

TALLINN UNIVERSITY OF TECHNOLOGY

School of Information Technology

Department of Software Science

ITC70LT

Raquel Tabuyo Benito 165626IVCM

FORENSIC ANALYSIS OF A STEAM BASED ONLINE GAME

Master's thesis

Hayretdin Bahsi

PhD

Senior Research Scientist

Tallinn 2018

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Tarkvarateaduse instituut

ITC70LT

Raquel Tabuyo Benito 165626IVCM

STEAMI INTERTIMÄNGU KOHTUEKSPERTIISILINE ANALÜÜS

Magistritöö

Hayretdin Bahsi

PhD

Vanemteadur

Tallinn 2018

Author's declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Raquel Tabuyo Benito

14.02.2018

Abstract

Currently, online gaming is a severe threat to the forensic community, as criminals started using it as a communication channel instead of the usual ones like WhatsApp or Facebook. In this thesis, we described a methodology built up after conducting an in-depth forensic analysis of the central artifacts of one of the most used video-games nowadays. We considered as valuable artifacts the ones who are related to the chatting feature of the game under study. Our research was developed by means of a network, volatile and disk analysis of Counter Strike Nexon Zombies video-game, a game that runs under Steam platform. We considered two well-stated cases of study which cover all the game's chat characteristics: chat inside and outside of the in-game rounds and the live chat done through YouTube Live Streaming.

Although it was not part of the scope of this thesis, we found a vulnerability (session hijacking) when analyzing the network, we reported to the developers as soon as it was found and they allowed us to include how we did it in our thesis.

The methodology developed after the analysis of the network capture, live acquisition, and post-mortem acquisition provides to the forensic community a complete guideline that can be used when dealing with a real criminal case in which there is a video-game involved with similar characteristics than this one.

This thesis is written in English and is 96 pages long, including 6 chapters, 57 figures and 5 tables.

Annotatsioon

Hetke seisuga on internetimängud tõsiseks ohuks kohtuekspertiisilisele kogukonnale, kuna kriminaalid on asunud kasutama internetimänge kommunikatsiooni vahendina, selle asemel, et kasutada tavalisi vahendeid nagu WhatsApp või Facebook.

Selles lõputöös me kirjeldasime metoodikat, mis on üles ehitatud tänapäeval kõige rohkem kasutatavatel videomängudel, kasutades põhjalikku kohtuekspertiisi. Me pidasime väärtuslikeks esemeteks neid funktsioone, mis on seotud jututoa funktsioonidega luubi all olevates mängudes.

Meie uuringud töötati välja Counter Strike Nexon Zombies videomängu võrgu, lenduva ja ketasanalüüsi abil, võttes arvesse kahte hästi väljakujunenud õpikeskkonda, mis katavad kõik mängu jututoa omadused: vestlus nii mängu sees kui ka väljaspool ja YouTube Live'i kaudu tehtud otseülekanded.

Kuigi see ei olnud otseselt lõputöö väite üks osadest, me siiski leidsime haavatava koha (seansi kaaperdamine) võrgustiku analüüsimise käigus, me raporteerisime selle arendajatele ning nemad lubasid meil seda fakti kasutada lõputöös.

Metoodika arenes peale võrgustiku analüüsi, elenus omandamine, ja surmajärgne omandamine, mis annab kohtuekspertiisi kogukonnale täieliku juhendi, mida saab kasutada tõelise kriminaalasja puhul, milles on tegemist videomängudega.

Käesolev lõputöö on kirjutatud inglise keeles ja see on 96 lehekülge pikk, kaasates 6 peatükki, 57 joonist ja 5 tabelit.

Table of abbreviations and terms

UNODC	United Nations Office on Drugs and Crime
PC	Personal Computer
OS	Operative System
CSNZ	Counter Strike Nexon Zombies
MFT	Master File Table
IM	Instant Message
RAM	Random Access Memory
PS	PlayStation
IMEI	International Mobile Station Equipment Entity
UID	Unique Identifier
NTFS	New Technology File System
ID	Identifier
CSRF	Cross-Site Request Forgery
RTMP	Real-Time Messaging Protocol

Table of contents

1.	Introduction.....	16
1.1.	Motivation.....	16
1.2.	Scope.....	18
1.2.1.	Contributions	19
1.2.2.	Limitations	19
1.3.	Chapters Summary.....	20
2.	Background information	21
2.1.	Theoretical and technical background	21
2.1.1.	Network forensics	21
2.1.2.	Post-mortem acquisition	21
2.1.3.	Live acquisition.....	22
2.1.4.	Windows forensics.....	22
2.1.5.	Data carving	24
2.1.6.	Steam & Counter Strike	24
2.2.	Literature Review.....	25
3.	Implementation	29
3.1.	Framework	29
3.2.	Forensic processes	30
3.2.1.	Tools used	31
3.3.	Cases of study	34
3.3.1.	Description.....	34
3.3.2.	Building the cases	36
3.3.3.	Workflow diagrams	38
4.	Methodology	41
4.1.	Network forensics analysis	41
4.1.1.	Common Network artifacts.....	41

4.1.2.	Session cloning - Vulnerability found	43
4.1.3.	YouTube Live Streaming specific artifacts	44
4.2.	Volatile memory analysis	46
4.2.1.	Volatile memory analysis with FTK Imager	46
4.2.2.	Volatile memory analysis with Volatility	53
4.3.	Disk analysis	55
4.3.1.	Windows Registry.....	55
4.3.2.	Shortcuts	58
4.3.3.	Prefetch	58
4.3.4.	Jump Lists	59
4.3.5.	Documents folder.....	60
4.3.6.	MFT	62
4.3.7.	CSNZ & Steam dedicated folders.....	62
4.3.8.	Recent files.....	67
4.3.9.	Thumbnails	67
4.3.10.	LogFile.....	68
4.3.11.	Web browsing information	68
4.3.12.	Analysis with Autopsy	70
5.	Analysis Results.....	72
5.1.	Primary artifacts.....	72
5.2.	Tables of Results.....	74
5.3.	Evidences	77
6.	Conclusion and Future Work	78
	References.....	80
	Appendix 1 – Game Screenshots Case 1	84
	Appendix 2 – Game Screenshots Case 2	87
	Appendix 3 – Live acquisition with FTK Imager.....	91

Appendix 4 – Physical disk acquisition with FTK Imager	92
--	----

List of figures

Figure 1. Workflow Case 1	39
Figure 2. Workflow Case 2	40
Figure 3. SteamUserID from network capture.....	42
Figure 4. Forensic User Information after Network analysis	43
Figure 5. YouTube Live Streaming Main server from Network capture	45
Figure 6. YouTube Live Streaming key from Network capture.....	45
Figure 7. YouTube Live Streaming encoder from Network capture	46
Figure 8. Steam UserName after volatile analysis.....	46
Figure 9. Steam NickName after volatile analysis.....	47
Figure 10. Steam Password after volatile analysis.....	47
Figure 11. Steam UserID after volatile analysis	47
Figure 12. CSNZ GameID after volatile analysis.....	48
Figure 13. Associated email after volatile analysis	48
Figure 14. Lobby Chat after volatile analysis.....	49
Figure 15. In-game chat after volatile analysis.....	50
Figure 16. YouTube chat after volatile analysis	50
Figure 17. Online friends after volatile analysis.....	51
Figure 18. Players in the game after volatile analysis	51
Figure 19. Room number after volatile analysis.....	51
Figure 20. Room password after volatile analysis	52
Figure 21. YouTube Channel after volatile analysis	52
Figure 22. YouTube streaming link after volatile analysis.....	52
Figure 23. YouTube streaming key after volatile analysis	52
Figure 24. YouTube streaming encoder after volatile analysis	53
Figure 25. CSNZ acronym after volatile analysis.....	53
Figure 26. GameName after volatile analysis.....	53

Figure 27. CSNZ mode after volatile analysis.....	53
Figure 28. List of processes after volatile analysis	54
Figure 29. Network connections after volatile analysis.....	55
Figure 30. CSNZ (I) in NTUSER.DAT	56
Figure 31. CSNZ (II) in NTUSER.DAT.....	56
Figure 32. CSNZ (I) in SOFTWARE	57
Figure 33. CSNZ (II) in SOFTWARE.....	58
Figure 34. Steam shortcut	58
Figure 35. CSNZ in Prefetch	59
Figure 36. CSNZ in Jump Lists	60
Figure 37. CSNZ in Documents.....	60
Figure 38. Screenshot in Documents	61
Figure 39. Streamed video in Documents.....	61
Figure 40. CSNZ in MFT	62
Figure 41. CSNZ dedicated folder	63
Figure 42. GameID log file.....	63
Figure 43. Nexon log file	64
Figure 44. Configuration file in Steam folder.....	65
Figure 45. Logged users in Steam folder.....	65
Figure 46. Player and Game log	65
Figure 47. Connection file in Steam folder.....	66
Figure 48. SessionIDs database	67
Figure 49. CSNZ in Recent folder	67
Figure 50. CSNZ in Thumbnails cache.....	68
Figure 51. CSNZ in LogFile	68
Figure 52. Cookies in Chrome.....	69
Figure 53. YouTube cache in Chrome.....	69

Figure 54. YouTube History in Chrome	69
Figure 55. YouTube Channel + link with autopsy.....	70
Figure 56. YouTube associated email with autopsy	70
Figure 57. Steam password with Autopsy	71

List of tables

Table 1. Player information under suspicion.	38
Table 2. Primary artifacts.....	72
Table 3. Case 1 Results.....	74
Table 4. Case 2 Results.....	75
Table 5. Evidences	77

1. Introduction

Over the last years, the video-games industry is spreading to new markets and is reaching new kind of players. Even though some people consider that video-game sector is addressed mostly to teenagers or a specific group of people, the reality is apparently the opposite.

Considering the 2017 video-game statistics shows that the average age of players is 35 years old [1]. Besides, the amount of money that video-games move along the last years is incredibly vast, the 2016 revenues in China were \$24,271,294,000 [1]; outstripping the USA by almost a billion dollar [1]. As we mentioned, online gaming is expanding to new sectors, a high percentage of gamers (54%) [1] play with others and find that video-games are useful for communicating with their friends (53%) [1], thanks to the online chat services that these games provide. Therefore, they are considered too as channels of communication, similar to Facebook or WhatsApp. Moreover, when recreating crimes scenes, the usage of video-games is recently an essential tool for forensic experts [2]. Therefore, online gaming is perceived currently not only as a way to enjoy the free time while playing with your friends but also as a way of communication or with other purposes rather than playing. Those purposes could be malicious.

