73 74 61 72 74 65 64 5F 61 74 B3 |duration..lc_started_at.|
00058128 32 30 31 39 2D 30 33 2D 32 35 20 31 37 3A 31 38 3A 34 38 AD 6E 63 5F 73 |2019-03-25 17:18:48.nc_s|
00058140 74 61 72 74 65 64 5F 61 74 B3 32 30 31 39 2D 30 33 2D 32 35 20 31 37 3A |tarted_at.2019-03-25 17:|
00058158 31 38 3A 34 38 B1 70 6F 77 65 72 5F 73 74 61 74 65 5F 73 74 61 72 74 A5 |18:48.power_state_start.|
00058170 41 77 61 6B 65 AF 70 6F 77 65 72 5F 73 74 61 74 65 5F 65 6E 64 A5 53 6C |Awake.power_state_end.Sl|
00058188 65 65 70 82 A8 73 79 73 5F 69 6E 66 6F 89 AE 61 70 70 6C 69 63 61 74 69 |eep..sys_info..applicati|
000581A0 6F 6E 5F 69 64 D7 00 01 00 00 00 00 00 10 24 A8 65 76 65 6E 74 5F 69 64 |on_id.........$.event_id|
000581B8 B3 73 79 73 74 65 6D 5F 73 74 61 74 65 5F 63 68 61 6E 67 65 AE 6F 70 65 |.system_state_change.ope|
000581D0 72 61 74 69 6F 6E 5F 6D 6F 64 65 00 AE 6C 63 5F 72 65 63 6F 72 64 65 64 |ration_mode..lc_recorded|
```

Hex Dump 7 - Connected Display Model:

```
00059200 70 70 65 64 D2 00 00 45 17 A9 45 64 69 64 42 6C |pped...E..EdidBl|
00059210 6F 63 6B C4 80 00 FF FF FF FF FF FF 00 09 D1 CE |ock.............|
00059220 78 45 54 00 00 03 19 01 03 80 35 1E 78 2E 6B 35 |xET.......5.x.k5|
00059230 A4 55 55 9F 27 0C 50 54 A5 6B 80 D1 C0 81 C0 81 |.UU.'.PT.k......|
00059240 00 81 80 A9 C0 B3 00 01 01 01 01 02 3A 80 18 71 |............:..q|
00059250 38 2D 40 58 2C 45 00 13 2B 21 00 00 1E 00 00 00 |8-@X,E..+!......|
00059260 FF 00 54 31 46 30 31 39 35 34 53 4C 30 0A 20 00 |..T1F01954SL0. .|
00059270 00 00 FD 00 32 4C 1E 53 11 00 0A 20 20 20 20 20 |....2L.S...     |
00059280 20 00 00 00 FC 00 42 65 6E 51 20 47 4C 32 34 36 | .....BenQ GL246|
00059290 30 0A 20 01 83 B2 45 64 69 64 45 78 74 65 6E 73 |0. ...EdidExtens|
000592A0 69 6F 6E 42 6C 6F 63 6B C4 80 02 03 22 F1 4F 90 |ionBlock....".O.|
```

forensic artefacts to be retrieved from the NAND memory of the Switch via automatic processing.

The awareness of the Nintendo Switch as a source of forensic evidence may not be considered within criminal threat models. The capability for law enforcement to collect evidence from laptops and mobile phones is well known. However, the capability for law enforcement to collect digital evidence from games consoles is not particularly well known. As such, criminals and terrorists use these services to communicate, which has allegedly attracted the attention of intelligence agencies, as Stevens (2015) has stated. As these devices are not included as part of the operational threat model there may be richer data that can be extracted.

6.1. Ethics

The use of exploitation to extract digital forensic evidence has some legal and ethical ramifications. Previous attempts to conduct exploitation on the Xbox, were met with lawsuits from Microsoft (Huang, 2003). The fact that exploitation of systems is commonly associated with criminal computer intrusion may present issues when provided in a court, as addressed by police guidance with regards to best practices for live acquisition of memory (Association of Chief Police Officers, 2012). This is due to verifiable chain of custody, that presents issues of data integrity. Write blockers and similar technology can be used for traditional file system forensics, to prove integrity. Whereas acquisition of forensic evidence acquired via exploitation may be challenged in court, as it is not possible to definitively prove that no evidence tampering has occurred. It must also be noted that in this case, the exploit chain does not alter the extracted data and the extracted data can be considered forensically sound.

With the extraction of personal data from these devices, it is important for us to adhere to the European General Data Protection Regulation. Our extraction of personal data from these devices should be included within the research exemptions concerning personal data, as described by the Information Commissioners Office (2020a). We also ensured that any extracted data was stored securely and deleted after our research and do not make any personally identifiable information public as a result of this research.

6.2. Limitations

The Tegra exploit that is required to gain code execution on the device was patched on a hardware level for all devices produced after June 2018. This still leaves all preceding devices vulnerable to exploitation. The fact that software required a hardware patch means that the vulnerability enabling exploitation and subsequently data extraction will persist indefinitely for these specific models. It is likely that subsequent models will be found to have vulnerabilities. All the data recovered could be alternatively obtained through a 'chip-off', with forensic integrity. But an exploit would still need to be deployed against the device to obtain keys for decryption.

