Forensic Analysis of the Nintendo 3DS NAND

By

Gus Pessolano, Huw O.L. Read, Iain Sutherland,
and Konstantinos Xynos

From the proceedings of

The Digital Forensic Research Conference

DFRWS 2019 USA

Portland, OR (July 15th - 19th)

Digital Investigation 29 (2019) S61–S70

Contents lists available at ScienceDirect

Digital Investigation

journal homepage: www.elsevier.com/locate/diin

DFRWS 2019 USA — Proceedings of the Nineteenth Annual DFRWS USA

Forensic Analysis of the Nintendo 3DS NAND

Gus Pessolano [a], Huw O.L. Read [a, b, *], Iain Sutherland [b, c], Konstantinos Xynos [b, d]

[a] *Norwich University, Northfield, VT, USA*
[b] *Noroff University College, 4608 Kristiansand S., Vest Agder, Norway*
[c] *Security Research Institute, Edith Cowan University, Perth, Australia*
[d] *Mycenx Consultancy Services, Germany*

ARTICLE INFO

Article history:

Keywords:
Nintendo 3DS
Games console
Physical extraction
Piracy
NAND
Dump

ABSTRACT

Games consoles present a particular challenge to the forensics investigator due to the nature of the hardware and the inaccessibility of the file system. Many protection measures are put in place to make it deliberately difficult to access raw data in order to protect intellectual property, enhance digital rights management of software and, ultimately, to protect against piracy. History has shown that many such protections on game consoles are circumvented with exploits leading to jailbreaking/rooting and allowing unauthorized software to be launched on the games system. This paper details methods that enable the investigator to extract system activity, deleted images, Internet history items, relevant friends list information, the console's serial number and plaintext WiFi access point passwords. This is all possible with the use of publicly available, open-source security circumvention techniques that perform a non-invasive physical dump of the internal NAND storage of the Nintendo 3DS handheld device. It will also be shown that forensic integrity is maintained and a detailed analysis is possible without altering original evidence.

1. Introduction

Games consoles with a range of connectivity and rich functionality are a valuable source of evidence with features such as Internet browsing capability, social media sharing and email/chat like conversation options (Conrad et al., 2010; Moore et al., 2014; Davies et al., 2015; Read et al., 2016). The original Nintendo 3DS console was released in Japan on February 26th, 2011 (NDS-Gear, 2019). More than six years after its release, the console has gone through numerous iterations including the Nintendo 3DS XL, New Nintendo 3DS, New Nintendo 3DS XL, Nintendo 2DS, and the New Nintendo 2DS XL.

The 3DS, Nintendo's highest selling active console with 73.53 million units sold worldwide (Nintendo, 2018) has already been involved in criminal activities (Ashcroft, 2013) as has its predecessor the DSi (Hanlon, 2012). It is aimed at a younger/family friendly audience; as of 29th January 2019 the ESRB has in its DS/DSi/3DS category 2587 games rated at Early Childhood/Everyone, but only 38 rated as Mature (17+) or above (ESRB, 2019). It is reasonable to conclude that a case involving a 3DS is more likely to involve minors. Therefore it warrants particular note from the digital forensic community to improve existing methods of data extraction and analysis to support, in the United States at least, the Daubert standard.

Alongside the growth of the popularity of the 3DS, interest in hacking and modifying the device has continued over the course of the console's lifespan (McClintic et al., 2018; Scires et al., 2018). On May 19th, 2017, a vulnerability known as the ‘boot9strap’ vulnerability was published online (3ds, 2019). This vulnerability, in conjunction with an exploit known as ‘ntrboot,’ allows for arbitrary code execution to occur before the console has booted to the system menu. This allows for a restorable backup of the system's NAND memory to be taken without booting the console, paving the way for a forensically sound methodology.

Like other game systems, the Nintendo 3DS uses a non-volatile NAND chip to store the system firmware and user settings information. The firmware is loaded from the NAND by the bootloader when the console is powered on (Gowrishankar, 2016). The 3DS console NAND is encrypted with a key that is specific to each console, meaning that the NAND cannot be analyzed without obtaining this decryption key. Although JTAG has been shown as a viable method to extract the NAND, it still requires hardware modification and some other means to obtain the decryption key to

[*] Corresponding author. Norwich University, Northfield, VT, USA.
E-mail address: hread@norwich.edu (H.O.L. Read).

https://doi.org/10.1016/j.diin.2019.04.015
1742-2876/

make the image readable. This forensic analysis will use a console purchased in the USA; it should be noted that some folder names on the NAND do differ slightly between regions. A valuable source of information for folder names may be found in 3dbrew (2015a).

2. Related work

Prior work by Read et al. (2016) in their paper entitled “A forensic methodology for analyzing Nintendo 3DS devices”, presented a summary of different features (Read et al., 2016: Table 1) that would be of interest to the forensic examiner including the web browser, camera, friend lists, activity log and game notes. What follows is a detailed forensic analysis methodology based upon empirical research providing a guide for an examiner to follow while performing a live analysis of the console. The authors also state that, by investigating the device in this fashion, it mitigated issues associated with dumping encrypted NAND images but “may have an impact on the state of the device” (Read et al., 2016). These impacts were minimized by adhering to UK ACPO guidelines to “minimise alterations, tampering and modifications of the original evidence to the extent possible” (Read et al., 2016).

Significant changes in the Nintendo 3DS hacking community has led the authors to less invasive forms of forensic analysis. The work conducted in this paper leading to the forensic acquisition of a 3DS would not be possible without the seminal work by Scires et al. (2018) and the tools created as a result of their research. Their paper made use of flaws discovered in the RSA signature verification of one of the boot ROMs (the ARM9 boot ROM known as “Boot9”) to cause firmware created by third parties to appear valid to the signature parser. Alternative firmware could notionally be used to redirect boot ROM code flow, execute a payload which then becomes a persistent exploit to allow extraction of information from protected areas of memory.

Further discoveries (Scires et al., 2018) made by analyzing the protected half of the Boot9 ROM revealed that, before it attempts to load a firmware image from the internal NAND, it will check if the device is closed or the physical sleep switch is ‘on’ (e.g., in the case of the 2DS) and whether the START, SELECT and X buttons are depressed. Boot9 checks if a DS cartridge is inserted and if so, attempts to load a signed firmware from it, bypassing the firmware on the NAND.

As confirmed in McClintic et al. (2018), the trust in the 3DS is the boot ROM which is burned into the System-on-Chip (SoC) during the manufacturing process. Many embedded systems use this method as it prevents 3rd parties from modifying the boot ROM as they are physically unable to write to the SoC, only read from it. In a similar fashion, limera1n exploit on early Apple iPhone devices could not be prevented without first updating the manufacturing process (iPhoneWiki, 2019).

When such vulnerabilities are found and are made exploitable, patching the issue consists of changing the SoC at the factory for all future revisions of the hardware. This is significant as the vulnerability will continue to exist in all Nintendo 3DS devices prior to discovery of the exploit (assuming Nintendo chooses to update the manufacturing process and change the boot ROM) providing a permanent method to perform digital forensic analysis.

