Contents lists available at ScienceDirect

Forensic Science International: Digital Investigation

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

DFRWS EU 2024 - Selected Papers from the 11th Annual Digital Forensics Research Conference Europe

Well Played, Suspect! – Forensic examination of the handheld gaming console “Steam Deck”

Maximilian Eichhorn [*], Janine Schneider, Gaston Pugliese

Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

ARTICLE INFO

Keywords:
Digital forensics
Gaming console forensics
Steam Deck
Steam client
SteamOS

ABSTRACT

The video game industry has been experiencing consistent growth, accompanied by an increase in the number of players. After the remarkable success of the Nintendo Switch, it comes as no surprise that various other manufacturers have ventured into developing their own handheld gaming consoles. As a consequence, it is likely that these types of devices will be found more frequently in households in the near future and that they will start to play a more important role in forensic investigations. In light of this, we conducted a forensic examination of Valve’s recent Steam Deck console to assist forensic investigators in retrieving and interpreting digital evidence obtained from such devices. The Steam Deck console runs on SteamOS and ships with a custom version of Valve’s highly popular Steam gaming platform. Our examination encompasses exploring the console’s architecture, the SteamOS operating system, and the pre-installed cross-platform Steam client. Using differential forensic analysis, we systematically identify forensically relevant artifacts on the handheld console and report on their locations and contents. Based on our findings, we developed Autopsy plugins for the automated extraction of forensic artifacts from images taken of Steam Deck devices.

1. Introduction

The number of video game players has been growing worldwide for years (DFC Intelligence, 2022) and experienced a further upswing during the COVID-19 pandemic (Chen et al., 2016). The ability to meet online with family and friends to play games together was an attractive alternative to meeting in the real world, especially during lockdowns. But apart from the COVID-19 pandemic, the video game industry is growing steadily and has now even overtaken the film and music industries in terms of sales figures and profits (Parreno, 2022).

Valve’s Steam platform is one of the most successful online platforms for playing and buying video games. Steam was initially launched to provide updates for Valve games but has become the most important digital game distribution platform (Chen et al., 2016). Apparently, Steam has become so popular that even Tesla announced to integrate Steam into their Model S and Model X vehicles (Tesla, 2022). In 2023, Steam set a new record with more than 32 million concurrent active users, of which over 10 million were playing games at the same time (SteamDB, 2023). Besides purchasing digital copies of video games, the platform offers several community features and thereby includes a social facet as well. For instance, Steam allows users to chat with friends via text or voice, invite friends to play, review games, or interact in community hubs.

In addition to its gaming platform, Valve also offers Steam-specific hardware, such as the Steam Controller (Valve, 2015a) or the Steam Link (Valve, 2015b). In the past, Valve has collaborated with HTC, with whom they have developed the VR headset HTC Vive. In 2022, Valve launched its first mobile gaming console, the Steam Deck (Valve, 2022a). It ships with custom versions of SteamOS and the Steam client. Although having a built-in touchscreen as well as analog buttons and pads for being used as a standalone gaming console primarily, the Steam Deck can be connected to an external monitor or TV and be used as a desktop computer as well.

Generally speaking, game consoles are becoming increasingly popular, especially handheld devices. The Nintendo Switch, for instance, has sold more than 128 million units since its release (VGChartz, 2023). For 2023, several new gaming handhelds from multiple vendors have been announced that will run on Windows and Android, such as the Asus ROG Ally, Logitech G Cloud, and Razer Edge. Also, Sony announced a handheld remote player called PlayStation Portal, which will be able to stream PlayStation 5 games.

Consequently, the already existing substantial and still growing user

[*] Corresponding author.
E-mail addresses: maximilian.eichhorn@fau.de (M. Eichhorn), janine.schneider@fau.de (J. Schneider), gaston.pugliese@fau.de (G. Pugliese).

https://doi.org/10.1016/j.fsidi.2023.301688

base of gaming platforms, as well as the continuing advent of new handheld consoles, render both of them increasingly compelling for digital forensics. Since it is likely that handheld consoles will be found more frequently in households in the near future, it is advisable for forensic investigators to learn how to extract and interpret digital evidence acquired from these electronic consumer devices.

In this paper, and for the first time, we therefore examine the handheld gaming console Steam Deck by Valve forensically. Valve's prominent market position, the widespread use of the proprietary Steam client, as well as the not yet adequately investigated SteamOS operating system, which the handheld gaming console is running on, make the Steam Deck a reasonable target for a thorough forensic analysis.

1.1. *Contributions*

Our goal was to find answers to the following fundamental research questions regarding the Steam Deck as a potential subject of investigation:

1. What kind of digital evidence can be obtained from the Steam Deck which may be of forensic relevance?
2. Where can the corresponding local artifacts be found on the Steam Deck's internal storage?

The contributions of this paper which have been made in the course of answering these questions are as follows:

- To the best of our knowledge, we are the first to examine the Steam Deck gaming console forensically.
- We performed a differential forensic analysis of images taken of the Steam Deck's internal storage to identify local artifacts systematically. Thereby, we localized several artifacts of SteamOS and of the proprietary Steam client, which are related to essential actions and events of both the user and the system.
- Based on our findings, we developed "Steam Deck Analyzer", a collection of Autopsy plugins to automatically extract local artifacts from Steam Deck images. The source code of the plugins is available at https://github.com/gpgls/autopsy-steam-deck.

1.2. *Outline*

This paper is structured as follows: First, we give an overview of related work in Section 2, and describe the fundamentals of the Steam Deck and its components in Section 3. Afterwards, we present the methodology of our forensic analysis in Section 4, which is followed by the respective results in Section 5. Finally, we discuss our findings in Section 6 before concluding the paper in Section 7.

2. Related Work

Read et al. (2016) analyzed the Nintendo 3DS and found several artifacts, such as friend lists and game notes, but also web browser, camera, and activity log artifacts. They proposed a methodology for live examinations of the Nintendo 3DS and stated that issues with the built-in encrypted NAND storage can be mitigated through live analysis.

Pessolano et al. (2019) used open-source security circumvention techniques to create a physical dump of the internal NAND storage of the Nintendo 3DS without making significant changes to the hardware. Through these techniques, they managed to extract system activity, deleted images, Internet history items, friends list information, the console's serial number, and plaintext Wi-Fi passwords.

Barr-Smith et al. (2021) presented an extraction process of forensic evidence from the Nintendo Switch gaming console. They were able to create a dump of the internal NAND storage of the device and discovered several key artifacts like personal information, the network connection history, and displays the device has been connected to. Furthermore, the authors developed a software tool to automatically dump and extract the content of the device's NAND storage and Autopsy plugins for the automation of the analysis process.

Khanji et al. (2016) described the challenges emerging from the non-availability of forensic tools for gaming consoles and developed a framework for examining gaming consoles like the Xbox One or PlayStation 4 in response.

Tabuyo-Benito et al. (2019) analyzed the chat artifacts of the Steam game "Counter Strike Nexon Zombies". For this purpose, the authors created two fictional cases where they analyzed the network traffic, the volatile memory, and the hard disk to gain insights about the communication within the video game as well as through a YouTube live stream.

Ebrahimi and Chen (2014) investigated how video games can be exploited to transfer information. The authors raised awareness that video game forensics is still in its infancy and, therefore, can be used by criminals to hide, display, and transfer data in video games without being noticed during an investigation. They also offered recommendations on how investigators should search for hidden data in games.

Chen et al. (2016) examined security and privacy issues of four popular computer video games and of the PlayStation 4 gaming console which arise from the exploitation of player customizations. The authors showed that each of their research objects had at least one exploitable feature which can be used to secretly transmit information while being difficult to detect by current forensic tools.