1.1. Motivation

Chat services offered by online video-games are nowadays becoming more and more popular among criminals as they consider them as the safest methods to communicate without being detected [3]. In 2015 after Paris terror attacks, security analysts investigated new communication channels that those terrorists could use, and they stated that they could have used PlayStation 4 as the way to exchange messages without being discovered, as it allows “party chats” [3]. Party chats in online gaming are known as chats separated from gameplay, which means that users do not have to play the game, in fact, to chat with others. In June 2013, the United Nations Office on Drugs and Crime (UNODC) issued a report reviewing cybercriminal’s methods for money laundering, and they stated that online gaming was a key method for that [4]. Furthermore, one of the most significant DDoS attacks of 2016, the *Mirai* botnet, was originated from a video-game called *Minecraft* [5].

Regarding devices for gaming, 56% of gamers prefer PCs rather than others [1], more specifically Windows 7 is the OS with the highest number of desktop and laptop users (Windows 7 has a 44.81% of market share against 28.19% of Windows 10) [6] [63]. The games with a higher number of players reside under the Steam platform, which has a total of 67 million monthly active players and controls between 50-70% of the gaming market [7]. Furthermore, in online gaming, it is trendy nowadays to perform live streaming gaming rounds, as they also allow users to chat and participate in the game. The most demanded tools are YouTube Live Streaming and Twitch. According to, Peter Warman, the CEO of Newzoo (research firm): “*Online video is the biggest thing to hit the games market since the launch of the iPhone in 2007*” [8]. In the research made by this company in 2016, it says that the number of worldwide gamers who watch online gaming content regularly is 470 million and that YouTube is still dominant in this market [8].

This thesis focused on Counter Strike Nexon Zombies (CSNZ) video-game, a game offered by Steam platform. The paid version of CSNZ, *Counter Strike: Global Offensive*, has been already involved in a criminal case, a gambling scandal, as there was an illegal betting market underneath, where a lot of teenagers participated [9]. Besides, it is the second most played game with higher revenues [1] and it is played in Windows PCs. CSNZ is set in a war environment; we think that it could be one of the perfect ways for criminals to communicate as they can hide inside this atmosphere. Moreover, CSNZ has two game modes, one of them is the mentioned war scenario (Zombie mode) and the other one is a scenario similar to Minecraft (Studio mode), a three-dimensional game with the goal of building entire worlds with pixelated blocks. In this kind of games, there are different kind of servers which personalize the user-experience and create a big network, which becomes really attractive to DDoS attackers, as we saw before (Mirai botnet [5]). Additionally, CSNZ allows players to do YouTube Live Streaming while playing the game, a contemplated feature in our thesis too.

Due to all of these facts, trying to obtain as much information as possible from this online game results in a significant contribution to digital forensics community and make a difference in the way forensic experts analyze a system when entering in a crime scene. They pay particular attention to artifacts originated from widely used Social Networks like Facebook [30], or Instant Messaging (IM) tools like Skype [29] or WhatsApp [25] [26] [27]. However, as we have seen, online games, which have an enormous audience

too, are offering this same channel of communication, which criminals can misuse. Consequently, valuable information could be lost if it is not appropriately analyzed or it is not taken into account. As we have seen, some criminal cases like money laundering and DDoS attacks have the usage of video-games as the main factor, but our contribution is focused on how to find artifacts related to the chatting features that online games provide.

Some forensic studies analyze video-games, but they focus primarily on post-mortem state analysis [41] and not in volatile or network traffic analysis.

1.2. Scope

The main purpose of this thesis is to conduct a forensic examination of Counter Strike Nexon Zombies (CSNZ) video-game installed in Windows 7 OS, utilizing different artifacts extracted from the game that are valuable from a forensic point of view, centering in the communication between players and excluding DDOS analysis or money laundering.

The artifacts related with the user, network connections, game traces and time-stamps, chat conversations and credentials are obtained from volatile memory, network traffic and the disk image. The analysis of this game was performed considering two cases, they were chosen based on the two game modes that this game provides (Zombie and Studio modes) and that a potential user could use in order to communicate with others:

- Case 1: two players inside Zombie mode (war scenario). One of them will send an invitation to the game-room to the other player. The chat will be done before entering the gameplay, inside the lobby. After that, both users will enter inside the Zombie scenario and will chat again. While playing, users will use the chat features that include text messages and voice audios.
- Case 2: two players inside Studio mode (“mining” scenario) with a defined password to enter into the game-room. One of them will perform a YouTube Live streaming while playing the game. An external viewer from the streaming video will chat with this player.

Therefore, the goal of my thesis can be defined as conforming a robust forensic analysis of all the artifacts that can be extracted from Counter Strike Nexon Zombies online video-

game and that can be useful and relevant in a forensic case and when presenting the information in court. The reason why we didn't choose to analyze the paid version is that the main difference is in the graphics quality, which is out of the scope of this study.

1.2.1. Contributions

The novelty of our study is that we performed a forensic analysis of one of the most used online video-games nowadays, not only from a post-mortem state, presented in other researches, but also an examination of volatile memory and network traffic. We found chat conversations, passwords, user's game information and time-stamps, as well as the possibility to perform session cloning. Besides, we considered as one of the cases of study the analysis of YouTube Live Streaming, which was not studied in previous academic studies concerning online gaming, and resulted in finding the streamed video with the chat conversation maintained between player and viewer. This will give to forensic experts some guidelines to follow in case they face with this game installed in the investigated system. There is a need to provide this kind of study to forensic community as there are a lot of facts that show that online gaming is being used for more than playing and having fun with friends so, if it is not considered, a way of communication between criminals could be unnoticed by forensic experts.

The forensic procedure developed in this thesis can be transferred to the same category of games and also to all the games that need Steam for log in, as this process is the same for all the games offered by this platform. The steps performed for finding the information can be generalized to online video-games with similar characteristics than the one we studied.

1.2.2. Limitations

The most serious concern of this study is that some data stored in the system and from the network was encrypted. However, we overcame this issue by finding relevant information in volatile memory or configuration Windows system files that helped to build a timeline for the case and usernames and passwords from the disk. Besides, we were able to find some session IDs, after performing the network analysis, that were used for doing session cloning and obtaining the user activity with respect to the game as well as sensitive information.

1.3. Chapters Summary

The thesis consists of six major chapters:

- Chapter 1 presents the motivation and scope of our research.
- Chapter 2 provides a glimpse into the similar academic works as well as some theoretical and technical background related to our study.
- Chapter 3 explains the framework that we have followed to perform the forensic examination, as well as the tools that we used and how we implemented our forensic procedure.
- Chapter 4 contains the methodology that we followed when doing the analysis, explained step-by-step and that could be used by future forensic experts as a guideline.
- Chapter 5 shows the results of the forensic analysis performed in our research.
- Chapter 6 defines our final conclusion and recommendations for future work.

2. Background information

This section contains a definition of some theoretical and technical concepts, in order to make this research paper more comprehensive and to help to its correct understanding. Furthermore, it gives an overview of similar academic studies which served as the basis for our research and reinforces our methodology.

2.1. Theoretical and technical background

Digital forensics involves the actions of acquiring and analyzing digital content (information) that will be used as evidence in court of law [10].

2.1.1. Network forensics

Network forensics refers to capturing, recording and analyzing all the network traffic or network events obtained in a forensic case [11]. According to Simson Garfinkel, network forensics can be performed in two ways [12]:

- **“Catch-it-as-you-can”**: capturing all the packets that passes through a certain host or point and storing them for a later analysis. It usually requires considerable amount of storage in an external device to save the data captured.
- **“Stop, look and listen”**: analyzing each packet in memory. It usually requires less amount of storage but faster processing resources.

As the video-game is played online and we performed a YouTube Live Streaming, network forensics becomes an essential task, as valuable data is exchanged in the network, like session IDs or the link to the video. Regarding the type of network forensics practice, we chose the first one because in this way we did not miss any packet out of our analysis.

2.1.2. Post-mortem acquisition

It refers to acquiring the disk image of the system after shutting it down. With this kind of acquisition, useful forensic data can be extracted, such as the device behavior, modified assets and their associated timestamps [13].

There are two types of acquisition of the disk image:

- **Logical acquisition:** obtaining the logical structure of the system and its related data. When doing logical acquisition, all the files from the system are copied but there are some limitations, because not all data can be extracted [14].
- **Physical acquisition:** obtaining the physical image of a hard disk is a bit-by-bit copy of the system. Therefore, apart from the logical image, it gets all the data from the device too, in other words, it gets located and unallocated space so there are more possibilities to obtain deleted data [14]. The unallocated space refers to the clusters of the disk where files are not stored, however they can contain deleted information from the disk partition which was not removed physically from the device yet [15].

Creating a duplicate copy of the disk by doing a physical post-mortem acquisition allowed us to obtain all the information stored on the hard drive. Therefore, we were able to know which information concerning the online game is stored locally in the system, such as the user information related to the game.

2.1.3. Live acquisition

Live acquisition refers to capturing volatile memory from the system. The data stored in memory won't be available in a post-mortem acquisition. This data can be running processes, event logs, registered services, network information or passwords [16].

As we mentioned previously, one of the limitations that we faced during our analysis is the encryption of the data. Consequently, by doing a live acquisition of the memory, we found relevant data for this study that could not be found when analyzing a post-mortem copy of the device, like the chat conversations.