These types of vulnerabilities allow opportunities to improve digital forensic analysis techniques as shown in this paper. However, the devices that are manufactured to deploy such exploits have come under close scrutiny in the past. Unofficial game cartridges, known colloquially as flashcarts (as they are user-writable) are commonly associated with software piracy. Of particular interest is the British court case between Nintendo Company Ltd. v. Playables Ltd. (Nintendo, 2010) which addresses the issue of copy infringement and copy-protection devices. The company, Playables Ltd. (based in the UK) imported a number of flashcarts (referred to as “game copiers” in Nintendo, 2010) which fit the proprietary cartridge connection in a previous-generation console, the Nintendo DS. One of the key arguments the defense made during the case was that they “argued that it [Playbles Ltd.] did not know that the devices would be used for this purpose and that the devices can be used for legal purposes” (Outlaw, 2010). The court rejected this defense as it did not provide legal cover for other illegal uses, i.e. software piracy. In section 296 of the 1988 Copyright, Designs and Patents Act (Copyright, Designs and Patents Act, 1988), an offence is committed if a person behind the sale of “any means the sole intended purpose of which is to facilitate the unauthorised removal or circumvention of the technical device” knows or believes it will be used to make illegal copies of software. The UK High Court said that Playables Ltd. did have this knowledge (Outlaw, 2010) and therefore in breach of the law. More recently in Canada in Nintendo of America Inc. v. King (2017), a similar judgement was made whereby the respondents were found to be liable for circumvention and/or infringement of the 2012 amendments to the 1985 Copyright Act. Of particular interest is section [33] which provides a list of the offending devices that were made available for sale; the first such item, R4i 3DS, is the immediate predecessor to the device used in this research (R4i+ 3DS) that made the forensic analysis presented in this paper possible.

3. Contribution

Although the forensic analysis of a Nintendo 3DS has been addressed before (Read et al., 2016) this paper makes a number of improvements to the field. The 3DS has the ability to store data on a memory card (Nintendo, n.d.) and on an internal NAND chip. This paper presents a forensically sound method of extracting the NAND and provides detailed analysis of different artifacts of interest to the digital forensic examiner. It demonstrates that the NAND disk image can be decrypted and analyzed entirely from a forensic

Table 1
Bookmark structure.

Length (hex)	Description
8	Timestamp, number of milliseconds since epoch (Jan. 1st, 2000). Default bookmarks are zeroed out.
4	Unknown
1	Counter. The byte increments the bookmark entry. First value is 0x00.
1	Unknown
1	Appears to have the value 0x01 if a default bookmark, 0x00 if a user added bookmark.
1	Unknown, always 0x01
200	Unknown, has data for default bookmarks, zeroed for user-generated.
400	URL (null-padded)
200	Bookmark name

Fig. 1. DS (L), R4i (M), 3DS (R).

workstation, without the use of the original 3DS console. The method is repeatable and verifiable, using best-practice hashing to confirm multiple extractions are digitally identical. After a review of available literature, it appears that this paper is possibly the first to make use of tools that circumvent security which are commonly associated with copyright infringement to develop more effective methods for the analysis of game console devices. Finally, by using hardware that is commonly associated with facilitating piracy (i.e., flashcarts), it can provide an interesting consideration on how law enforcement can use such tools as it becomes more difficult to purchase or access them in their own country.

4. Method and tools

The first challenge to the investigator is assembling the right collection of tools. As discussed in the literature review, 3DS consoles may be instructed to launch alternative firmware from a standard DS cartridge. Nintendo does not provide any cartridges that have this capability for obvious reasons, however there are several third-party sources that have independently developed and sold such devices. The R4i+ 3DS/DS (henceforth R4i) cartridge was used by the authors and found to be particularly effective at performing the digital forensic tasks. The flashcarts cartridges have the unique ability of being writable in nature. Unfortunately, such devices are commonly associated with copyright infringement (Nintendo, 2017, Nintendo Support, 2019). Technical information about the flashcart may be seen in the manufacturer's website (r4ids, 2019). It should be noted, that the information on this particular webpage also appears to demonstrate how to use the device for copyright infringement. The authors do not condone these illegal activities, but rather provide the reference to the website for the forensic investigator to understand the usage of the R4i flashcart for lawful means and for digital forensic analysis purposes only.

The R4i flashcart is presented in Fig. 1 and Fig. 2. For comparison purposes in Fig. 1, a Nintendo DS cartridge appears on the left, a Nintendo 3DS cartridge appears on the right. Within the R4i flashcart, there are a few important areas to highlight. Fig. 2 shows the flashcart itself, with a microSD slot in the upper-right of the image. The microSD slot in the R4i was not used in the experiments, rather all microSD (for devices designated as new) and SD (for all original 3DS devices) cards were inserted into the console itself. Furthermore, there is a small white switch just underneath the microSD slot on the R4i. The switch can toggle between an N or a D. D enables DS functionality (not used in this paper), N enables the NTRboot functionality (i.e. the ability to boot from a DS, rather than 3DS, cartridge if the conditions (specific button presses) discussed in Scires et al. (2018) are met). It is important for the forensic investigator to note that the switch should be set to N which will allow dumping the internal NAND and the key required for decryption.

4.1. Preparing the flashcart

The R4i card lets the investigator redirect the boot process from the internal firmware to a microSD/SD card in the Nintendo 3DS. The MicroSD/SD card must be prepared in a particular fashion; detailed guides exist online (3DS Guide, 2019); the general process is as follows:

1. A reflashable cartridge is prepared with the ntrboot files (the R4i used by the authors is pre-flashed with ntrboot_flasher (ntrteam, 2019)).
2. The original microSD/SD card in the console is forensically imaged and set aside. A new microSD/SD is prepared with the following tools:
   a. boot9strap (Scires, 2017b) which enables Boot9 code execution
   b. decrypt9WIP (decrypt9, 2017) which performs the actual dumping of NAND and the system's encryption key.

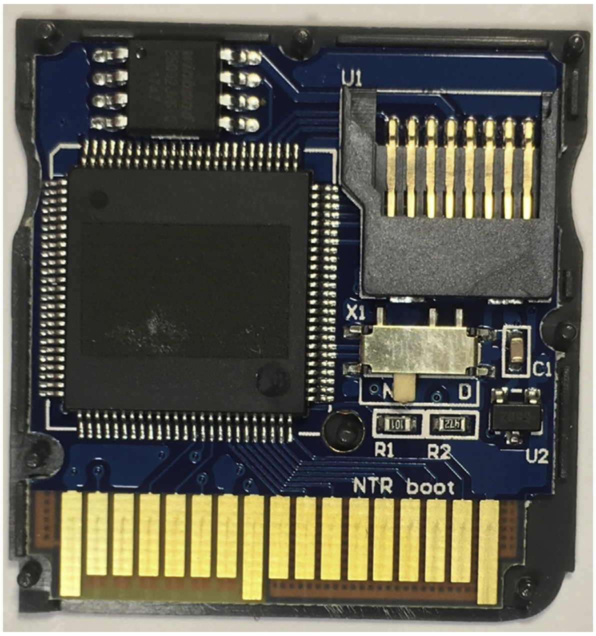


Fig. 2. Inside the R4i.