3. Background

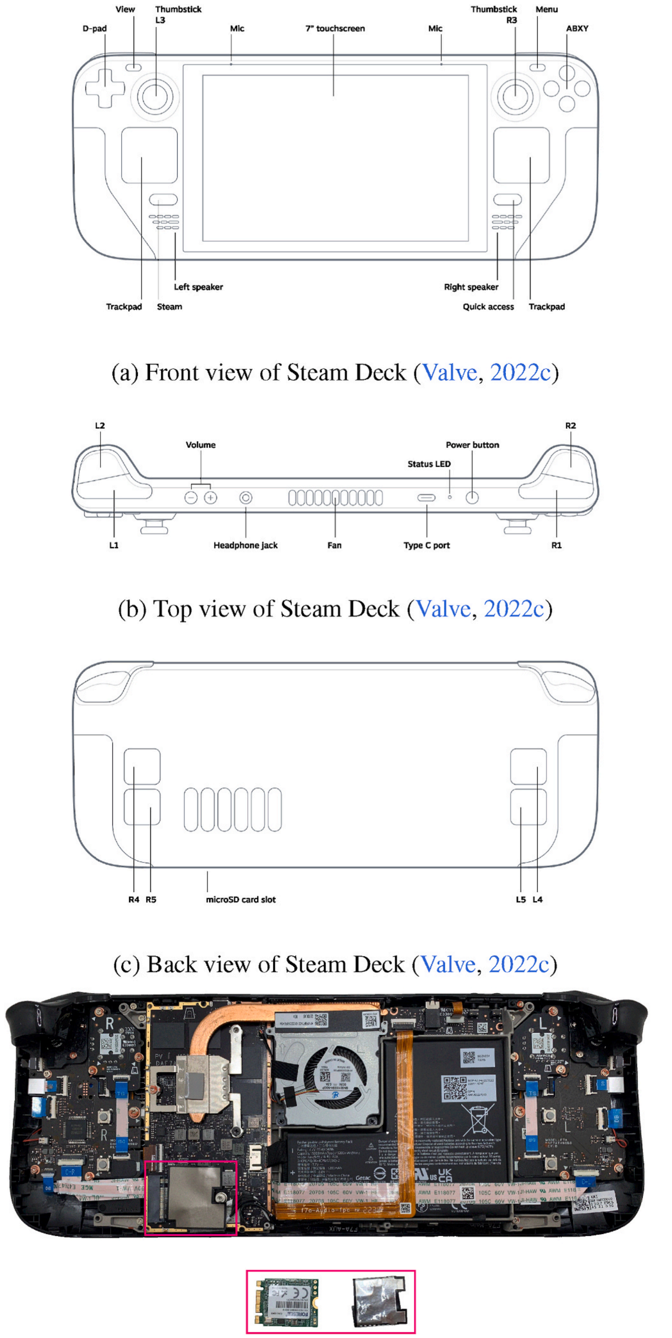
Valve is a video game developer, publisher, and digital distributor founded in 1996 (Dempsey, 2022; Le, 2023): After the success of "Half-Life", their first game in 1998, Valve launched the Steam client in 2003 and thus became its own publisher. In 2015, Valve entered the hardware market with the Steam Machine and remained active in that area ever since. In 2022, Valve released their first handheld gaming console named Steam Deck, which runs SteamOS 3.0 and the Steam client.

3.1. *Steam Deck*

The Steam Deck is offered in three variants with varying internal storage (Valve, 2022a, 2022c): 64 GB eMMC, 256 GB NVMe SSD, or 512 GB "high-speed" NVMe SSD. The internal storage of the eMMC model is indicated to be PCIe Gen 2 x1, while that of the two NVMe SSD models is indicated to be PCIe Gen3 x4 or x2. All models feature a custom accelerated processing unit (APU) by AMD, which consists of a Zen2 CPU (2.4–3.5 GHz, 4 cores, 8 threads) and a GPU based on 8 RDNA 2 computing units (1.0–1.6 GHz). Further, the Steam Deck has 16 GB of LPDDR5 on-board RAM, a 7" IPS LCD touchscreen (1280 × 800, 60Hz, 400 nits), and supports "socketed" 2230 M.2 modules for internal storage as well as microSD cards for additional storage. Connectivity-wise, the Steam Deck supports Bluetooth 5.0, dual-band Wi-Fi (2.4 GHz, 5 GHz, IEEE 802.11a/b/g/n/ac), and 2x2 MIMO. Its 3.5 mm audio jack can be used to connect headphones or headsets, and its USB C port (3.2 Gen 2y) supports DisplayPort 1.4 and 45W power supply (PD3.0). The Steam Deck has a double microphone array installed on the front top of the console, and while it has an ambient light sensor, additional sensors or a camera are not built in. The modular design of the Steam Deck simplifies repairs, and Valve collaborates with iFixit (2022) to provide instructions and spare parts. Fig. 1 shows the front, top, and back view of the external device components and an internal view revealing the location of the 2230 M.2 eMMC/SSD module.

3.2. *SteamOS*

The Steam Deck runs a Linux-based operating system called SteamOS, which was initially developed for the Steam Machine (Le,



(a) Front view of Steam Deck (Valve, 2022c)

(b) Top view of Steam Deck (Valve, 2022c)

(c) Back view of Steam Deck (Valve, 2022c)

(d) Removed rear casing component and shielded eMMC/SSD (●)

Fig. 1. Device components of the Steam Deck.

2023; Valve Developer Community, 2023), and can be downloaded from the vendor's website (Valve, 2022e). The first two versions of SteamOS, namely 1.0 (2013, "alchemist") and 2.0 (2019, "brewmaster"), were based on Debian 7 and 8, respectively, and used the GNOME desktop environment (Steam Community, 2013; Valve, 2019). The version of SteamOS shipping with the Steam Deck is a preview version of SteamOS 3.0 called "holo" and based on Arch using the KDE Plasma 5 desktop environment.

When using the term SteamOS, we refer to SteamOS 3.0 (holo). Currently, SteamOS can not be downloaded as an ISO image; only a specific recovery image is offered by Valve (2019, 2022b,d,e). Although there are community-driven ISO images available (which differ in structure from the official SteamOS), SteamOS heavily depends on the used hardware and can therefore only be installed on specific computer setups. Furthermore, to the best of our knowledge, all currently available SteamOS documentations are unofficial.

SteamOS can be used in two modes: the gaming mode and the desktop mode. By default, SteamOS automatically starts into the gaming mode, providing a graphical user interface (GUI) optimized for gaming. The desktop mode provides a conventional KDE Plasma desktop so that the Steam Deck can be used as a conventional computer. Since SteamOS is based on Linux, Proton is used to support Windows-based games without native Linux ports. Currently, SteamOS does not support any type of native data encryption.

3.3. Steam Client

Steam was initially launched as a software client to provide automatic updates for Valve games (Wilde and Sayer, 2022): In 2004, the first game (Half-Life 2) was digitally offered on Steam, which required users to install the Steam client. In late 2005, Steam became a third-party distribution platform through contracts with other game developers. To integrate Steam into games and third-party software, Valve released the Steamworks API and SDK in 2008 (Caron, 2008). By now, the proprietary Steam client is available for Windows, Linux, macOS, Android, and iOS.

A key purpose of Steam is the publishing and distribution of games and software via its digital storefront, the Steam Store. After a purchase, the respective game or software is added to a virtual library. Steam supports several methods for direct payment, but also enables users to add funds to their Steam Wallet via prepaid vouchers, such as Steam Gift Cards or paysafecards.

Main features of Steam are, inter alia, saving game progress in the Steam Cloud, adding friends and joining their games, customizing of profile pages, participating in groups and forums, communicating via text and voice chats, or reviewing and rating games. While playing, users can access certain Steam functions via an in-game overlay, for instance, to take screenshots or use the built-in web browser without exiting the game. Also, Steam users can share games with family members or close friends by enabling library access through authorized machines or share their game modifications with the community via Steam Workshop.