2.1.4. Windows forensics

Windows forensics is also important for the purpose of our thesis, as it is focused on Windows PCs. Therefore, Windows forensics refers to creating an in-depth forensics knowledge about Microsoft Windows operating systems [17]. Some of the interesting artifacts that should be considered are:

- **Windows registry keys:** According to Microsoft Documentation, *“a hive is a logical group of keys, subkeys, and values in the registry that has a set of supporting files containing backups of its data”* [18]. The standard hives are:

- NTUSER.DAT: it contains the activity of the user.
 - SAM: it contains information about user and group profiling.
 - SOFTWARE: it contains information about the system configuration and devices connected.
 - SYSTEM: it contains information about the system configuration and devices connected.
- **Shortcut:** it is a connection that points to a file in the system [19] and that allows the user to find it quickly and easily. In forensics, it can show the existence of a file or program, even though, it was removed.
 - **Prefetch:** it pre-loads data into memory before it is required by an application. At boots up, it loads portions of programs that are commonly run in the system [20]. It is also an indicator of run programs from a forensic point of view.
 - **Jump Lists:** it is a feature that shows all the pinned files. Besides, it also shows the last visited files with respect to a software [10].
 - **LogFile:** it is a file that has a record of the events or software that was run in the system [21].
 - **MFT:** short of Master File Table. It is an index where all the files in an NTFS volume are stored. In forensics, it is relevant as it contains the file name, some file attributes and pointers where those files are stored [22].
 - **Thumbnails:** reduced-size representation of a picture, video or page that it is used to help in identifying a file by its contents. It is an option used in Windows Explorer so that if we click on a thumbnail, that specific file will be opened [23].
 - **Recent Files:** it is a folder that contains links to the most recent accessed or opened files by an user [10].
 - **Web browsing information:** website information can be stored in form of cookies [10]. It is also important to consider the information about the Internet user activity stored in the browsers such as Internet Explorer, Google Chrome or Firefox.

Considering all the information that can be obtained from Windows Forensics artifacts, the analysis of them is an essential element in our thesis, as there are a lot of artifacts that shows the user activity with respect to the game.

2.1.5. Data carving

Data carving is the process of collecting data from unallocated space using file signatures that identify those files, known as magic numbers. Those signatures are header or footer type-specific of each file. File system structures are not used during the carving process [24]. As we mentioned in the definition of post-mortem acquisition, the unallocated space refers to the clusters of the disk where files are not stored, however they can contain deleted information from the disk partition which was not removed physically from the device yet [15]. Therefore, there could be interesting information, from a forensic point of view, that can be extracted from those clusters and that could indicate the usage of the game by the user. However, we didn't obtain anything relevant after performing the analysis.

2.1.6. Steam & Counter Strike

Counter Strike Nexon Zombies resides under Steam platform. To run the game, first of all, the user must authenticate himself on Steam by running the Steam application installed on his computer. After logging with his credentials (username and password), the user can start the game.

Steam is a digital distribution platform for PC gaming developed by Valve Corporation. It has a variety of different video-games and offers multiplayer gaming, video streaming and social networking services. With Steam the user can install and automatically update video-games in his device. Additionally, Steam includes a list of friends and community groups, the possibility of saving the data in the cloud, chat and in-game voice functionalities [64].

Regarding Counter Strike video-game, it is a multiplayer video-game that belongs to the category of first-person shooters. The topic of this game is that players are divided between two teams: terrorists (commit terror attacks) and counter-terrorists (prevent terror attacks). Counter Strike Nexon Zombies is a free-to-play spin-off of this game with a zombie-themed. The main difference with the paid version is that it has a poor user interface and old graphics [65]. The game allows sending text and audio messages between players.

2.2. Literature Review

Although most of the academic works are about chat messaging applications or programs, it is useful for the purpose of this study the way they analyzed the files. Besides, thanks to those studies, we discovered which files must be considered to do not get lost with the bunch of information that we faced. Moreover, our thesis claimed to go further, because, as we will see in the following paragraphs, the only related work about forensic analysis of online gaming is about a game which is not popular anymore and it was only focused on post-mortem analysis of the disk [41].

Most relevant forensic examinations that have been addressed recently, are related with IM (Instant Message) tools, social networks or web resources like WhatsApp [25] [26] [27], LINE [28], Skype [29], Facebook [30], Telegram [31], Tango [32], ChatSecure [33], or WeChat [34].

WhatsApp is one of the most used IM applications. There are a lot of forensic studies that cover its analysis, but we have considered the one performed by Anglano [25] because he obtained information about user's contacts, settings and preferences and a lot of information about the communication information exchanged, which is important for our thesis. In his research, he found that there was a backup database of the messages that can be easily decrypted since the default encryption key is used in all the devices. Even though it was not possible to see the content of the message, he discovered the status of each message, geolocation coordinates, contact cards, partners of the messages (group/broadcast) and the messages sent and received in the device. Lp Jhala Ky [27] designs a method in which it is possible to obtain, from RAM, the password to decrypt the messages of WhatsApp database of the device. Moreover, in the research performed by Karpisek, et al. [26], they were able to decrypt the network traffic and obtain forensic artifacts related with WhatsApp calls (WhatsApp phone number, WhatsApp server IPs, WhatsApp audio codec, WhatsApp call duration and termination).

LINE is another IM application that allows having secret or hidden conversations. For our study, it is also relevant the study performed by Iqbal, et al. [28] because they were able to retrieve not only all the information about not-hidden conversations but also these secret conversations if the app was force closed before the end of a TimeSet. They partially recovered those hidden conversations, as it is only possible to obtain them if a

specific event happens. The post-mortem forensic analysis performed by Satrya, et al. [35] of private chats in LINE, Telegram and KakaoTalk was also considered in our thesis.

Skype is a software that provides video chat and voice calls, as well as text messages exchanging. The information stored about the messages can be recovered as the database is stored in the system. Besides, in the research made by McQuaid [29], he found the way how to obtain and listen the voicemail and some forensic artifacts stored in unallocated space and volatile memory, which reinforces our idea of analyzing those options.

Facebook is an online social media and one of the most used social networking service. Wong, et al. [30] found that it is possible to obtain chat information, Facebook posts and pictures from RAM and cache browser. Furthermore, in the research done by Yang, et al. [36], they found that inside Windows system files like \$LogFile, \$MFT and \$UsnJrnl, there were indicators about filenames, directory paths and timestamps of the downloaded Facebook files. As we will focus on Windows Forensics, those studies are also relevant to us.

Telegram is another IM application that competes with WhatsApp to have the highest number of users, as they provide similar services. Regarding the Telegram forensic analysis, for us, it could be valuable the online and offline analysis performed by Satrya, et al. [31] as they were able to acquire application and user activity, contact information, messages exchanged (including audio files), shared files, shared location and deleted communication information.

Tango is a software which has similar features to Skype. Although in the research made by Sgaras, et al. [32] they could not access the databases. Besides, they performed a Session cloning for extracting more artifacts related to the messages exchanged.

ChatSecure is an IM application with encryption features. Anglano, et al. [33] developed an algorithm to decrypt the databases based on the passphrase found in volatile memory and they joined some tables to obtain valuable artifacts and correlate them to establish a case. However, in their research, they couldn't recover deleted data.

WeChat is an IM application with encrypted messages. However, Wu, et al. [34] were capable of decrypting the messages based on the IMEI of the phone and UID, which were

stored in some configuration files. This research could provide us some guidelines when dealing with encryption and configuration files.

Barghuthi and Said [37] carried out a network forensic analysis of the most used messages applications, like Skype or Facebook. Although in some of them, the traffic was not encrypted, we can see from the rest of previous studies that one of the standard issues is the encryption, which was solved in most of them by finding information in volatile memory; this case also happens to our research too. Moreover, those academic works under Windows forensics helped us when deciding which files we should analyze to find relevant forensic data.

Considering that the main subject of our thesis is a video-game, the academic works about video-games consoles are also relevant to us, as they propose some methodologies when dealing with the games executed in a system. Davies, et al. [38] offer some steps that must be followed when trying to recover data from PS4. In the study written by Khanji, et al. [39] they propose best manners when analyzing PS4 and XboxOne, in our thesis it is practical to know the best approach when doing a live acquisition or network forensics analysis. Besides, they showed some forensic tools like Autopsy or FTK Imager that help when finding forensic information stored in the evidence. However, the most interesting research with respect to video-games consoles from the purpose of our thesis is the one performed for XboxOne by Moore, et al. [40]. It includes network forensics analysis, data carving, imaging acquisition and analysis with Autopsy (based on keyword search) but they didn't consider the chatting feature of the games. XboxOne has a similar file system concerning Windows 7 (NTFS), which is the OS used in our cases of study. More specifically, in this research, they found that some files in the system were modified at the same time they were playing with the XboxOne, and also that there were created some new records for backups. In the network forensics part of this research, they were able to set a timeline of the case just by considering the traffic generated when playing the game.

According to online gaming, there is a research made in 2010 by Larry E. Daniel [41] about *Everquest II* video-game. He performed a post-mortem analysis and studied which forensic artifacts can be extracted from this video-game. From our point of view, it was pretty trivial as the valuable forensic data was saved in the system as TXT format and it was not encrypted, so it was easily human readable. Besides, this game is not one of the most famous among gamers anymore and have a reduced number of players nowadays,

comparing to Counter Strike. The study we want to perform claims to go further with a more popular game, with encrypted data, and considering volatile and network artifacts; whereas in Daniel's study, he only focused on the post-mortem data saved in the system and consider volatile data neither network artifacts.

Considering all of these studies, our thesis improves them with a more popular game between users and in terms of market revenues. Our thesis provides a broader analysis, not only from a post-mortem acquisition of the system but also with a live acquisition, network forensics, and Windows forensics analysis. Besides, the decryption techniques used in some forensic studies performed in some popular instant messaging applications (based on volatile memory analysis) are taken into account; as well as the files considered in those studies with respect to Windows systems.

3. Implementation

In this chapter, we describe the forensic framework that we followed for the correct development of the experimental part of this thesis. Additionally, the list of tools that we have used is presented and the cases that were under study.

3.1. Framework

The framework that we will follow for dealing with the evidences is the McKemmish forensic framework [42]. This forensic process consists on the following steps:

- 1) **Identification of digital evidence:** being acquainted with what evidence is going to be handled, where and how it is stored; in order to know accurately how to behave during its recovery. In addition, the forensic expert must be aware of the type of information that will be acquired and the format in which it is stored in the device, so that the correct technology will be used during its extraction.
- 2) **Preservation of digital evidence:** it is one of the most important steps during this forensics process. As it is possible that the forensics process is under inspection in a court of law, it is mandatory that the whole process is accomplished in the least intrusive way. There are circumstances where the change of the evidence data is inevitable however, those changes have to be reduced to minimum. In those unavoidable circumstances, it is essential that the nature of those changes in the data are explained clearly, as well as the reasons why those changes occur. The changes refer not only to the ones made to the data itself but also to any physical change that the evidence could be affected by, for example, when accessing a physical device, sometimes it is necessary to perform some physical changes on it in order to access to the data stored.
- 3) **Analysis of digital evidence:** it includes the extraction, processing and interpretation of the data extracted from the device. Sometimes after extracting the data, its processing is required in order to be human readable. If it is not the case, the processing can be skipped.
- 4) **Presentation of digital evidence:** it refers to the presentation of it in a court of law. A good presentation also depends on the expertise and qualification of the

presenter as well as on the integrity of the processed applied to produce the evidence. The primary requirement is that this whole forensic process must be legally acceptable.