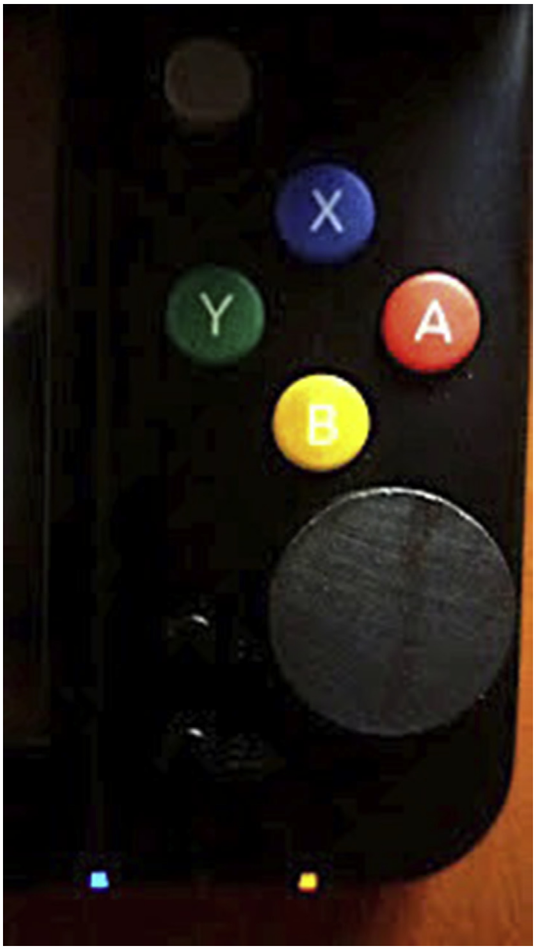
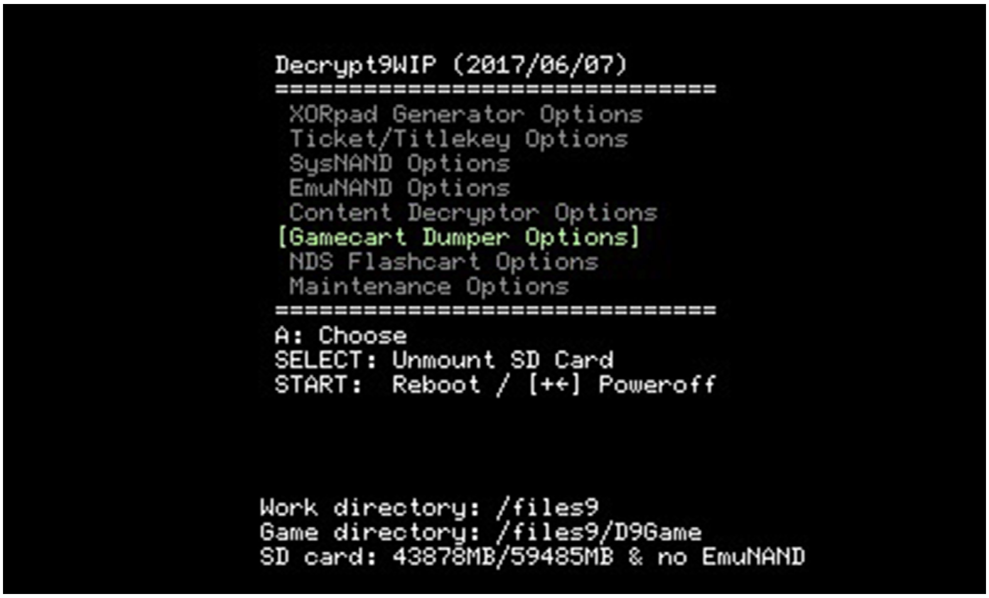


Fig. 3. Magnet location.

4.2. Dumping the NAND

The actual process of obtaining a dump of the internal NAND is as follows:

1. Ensure the console is in an off state (not suspend).
2. Fold-open the console (ignore if using a Nintendo 2DS).
3. Use a magnet to trigger the sleep switch on the 3DS. Fig. 3 identifies where the switch is, just below and right of the yellow B button. If analyzing an original Nintendo 2DS, enable the sleep switch instead (no magnet required).
4. With the magnet in place (or sleep switch enabled on the original 2DS), hold START, SELECT, X and the POWER button to turn on the console. These should be held for about 10 seconds and and then can be released. The magnet (or sleep switch) can be removed at this juncture. This can take some practice, it is advised that the investigator familiarise themselves on another console to limit unintentional alteration of their evidence.
5. If the process worked correctly, the analyst should be presented with a screen similar to Fig. 4. The Decrypt9WIP application, initialised from the microSD/SD card, will perform three important functions for the forensic analyst:
   a. Backup of the internal NAND via `SysNAND Options - > SysNAND Backup/Restore ... - > NAND Backup`. This creates two files, the NAND.bin and a NAND.bin.sha (SHA256 value).
   b. Verification of the NAND dump via `SysNAND Options - > SysNAND Backup/Restore ... - > Validate NAND Dump`.
   c. Backup of the unique encryption key (XOR) needed to decrypt the NAND dump via `XORpad Generator Options - > CTRNAND Padgen`, nand.fat16.xorpad.
6. Once the NAND and the encryption key have been dumped to the microSD/SD card and the NAND dump verified, the console can be powered off by pressing LEFT and START on the main Decrypt9WIP menu.
7. Once the console is off; the microSD/SD card can be removed and put into the forensic workstation. The SHA256 of the NAND.bin may be created and compared to the value stored in NAND.bin.sha.



```
 Decrypt9WIP (2017/06/07)
 ==============================
  XORpad Generator Options
  Ticket/Titlekey Options
  SysNAND Options
  EmuNAND Options
  Content Decryptor Options
 [Gamecart Dumper Options]
  NDS Flashcart Options
  Maintenance Options
 ==============================
 A: Choose
 SELECT: Unmount SD Card
 START:  Reboot / [+←] Poweroff




Work directory: /files9
Game directory: /files9/D9Game
SD card: 43878MB/59485MB & no EmuNAND
```

Fig. 4. Decrypt9WIP tool for dumping NAND.

4.3. Decrypting NAND contents

At this point, the analyst will have three files, the encrypted NAND.bin, the SHA256 hash stored in NAND.bin.sha and the encryption key nand.fat16.xorpad. The NAND.bin file itself contains two partitions, the backwards-compatible DSi partition and the 3DS partition (3dbrew, 2017b). The 3DS partition, or CTRNAND, contains artifacts of interest to the examiner. To decrypt the NAND.bin for use in digital forensic tools, the dump contents must first be passed through an XOR function. This may be achieved using a dedicated XORing application, like 3DSFAT16Tool (3DSFAT16tool) as seen in Fig. 5. With the process taken thus far, the tool will take the NAND.bin as input, the nand.fat16.xorpad as input, and XOR the CTRNAND partition to produce a decrypted version in a raw, uncompressed format and dump the results into a new FAT16 image (e.g., using the -d dump flag, as shown in Fig. 5). The filesystem is recognized as FAT16 which is readily understood by many digital forensic tools.