To protect Steam accounts, Steam offers the so-called Steam Guard as an optional second factor for secure authentication. Either via email or via Steam's mobile app, users receive time-limited codes which they have to enter during authentication processes.

The Big Picture mode optimizes the Steam GUI for high-definition TVs (Valve, 2012) and served as basis for the Steam Deck's gaming mode. For comparison, Fig. 2 shows the GUIs of the Steam desktop client and of Steam Deck's gaming mode.

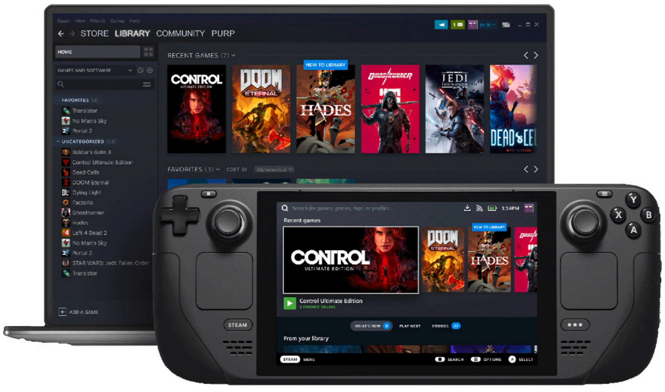


Fig. 2. Steam GUI on desktop client and Steam Deck (Valve, 2022a).

4. Methodology

In this section, we describe how we examined an initial launch version (Steam Community, 2023) of the Steam Deck device (64 GB eMMC model) running the latest version of SteamOS at the time of analysis, namely 3.3.2 (build 20221005.1, kernel 5.13.0-valve21.3-1-neptune).

4.1. Baseline Image

After unpacking the sealed Steam Deck device, we removed the rear casing component to extract the built-in eMMC storage before starting the device for the first time. As advertised by Valve (2022a), our teardown revealed a socketed 2230 M.2 module (FORESEE E2M2 064G FE2H0M064G-B5X10). As a baseline for further analysis, we obtained a bitwise copy of the eMMC module in factory state using an M.2-to-USB adapter for PCIe-based SSDs and a Tableau Forensic Imager TX1.

4.2. Creation of Test Data

Since using the Steam Deck requires logging in with a Steam account, and since a Steam friend is needed for examining the artifacts of certain actions, we first created two new Steam accounts, namely `janedoe226` and `johndoe2261`. The Steam account `janedoe226` is always logged in on the Steam Deck, while the account `johndoe2261` is logged in on a separate computer using the Steam desktop client.

To create test data, we defined sets of actions, each consisting of one or more activities which are likely to create relevant artifacts. Our collection of action sets, denoted as $\Sigma = \{\sigma_0, \ldots, \sigma_m\}$, covers common usage actions that can be performed through the GUI of the Steam Deck. A summary of all action sets is shown in Table 1.

Although we did not perform any actions before creating the baseline image in factory state (Section 4.1), we denote the corresponding action set (of doing nothing) as σ_0. The other action sets are summarized as follows: Installing initial updates and connecting to Wi-Fi (σ_1), logging in with a Steam account (σ_2), installing the game “Counter-Strike: Global Offensive” (CS:GO)[1] and taking screenshots while playing it (σ_3), uninstalling the game (σ_4), adding a friend after adding a fund[2] of €5 via *paysafecard* as well as €5 via Steam Gift Card to the Steam Wallet (σ_5), using the chat for text messaging (σ_6), using the voice chat (σ_7), deleting a friend (σ_8), and factory resetting the device (σ_9).

Table 1
Collection of action sets $\Sigma = \{\sigma_0, \ldots, \sigma_m\}$ performed to generate test data on the Steam Deck.

Set	Name	Short Description
σ_0	Factory Image	No actions; baseline right after unpacking the Steam Deck, i.e., never powered on (see Section 4.1).
σ_1	Initial Update	Power on device for the first time, connect to Wi-Fi, and install initial update.
σ_2	First Login	Connect to new Wi-Fi, log in with the Steam account janedoe226 for the first time, set Wi-Fi and Steam account to be remembered, reboot, and install updates.
σ_3	Install & Play Game	Enter Steam store, search for “counter strike go”, install “CS: GO”, take two screenshots while playing.
σ_4	Uninstall Game	Browse to library, uninstall “CS:GO”.
σ_5	Add Friend	Add €5 to Steam Wallet via paysafecard, add €5 to Steam Wallet via Steam Gift Card, search for “johndoe2261” in search bar (category “friends”), add johndoe2261 as a friend via friend code (1466794558).
σ_6	Use Chat	Write text messages with johndoe2261 in both gaming and desktop mode of SteamOS.
σ_7	Use VoIP	Open voice chat with johndoe2261, talk for about a minute, and leave the chat.
σ_8	Delete Friend	Remove johndoe2261 as friend.
σ_9	Factory Reset	Reset Steam Deck back to factory state.

4.3. Differential Analysis

The action sets $\Sigma = \{\sigma_0, \ldots, \sigma_9\}$ summarized in Table 1 have been executed in sequential order. Between each action set, the Steam Deck was powered off to obtain a bitwise copy of the eMMC storage (cf. Section 4.1). As a result, we obtained 10 snapshots of the Steam Deck’s eMMC at different points in time. Besides initially exploring the corresponding images manually and exploratively using FTK Imager, The Sleuth Kit (TSK), and Autopsy, we opted for a differential forensic analysis (Garfinkel, 2009, 2012; Garfinkel et al., 2012) to identify relevant traces on the file system more systematically.

In our case, while adapting the notation of Dewald (2012, 2015) to describe digital evidence using a finite-state machine system model, let the entirety of snapshots taken of the Steam Deck’s eMMC be a system $\mathscr{S}$ which consists of temporally successive states $s_0, \ldots, s_n$ as shown in Fig. 3. Further, let the entirety of state transitions between each two consecutive states be defined by the collection of action sets $\Sigma = \{\sigma_0, \ldots, \sigma_m\}$, where $m = |\mathscr{S}| - 1 = n - 1$. The time when a state $s_i \in \mathscr{S}$ was captured is denoted as t_{s_i} with $\forall i \in \{1, \ldots, n\} : t_{s_i} > t_{s_{i-1}}$. The observable differences between two states s_{k+1} and s_k are denoted as $\Delta(s_{k+1}, s_k)$.

The differential forensic analysis was carried out using `idifference2.py` (Garfinkel, 2023) to determine the differences between two given file system images; more specifically (Garfinkel, 2012): files that have been (i) created, (ii) deleted, or (iii) modified content-wisely without a change to the modification timestamp, as well as files whose (iv) modification timestamp changed without changes to the file contents. We calculated the differences $\Delta(s_{k+1}, s_k)$ for $s_k \in \mathscr{S}$ (Fig. 3), and parsed the resulting XML reports of `idifference2.py` to group the aforementioned types of file changes per file path.

The state transitions between each two consecutive snapshots of the Steam Deck’s eMMC yielded between 150 and 104,456 distinct file paths which have been affected by changes. By compiling a filter list to exclude files and directories which are perceptibly not associated with

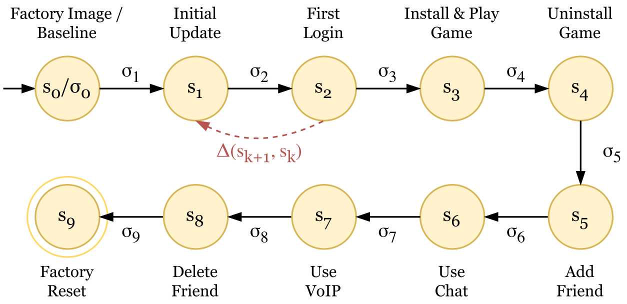


Fig. 3. States $s_0, \ldots, s_9$ of Steam Deck snapshots, including state transitions induced by action sets $\sigma_0, \ldots, \sigma_9$ and the pairwise differences between two states $\Delta(s_{k+1}, s_k)$.