3.2. Forensic processes

We want to enable the acquisition of realistic data similar to the one that we would find in real world investigations, consequently, we decided to follow the McKemmish approach conscientiously with all the processes involved in this thesis:

- **Network forensics:** we captured the network traffic during the two cases of study. For the proper acceptance in court of the network evidences, we disconnected the system from any external connection after capturing the traffic, so that we avoided any kind of modification that could let in an invalid evidence in court.
- **Live acquisition:** we performed an acquisition of volatile data (RAM memory dump) but being aware that we should create the least amount of changes in the system under inspection. Besides, the changes are explained and reported in order to ensure a precise preservation of the evidence and admissibility in court.
- **Post-mortem acquisition:** we acquired the hard disk in terms of **physical acquisition** due to the fact that in this way, we obtain a bigger picture of the system as with this type of acquisition, we have a complete copy of the disk (including the unallocated space) and we can see all the information about the game stored in the system locally.
- **Windows forensics:** the OS of the disk acquired is a Windows 7 OS; consequently, we performed an in-depth analysis of it based on Windows forensics manners.

Once we extracted all the data, we continued with the interpretation of the evidences: network artifacts obtained from the previous traffic capture; the analysis of the RAM memory; windows registry; dedicated folder of the game in the system; windows system files and folders (\$MFT, \$LogFile, Prefetch folder, shortcuts, Recent folder, JumpLists, thumbnails and web information).

After interpreting the data, we tried to find enough information in order to build the timeline of the case, when the user connected, with whom the user played, the list of

friends, chat information, connection time-stamps, history of the game and user information like username and passwords. When analyzing the network information, we have considered the session IDs so that the session cloning option has been contemplated.

Regarding the last step of the McKemmish framework, the presentation of digital evidence, we decided to skip it as this thesis is more focused on the analysis of the data and the information that can be retrieved about the game. However, we tried to follow all the best manners when dealing with evidences so that they could be admissible at the court of law.

3.2.1. Tools used

The tools used for the correct development of this thesis are described in this section. We decided to make use of free tools that are frequently used by the forensic expert's community. These tools were obtained from previous academic works and that are popular in forensics. We will classify them based on the forensic processes from which they were used:

Network forensics

During the network forensics analysis, the tools that were used for the capture of the traffic and the analysis of it are:

- Wireshark: it is a packet analyzer tool. It is used for network traffic capturing, analysis and troubleshooting. It is similar to *tcpdump* but Wireshark offers a graphical interface with options of filtering and sorting [43] [44]. Version: 2.2.4
- NetworkMiner: it is a network forensic tool that can be used as a passive network sniffer that detects OS, hostnames, sessions, open ports, among other features from the traffic captured [45]. Version: 2.2

Live acquisition

We acquire the RAM of the system and the following analysis of the data extracted was performed with these tools:

- FTK Imager: it is tool for data preview and imaging. The functionalities that it provides are: creating forensic images without modifying the original evidence, previewing files and folders, exporting files from forensic images, seeing and recovering deleted files, creating file hashes, etc [46]. In order to guarantee that the system under suspicion is affected to the lesser extent when acquiring the RAM, we had installed FTK Imager in an external USB so that we don't have to configure anything in the system. Version: 3.4.3.3
- Volatility: it is a memory forensics framework that analyzes the runtime state of the system by considering the data stored in volatile memory (RAM) [47]. Version: 2.6

Post-mortem acquisition

We acquire a copy of the hard disk of the system. The tool used for the physical acquisition of the disk is:

- FTK Imager: it is tool for data preview and imaging. The functionalities that it provides are: creating forensic images without modifying the original evidence, previewing files and folders, exporting files from forensic images, seeing and recovering deleted files, creating file hashes, etc. [46]. Version: 3.4.3.3

Windows forensics

For the analysis of the files inside the disk under a Windows 7 OS, the tools that we used for the proper interpretation of the data are:

- FTK Imager: it is tool for data preview and imaging. The functionalities that it provides are: creating forensic images without modifying the original evidence, previewing files and folders, exporting files from forensic images, seeing and recovering deleted files, creating file hashes, etc. [46]. Version: 3.4.3.3
- Autopsy: it is a graphical interface tool that deploys plugins used in the *Sleuth Kit* for the analysis of the disk. Some of the modules are: timeline analysis, hash

filtering, keyword search, web artifacts, data carving, multimedia and indicators of compromise [48]. Version: 4.5.0

- Mft2csv: it extracts the \$MFT information and stores it in a CSV file [49]. Version: 2.0.0.41
- Microsoft Excel: it is a spreadsheet developed by Microsoft. We used it to read the CSV output from Mft2csv.
- NAR extractor: it is a modding tool designed specifically for Counter-Strike [50]. Version: 2.0.
- JumpListView: it is a tool that displays the information stored inside the Jump Lists of Windows 7/8 OS. It shows the filename, the date/time of the opened file, the ID of the program that opened the file, etc [51]. Version: 1.15
- WinPrefetchView: it is a tool that displays the information stored in the Prefetch folder. It allows to see which files were used by a specific file inside Prefetch folder [52]. Version:1.35
- RecentFilesView: it is a tool that displays a list of the Recent Files [53]. Version1.33
- AccessData Registry Viewer (Demo Mode): it is a tool that displays the content of Windows OS registry hives [54]. Version: 1.8.0.5
- Regripper: it is a tool that extracts the data from Windows OS registry hives. It is based on plugins that shows different type of information [55]. Version:2.8
- Notepad: it is a text editor for Microsoft Windows, pre-installed in the OS.
- Notepad++: it is a text editor that supports several languages [56]. Version: 7.5.2
- Thumbcache Viewer: it is a tool that displays the data stored (thumbnails) and its related metadata inside Thumbcache of Windows OS [57]. Version: 1.0.3.4
- VLC media player: it is a tool that plays multimedia files, DVDs, Audio CDs, VCDs, and various streaming protocols [58]. Version:2.2.8
- ChromeCacheView: it is a tool that displays information stored inside the Cache folder of Google Chrome Browser. This information is: URL, Content type, File size, Expiration time, Server name, etc. [59]. Version: 1.77
- ChromeHistoryView: it is a tool that displays the list of all the visited Web pages with Google Chrome Browser. For each webpage, it shows: URL, Title, Visit Date/Time, Number of visits, etc. [60]. Version: 1.32

- ChromeCookiesView: it is a tool that displays the list of cookies stored by Google Chrome Browser. For each cookie, it shows: Host name, Path, Name, Value, Last Accessed Time, Creation Time, Expiration time, etc. [61]. Version: 1.46
- DB Browser for SQLite: it is a tool that opens, creates, design and edits database files compatible with SQLite [62]. Version: 3.10.1

3.3. Cases of study

This thesis is focused on two cases of study. These cases were chosen based on the two game modes (Zombie and Studio mode) that Counter Strike Nexon Zombies provides and that a potential user could use in order to communicate with others. Besides, in one of the cases, we also considered the YouTube Live Streaming option offered by this game due to the fact that the user could use it to exchange messages.

They could be representative in a forensic scenario in a way that a suspect or suspects under investigation could use one of these modes or the YouTube Live Streaming option to send and receive messages. In this way, they can exchange relevant information for the forensic case trying to hide themselves as players of the game in a war scenario (Zombie mode) or mining scenario (Studio mode). In these modes, this exchange could be done before (in the lobby chat) or during an in-game round or by performing a YouTube Live Streaming while playing in one of the modes.

3.3.1. Description

The two cases simulate how a user can send and receive messages to/from his friends while playing CSNZ. The first case is recreated in war scenario (Zombie mode) and the second case is related with the mining scenario (Studio mode). The approach they were implemented and the workflow in order to simulate, in the most realistic way, how users could communicate in the game are described as follows:

Case1

There are two players involved. One of them is called “TTUPlayer” and the other is called “UC3MPlayer”. These players are part of the same “family” in the game, which is the way of being friends inside the game (list of friends). The following actions are taken in this case:

- 1) Both players login into Steam platform with their own credentials and then start CSNZ game.
- 2) “TTUPlayer” starts a chat conversation in the lobby of CSNZ with “UC3MPlayer”. This conversation is the following:

TTUPlayer: Hello I am ttuplayer
UC3MPlayer: Hi, do you have my things?
TTUPlayer: Yes, I do
TTUPlayer: Let's play
UC3MPlayer: ok

- 3) Both players start an in-game round in Zombie mode. “TTUPlayer” creates a Zombie private room for playing and sends an invitation to the family member “UC3MPlayer”. “UC3MPlayer” accepts the invitation and whenever they are ready to play, the Zombie mode game starts.
- 4) Inside the game round, both players initiate another chat that includes the following messages:

TTUPlayer: Hi
UC3MPlayer: I am UC3M

Besides, “TTUPlayer” sends a voice message to “UC3MPlayer” saying “*I am TTUPlayer*”.

- 5) Finally, both players exit the Zombie in-game round and log out of the game.

Appendix 1 gives details about how “TTU Player” sees the workflow of case 1 in Counter Strike Nexon Zombies.

Case2

There are two players involved and one viewer of the YouTube Live Streaming. One of the players is called “TTUPlayer” and the other is called “UC3MPlayer”, the viewer is called “Juanma”. The players are part of the same “family” in the game, which is the way of being friends inside the game (friends list). The following actions are taken in this case:

- 1) Both players login into Steam platform with their credentials and then start CSNZ game.