6.3. Future Work

There is unlikely to be any mitigation of this exploit for previously distributed versions of the Switch as the Tegra bootRom exploit was based on an unpatchable hardware bug. However, Nintendo did patch the bug on a hardware level for devices released after disclosure of the exploit. Additionally, the ingenuity of the homebrew community is likely to ensure that future versions of the Nintendo Switch will be exploited. It must also be noted that the exploitation of the Nintendo Switch is not solely limited to the Tegra exploit. An updated list of vulnerabilities in the hardware and software of the Switch is accessible in the SwitchBrew Wiki (2020b) and also at GBATemp (2020).

There are also new consoles being released, such as the Nintendo Switch Lite, the Switch 2 and other derivative

models. As the underlying memory architecture will likely be the same, this presents an opportunity for future forensic research, building on our initial work.

6.4. Future of Console Forensics

As the amount of functionality and therefore provided metadata used by portable consoles increases, the capability for effective extraction of forensic evidence also increases.

The extraction of forensic evidence via memory dumps has been possible on most mass market consoles. Though the evolution of modern memory protection mechanisms such as Address Space Layout Randomisation and Data Execution Prevention have made exploitation and gaining remote code execution to extract forensic data more complex, most mass market consoles to date have been exploited.

7. Conclusions

We have shown that it was possible to recover data from the Nintendo Switch console in the form of a few key forensic artefacts. We also evaluated the potential evidential value of these artefacts in an investigation. Many of these forensic artefacts persisted between factory resets of the device.

We automated this recovery capability and provided the code enabling this extraction through 10 Autopsy ingest modules, enabling forensic analysts to replicate our research for criminal or corporate forensic investigations. This also allows future forensics research to build upon the findings of our research.

The exploitation, extraction and analysis of forensic data from gaming consoles is not entirely novel. However, our instructions and tools for the forensic analysis of the Switch are novel and provide capabilities for analysts. This method may be applicable to other portable consoles released by Nintendo as part of the Switch product line.

Acknowledgment

The authors would like to thank the Defence Science and Technology Laboratory (DSTL) for providing funding for this research under the 'Digital Crime Scene Forensics Research' research area. Thanks also to Chris Hargreaves for helpful advice and guidance and to Felix Freiling for shepherding this paper.

References

Association of Chief Police Officers, 2012. ACPO Good Practice Guide for Digital Evidence. Technical Report. URL: https://www.digital-detective.net/digital-forensics-documents/ACPO_Good_Practice_Guide_for_Digital_Evidence_v5.pdf.

Barney, A., 2020. Github - Thealexbarney/LibHac. URL: https://github.com/Thealexbarney/LibHac/releases.

Basis Technology, 2020. Autopsy Forensic Browser Developer's Guide and API Reference - Module Development Overview. URL: https://www.sleuthkit.org/autopsy/docs/api-docs/4.9.0/platform_page.html.

BBC News, 2020. Nintendo Wins 1.5m in Switch Hacking Case. URL: https://www.bbc.co.uk/news/technology-54386985.

Carrier, B., 2005. File System Forensic Analysis. Addison-Wesley Professional.

Doolan, L., 2019. Nintendo Switch's Online Activity Leads To Break In Murder Case Mystery. URL: http://www.nintendolife.com/news/2019/12/nintendo_switchs_online_activity_leads_to_break_in_murder_case_mystery.

EEDAR, 2018. The Evolving Demographics of Nintendo Switch Owners. URL: https://www.eedar.com/post/2018-q4-round-up.

Eliboa, 2020. Github - Eliboa/TegraRcmGUI. URL: https://github.com/eliboa/TegraRcmGUI.

Fukami, A., Ghose, S., Luo, Y., Cai, Y., Mutlu, O., 2017. Improving the reliability of chip-off forensic analysis of NAND flash memory devices. DFRWS 2017 EU - Proceedings of the 4th Annual DFRWS Europe 20, S1–S11. URL: http://dx.doi.org/10.1016/j.diin.2017.01.011, doi:10.1016/j.diin.2017.01.011.

GBATemp, 2020. List of Switch Exploits. URL: https://wiki.gbatemp.net/wiki/List_of_Switch_exploits.

Greca, R., 2020. Github - Renangreca/Switch-Screenshots. URL: https://github.com/RenanGreca/Switch-Screenshots.

Hargreaves, C., Patterson, J., 2012. An automated timeline reconstruction approach for digital forensic investigations. Proceedings of the Digital Forensic Research Conference, DFRWS 2012 USA 9, S69–S79. URL: http://dx.doi.org/10.1016/j.diin.2012.05.006, doi:10.1016/j.diin.2012.05.006.