[1] We selected “CS:GO” as exemplary game, because it is the most played game on Steam by the hourly average number of players (GitHyp, 2023).

[2] Mandatory for adding friends due to restrictions posed on limited user accounts aiming to prevent “spamming, phishing, and other abuse” (Valve, 2023c).

our performed action sets in Table 1, we reduced the number of file paths to be examined manually for identifying forensic artifacts by 33.5–99.43% to 89–719 file paths per action set.

5. Results

In this section, we report the results of our examination, addressing the partition layout of Steam Deck's eMMC and the artifacts identified on the file system. Finally, we present the Autopsy plugins we developed based on our findings.

5.1. *Partition Layout*

The partition layout consists of 8 partitions, namely `esp`, `efi-A`, `efi-B`, `rootfs-A`, `rootfs-B`, `var-A`, `var-B`, and `home` (see Table 2). SteamOS uses a custom RAUC update client (Li, 2023) to have a redundant set of the partitions `efi`, `rootfs`, and `var` available for performing robust updates. Instead of running updates on an active partition, new updates are installed on the redundant but inactive partition, and the bootloader controls which partition is booted by switching between them (Luebbe and Joerns, 2023). Changes made to the old partition set are cloned, such as configuration files in the `etc` directory. The early boot `esp` partition and the user space `home` partition are shared.

Each EFI partition is formatted as FAT16. Out of the box, both root partitions have been formatted as ext4 in our case. After initial startup and update, the primary root partition was formatted as Btrfs, and the secondary root partition remained formatted as ext4. But after another update, the file system of the secondary root partition also changed to Btrfs. All other partitions have been formatted as ext4. None of the partitions were encrypted.

5.2. *Forensic Artifacts*

We now present the forensic artifacts which we identified using the method described in Section 4. A summary of detailed directory and file paths is shown in Table 3. In the following, we address found traces categorically while omitting the directory paths for better readability.

5.2.1. *Boot Partitions*

The details about SteamOS' A/B system (cf. Section 5.1) are found in the two configuration files `A.conf` and `B.conf`. More specifically, the configuration files reveal which partition set is currently active, the timestamp of the last boot, and a counter for the number of boot processes. An excerpt from the file contents of `A.conf` is shown in Listing 1.

Table 2
Partition layout of the eMMC storage of the Steam Deck (64 GB model) as obtained via `mmls`.

	Slot	Start	End	Length	Desc.
00	Meta	0000000000	0000000000	0000000001	Safety Table
01	–	0000000000	0000002047	0000002048	Unallocated
02	Meta	0000000001	0000000001	0000000001	GPT Header
03	Meta	0000000002	0000000033	0000000032	Partition Table
04	000	0000002048	0000133119	0000131072	esp
05	001	0000133120	0000198655	0000065536	efi-A
06	002	0000198656	0000264191	0000065536	efi-B
07	003	0000264192	0010749951	0010485760	rootfs-A
08	004	0010749952	0021235711	0010485760	rootfs-B
09	005	0021235712	0021759999	0000524288	var-A
10	006	0021760000	0022284287	0000524288	var-B
11	007	0022284288	0120831964	0098547677	home
12	–	0120831965	0120831997	0000000033	Unallocated

Table 3
Overview of relevant traces on the Steam Deck and SteamOS.

Directory/File Name	Examples of Traces
home/deck/.config/	
ktimezonedrc	Timezone
user-dirs.locale	Language
home/deck/.local/share/Steam/	
config/config.vdf	Steam username and ID
config/libraryfolders.vdf	Installed Steam apps (ID)
config/loginusers.vdf	Steam username, last login
config/htmlcache/Cookies	Cookies
config/htmlcache/QuotaManager	Storage quotas of hosts
config/htmlcache/Cache	Cached web resources incl. URLs
logs/appinfo_log.txt	Installs and updates of Steam apps
logs/bootstrap_log.txt	Launch Steam client
logs/content_log.txt	Steam app names & IDs, uninstalls
logs/durationcontrol_log.txt	Begin of playing games
logs/systemmanager.txt	Steam ID, login/shutdown events
logs/timedtrial_log.txt	Online/offline status of user
logs/webhelper.txt	Failed search strings, URL artifacts
logs/*_log.txt	*Further log files*
steamapps/appmanifest_<APPID>.acf	Steam app name and ID
home/deck/.local/share/Steam/package/	
steam_client_[…]_ubuntu12.manifest	Steam client version
home/deck/.local/share/Steam/userdata/	
<FRIENDID>/config/localconfig.vdf	Steam name, Friend ID & friends, installed Steam apps (ID), last played timestamp for apps
<FRIENDID>/760/screenshots.vdf	Metadata of screenshots
<FRIENDID>/760/remote/<APPID>/screenshots/	Screenshot images and thumbnails
home/deck/.steam/	
registry.vdf	Autologin Steam username, Steam app names & IDs
rootfs-[AB]/	
etc/lsb-release	SteamOS codename
usr/share/steamos-efi/steamcl-version	Steam client version
var-[AB]/lib/	
iwd/<SSID>.8021x	WPA-802.1X credentials
iwd/<SSID>.psk	WPA-PSK credentials
NetworkManager/NetworkManager.state	Status of connectivity interfaces
NetworkManager/internal-<UUID>.lease	Local IP address
pacman/local/linux-.*-valve.*	Kernel version (filename)
var-[AB]/lib/overlays/etc/upper/	
.devkit-service-on-us-update	SteamOS codename, build, version
NetworkManager/[…]/<SSID>.nmconnection	WPA-802.1X/-PSK credentials
esp/SteamOS/conf/	
A.conf	Boot count/status of partition set A
B.conf	Boot count/status of partition set B
esp/efi/steamos/	
factory-reset/	Empty directory after factory reset
steamcl-version	Steam client version

```
boot-other: 0     // here, for A: 0=active, 1=inactive
boot-count: 6
boot-time: 20221110143544
comment:[2022-11-10 15:35:44 +0100]bootconf mode: boot-ok
```

Listing 1: Excerpt from `A.conf` revealing an active set of boot partitions, count of boots, and the timestamp of the last boot.

5.2.2. *Device*

The codename of the installed SteamOS version (e.g., "Holo", cf. Section 3.2) can be retrieved from the two files `lsb-release` and `.devkit-service-on-os-update`. The latter file contains additional information about SteamOS, including the build ID (e.g., "20221005.1") and version number (e.g., "3.3.2") previously mentioned in Section 4.

Depending on the active partition set, the kernel version (e.g., "5.13.0-valve21.3-1-neptune", cf. Section 4) can be identified based on the corresponding file located in the `lib/pacman/local` directory on the var-[AB] partition whose name starts with “linux-” and contains the substring “valve”.

The version of the installed Steam client can be found in the steam_client_steamdeck_stable_ubuntu12.manifest file and is indicated by an epoch timestamp (e.g., “1668654410” → “2022-11-17 04:06:50”). Another version number of the Steam client is indicated in the steamcl-version file which, however, was not identical in our test (e.g., “20220615.1”).

The state of connectivity interfaces of the Steam Deck device is stored in the NetworkManager.state file (i.e., networking, wireless, WWAN). Moreover, the device’s local IP addresses are persisted in internal-<UUID>.lease files.