- 2) “TTUPlayer” enables the option of YouTube Live Streaming (enable broadcast option) by login with his Google account credentials.
- 3) “TTUPlayer” logs into his YouTube Channel by using Google Chrome Browser so that he can use the chat feature of YouTube Live Streaming. His nickname on YouTube is “TTU Thesis”.
- 4) Both players start an in-game round in Studio mode. “TTUPlayer” creates a Studio private room for playing and sets a password for entering the room. “UC3MPlayer” knows the password and enters this room. Whenever they are ready to play, the Studio mode game starts.
- 5) “TTUPlayer” starts a YouTube Live Streaming. Since this point, “UC3MPlayer” does nothing (he does not participate in the game either exchange messages).
- 6) “Juanma” starts watching the streaming video through the YouTube Channel of “TTUPlayer” (under the name of “TTU Thesis”) and sends a message to the chat.
- 7) “Juanma” and “TTU Thesis” chat between each other by using the chatting feature of YouTube Live Streaming. The conversation is made through Google Chrome Browser, but the messages are also shown in the streamed video of the Studio in-game round. The exchanged messages are:

Juanma: Hello I am TTU Player 2
TTU Thesis: Do you have my thing?
Juanma: Yes, I do
TTU Thesis: Ok, bye
Juanma: bye

- 8) “TTUPlayer” ends the broadcast and exits the game. “UC3MPlayer” exits the game too.
- 9) “UC3MPlayer” and “TTUPlayer” log out.

Appendix 2 gives details about how “TTU Player” sees this workflow in Counter Strike Nexon Zombies.

3.3.2. Building the cases

Both cases run under VMWare virtual machines. We created two different virtual machines for each example. The reason why we decided to use a virtual machine instead of having the game installed directly in our system is that, in this way, we can have the game isolated and there was not any interference at all with it from external programs. Additionally, thanks to snapshots, we can always go to a previous state if we crush the program.

Regarding the specifications, we used VMWare Fusion 10.0.1 for each case with a Windows 7 Professional guest OS installed. The host system was a macOS High Sierra MacBook Air 2,2 GHz Intel Core i7 with 8GB of memory. In each virtual machine, there was 2GB of memory and 20 GB of disk space. Each virtual machine has a different IP concerning host's IP. For the network capturing, we executed Wireshark in the host system, so that we captured the network traffic in the middle of the communication of both users (man-in-the-middle).

In regard to the game, we installed Steam platform for Windows, as the game cannot be run without it. After that, the game was downloaded from Steam and installed in the Windows systems. For the second case, we also installed Google Chrome Browser since we performed a YouTube Live Streaming. The rest of programs installed in the system are the ones that are pre-included in Windows like Internet Explorer or Notepad.

Due to the fact that this thesis is based on the communication between players while using the game, we have used two virtual machines for each player. As there are two cases of study, there are a total of 4 virtual machines for this thesis.

We had to create 2 Steam accounts and 2 players of CSNZ (one for "TTUPlayer" and another for "UC3MPlayer"). The login process is done through Steam, and then the game starts. We acquired the disk image and RAM from the device (virtual machine) of "TTUPlayer" as he was the suspect in our forensic study. We created an Outlook email account for this player as well as a Google account for the YouTube Live Streaming. In addition, we created a YouTube Channel for "TTUPlayer" with "TTU Thesis" as the nickname. The viewer of the streaming, "Juanma", is subscribed to this channel. This option is not necessary for watching streamed videos in YouTube; however, we decided that the viewer is also a subscriber because we would receive a notification as soon as "TTU Thesis" starts doing live streaming. All this information is displayed in the

following table, to know precisely the type of information that could be recovered from this player. In next chapters, we will show which of them we were able to obtain after doing the analysis.

Table 1. Player information under suspicion.

Game name	Counter Strike Nexon Zombies
Game acronym	CSNZ
UserName Steam account	thttu
Steam User ID	76561198404618625
NickName Steam account	TTU - Thesis
Password Steam account	pAssWd123
Associated email in Steam account	ttu.thesis@outlook.com
CSNZ username	TTUPlayer
CSNZ Game ID	273310
Google account email	ttu.thesis@outlook.com
Google account username	TTU Thesis
Google account password	youtubeaccount1
YouTube Channel name	TTU Thesis
YouTube Channel identifier	UC-pPH8RIVmlFBMbJjUcPZzw
YouTube Live Streaming key	2ppw-5x2j-cz0z-6rrv
YouTube Live Streaming video	https://www.youtube.com/watch?v=Qd4OL0t3bBw
CSNZ Player 2	UC3MPlayer
YouTube Live Streaming viewer	Juanma
Zombie in-game room number	30220
Studio in-game room number	55349
Password Studio room	2tudio

3.3.3. Workflow diagrams

As we commented, the user under suspicion is “TTUPlayer”. In the following diagrams, we present the workflow from the “TTUPlayer” perspective and the moments where the acquisition processes (network capture, live and post-mortem acquisition) took place. The messages that appear in green color means the one sent by “TTUPlayer” and the ones

received are in orange color. In the second diagram, after “UC3MPlayer” enters the password and initiates the Studio in-game round, he does not participate in the game anymore and does not send messages, so there is no more workflow related with this player from “TTUPlayer” perspective.