Hosani, H.A., Yousef, M., Shouq, S.A., Iqbal, F., 2020. State of the Art in Digital Forensics for Small Scale Digital Devices, in: 2020 11th International Conference on Information and Communication Systems (ICICS), pp. 72–78. doi:10.1109/ICICS49469.2020.239531.

Huang, A.B., 2003. Hacking the Xbox: An Introduction to Reverse Engineering. URL: https://bunniefoo.com/nostarch/HackingTheXbox_Free.pdf.

Information Commissioners Office, 2020a. Guide to the General Data Protection Regulation (GDPR): Exemptions. URL: https://ico.org.uk/for-organisations/guide-to-data-protection/guide-to-the-general-data-protection-regulation-gdpr/exemptions/.

Information Commissioners Office, 2020b. What is Personal Data? URL: https://ico.org.uk/for-organisations/guide-to-data-protection/guide-to-the-general-data-protection-regulation-gdpr/key-definitions/what-is-personal-data/.

Lagerholm, F., van den Berg, J., Friel, R., Axelsson, S., 2020. Forensic Analysis of the Nintendo Switch. URL: https://www.diva-portal.org/smash/get/diva2:1441718/FULLTEXT02.

Modux Labs, 2020. Github - modux/Nintendo-Switch-Forensics. URL: https://github.com/modux/Nintendo-Switch-Forensics/tree/master/memory-dump-utils.

Murphy, A., Kitching, C., Clare, H., 2019. How Brothers Murdered by Mum Sarah Barrass Tried to Use Nintendo Switch to Warn Pals. URL: https://www.mirror.co.uk/news/uk-news/how-brothers-murdered-mum-sarah-20840610.

National Crime Agency, 2017. Pathways Into Cyber Crime , 18URL: https://www.nationalcrimeagency.gov.uk/who-we-are/publications/6-pathways-into-cyber-crime-1/file, doi:0217-HO.

Nintendo, 2020. Dedicated Video Game Sales. URL: https://www.nintendo.co.jp/ir/en/finance/hard_soft/index.html.

Nintendo Homebrew Discord Server, 2020. Nintendo Homebrew Switch Guide. URL: https://nh-server.github.io/switch-guide/faq/.

Nvidia Support, 2018. Security Notice: NVIDIA Tegra RCM Vulnerability. URL: https://nvidia.custhelp.com/app/answers/detail/a_id/4660/.

Rabaiotti, J.R., Hargreaves, C.J., 2010. Using a software exploit

to image RAM on an embedded system. Digital Investigation 6, 95–103. URL: http://dx.doi.org/10.1016/j.diin.2010.01.005, doi:10.1016/j.diin.2010.01.005.

Read, H., Thomas, E., Sutherland, I., Xynos, K., Burgess, M., 2019. Forensic Analysis of the Nintendo 3DS NAND. Digital Investigation 29, S61–S70. URL: https://linkinghub.elsevier.com/retrieve/pii/S1742287619301641, doi:10.1016/j.diin.2019.04.015.

Read, H., Thomas, E., Sutherland, I., Xynos, K., Read, H., Thomas, E., Sutherland, I., Xynos, K., Burgess, M., 2018. A Forensic Methodology for Analyzing Nintendo 3DS Devices URL: https://hal.inria.fr/hal-01758689/document.

Rogers, M., 2020. Forensic Evidence and Cybercrime. Springer International Publishing, Cham. pp. 425–445. URL: https://doi.org/10.1007/978-3-319-78440-3_13, doi:10.1007/978-3-319-78440-3_13.

Sadayuki, F., 2012. Msgpack. URL: https://msgpack.org/.

Stevens, T., 2015. Security and Surveillance in Virtual Worlds: Who Is Watching the Warlocks and Why? International Political Sociology 9, 230–247. doi:10.1111/ips.12094.

Stojadinovic, R., 2020. Github - Rajkosto/Biskeydump. URL: https://github.com/rajkosto/biskeydump.

SwitchBrew Wiki, 2020a. Flash Filesystem. URL: https://switchbrew.org/wiki/Flash_Filesystem.

SwitchBrew Wiki, 2020b. Switch SystemFlaws. URL: https://switchbrew.org/wiki/Switch_System_Flaws.

Tanguy, G., Gabriel, H., Roussel-Tarbouriech, I., Menard, N., True, T., TiniVi, Reisyukaku, 2019. Methodically defeating nintendo switch security. arXiv , 1–12URL: https://arxiv.org/pdf/1905.07643.pdf, arXiv:1905.07643.

Temkin, K., 2018. Vulnerability Disclosure: Fusée Gelée. URL: https://misc.ktemkin.com/fusee_gelee_nvidia.pdf.

Turnbull, B., 2008. Forensic Investigation of the Nintendo Wii : A First Glance. Small Scale Digital Device Forensics Journal 2, 1–7. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.438.481&rep=rep1&type=pdf.

United Nations Office on Drugs and Crime, 2013. Comprehensive Study on Cybercrime. Technical Report February. URL: http://www.unodc.org/documents/organized-crime/UNODC_CCPCJ_EG.4_2013/CYBERCRIME_STUDY_210213.pdf.