Aside from that, the configured timezone can be determined from the ktimezonedrc file (e.g., “Europe/Paris”), and the locale from the user-dirs.locale file (e.g., “en_US”).

5.2.3. *Wi-Fi*

Artifacts of the two Wi-Fi connections established during our test (σ_1, σ_2) were persisted in the iwd directory. Depending on whether the authentication security is WPA-PSK (pre-shared key, personal) or WPA-802.1X (enterprise), the corresponding artifact file is named <SSID>.psk or <SSID>.8021x, respectively, and contains the SSID and credentials (e.g., PSK passphrase, or 802.1X identity and password). The NetworkManager’s <SSID>.nmconnection files represent another source for obtaining Wi-Fi credentials (Listing 2).

```
[wifi]
ssid=[SSID]
[wifi-security]
psk=[PLAINTEXT_PASSWORD]
```

Listing 2: Excerpt from <SSID>.nmconnection containing WPA-PSK credentials.

5.2.4. *Users*

The Steam account used to log in into the Steam client on SteamOS when setting up the Steam Deck device (σ_2) can be retrieved from the two files loginusers.vdf and config.vdf. While the latter file only reveals the account name and Steam ID (Listing 3), the former file indicates additional information, such as the account’s persona name, that the password was set to be remembered (cf. σ_2), and the timestamp of the last login. Additionally, successful login and shutdown events related to user accounts can be retrieved from the systemmanager.txt file.

```
// excerpt: /deck/.local/share/Steam/config/loginusers.vdf
"users" { "76561199427431261" { [...]
    "AccountName"                 "janedoe226"
    "PersonaName"                 "janedoe226"
    "RememberPassword"            "1"
    "AllowAutoLogin"              "1"
    "MostRecent"                  "1"
    "Timestamp"                   "1669978678"
} }

// excerpt: /deck/.local/share/Steam/config/config.vdf
"Accounts"{ "janedoe226" { "SteamID" "76561199427431261" }}

// excerpt: deck/.local/share/Steam/logs/systemmanager.txt
[2022-11-09 13:54:08] CSystemStatsNewSessionJob
    SteamID: 76561199427431261 SessionAppId: 1675200
[2022-11-09 13:54:08] Stats Session Created
     8659548853633 [...]
[2022-11-09 14:13:04] udev: shutting down
[2022-11-09 14:13:04] udev: shut down complete
```

Listing 3: Excerpts from loginusers.vdf, config.vdf and systemmanager.txt revealing IDs and names of Steam accounts, the timestamps of the last logins, and logs of successful login and shutdown events.

5.2.5. *Games and Apps*

The registry.vdf file provides a list of installed applications, including their corresponding App IDs and an indication which account is set as “AutoLoginUser” (janedoe226 in our case, σ_2). The list also contains pre-installed applications related to SteamOS, such as “Steam Input Configs” (App ID 241100), “Steam Linux Runtime” (App ID 1070560), or “Steam Linux Runtime - Soldier” (App ID 1391110). Unlike for our installed game “CS:GO” (App ID 730; σ_3), the names of the pre-installed applications have not been indicated in registry.vdf, but they can be fetched from a public Steam API (Valve, 2023b). Further, registry.vdf indicates whether the listed applications are currently updating, installed, or running, as well as the language setting of the Steam account. After uninstalling “CS:GO” (σ_4), the game was still listed in registry.vdf but indicated as currently not installed. An exemplary excerpt of registry.vdf is shown in Listing 4.

```
"language"                "english"
"AutoLoginUser"           "janedoe226"
"apps" { [...]
    "730" {
        "name"            "Counter-Strike: Global Offensive"
        "Installed"       "1"
        "Updating"        "0"
        "Running"         "0"
    } [...]
    "1070560" {           // no name, but: Linux Runtime
        "Installed"       "1"
        "Updating"        "0"
        "Running"         "0"
    } [...]
} [...]
```

Listing 4: Excerpt from registry.vdf revealing the Steam account set for auto login and its language setting as well as App IDs and names of installed games and applications and their current status.

For installed games and applications, corresponding appmanifest_<APPID>.acf files (e.g., Listing 5) can be found in deck/.local/share/Steam/steamapps/which indicate, inter alia, the timestamp of the last update alongside the App ID and name as well as the Steam ID of the owner’s account.

```
"AppState" { [...]
    "appid"             "730"
    "name"              "Counter-Strike: Global Offensive"
    "LastUpdated"       "1668088436"
[...] }
```

Listing 5: Excerpt from appmanifest_730.acf revealing the name and ID of the game “CS:GO” as well as the timestamp of its last update.

A further source for obtaining a list of App IDs is the libraryfolders.vdf file which is user-specific and not device-specific. The localconfig.vdf file reveals additional App IDs of pre-installed system applications like the Steam client itself (App ID 7). For games, the localconfig.vdf file indicates the playtime and the timestamp of the last play (Listing 6); this applies even after the game was uninstalled (cf. σ_4). If an installed game or application has cloud capabilities, the quota usage, the sync state, or the last launch and exit of “autocloud” may be indicated as well.

```
[...] "apps" { [...]
    "730" {                         // CS:GO
        "LastPlayed"            "1668091587"
        "autocloud" {
            "lastlaunch"        "1668090985"
            "lastexit"          "1668091587"
        }
        "Playtime2wks"          "22"
        "Playtime"              "22"
    } [...]
}
"LastPlayedTimesSyncTime"     "1668091587" [...]
```

Listing 6: Excerpt from `localconfig.vdf` indicating game and application information like App IDs, timestamps of last plays and cloud synchronizations as well as playtime.

5.2.6. *Screenshots*

The pre-installed application “Steam Screenshots” (App ID 760) is responsible for taking screenshots during games, such as the two we have taken while playing (σ_3) and which have not been deleted after uninstalling the game (σ_4).

Within nested directories named after the App ID of “Steam Screenshots” (i.e.,`760`) and the user’s Friend ID (e.g., “1467165533”) in `deck/.local/share/Steam/userdata` on the `home` partition, the screenshot application stores the individual screenshots and their thumbnails as JPG files using the date and timestamp as file names.

Additionally, the `screenshots.vdf` file contains a list of metadata for existing screenshot files, including the width or height of the images as well as a creation timestamp and the App ID the screenshot is associated with (see Listing 7). Depending on the game, the metadata may also reveal in-game information, such as the map that was played when the screenshot was taken (e.g., “de_cbble” in our case).

```
"Screenshots" { "730" { "0" { [...]
  "filename"      "730/screenshots/20221110154313_1.jpg"
  "thumbnail"     "730/screenshots/thumbnails/
     20221110154313_1.jpg"
  "externalfilename"     "/home/deck/.local/share/
     Steam/steamapps/common/Counter-Strike Global
     Offensive/csgo/screenshots/de_cbble0002.jpg"
  "Location"      "de_cbble"     // played map
  "width"         "1280"
  "height"        "800"
  "gameid"        "730"          // CS:GO
  "creation"      "1668091393"
} [...] } [...] }
```

Listing 7: Excerpt from `screenshots.vdf` revealing a screenshot of the game “CS:GO” (App ID `730`) taken while playing the map “Cobblestone” (`de_cbble`).

5.2.7. *Friends*

The `localconfig.vdf` file (cf. Section 5.2.5) also contains the list of friends of our account. Here, friends are indicated by their Friend IDs with whom they can be added by others, and the file also provides a history of their public names.