5. Forensic analysis

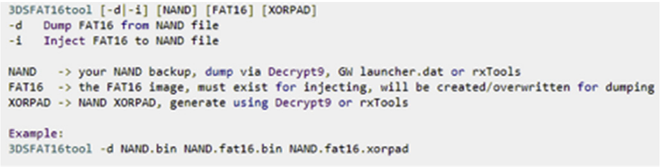
This paper seeks to perform a number of experiments of interest to the forensic examiner to determine what evidence may be left on a 3DS without having to boot or investigate the device live as in Read et al. (2016). A second-hand version of the new Nintendo 3DS was used for the experimentation; findings were later confirmed against a newly-purchased 3DS XL, an original 3DS and an original 2DS. The second-hand version was updated to firmware version 11.6.0—39U which was the most up-to-date at the time the research was conducted. The console was used to play several games and had around 50 friends listed in the friends list, and was connected to different WiFi access points.

OSFMount (PassMark Software, 2019) was used to enable a Windows based system to mount the decrypted CTRNAND partition as read-only. AccessData's Forensic Toolkit 7.0 was used to perform forensic analysis. Fig. 6 presents the folder structure of the CTRNAND.

The main folder of interest is the data folder. Within the data folder, is a directory made up of alphanumeric characters which represents a SHA256 hash of console-unique data (ID0 in 3dbrew, 2017b) providing a unique value. The extdata and sysdata folders are found within this ID0 directory.

5.1. Analyzing the extdata folder

The extdata (or extra data) folder stores additional, arbitrary data for an application (3dbrew, 2015b). The directory structure may be seen in Fig. 7. A single folder titled 00048000 is present within the extdata folder and contains a number of sub-folders which represent several applications. Within this are a number of folders (beginning with "f") that represent the data storage for built-in applications. Finally, each of these have a single folder, 00000000, which contains the actual content. By default, there are two files 00000001 and 00000002 which are part of the core operating system. The timestamps reflect an earlier date (2002 on the new 3DS XL and new 3DS, 2001 on the original 3DS and 2011 in the 2DS devices tried) and do not appear to be updated based on usage. If present, 00000003 and above are created by a user's actions and have a timestamp reflecting the action taken by an individual.

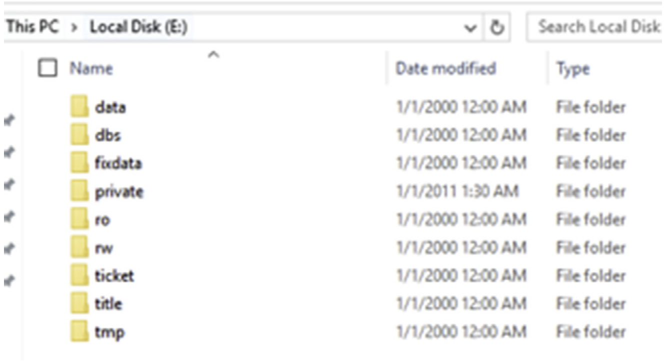


```
3DSFAT16tool [-d|-i] [NAND] [FAT16] [XORPAD]
-d   Dump FAT16 from NAND file
-i   Inject FAT16 to NAND file

NAND   -> your NAND backup, dump via Decrypt9, GW launcher.dat or rxTools
FAT16  -> the FAT16 image, must exist for injecting, will be created/overwritten for dumping
XORPAD -> NAND XORPAD, generate using Decrypt9 or rxTools

Example:
3DSFAT16tool -d NAND.bin NAND.fat16.bin NAND.fat16.xorpad
```

Fig. 5. 3DSFAT16tool.



Name	Date modified	Type
data	1/1/2000 12:00 AM	File folder
dbs	1/1/2000 12:00 AM	File folder
fixdata	1/1/2000 12:00 AM	File folder
private	1/1/2011 1:30 AM	File folder
ro	1/1/2000 12:00 AM	File folder
rw	1/1/2000 12:00 AM	File folder
ticket	1/1/2000 12:00 AM	File folder
title	1/1/2000 12:00 AM	File folder
tmp	1/1/2000 12:00 AM	File folder

Fig. 6. Contents of the NAND

Of particular interest to the examiner are the contents of the camera app (f0000001) and the sound recorder app (f0000002).

5.2. Camera app - f0000001

Within the camera app directory f0000001, the subdirectory 00000000 contains files of interest to the examiner. As mentioned above, 00000001 and 00000002 are present by default. AccessData's FTK was used to analyze the remaining files. The 0000003 and 00000004 files are created after the camera app has been used and remain even if all pictures have been deleted. The timestamp of 00000003 is updated when a picture is taken or when a picture is deleted. 00000004 appears to reflect the initialization of the camera app and does not change thereafter.

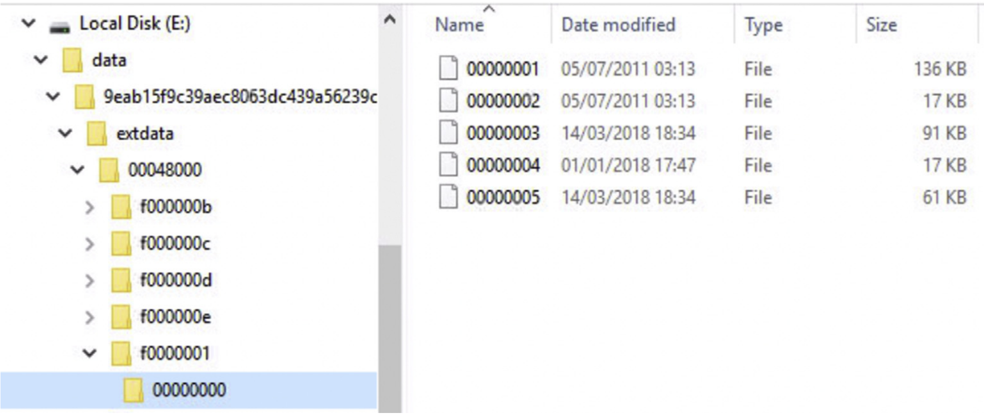
If present, 00000005 and above contain embedded pictures (one per file - n.b. "3D" images appear as two embedded files, slightly off-center from each other) in JPEG format (i.e, file header 0xFFD8FFE1). FTK successfully carved all the images taken. The images were then deleted using the camera app's delete function. Another NAND image was obtained and put through a second file carve. FTK was able to carve out all the deleted pictures.

The images themselves contain a substantial amount of metadata in the EXIF headers. Using ExifTool (Harvey, 2019) in the following fashion `exiftool -a -u -g1 filename.jpg` a wealth of metadata was obtained. Of particular interest within the "ExifIFD" group, create date and date/time original tags provided the creation date of the image. Within the Nintendo group, time stamp (which again reflected the creation time of the image) and a tag labelled "internal serial number" were obtained.

5.3. Sound app - f0000002

Within the sound app directory f0000002, the subdirectory 00000000 contains files of interest. As before, 00000001 and 00000002 are present by default and are of little evidentiary value. Files 00000003 and above contain sound recordings made by the built-in microphone. They are in the m4a format (i.e., file header 0x66747970).