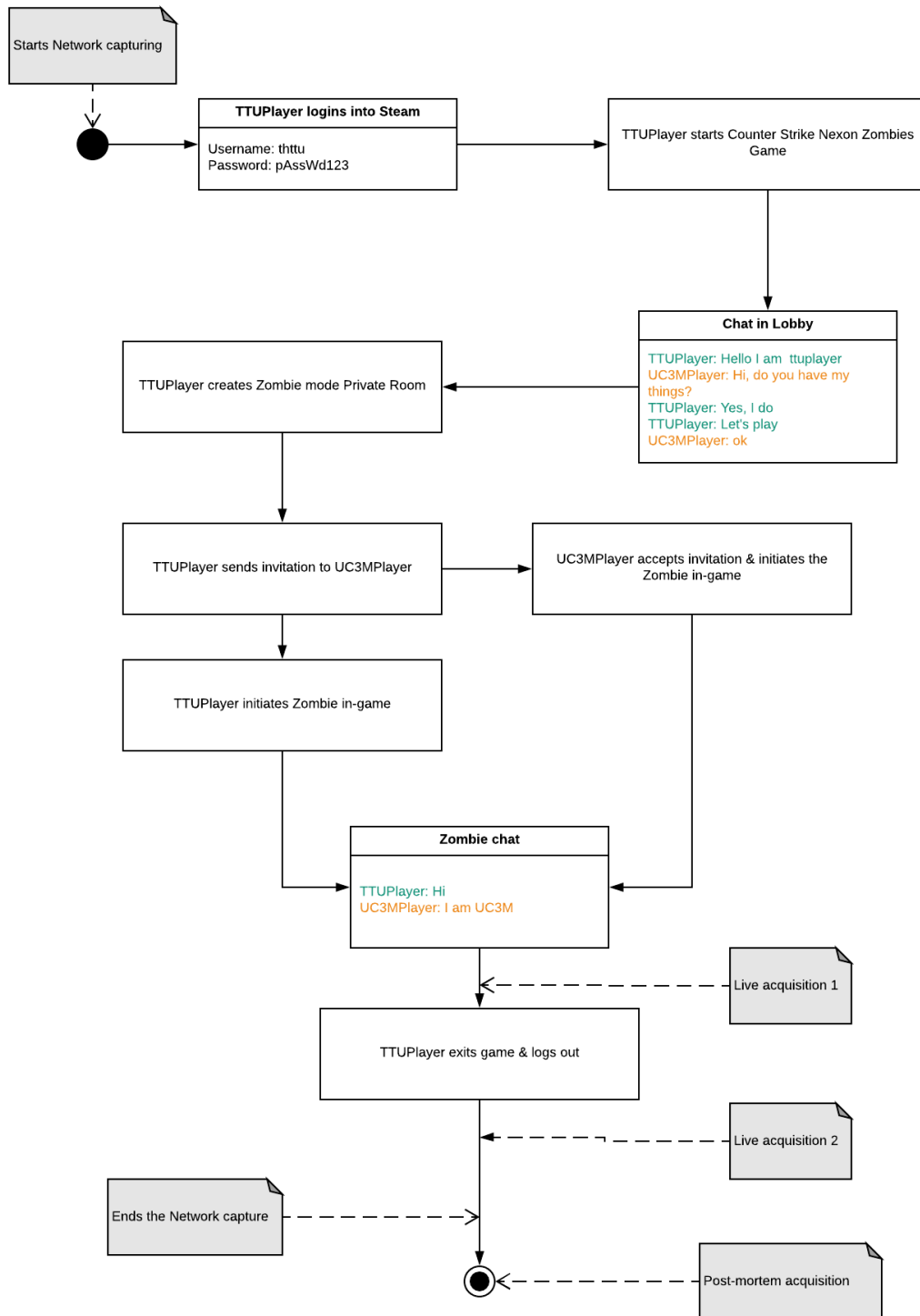


Figure 1. Workflow Case 1

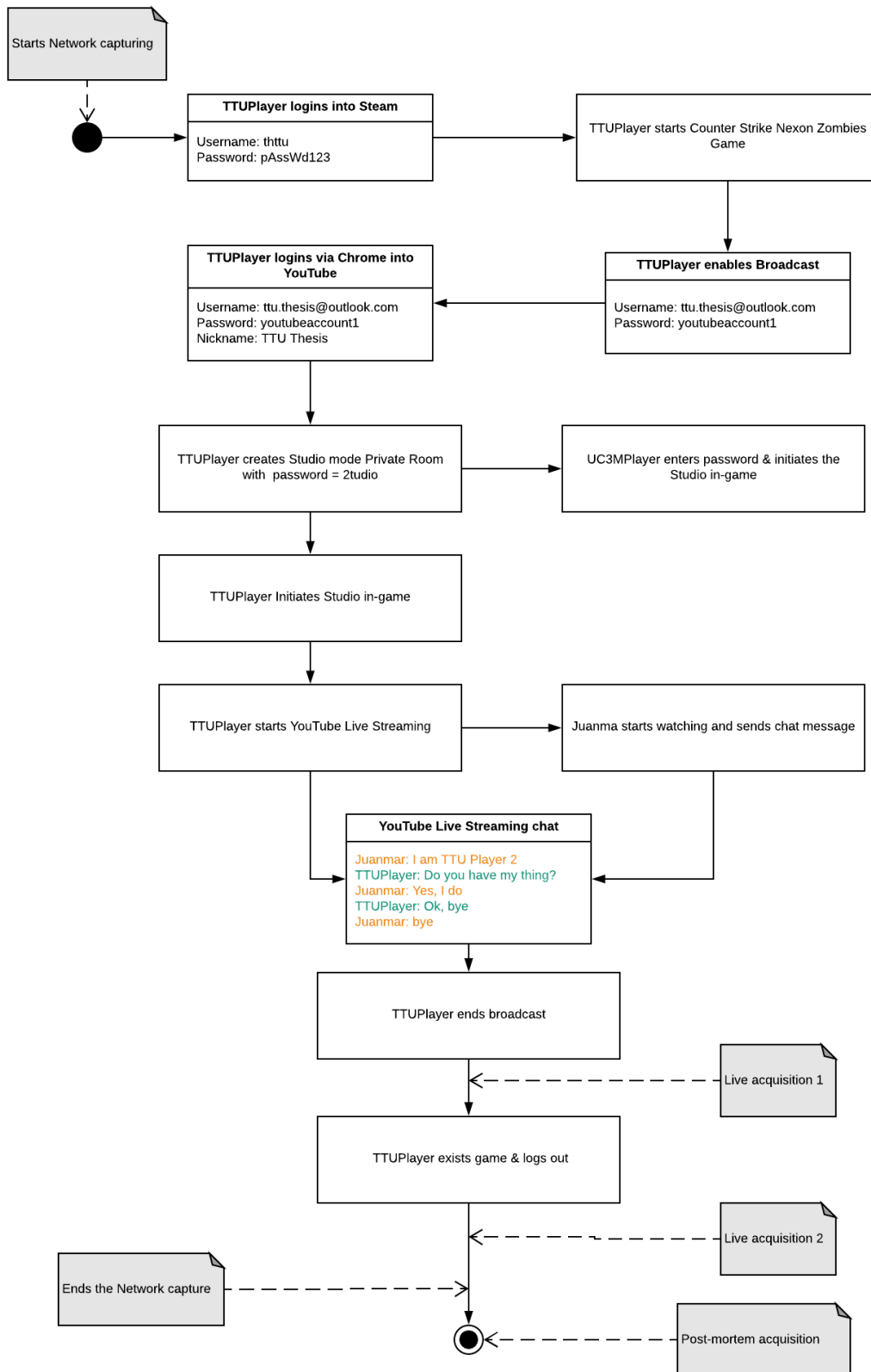


Figure 2. Workflow Case 2

4. Methodology

We have built a guideline for forensic community based on the analysis of both cases in terms of live acquisition, network capturing and post-mortem acquisition. The methodology is applicable for both cases, if there is any particular difference, it is explained in the specific section.

4.1. Network forensics analysis

4.1.1. Common Network artifacts

After capturing the traffic of the network with Wireshark, we have found some relevant artifacts that tie the user to CNSZ game. Even though it was not part of the scope of this thesis, we found a vulnerability that it is useful for finding more forensic information.

The process of identifying in Steam a user after authentication is done through cookies. After authentication, the user is redirected by default to the Steam Store website, the rest of Steam websites (community and help site) use the same cookies for user identification (user sessions). Two main cookies are sent over HTTP:

- **sessionid**: it is a CSRF token. The first time someone accesses to one of its websites, Steam assigns this cookie randomly. It can be any value; the only requirement is that it has to match with the session parameter of the POST requests of that person. Therefore, as it is not linked to any account or specific session, a user doesn't need to authenticate himself first for getting this cookie value.
- **steamLogin**: It is built with the 16-numerical characters of the `Steam_User_ID` + `%7C%7C` (two pipe characters) + 40-character uppercase session token in hexadecimal. This cookie is only generated after the user authenticates.

Therefore, from the `steamLogin` cookie sent via HTTP, we can obtain the **Steam User ID**. With this value, we can visit the website shown below and obtain more forensic information such as: **NickName user account; games played; last time they were played and total number of hours played; current status of the user (Online/Offline) and last time being online; and personal information that the user wants to share.**

<http://steamcommunity.com/profiles/<Steam User ID>>

Considering the *Stop, look and listen* strategy of Network forensics analysis, we can define the way to obtain this cookie. In Wireshark, after opening the packet capture obtained previously, we click on *Edit* → *Find a packet*. Next, we type “steamLogin” in the text box and select *String*, *Packet Details* and *Narrow and Wide* as the options for the Wireshark search. Click on *Find* and the packet that contains this cookie will appear.

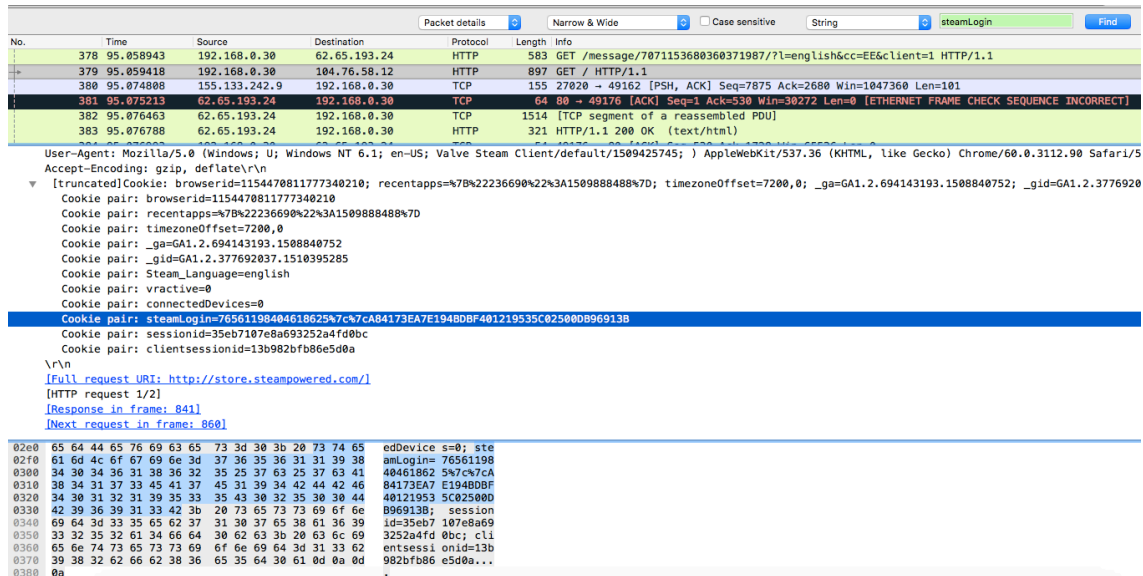


Figure 3. SteamUserID from network capture

From the above picture, we can assure that the Steam User ID is 76561198404618625. If we go to the webpage mentioned previously, we will obtain more forensic information about the user: <http://steamcommunity.com/profiles/76561198404618625>

- NickName Steam user account: TTU – Thesis
- Game(s) played: Counter Strike Nexon Zombies (19.1 total amount of played hours and played last time on 14th of January).
- Current status: Offline (last online 20 days ago).
- Personal information: No information given

It is important to notice that the last time a game was played, does not have to match with the last time of being online. The reason why this could happen is that the user can be online in Steam but without playing any game.

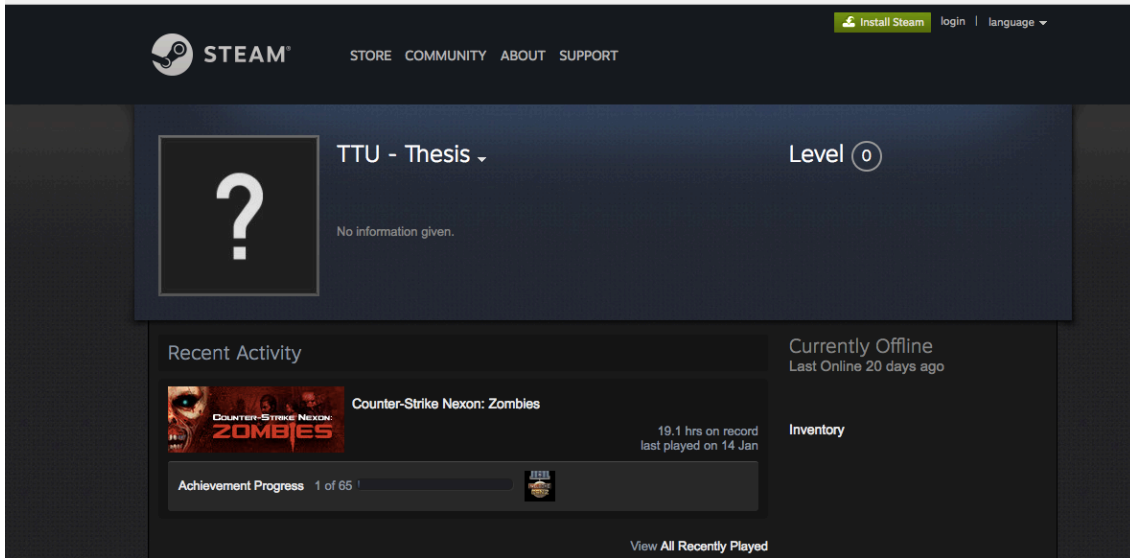


Figure 4. Forensic User Information after Network analysis

4.1.2. Session cloning - Vulnerability found

Although it was not part of the scope of this thesis, we found a vulnerability of Session hijacking. When trying to perform a session cloning, as it was done in other studies [32], we found that it was possible to hijack the session of an authenticated user. This **vulnerability** was reported to Valve Corporation via email and they only allowed us to include the explanation about how we did it in this thesis. The report and the fix could not be included. The session hijacking is described below.

Considering the fact that the `sessionid` doesn't depend on the authenticated user and that there are some cookies transmitted over HTTP, after HTTPS authentication, which are not changed, we found a way to clone the session of the user while he has it opened. The steps performed for the session hijacking were:

- 1) With the Wireshark capture, search inside the packet details for the string "steamLogin". Copy it and all its related cookies as "printable text".
- 2) Create an account on Steam (whatever user and password) and login to generate the steamLogin cookie. We logged with Google Chrome Browser as we used "Edit this cookie" extension in order to inject the cookies related with the user session we want to clone, but it is possible to do it with other browsers and add-ons.
- 3) Click on "Edit this cookie" extension, we modify the cookies `_gad`, `sessionID` and

steamLogin with the ones that we obtained from the Wireshark capture (Step 1), in this way, we are injecting other user cookies to hijack his session. Click on the green icon to save those cookies.