In our case, we added the Steam account `johndoe2261` as a friend using their Friend ID “1467165533” (σ_5). As shown in Listing 8, `localconfig.vdf` contains our friend’s public name “johndoe226” and their Friend ID, alongside our account’s own friend ID and public name.

```
"friends" {
  "1467165533" {  // our account "janedoe226"
    "name" "janedoe226"  // public name = account name
    "NameHistory" { "0" "janedoe226" }
  }
  "PersonaName" "janedoe226"  // our account "janedoe226"
  "1466794558" {  // our friend's account "johndoe2261"
    "name" "johndoe226"  // public name != account name
    "NameHistory" { "0" "johndoe226" }
  }
}
```

Listing 8: Excerpt from `localconfig.vdf` showing Friend IDs and public names of our own and our friend’s account.

Although we removed this friend again during our test (σ_8), they remained in the friends list in `localconfig.vdf`. Methodologically conditioned, we cannot exclude the possibility that this was a temporal artifact which may vanish after a reboot or further cloud synchronizations.

5.2.8. *Steam Wallet*

To lift the restrictions for limited user accounts (Valve, 2023c) before adding a friend to our Steam account (σ_5), we added funds to our Steam Wallet: €5 via Steam Gift card (i.e., using a first-party method), and €5 via paysafecard (i.e., using a third-party method). We found no local traces of adding €5 via Steam Gift card to our Steam Wallet, and assume that this is because the redeem code is directly sent from the Steam client to the Steam cloud without detours.

For the third-party method, however, we found local traces. During the redeem process, we were redirected to the paysafecard website, and the Chromium-based WebView in the Steam client apparently created web browsing artifacts. High-level browsing artifacts regarding not further specified interactions with the host `paysafecard.com` can be found in the SQLite databases `Cookies` and `QuotaManager` in the `htmlcache` directory on the `home` partition.

A rather accidental browsing artifact, as it is conditioned by the occurrence of a warning related to the link preloading of a font resource by the paysafecard website, was found in the `webhelper.txt` log file and discloses both the amount and currency of the added fund (see Listing 9).

```
[2022-11-22 13:11:36] WARNING: https://customer.cc.
     at.paysafecard.com/rest/payment/panel?mid=
     [REDACTED]&mtid=[REDACTED]&amount=5.00&currency=EUR&
     language=en:0: The resource https://customer.cc
     .at.paysafecard.com/psccustomer/payment/fonts/
     roboto-v15-latin-300.woff2 was preloaded using
     link preload but not used within a few seconds
     from the window's load event. [...]
```

Listing 9: Excerpt from `webhelper.txt` showing a warning-related artifact of the added fund of €5 via paysafecard.com.

Looking for a more reliable artifact, the cache artifacts in the `htmlcache` directory can be parsed, for instance, with ChromeCacheView by Nirsoft. Among others, the URL of the redeem process including the fund amount and currency was found (see Listing 10). Artifacts related to the billing information provided during the redeem process on the paysafecard website (i.e., name and postal address), however, could not be found.

```
https://store.steampowered.com/paypal/launchauth/?
    webbasedpurchasing=1&transid=[REDACTED]&authurl=
    https%3A%2F%2Fcustomer.cc.at.paysafecard.com%2
    Fpsccustomer%2FGetCustomerPanelServlet%3Fmid%3D
    [REDACTED]%26mtid%3D[REDACTED]%26amount%3D5.00%26
    currency%3DEUR%26language%3Den&s=[REDACTED]
```

Listing 10: Cache artifact of redeeming €5 via paysafecard.

5.2.9. *Communication*

For the text chat (σ_6) and voice chat (σ_7), we found no local artifacts on the Steam Deck concerning the communication contents. The cache in the aforementioned `htmlcache` directory of Steam (cf. Section 5.2.8) only contained URL-based artifacts regarding the chat client GUI, such as "https://steam-chat.com/chat/clientui/".

Considering remote artifacts after our initial generation of test data, we found that Steam offers a log of chat messages on the account data page (Valve, 2023a), as shown in Fig. 4. However, it should be noted that chat logs are only preserved for two weeks, as well as that access requires the user credentials and maybe even a second factor if Steam Guard is enabled to protect the user's account (cf. Section 3.3).

5.2.10. *Logs*

The `Steam` directory on the `home` partition contains several log files which reveal timestamps and contents of various events during device usage. For instance, the startup of the Steam client is logged in the `bootstrap_log.txt` file, timestamps of when games have been started (σ_3) are logged in the `durationcontrol_log.txt` file, and the online status of users is logged in the `timedtrial_log.txt` file (see Listing 11). Names or IDs of corresponding Steam accounts, however, are not indicated for these events.

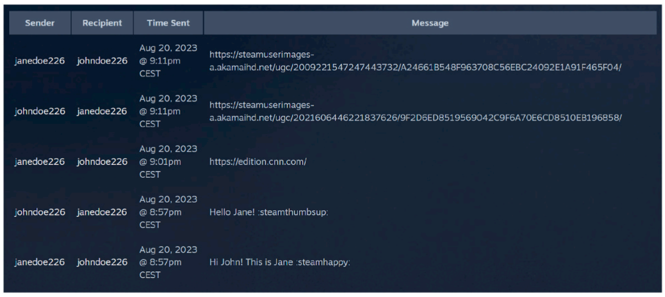
```
excerpt: bootstrap_log.txt
[2022-11-10 15:30:20] Startup - Steam Client
    launched with: '/home/deck/.local/share/Steam/
    ubuntu12_32/steam' '-steamdeck' '-steamos3'
    [...]

excerpt: durationcontrol_log.txt (app 730 = CS:GO)
[2022-11-10 15:05:42] ISteamUser::
    BSetDurationControlOnlineState:
    New state 3 for app 730, starting timers

excerpt: timedtrial_log.txt
[2022-11-10 15:34:50] User is online
[2022-11-10 15:35:11] User is offline
```

Listing 11: Excerpts from log files in `Steam/logs/`directory on the `home` partition regarding user and game activities.

Regarding further game-related activities, the `appinfo_log.txt` file logs which games have been installed or updated, and the `content_log.txt` file reveals the name of installed games as well as initiated uninstallations (see Listing 12).



Sender	Recipient	Time Sent	Message
janedoe226	johndoe226	Aug 20, 2023 @ 9:11pm CEST	https://steamuserimages-a.akamaihd.net/ugc/2009221547247443732/A24661B548F963708C56EBC24092E1A91F465F04/
johndoe226	janedoe226	Aug 20, 2023 @ 9:11pm CEST	https://steamuserimages-a.akamaihd.net/ugc/2021606446221837626/9F2D6ED8519569042C9F6A70E6CD8510EB196858/
janedoe226	johndoe226	Aug 20, 2023 @ 9:01pm CEST	https://edition.cnn.com/
johndoe226	janedoe226	Aug 20, 2023 @ 8:57pm CEST	Hello Jane! :steamthumbsup:
janedoe226	johndoe226	Aug 20, 2023 @ 8:57pm CEST	Hi John! This is Jane :steamhappy:

Fig. 4. Remote source for logs of chat messages (Valve, 2023a).

```
excerpt: appinfo_log.txt
[2022-11-10 14:30:50] RequestAppInfoUpdate: AppIDs 730
[2022-11-10 14:30:50] Requesting 1 apps, 0 packages
    (meta data, 0 prev attempts)
[2022-11-10 14:30:50] UpdatesJob: finished OK, apps
    updated 0 (0 KB), packages updated 0 (0 KB)

excerpt: content_log.txt
[2022-11-10 14:31:33] AppID 730 update
    started : download 0/13984414848, store 0/0,
    reuse 0/0, delta 0/0, stage 0/33039436161
[2022-11-10 14:31:33] AppID 730 state changed :
    Update Required,Update Queued,Update Running,
    Update Started,
[2022-11-10 14:31:33] AppID 730 update changed :
    Running Update,Downloading,Staging,
[...]
[2022-11-10 15:05:01] AppID 730 state changed : Fully
    Installed,
[...]
[2022-11-16 13:15:01] AppID 730 scheduler update :
    Priority Paused, not played for 509314 seconds,
     update disabled for 0 seconds
[2022-11-16 13:15:01] AppID 730 state changed :
    Update Required,Fully Installed,Update Paused,
    (Update delayed for 59153 secs)
[2022-11-16 13:15:01] AppID 730 state changed :
    Update Required,Fully Installed,Files Missing,
    Uninstalling, (Update delayed for 59153 secs)
[2022-11-16 13:15:01] AppID 730 state changed :
    Uninstalled, (Update delayed for 59153 secs)
[2022-11-16 13:15:01] AppID 730 finished uninstall (No
    Error)
```

Listing 12: Excerpts from log files in `Steam/logs/`directory on the `home` partition regarding game-related activities.