PhotoRec was able to extract the m4a files. The sound clips were then deleted, and the console re-imaged. PhotoRec was able to recover all the deleted audio files. ExifTool (Harvey, 2019) provided



- Local Disk (E:)
  - data
    - 9eab15f9c39aec8063dc439a56239c
      - extdata
        - 00048000
          - f000000b
          - f000000c
          - f000000d
          - f000000e
          - f0000001
            - 00000000

Name	Date modified	Type	Size
00000001	05/07/2011 03:13	File	136 KB
00000002	05/07/2011 03:13	File	17 KB
00000003	14/03/2018 18:34	File	91 KB
00000004	01/01/2018 17:47	File	17 KB
00000005	14/03/2018 18:34	File	61 KB

Fig. 7. 3DS extdata folder structure.

an interesting insight into the metadata; the intuitively-named entries with “date” in the title were all consistent with one another, but were all incorrect by several years. A manual conversion with epoch set at Jan. 1, 2000 (see analysis of the Web Browser below) did not correct the skew. However, the investigator will want to take note of the Title entry; it presented the correct date in M/D/Y H:M:S in string format.

5.4. Analyzing the sysdata folder

The sysdata (or system savedata) provides storage for the applications on the 3DS. Every module/application on the NAND has a savegame associated with it. Within the sysdata folder are a number of subdirectories which represent the different applications (3DBrew, 2015a). Within each a save file, 00000000 may be identified. The timestamps are not particularly helpful. Using the classifications available at 3dbrew (2015a), system modules (those beginning with 0001xxxx in Fig. 8) have a factory default set between the year 2000 and 2001 and do not change when used. The system applications and applet saves (those beginning with 0002xxxx in Fig. 8) have a timestamp set to their first initialization and do not change. At best, an investigator could deduce when the latter were first launched based on the timestamps.

There are 11 (12 in recent updates) different system module savegames, all of which have an entry in sysdata. There are 22 system application savegames, only those which have been explicitly opened will appear in sysdata. The system modules are functions like Spotpass (3dBrew, 2017c), whereas the applications include the Internet Browser and the microSD card management application. Analysis was performed on both the system modules and system applications. The windows strings (Russinovich, 2016) command was the main tool used to parse the files into human-readable text.

5.5. Friends – 00010032

The Friends system module savegame can be located under the 00010032 folder. Within this folder is the standard 00000000 file (3DBrew, 2017d). Analyzing this file with strings yielded the names and publicly displayed messages of all friends that have been added to the console. In addition the 3DS console's Serial number, username, and public message were all viewable.

5.6. microSD management – 00020241

The microSD Management application is specific to the New Nintendo 3DS line of 3DS consoles only. Since the microSD card of the New Nintendo 3DS is kept behind a screwed-in back cover, Nintendo offers this application, which temporarily turns a New Nintendo 3DS into a device that can be accessed by a SMB network connection. No system/NAND files can be accessed in this manner, only files stored on the console's microSD card (3DBRew, 2018b).

The microSD Management application savegame can be located under the 00020241 folder. This is by far one of the smallest savegame files of any application or system module. Within this folder is the standard 00000000 file. Analyzing this file with the strings command yields a number of things, including the SSID of any network stored within the console. The file also contains the name the device is given when joining the network (in this case, ‘3DS-6599’) and the device's required login name and PIN to gain access to it, in this case ‘User’ and ‘77XX.’ The ‘OTHER’ field is also the name of the workgroup in the console's respective network.

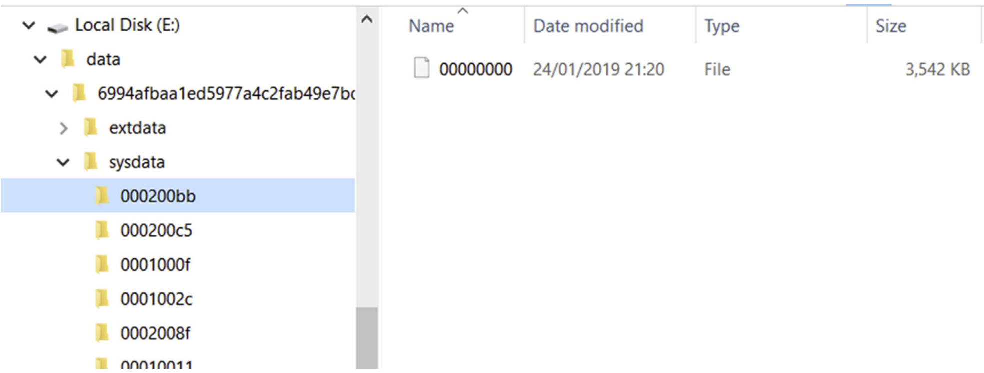
5.7. Internet browser - 000200bb

The New and original 3DS systems have different storage formats for the web browser (3DBrew, 2018). The original 3DS has not been analyzed at the time of writing due to time constraints in the research. In the New Nintendo 3DS console, the Internet Browser application savegame can be located under the 000200bb folder; A file carve after browsing the web does not reveal any images of the web pages visited. This suggests the 3DS does not keep a local webcache like desktop web browsers. However, it is possible to extract the bookmarks (name, URL, date & time of creation) and the history (name, URL, data & time of visit). Using the 3DS Save File Extraction Tool, in particular disa-extract.py (wwylele, 2018), the file t.bin can be extracted from 00000000.

It was found that the analyst can investigate 00000000 directly after analysis of t.bin as this extracted file is readable in its host. The investigator should search for the header of t.bin inside 00000000, 0x100000080DF0A00. From this offset, the start of the first bookmark entry can be found 0xD8 bytes later, the first value is 0000000000000000 0000000000000101. Table 1 describes the structure of a bookmark entry (0 × 810 bytes).

After the last bookmark, the next timestamp is zero, but the counter and the following value are 0xFF. This is then repeated every 0 × 810 bytes until a final value of 0xFFFFFFFF00007E43 (followed by 8 null bytes) is reached. Based on the offsets, it appears there is enough capacity (0x31E30 bytes) for 0x63 (99 decimal) bookmarks on the new Nintendo 3DS. This is confirmed in the bookmarks function of the web browser (xx/99).

The history follows a near-identical structure as in the above



Local Disk (E:)
data
6994afbaa1ed5977a4c2fab49e7bc
extdata
sysdata
000200bb
000200c5
0001000f
0001002c
0002008f
00010011

Name	Date modified	Type	Size
00000000	24/01/2019 21:20	File	3,542 KB

Fig. 8. sysdata (system savedata) directory structure.

table. Immediately after the bookmark final value, the first 8 bytes represent the timestamp. Of the second 8 bytes, the first 3 are either 0x010101 or 0x010100 (remainder are null). If the third byte is 0x01, it indicates the next 0 × 200 bytes have data. The remainder of the structure is identical to the bookmarks.

After the last history item, the next timestamp is null, the next byte is 0x01. The remainder of the entry (0 × 807 bytes) is filled with null bytes. This repeats until the value 0x0100000001000000 appears. Based on the offsets, it appears there is the same capacity (0x31E30 bytes) as with bookmarks, for 0x63 (99 decimal) items on the new Nintendo 3DS.