- 4) Click on the arrow-Back to go to the previous initial page of login and click on refreshing the page. We will be logged as our desired user so that the session hijacking results successfully. We can see **all the information related to that user**: list of friends, games played, last connection, status (online), etc.

NOTE: It is important to point out that this method is related to sessions, so if the desired user logs out, we won't be able to continue using his session.

From a forensic point of view, the possibility to clone the session give us a lot of information about the user: the list of friends in Steam, games played, last connection, status (online) and Steam chat messages.

4.1.3. YouTube Live Streaming specific artifacts

From the second case, we can see that in the network capture are references that show that the suspect was doing a YouTube Live Streaming.

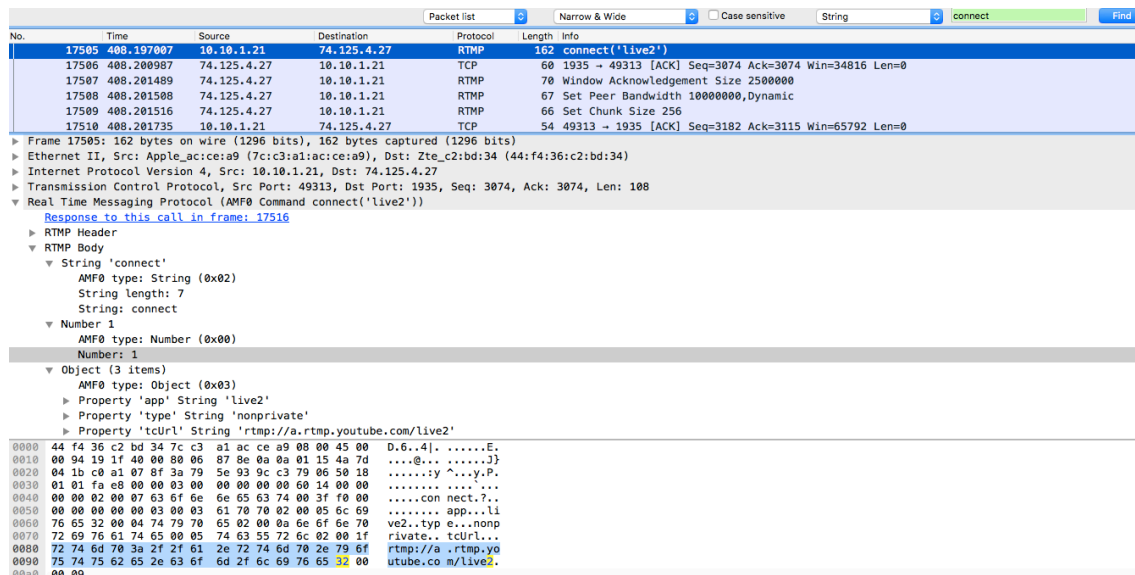
There are RTMP packets (without TLS/SSL encryption), which are used for live streaming. More specifically, there must be a packet that shows the connection with the **main server of YouTube**, the method used for this purpose is: `connect('live2')` and the associated URL is:

`rtmp://a.rtmp.youtube.com/live2`

In Wireshark, after opening the packet capture obtained previously, we click on *Edit* → *Find a packet*. Next, we type “connect” in the text box and select *String*, *Packet list* and *Narrow and Wide* as the options for the Wireshark search. Click on *Find* and the packet that contains the URL will appear.

Besides, in the following packets, it also appears the **YouTube Live Streaming key** that it is being used for the streaming, it is show in the method called `releaseStream('<key>')`. This key is unique for each user, and even it is not possible to watch the video with it if we capture it, it can be used for doing a YouTube Live

Streaming with the account of other user. The method to obtain this key is the same as before, but searching for “releaseStream”.

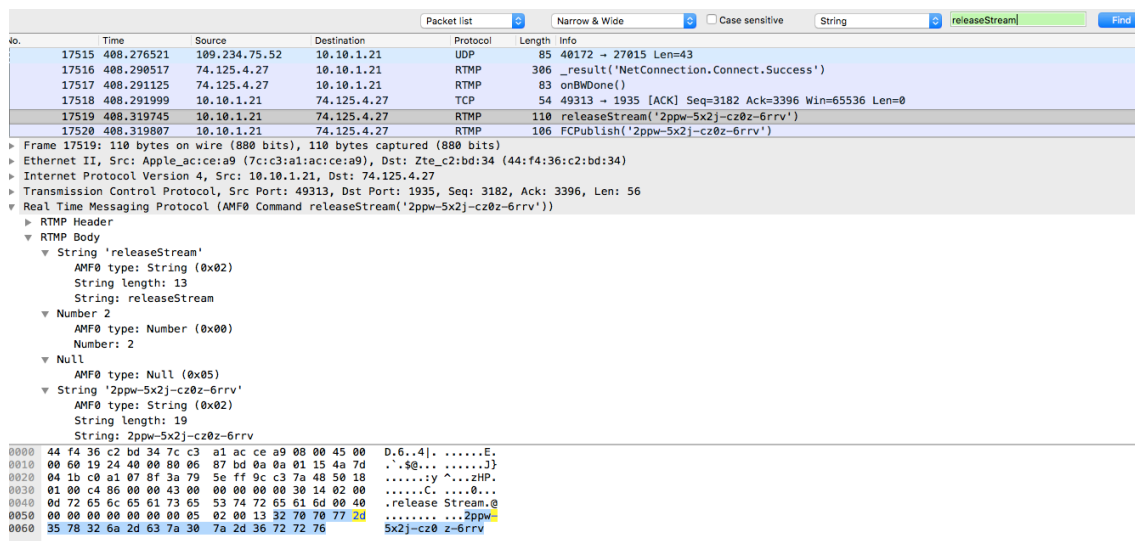


The image shows a Wireshark network capture of an RTMP connection. The packet list shows a 'connect' packet (17505) from 10.10.1.21 to 74.125.4.27. The packet details pane shows the RTMP body with a 'connect' command. The packet bytes pane shows the raw data of the packet.

No.	Time	Source	Destination	Protocol	Length	Info
17505	408.197007	10.10.1.21	74.125.4.27	RTMP	162	connect('live2')
17506	408.200987	74.125.4.27	10.10.1.21	TCP	60	1935 → 49313 [ACK] Seq=3074 Ack=3074 Win=34816 Len=0
17507	408.201489	74.125.4.27	10.10.1.21	RTMP	70	Window Acknowledgement Size 2500000
17508	408.201508	74.125.4.27	10.10.1.21	RTMP	67	Set Peer Bandwidth 1000000,Dynamic
17509	408.201516	74.125.4.27	10.10.1.21	RTMP	66	Set Chunk Size 256
17510	408.201735	10.10.1.21	74.125.4.27	TCP	54	49313 → 1935 [ACK] Seq=3182 Ack=3115 Win=65792 Len=0

Frame 17505: 162 bytes on wire (1296 bits), 162 bytes captured (1296 bits) on interface 0
 Ethernet II, Src: Apple, ac:ce:a9:7c:c3:a1, Dst: Zte, c2:bd:34 (44:f4:36:c2:bd:34)
 Internet Protocol Version 4, Src: 10.10.1.21, Dst: 74.125.4.27
 Transmission Control Protocol, Src Port: 49313, Dst Port: 1935, Seq: 3074, Ack: 3074, Len: 108
 Real Time Messaging Protocol (AMF0 Command connect('live2'))
 Response to this call in frame: 17516
 RTMP Header
 RTMP Body
 String 'connect'
 AMF0 type: String (0x02)
 String length: 7
 String: connect
 Number 1
 AMF0 type: Number (0x00)
 Number: 1
 Object (3 items)
 AMF0 type: Object (0x03)
 Property 'app' String 'live2'
 Property 'type' String 'nonprivate'
 Property 'url' String 'rtmp://a.rtmp.youtube.com/live2'

Figure 5. YouTube Live Streaming Main server from Network capture



The image shows a Wireshark network capture of an RTMP releaseStream packet. The packet list shows a 'releaseStream' packet (17519) from 10.10.1.21 to 74.125.4.27. The packet details pane shows the RTMP body with a 'releaseStream' command. The packet bytes pane shows the raw data of the packet.

No.	Time	Source	Destination	Protocol	Length	Info
17515	408.276521	109.234.75.52	10.10.1.21	UDP	85	40172 → 27015 Len=43
17516	408.290517	74.125.4.27	10.10.1.21	RTMP	306	_result('NetConnection.Connect.Success')
17517	408.291125	74.125.4.27	10.10.1.21	RTMP	83	onBWDone()
17518	408.291999	10.10.1.21	74.125.4.27	TCP	54	49313 → 1935 [ACK] Seq=3182 Ack=3396 Win=65536 Len=0
17519	408.319745	10.10.1.21	74.125.4.27	RTMP	110	releaseStream('2ppw-5x2j-cz0z-6rrv')
17520	408.319807	10.10.1.21	74.125.4.27	RTMP	106	FCPublish('2ppw-5x2j-cz0z-6rrv')

Frame 17519: 110 bytes on wire (880 bits), 110 bytes captured (880 bits) on interface 0
 Ethernet II, Src: Apple, ac:ce:a9:7c:c3:a1, Dst: Zte, c2:bd:34 (44:f4:36:c2:bd:34)
 Internet Protocol Version 4, Src: 10.10.1.21, Dst: 74.125.4.27
 Transmission Control Protocol, Src Port: 49313, Dst Port: 1935, Seq: 3182, Ack: 3396, Len: 56
 Real Time Messaging Protocol (AMF0 Command releaseStream('2ppw-5x2j-cz0z-6rrv'))
 RTMP Header
 RTMP Body
 String 'releaseStream'
 AMF0 type: String (0x02)
 String length: 13
 String: releaseStream
 Number 2
 AMF0 type: Number (0x00)
 Number: 2
 Null
 AMF0 type: Null (0x05)
 String '2ppw-5x2j-cz0z-6rrv'
 AMF0 type: String (0x02)
 String length: 19
 String: 2ppw-5x2j-cz0z-6rrv