While adding a friend (σ_5, cf. Section 5.2.7) before using the Friend ID to add the account, we typed in the name into the search field of the Steam client UI. This activity was logged in the `webhelper.txt` file (see Listing 13).

```
[2022-11-17 13:08:33] WARNING: https://steamloopback
    .host/sp.js:2: Throwing away stale suggestions
    for joh
[2022-11-17 13:08:35] WARNING: https://steamloopback
    .host/sp.js:2: Throwing away stale suggestions
    for johndo
[2022-11-17 13:08:37] WARNING: https://steamloopback
    .host/sp.js:2: Throwing away stale suggestions
    for johndoe2
```

Listing 13: Excerpts from `webhelper.txt` showing artifacts of a search query looking for another Steam user's profile.

5.2.11. *Factory Reset*

After performing a factory reset of the Steam Deck (σ_9), an empty directory `esp/efi/steamos/factory-reset` was created which may serve as an indicator that a reset occurred. Since the two configuration files `A.conf` and `B.conf` (cf. Section 5.2.1) were not affected by the reset, the last boot times and boot counts may serve as further indicators.

Apart from that, all aforementioned sources of forensic artifacts have

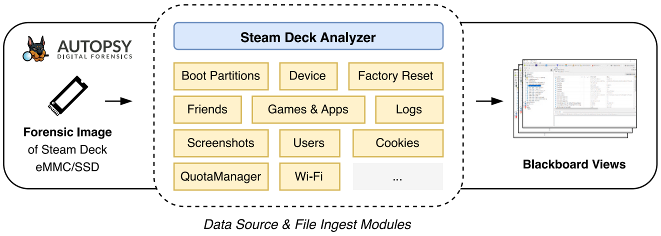


Fig. 5. Overview of the Steam Deck Analyzer plugin collection for Autopsy.

been either deleted or reset to default contents.

5.3. Autopsy Plugin Collection

Based on the forensic artifacts identified in Section 5.2, we implemented a collection of plugins for Autopsy called "Steam Deck Analyzer" (see Fig. 5) to facilitate the extraction of artifacts in practice. On both Windows 10 and 11, we tested all plugins on the two latest versions of Autopsy at the time of writing, namely 4.20 and 4.21.

Due to Autopsy's Jython support, plugins can be written in Python 2.7 and access Autopsy's Java interfaces (BasisTech, 2023). We developed so-called file ingest modules and data source ingest modules. The former type is applied on any file found by Autopsy on a given data source, and the latter type has to iterate over a given data source by itself to locate files of interest. The artifacts extracted by the individual modules from contents of files identified as relevant are then rendered in so-called blackboard views of the Autopsy GUI (i.e., in tabular form with a varying number of predefined columns depending on the individual artifact type).

The plugin collection consists of the following individual modules to extract artifacts categorically: (1) Boot Partitions, (2) Device, (3) Factory Reset, (4) Friends, (5) Games & Apps, (6) Logs, (7) Screenshots, (8) Users, (9) Web: Cookies, (10) Web: QuotaManager, and (11) Wi-Fi.

During development, we applied the plugins on all images we have taken of our Steam Deck device during test data generation (cf. Section 4.2) to verify their functionality. After completing development, we evaluated the applicability of our plugins on a yet unseen Steam Deck image which we generated as follows. Using the Steam Web interface, we re-added our friend to our friends list (σ_5). Since our Steam Deck device has been reset to factory state previously (σ_9), we added a new Wi-Fi connection while setting up the device (σ_1, σ_2), installed the latest updates (σ_1), and logged in with our Steam account (σ_2). Afterwards, we used the funds previously added to our Steam Wallet (σ_5) to buy two paid games from the Steam Store, downloaded another free game, and took several screenshots while playing them (σ_3).

During this practical test of using a yet unseen image of our Steam Deck (SteamOS 3.4.10, kernel version "linux-neptune-5.13.0.valve37-1"), we verified that all intended artifacts have been extracted successfully by our collection of plugins. Moreover, since the first forensic image of our Steam Deck during test data generation (cf. Section 4) was taken in November 2022, our plugins exhibited robustness against minor version updates of SteamOS until September 2023.

6. Discussion

Our method presented in Section 4 enabled us to identify a reasonable but non-exhaustive list of forensic artifacts on the Steam Deck device we examined. Naturally, a different or more atomically defined collection of action sets (cf. Section 4.2), a different approach than differential forensic analysis (cf. Section 4.3), or different versions of SteamOS and the Steam client may reveal varying sets of artifacts.

Attempting to choose action sets for test data generation containing mostly activities with a higher likelihood of creating local artifacts (cf. Table 1), we intentionally focused less on, for instance, community- and cloud-related functionalities of Steam, such as publicly posting game reviews or comments on friends' profiles. Although we have touched on the subject of remote artifacts (cf. Section 5.2.9), it should be explored in more depth (e.g., SteamDB (2022)). Furthermore, due to our methodology, we specifically did not look for well-known Linux-based artifacts or investigate the differences between Arch Linux and SteamOS. Instead, we focused on artifacts created by certain Steam Deck-specific actions.

Resource-wise, the evaluation of the Autopsy plugins that we developed was of rather functional nature (cf. Section 5.3). Even though we consider our implementation a decent baseline, future improvements and enhancements shall be verified on a heterogeneous set consisting of many Steam Deck devices from different users with diverse usage behaviors. To counteract this current limitation in perspective by enabling the community to analyze their own devices, we publish the source code of the plugins.

For future work, we encourage further examinations of certain aspects of the Steam ecosystem from a forensic perspective, such as the cross-platform differences between behavior and artifacts of the Steam client, or the artifacts of peripherals and storage media connected to the Steam Deck.

7. Conclusion

In this paper, we forensically examined the Steam Deck handheld gaming console which runs on SteamOS and ships with the quite widespread Steam client. Using differential forensic analysis, we identified local artifacts on the device systematically based on an iterative procedure of performing a certain set of actions to generate test data, taking a forensic image of the storage module, and determining differences between consecutive images on file system level.

Our examination revealed that a variety of forensically relevant artifacts can be obtained from Steam Deck devices. Among these artifacts are, for instance, Wi-Fi credentials, IDs and names of user accounts and friends, screenshots, ownership and installation state of games, traces of redeemed funds for the Steam Wallet via third-party provider, timestamps of usage and gaming activities through log files, or cache and search query artifacts of the Steam client. Based on our findings, we developed a collection of plugins for Autopsy to automate the extraction of artifacts and to facilitate investigations of Steam Deck devices in practice.

CRediT authorship contribution statement

Maximilian Eichhorn: Conceptualization, Methodology, Validation, Formal Analysis, Investigation, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization, Project administration. **Janine Schneider:** Conceptualization, Methodology, Validation, Formal Analysis, Investigation, Resources, Data Curation, Writing - Original Draft, Writing - Review & Editing Visualization, Supervision, Project administration, Funding acquisition. **Gaston Pugliese:** Conceptualization, Methodology, Software, Validation, Formal Analysis, Investigation, Resources, Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization, Supervision, Project administration, Funding acquisition.