After analysis of the web browser, the built-in reset functionality was tested (Fig. 9). An image of the NAND was taken and the method above was used to successfully retrieve all bookmarks and history entries. AccessData FTK did mark the folder 000200bb was deleted but the data could be recovered.

5.8. *Config services - 00010017*

The Config Services system module savegame can be located under the 00010017 folder. Within this folder is the savegame file 00000000 (3DBrew, 2017a). After an investigation of the WiFi configuration entries, it was discovered that WiFi SSIDs and their passwords are stored in plaintext. An investigator may search for the start of the embedded config file with 0x4100E44100, noting the offset. Then, following guidance in 3DBrew (2017a), search for the WiFi configuration slot block IDs. The offset to the WiFi entry can be located by adding offset in the block entry to the one noted earlier.

The 3DS was added to three access points, their passwords accessible in the manner described above. Two of the entries were erased, the console was re-imaged. One of entries was recoverable (see Fig. 10); the others could not be found (neither by explicitly searching for the known SSID nor the known password).

5.9. *Further experiments*

5.9.1. *Impact of booting system*

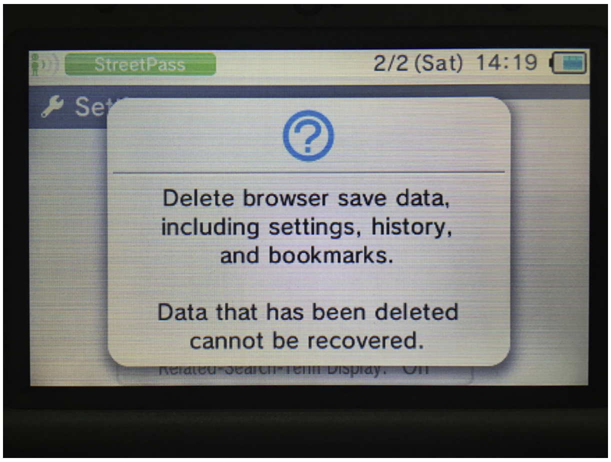
Booting the 3DS normally into the main menu and then powering it is enough to change the hash of the image. Given that the keypress combinations can easily be missed causing the console to boot, a comparison was made between a pristine capture and those created after booting. A hash set was collected from the files in the pristine image. This set was then applied using the known file filter (KFF) functionality in AccessData's FTK to ignore known files in subsequent images and remove them from view. Less than 1.7% of files were different in successive iterations of booting the console. Changes were observed in *sysdata* related to live configuration settings (*00010017* and *00010022*), online services SpotPass (*000100034*) and News (*00010035*), and the Home Menu application (*0002008F*). Changes observed in *extdata* are similar, with Home Menu and SpotPass storage (*f000000d*), SpotPass notification storage (*f0000009*), and play/usage records (*f000000b*).

5.9.2. *Analysis of title.db*

On the root of the decrypted NAND disk image is a folder dbs or databases. Within this folder is several files, one of which is the title.db file. Analyzing the title.db file with strings does not reveal much other than lots of 'CTR-N-HXXX' entries. These strings are actually the installed titles that are present on the console. While most of these are default system titles, some of them can actually be installed games (Decrypt9, 2019). For example, 'CTR-P-AQEE' is the system title for The Legend of Zelda™: Ocarina of Time™ 3D. The last four letters of the system title can be searched in 3dsdb.com to reveal the game's actual title. This can be helpful in seeing exactly what software was installed on the console, without having to live boot the console to check.

5.9.3. *Restoring a NAND dump*

The process of decoding the core files on a Nintendo 3DS is a lengthy process, but does provide a greater level of detail by looking

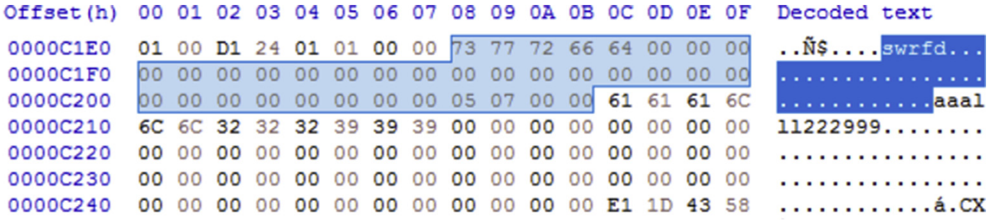


Delete browser save data, including settings, history, and bookmarks.

Data that has been deleted cannot be recovered.

Related-Search-Term Display: Off

Fig. 9. Reset function in the new 3DS browser.



```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text
0000C1E0  01 00 D1 24 01 01 00 00 73 77 72 66 64 00 00 00  ..Ñ$....swrfd...
0000C1F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000C200  00 00 00 00 00 00 00 00 05 07 00 00 61 61 61 6C  ............aaal
0000C210  6C 6C 32 32 32 39 39 39 00 00 00 00 00 00 00 00  ll222999........
0000C220  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000C230  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0000C240  00 00 00 00 00 00 00 00 00 00 00 00 E1 1D 43 58  ............á.CX
```

Fig. 10. SSID swrfd, Password aaalll222999.

at raw data. However, there are reasons why a live investigation could also be performed; it provides quicker interpretation of the data using native tools built into the game console, and reduces the lag time between the release of a new application and the decoding/validation of the raw files by the community. Furthermore, there is information of significance on a microSD/SD card that has not been the focus of this paper. An investigator could follow guidance in Read et al. (2016) to retrieve information in a live fashion to help them with current casework.

This experiment restored an earlier NAND taken from the device. Although best practice when working with traditional media (e.g. hard drives, USB sticks, etc.) is to clone data to another system, this is difficult with a Nintendo 3DS game console. Each 3DS has a unique key (which is dumped to nand.fat16.xorpad) which cannot readily be uploaded to another device. Although it is technically possible to transfer the data for analysis on another system (Plailect, 2019), the process involves installation of several applications on the original console to prepare for transfer which will alter the original NAND data considerably and therefore has the potential to overwrite evidence.

The NAND.bin, NAND.bin.sha and nand.fat16.xorpad need to be placed on the microSD/SD card. Using the Decrypt9WIP tool function located at SysNAND Options - > SysNAND Backup/Restore … - > NAND Restore, and following the instructions in Fig. 11 will begin the restore process.

After it was flashed, a new NAND dump was immediately captured without launching the 3DS firmware. After hashing, it was found that the SHA256 of the reflashed and reimaged dump was the same as the original. In this fashion, an investigator could forensically image the 3DS as described in Method and Tools, proceed to turn the device on and investigate live, then restore the initial capture to confirm findings.

5.10. Impact of system erase

The 3DS has a format system memory function (Fig. 12). Its impact on the 3DS was assessed by using files with known signatures that could be data-carved (JPEG pictures taken with the Camera app), erasing using the function, and then imaging. After analysis, FTK confirmed the filesystem had been altered (the Camera app location described earlier did not contain any user-generated files i.e. those of *00000003* and above). However, the data carve function was able to recover all the pictures from unallocated space. Furthermore, each Nintendo 3DS has a unique xor key (dumped as the nand.fat16.xorpad file) which remains constant after a system erase takes place.

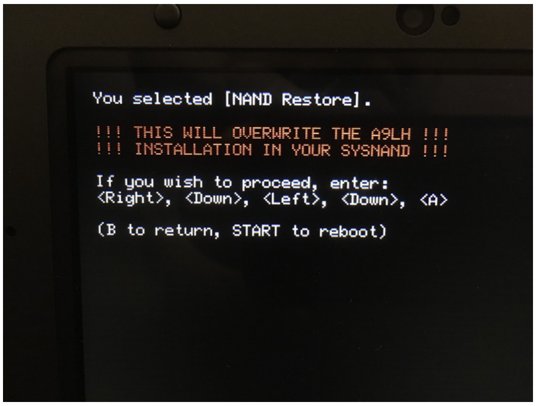
6. Discussion

The methods outlined above enable an investigator to extract and decrypt the contents of a Nintendo 3DS NAND memory chip. This provides access to a number of key sources of information including: deleted images, internet history items, relevant friends list information, console serial number and plaintext access point passwords. This can be achieved without fully booting the device. The memory extraction was repeated (without booting the Nintendo 3DS between extractions) and the successive SHA256 hashes proved to be the same indicated that the process is consistent and does not alter the NAND contents. This suggests a forensically sound method of imaging the console.

This is therefore a more forensically sound method than that of accessing the device via the user interface and therefore provides a more in depth method than Read et al. (2016). However this method requires the use of tools and techniques developed by the hacking and modding communities that seek to expand the capabilities of devices like the Nintendo 3DS. It also relies on the use of a flashcart device that can no longer be sold or supplied in certain jurisdictions.

There are many companies that operate in the cyber security/forensics domain, developing analysis tools for forensic investigators. Clearly these companies need to understand the devices that they analyse and operate on, be it a car, IoT system or mobile device. An in depth analysis would be needed to be able to identify the key areas within a device to ensure the information is extracted and interpreted correctly. Bugs and vulnerabilities in the devices might be employed to extract information. In the README, supplied with the software used in this process, the software author notes much of the iOS-related code is very similar to that used in the jailbreaking scene—a community of iPhone hackers that typically breaks into iOS devices and release its code publicly for free. Techniques that have already been used by other investigators to extract data from mobile devices, including law enforcement in some jurisdictions (Chang et al., 2015).

The question then is that if these tools and techniques are known to the hacking and modding community, should they be ignored by the forensics community. The authors do not approve or condone any action that result in the infringement of intellectual property rights, it is suggested that ignoring the fact that these tools



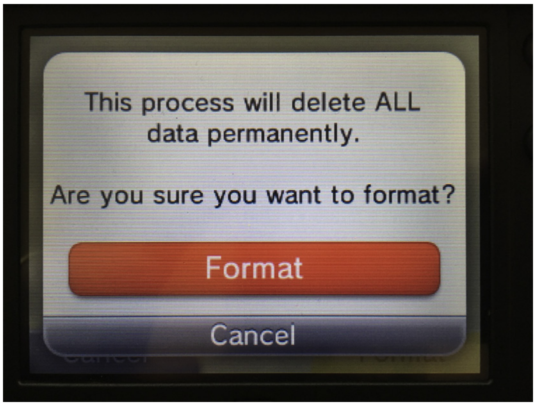
```
You selected [NAND Restore].

!!! THIS WILL OVERWRITE THE A9LH !!!
!!! INSTALLATION IN YOUR SYSNAND !!!

If you wish to proceed, enter:
<Right>, <Down>, <Left>, <Down>, <A>

(B to return, START to reboot)
```

Fig. 11. Decrypt9WIP restore NAND function.


This process will delete ALL data permanently.

Are you sure you want to format?

Fig. 12. Format system memory.

(which are in the public domain) exist serves no useful purpose. The code is open source and can be examined and tested as in this paper and not to use these resources disadvantages the forensic examiner.

7. Conclusions

A collection of hacking tools designed to enable modifications and unlawful game play on 3DS can be used to extract a NAND dump of the console. This technique can be performed to provide a backup of the console's memory. Although the NAND image is encrypted with a console specific encryption key, it can be decrypted due to advancements made by the console hacking community. This software is open source and can be examined to ensure veracity.

Once the contents are extracted and decrypted a myriad of system and application modules can be investigated. Relevant deleted pictures, audio and videos, deleted Internet history items, relevant friendslist information, the console's serial number, plaintext access point passwords, can be extracted and analysed offline. A great deal of information can be obtained through the decrypted NAND analysis. Much of this information can not be accessed via other means and it could be that vital evidence might be missed without the NAND level analysis.

8. Future work

As discussed in the contribution section, there do not appear to be any other digital forensics research papers that rely upon tools typically associated with illegal activities (i.e. copyright infringement through use of flashcarts) to obtain non-invasive, forensically-sound methods of data extraction. The authors would like to explore the broader implications of using such tools when retailers have been found in breach of the law for selling products circumventing copy-protection measures. Furthermore, many of the forensic artifacts identified from the 3DS NAND came from manual parsing and extraction. The authors hope to develop automated tools that can extract information of forensic significance for the analyst in the future.

References

3DBrew, 2015a. System SaveData. Available online at: https://www.3dbrew.org/wiki/System_SaveData. (Accessed 23 January 2019).

3DBrew, 2015b. ExtData. Available online at: http://www.3dbrew.org/wiki/Extdata. (Accessed 26 January 2019).

3DBrew, 2017a. Config savegame. Available online at: https://www.3dbrew.org/wiki/Config_Savegame. (Accessed 26 January 2019).

3DBrew, 2017b. Flash filesystem. Available online at: http://www.3dbrew.org/wiki/Flash_Filesystem. (Accessed 26 January 2019).

3DBrew, 2017c. Spotpass. Available online at: https://www.3dbrew.org/wiki/SpotPass. (Accessed 26 January 2019).

3DBrew, 2017d. FRD savegame. Available online at: https://www.3dbrew.org/wiki/FRD_Savegame. (Accessed 26 January 2019).

3DBrew, 2018. Internet browser. Available online at: https://www.3dbrew.org/wiki/Internet_Browser#New3DS. (Accessed 23 January 2019).

3DBrew, 2018b. microSD Management. Available online at: https://www.3dbrew.org/wiki/MicroSD_Management. (Accessed 26 January 2019).

3ds, 2019. Installing boot9strap (Homebrew launcher). Available online at: https://3ds.hacks.guide/installing-boot9strap-(homebrew-launcher).html. (Accessed 24 January 2019).

3DS Guide, 2019. Available Online at: https://3ds.guide. (Accessed 23 January 2019).

3DSFAT16tool. Available online at: https://github.com/d0k3/3DSFAT16tool/tree/v2. (Accessed 24 January 2019).

Ashcroft, B., 2013. Accused child predator allegedly used Nintendo's Swapnote service. Kotaku Available Online at: https://kotaku.com/child-predators-were-using-nintendos-swapnote-service-1459304126. (Accessed 3 February 2019).

Chang, Y.-T., Teng, K.-C., Tso, Y.-C., Wang, S.-J., 2015. Jailbroken iPhone forensics for the investigations and controversy to digital evidence. J. Comput. 26 (2), 19–33. July 2015 26.