Acknowledgments

We thank the anonymous reviewers for their helpful comments. This work has been supported by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) as part of the Research and Training Group 2475 "Cybercrime and Forensic Computing" (grant number 393 541 319/GRK2475/1-2019), and by the Bavarian Ministry of Science and Arts as part of the project "Security in Everyday Digitization" (ForDaySec).

References

Barr-Smith, F., Farrant, T., Leonard-Lagarde, B., Rigby, D., Rigby, S., Sibley-Calder, F., 2021. Dead Man's Switch: Forensic Autopsy of the Nintendo Switch. Forensic Sci. Int.: Digit. Invest. 36, 301110.

BasisTech, L.L.C., 2023. Autopsy – Python Dev setup. URL: https://sleuthkit.org/autopsy/docs/api-docs/latest/mod_dev_py_page.html. .

Caron, F., 2008. Steamworks SDK now live. URL: https://arstechnica.com/gaming/2008/05/steamworks-sdk-now-live/. .

Chen, L., Shashidhar, N., Rawat, D., Yang, M., Kadlec, C., 2016. Investigating the security and digital forensics of video games and gaming systems: a study of PC games and PS4 console. In: 2016 International Conference on Computing, Networking and Communications. ICNC), pp. 1–5. https://doi.org/10.1109/ICCNC.2016.7440557.

Dempsey, P., 2022. TheTeardown: The Gaming Specialist Has Translated Its Open-Hardware Platform into a Console. Eng. Technol. 17, 1–5. https://doi.org/10.1049/et.2022.0419.

Dewald, A., 2012. Formalisierung digitaler Spuren und ihre Einbettung in die Forensische Informatik. Ph.D. thesis. Friedrich-Alexander-Universität Erlangen-Nürnberg.

Dewald, A., 2015. Characteristic evidence, counter evidence and reconstruction problems in forensic computing. IT Inf. Technol. 57, 339–346.

DFC Intelligence, 2022. Number of PC Gaming Users Worldwide from 2008 to 2024. URL: https://www.statista.com/statistics/420621/.

Ebrahimi, M., Chen, L., 2014. Emerging Cyberworld Attack Cectors: Modification, Customization, Secretive Communications, and Digital Forensics in PC Video Games. In: 2014 Int. Conference on Computing, Networking and Communications (ICNC), pp. 939–944. https://doi.org/10.1109/ICCNC.2014.6785463.

Garfinkel, S., 2009. Automating Disk Forensic Processing with SleuthKit, XML and Python. In: 4th Int. IEEE Workshop on Systematic Approaches to Digital Forensic Engineering, IEEE, pp. 73–84.

Garfinkel, S., 2012. Digital forensics XML and the DFXML toolset. Digit. Invest. 8, 161–174.

Garfinkel, S., 2023. Dfxml – idifference2.py. URL: https://github.com/dfxml-working-group/dfxml_python/blob/main/dfxml/bin/idifference2.py.

Garfinkel, S., Nelson, A.J., Young, J., 2012. A general strategy for differential forensic analysis. Digit. Invest. 9, 50–59.

GitHyp, 2023. Most played games on Steam in 2022 by hourly average number of players. URL: https://www.statista.com/statistics/656319/.

iFixit. Steam Deck – Replacement Guides. URL: https://www.ifixit.com/Device/Steam_Deck.

Khanji, S., Jabir, R., Iqbal, F., Marrington, A., 2016. Forensic analysis of Xbox one and PlayStation 4 gaming consoles. In: 2016 IEEE International Workshop on Information Forensics and Security (WIFS), IEEE, pp. 1–6.

Le, L., 2023. A Complete History of Valve: Founding, Hits, and Failures. URL: https://history-computer.com/a-complete-history-of-valve-founding-hits-and-failures/.

Li, D., 2023. SteamOS Teardown. URL: https://github.com/randombk/steamos-teardown.

Luebbe, J., Joerns, E., 2023. RAUC Documentation – Updating your Embedded Device. URL: https://rauc.readthedocs.io/en/latest/updating.html.

Parreno, R., 2022. Gaming Is Five Times Bigger than Movies Now. https://gameranx.com/updates/id/416500/.

Pessolano, G., Read, H.O.L., Sutherland, I., Xynos, K., 2019. Forensic Analysis of the Nintendo 3DS NAND. Digit. Invest. 29, 61–70.

Read, H., Thomas, E., Sutherland, I., Xynos, K., Burgess, M., 2016. A Forensic Methodology for Analyzing Nintendo 3DS Devices. In: IFIP International Conference on Digital Forensics. Springer, pp. 127–143.

Steam Community, 2013. SteamOS FAQ. URL: https://steamcommunity.com/groups/steamuniverse/discussions/1/648814395741989999/. .

Steam Community, 2023. Recognize new revision. URL: https://steamcommunity.com/app/1675200/discussions/0/3815166007135516910/. .

SteamDB, 2022. SteamTracking-GDPR – account data. URL: https://github.com/SteamDatabase/SteamTracking-GDPR/blob/master/steam_accountdata.md. .

SteamDB, 2023. Steam has reached 10 million concurrent in-game players for the first time, as well as 32 million concurrently online users today. URL: https://twitter.com/SteamDB/status/1611821827948531714. .

Tabuyo-Benito, R., Bahsi, H., Peris-Lopez, P., 2019. Forensics Analysis of an On-line Game over Steam Platform. In: Digital Forensics and Cyber Crime: 10th International EAI Conference, ICDF2C 2018, New Orleans, LA, USA, September 10–12, 2018, Proceedings 10. Springer, pp. 106–127.

Tesla, 2022. Steam is here—bringing thousands of games to new Model S & X vehicles. URL: https://twitter.com/Tesla/status/1602789357156536321. .

Valve. Steam Big Picture Available Now. URL: https://store.steampowered.com/oldnews/9495.

Valve, 2015a. Steam Controller On Steam. URL: https://store.steampowered.com/app/353370/Steam_Controller/. .

Valve, 2015b. Steam Link On Steam. URL: https://store.steampowered.com/app/353380/Steam_Link/. .

Valve, 2019. Steam OS – version repository. URL: https://repo.steampowered.com/steamos/dists/. .

Valve, 2022a. Steam Deck. URL. https://store.steampowered.com/steamdeck.

Valve, 2022b. Steam Deck – SteamOS Holo Recovery. URL: https://steamdeck-images.steamos.cloud/recovery/.

Valve, 2022c. Steam Deck – Tech Specs. URL: https://www.steamdeck.com/en/tech.

Valve, 2022d. Steam Deck Recovery Instructions. URL: https://help.steampowered.com/en/faqs/view/1B71-EDF2-EB6D-2BB3. .

Valve, 2022e. Steam OS – Build your own Steam Machine. URL: https://store.steampowered.com/steamos/buildyourown. .

Valve, 2023a. Steam – accountdata/GetFriendMessagesLog. URL: https://help.steampowered.com/en/accountdata/GetFriendMessagesLog. .

Valve, 2023b. Steam API – ISteamApps/GetAppList. URL: https://api.steampowered.com/ISteamApps/GetAppList/v2. .

Valve, 2023c. Steam Support - Limited User Accounts. URL: https://help.steampowered.com/en/faqs/view/71D3-35C2-AD96-AA3A. .

Valve Developer Community, 2023. SteamOS. URL: https://developer.valvesoftware.com/wiki/SteamOS. .

VGChartz, 2023. Video game console sales worldwide for products total lifespan. as of September 2023. URL: https://www.statista.com/statistics/268966/.

Wilde, T., Sayer, M., 2022. The 19-year evolution of steam. URL: https://www.pcgamer.com/steam-versions/. .