Conrad, S., Dorn, G., Craiger, P., 2010. In: Choi, K., Shenoi, S. (Eds.), Forensic Analysis of a Playstation 3 Console, Advanced in Digital Forensics VI. Springer, NY ch. 5, 2010.

Copyright. Designs and Patents Act 1988 (Chapter 48), UK. Available online: https://www.legislation.gov.uk/ukpga/1988/48/section/296ZD. (Accessed 3 February 2019).

Davies, M., Read, H., Xynos, K., Sutherland, I., 2015. Forensic analysis of a Sony PlayStation 4: a first look. Digit. Invest. 12 (1), 81–89, 2015.

Decrypt9, 2019. Available Online: Multipurpose Content Dumper and Decryptor for the Nintendo 3DS. https://github.com/d0k3/Decrypt9WIP. (Accessed 24 January 2019).

ESRB, 2019. Entertainment software rating board. Available Online: https://www.esrb.org/. (Accessed 29 January 2019).

Gowrishankar, K., 2016. NAND Dumping 2DS/3DS/3DS XL/N3DS/N3DS XL! GBA-TEMP. https://gbatemp.net/threads/tutorial-noob-friendly-nand-dumping-2ds-3ds-3ds-xl-n3ds-n3ds-xl.414498/.

Hanlon, C., 2012. Quick-thinking girl, 10, traps paedophile by using her games console to take picture of him molesting her. Available Online: https://www.dailymail.co.uk/news/article-2121454/Quick-thinking-girl-10-traps-evil-paedo-phile-games.htm lDaily Mail. (Accessed 28 March 2012).

Harvey, P., 2019. Read, write and edit meta information. Available Online: https://www.sno.phy.queensu.ca/~phil/exiftool/. (Accessed 2 February 2019).

iPhoneWiki, 2019. limera1n. Available online: https://www.theiphonewiki.com/wiki/Limera1n. (Accessed 29 January 2019).

McClintic, M., Maloney, D., Scires, M., Marcano, G., Norman, M., 2018. Keyshuffling attack for persistent early code execution in the Nintendo 3DS secure bootchain. Cornell Univ. arXiv J. Available online at: https://arxiv.org/abs/1802.00092. (Accessed 26 January 2019).

Moore, J., Baggili, I., Marrington, A., Rodrigues, A., 2014. Preliminary forensic analysis of the Xbox one. Digit. Invest. 11, 5765, 2014.

NDS-Gear, 2019. Nintendo 3DS console family. Available online at: http://www.nds-gear.com/nintendo-3ds-console-family/. (Accessed 26 January 2019).

Nintendo (n.d.). What SD cards and microSD cards are compatible? Available Online: https://en-americas-support.nintendo.com/app/answers/detail/a_id/274/~/what-sd-cards-and-microsd-cards-are-compatible%3F. (Accessed 29 January 2019).

Nintendo of America v King, 2010. FC 246, Canada. Available Online: https://decisions.fct-cf.gc.ca/fc-cf/decisions/en/item/223922/index. (Accessed 3 February 2019).

Nintendo, 2018. Dedicated video game sales units. Available Online: https://www.nintendo.co.jp/ir/en/finance/hard_soft/index.html. (Accessed 29 January 2019).

Nintendo of America v King, 2017. FC 246, Canada. Available Online: https://decisions.fct-cf.gc.ca/fc-cf/decisions/en/item/223922/index.do. (Accessed 3 February 2019).

Nintendo Support, 2019. What SD cards and microSD cards are compatible? Available online at: https://en-americas-support.nintendo.com/app/answers/detail/a_id/274/~/what-sd-cards-and-microsd-cards-are-compatible%3F. (Accessed 26 January 2019).

Ntrteam, 2019. ntrboot_flasher - a tool to flash that bootrom-hax goodness to your flashcart. Available onlint at: https://github.com/ntrteam/ntrboot_flasher. (Accessed 2 February 2019).

Outlaw, 2010. Nintendo mod chip seller infringed copyright. rules High Court Available online at: https://www.out-law.com/page-11268. (Accessed 31 January 2019).

PassMark Software, 2019. OSMount. https://www.osforensics.com/tools/mount-disk-images.html.

Plailect, 2019. CTRTransfer guide. Available online at: https://3ds.hacks.guide/ctrtransfer.html. (Accessed 2 April 2019).

r4ids, 2019. R4i Gold 3DS plus. Available online at: http://www.r4ids.cn/r4ids-e.htm. (Accessed 29 January 2019).

Read, H., Thomas, E., Sutherland, I., Xynos, K., Burgess, M., 2016. A forensic methodology for the analysis of a Nintendo 3DS. In: Twelfth Annual IFIP WG 11.9 International Conference on Digital Forensics, New Delhi, India January 4-6,

2016. www.ifip119.org.

Russinovich, M., 2016. Strings v2.53. Available online at: https://docs.microsoft.com/en-us/sysinternals/downloads/strings. (Accessed 2 February 2019).

Scires, M., 2017b. Boot9strap. Available online at: https://github.com/SciresM/boot9strap. (Accessed 26 January 2019).

Scires, M., Mears, M., Maloney, D., Norman, M., Tux, S., Monroe, P., 2018. Attacking the Nintendo 3DS boot ROMs. Cornell Univ. arXiv J. Available online at: https://arxiv.org/abs/1802.00359. (Accessed 26 January 2019).

Wwylele, 2018. 3DS save file extraction tools. Available online at: https://github.com/wwylele/3ds-save-tool. (Accessed 2 February 2019).

Mr. Gus Pessolano BS is currently a GCIP-certified Cyber Security Analyst at the Vermont Electric Power Company. He graduated Summa Cum Laude from Norwich University in 2017 with a B.S. in Computer Security and Information Assurance and a minor in Mathematics. During his last semester he forensically investigated the Nintendo 3DS games console. After graduating, he pursued his interest in digital forensics by analysing android mobile applications. He enjoys hack-a-thons and has competed in several Red Team exercises.

Dr. Huw Read BSc PhD is a Professor at Norwich University in Vermont, USA and the director for the Centre of Advanced Computing and Digital Forensics (NUCAC-DF). Dr. Read began his academic career in 2004 at the University of South Wales (UK) and has taught a number of specialist courses in digital forensics and cyber security. For over 15 years he has worked alongside industry and government on a number of cyber-related projects, partnering with diverse teams to design solutions to complex security problems. Dr. Read is actively engaged within the field, having published a number of peer-reviewed research articles and attracting grant funding for research and scholarship.

Professor Dr. Iain Sutherland BSc MSc PhD MBCS is currently Professor of Digital Forensics at Noroff University College in Kristiansand, Norway. He is a recognised expert in the area of computer forensics and data recovery. He has authored numerous articles ranging from forensics practice and procedure to network security. In addition to being actively involved in research, he has acted as a consultant on forensic and security issues for both UK police forces and commercial organisations. His current research interests lie in the areas of computer forensics and computer security.

Professor Dr. Konstantinos Xynos BSc MSc PhD has a strong interest in embedded devices, IoT and games consoles. Not only does he observe a device's security aspects but also the potential forensic value. He continues to pursue an active research role investigating hardware and software challenges that encompass these devices and their technological advances.