Figure 6. YouTube Live Streaming key from Network capture

Moreover, we also recovered the **encoder** that was used for the video. Thanks to this information, the forensic expert can watch the video if there is any possibility to extract it from the copy of the disk, as we will see in the Post-mortem analysis section. The encoder used was VLC and can be found by doing right-click in one of the RTMP packets in Wireshark and clicking on *Follow TCP Stream*. The conclusion achieved after this finding is that the video can be watched with VLC multimedia player.

```

- core 148 r2762 90a61ec H.264/MPEG-4 AVC codec - Copyleft 2003-2017 - http://www.videolan.org/x264.html options: cabac=1 ref=3 deblock=1:0:0 analyse=0x3:0x1
1 trellis=1 8x8dct=1 cqm=0 deadzone=21,11 fast_pskip=1 chroma_opt_offset=2 threads=3 lookahead_threads=1 sliced_threads=0 nr=0 decimate=1 interlaced=0 bluray_co
t=1 weight=0 open_gop=0 weightp=2 keyint=250 keyint_min=25 scenecut=40 intra_refresh=0 rc_lookahead=40 rc=crf mbtree=1 crf=23.0 cqp=0.0 gpmq=0.0 gpmq=69.0
p_ratio=1.40 aq=1.1:0.0, e...'.D.....Np*...'.W...T.Asg...R.7!..$....e...jE..Z.xp.....77..l.lq>..6...ep.....(*.8..uK.#...!
!A.....>.A.V...F...K.0.l.e!.....v...q $.U...].7=<...7X...D..d.C...>...Y.
7.0 vx 2 f 1 M...e...A2 M f e T...A...D...B...H...I...C...F...I...A...5...I...C...e...C...B...I...D...M...e...>...H...I...>.../...

```

Figure 7. YouTube Live Streaming encoder from Network capture

4.2. Volatile memory analysis

After acquiring 2 evidences of the RAM with FTK Imager installed in an external USB: one before and the other after logging out from the game, we have analyzed both with FTK Imager and Volatility framework. To see the details about how to acquire a memory image with FTK Imager, please check Annex 3. The main difference between them was that it was possible to extract the chat messages only if the user didn't log out.

4.2.1. Volatile memory analysis with FTK Imager

After adding the evidence, the methodology that we have developed in order to find easily forensic artifacts after a live acquisition is based on keywords, similarly to the study performed about the XboxOne with Autopsy [40].

FTK Imager has integrated a search tool so, by doing right-click at the beginning of the memory dump and click on *Find* we can search any word. It is also possible to search a regular expression. The keywords and regular expressions that are useful for finding valuable forensic data with respect to the game are explained below.

- **Username Steam account**

For finding the username that the user has when login into Steam, use the keyword: **SteamUser**

if20	52	4F	4F	54	44	52	49	56-45	3D	43	00	00	00	00	00	ROOTDRIVE=C:...
if30	5B	D7	22	4C	00	00	00	88-53	74	65	61	6D	55	73	65	[* "L...SteamUse
if40	72	3D	74	68	74	74	75	00-54	D7	22	4C	00	00	00	89	r=thttu-T*"L...
if50	53	79	73	74	65	6D	44	72-69	76	65	3D	43	3A	00	00	SystemDrive=C:...

Figure 8. Steam UserName after volatile analysis

- **NickName Steam account**

The nickname that the user has in Steam could be different than the name that the user has when login. In other to don't get confused with this fact, the keyword that should be used to find it quickly is: **PersonaName**

35101d10	EE F1 EE F1 22 50 65 72-73 6F 6E 61 4E 61 6D 65	îîîî"PersonaName
35101d20	22 09 09 22 54 54 55 20-2D 20 54 68 65 73 69 73	"..-TTU - Thesis
35101d30	22 0A 09 09 22 50 65 72-73 6F 6E 61 53 74 61 74	"..-PersonaStat
35101d40	65 4A 65 73 69 72 65 6A-8A DD 76 00 FF F1 FF F1	eDesired.V!..f&f&

Figure 9. Steam NickName after volatile analysis

▪ Password Steam account

The password that the user has when doing login in Steam is stored in volatile memory in plaintext. The way to find it is just by searching for the word: password=

4099a050	00 00 00 00 00 00 00 00-00 00 00 00 92 61 FE B5apu
4099a060	68 74 74 70 73 3A 2F 2F-68 65 6C 70 2E 73 74 65	https://help.ste
4099a070	61 6D 70 6F 77 65 72 65-64 2E 63 6F 6D 2F 65 6E	ampowered.com/en
4099a080	2F 77 69 7A 61 72 64 2F-41 6A 61 78 43 68 65 63	/wizard/AjaxChec
4099a090	6B 50 61 73 73 77 6F 72-64 41 76 61 69 6C 61 62	kPasswordAvailab
4099a0a0	6C 65 2F 3F 73 65 73 73-69 6F 6E 69 64 3D 33 39	le/?sessionid=39
4099a0b0	66 61 30 34 37 63 66 30-30 66 30 32 63 35 62 61	fa047cf00f02c5ba
4099a0c0	33 33 35 39 61 63 26 77-69 7A 61 72 64 5F 61 6A	3359ac&wizard_aj
4099a0d0	61 78 3D 31 26 70 61 73-73 77 6F 72 64 3D 70 41	ax=1&password=pA
4099a0e0	73 73 57 64 31 32 33 00-00 00 00 00 00 00 00	ssWd123
4099a0f0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00	

Figure 10. Steam Password after volatile analysis

▪ Steam User ID

Every user in Steam has an associated identification number, as we saw from the network part. This value is also stored in volatile memory. The keyword is: steamid

1c4c1670	6D 00 65 00 49 00 64 00-3D 00 32 00 37 00 33 00	m·e·I·d·=-2·7·3·
1c4c1680	31 00 31 00 30 00 00 00-53 00 54 00 45 00 41 00	1·1·0·-·S·T·E·A·
1c4c1690	4D 00 49 00 44 00 3D 00-37 00 36 00 35 00 36 00	M·I·D·=-7·6·5·6·
1c4c16a0	31 00 31 00 39 00 38 00-34 00 30 00 34 00 36 00	1·1·9·8·4·0·4·6·
1c4c16b0	31 00 38 00 36 00 32 00-35 00 00 00 53 00 74 00	1·8·6·2·5·-·S·t·
1c4c16c0	65 00 61 00 6D 00 50 00-61 00 74 00 68 00 3D 00	

Figure 11. Steam UserID after volatile analysis

With this ID, the forensic expert can go to <http://steamcommunity.com/profiles/<Steam User ID>>, as it is described in the network analysis section of this chapter and obtain more forensic information such as the last time played of the game, nickname of the user, etc.

▪ CSNZ Game ID

Every game that is integrated into Steam platform has a number that uniquely identifies it. For Counter Strike Nexon Zombies is 273110. It can be found with the keyword: steamGameID

4c4c1660	00 00 53 00 74 00 65 00-61 00 6D 00 47 00 61 00	-S-t-e-a-m-G-a-
4c4c1670	6D 00 65 00 49 00 64 00-3D 00 32 00 37 00 33 00	m-e-i-d=-2-7-3-
4c4c1680	31 00 31 00 30 00 00 00-53 00 54 00 45 00 41 00	1-1-0--S-T-E-A-

Figure 12. CSNZ GameID after volatile analysis

▪ Associated email address

Each user has an email account associated with this Steam account and consequently, with the Counter Strike Nexon Zombies game. This email is also stored in volatile memory. A way to find the email account will be through a regular expression. We propose the following one: \b[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,}\b

0	00 00 00 00 00 00 00 00-61 70 69 2E 73 74 65 61api.stea
0	6D 70 6F 77 65 72 65 64-2E 63 6F 6D 00 16 00 00	mpowered.com----
0	74 74 75 2E 74 68 65 73-69 73 40 6F 75 74 6C 6F	ttu.thesis@outlo
0	6F 6B 2E 63 6F 6D 00 00-37 35 62 39 4E 49 32 4C	ok.com--75b9NI2L

Figure 13. Associated email after volatile analysis

▪ Chat messages

From volatile memory, it is possible to obtain the messages sent and received by the user but only if he/she didn't log out when taking the memory dump.

In Counter Strike Nexon Zombies, the user can exchange messages directly without starting a game round, inside in-game rounds and also while doing YouTube Live Streaming. In both cases, messages can be sent to all the players, to your family group, or inside the party group. In this way, this different kind of people that can be addressed in a message is the way to find the information about messages. The way to know which group the message was sent to is by considering the following layouts that we have developed:

- Chat in the lobby: (Type of receiver) [Nickname of the sender] : message
- In-game chat: [Type of receiver] Nickname of the sender : message
- YouTube chat: [YOUTUBE] Nickname of the sender : message

Therefore, the keywords for looking for chat messages are based on the type of the receiver: All, Family, Party and YOUTUBE. Once we found the messages, we can state where it was sent to (lobby, in-game or YouTube) based on the previous layouts. Below we include three pictures of each type of chat situation.

```

18dfcc40 54 9B EB 0D 00 00 00 00-00 00 00 00 00 00 00 00 T-ë-----
18dfcc50 00 00 00 00 88 01 00 00-00 00 00 00 00 00 00 00 .....
18dfcc60 88 7E 35 53 73 CB 00 8C-28 00 46 00 61 00 6D 00 ..~5SsE- (·F·a·m·
18dfcc70 69 00 6C 00 79 00 29 00-5B 00 54 00 54 00 55 00 i·l·y·)·[·T·T·U·
18dfcc80 50 00 6C 00 61 00 79 00-65 00 72 00 5D 00 20 00 P·l·a·y·e·r·]··
18dfcc90 48 00 65 00 6C 00 6C 00-6F 00 20 00 49 00 20 00 H·e·l·l·o· ·I··
18dfcca0 61 00 6D 00 20 00 74 00-74 00 75 00 70 00 6C 00 a·m· ·t·t·u·p·l·
18dfccb0 61 00 79 00 65 00 72 00-0A 00 28 00 46 00 61 00 a·y·e·r····(·F·a·
18dfccc0 6D 00 69 00 6C 00 79 00-29 00 5B 00 55 00 43 00 m·i·l·y·)·[·U·C·
18dfccd0 33 00 4D 00 50 00 6C 00-61 00 79 00 65 00 72 00 3·M·P·l·a·y·e·r·
18dfcce0 5D 00 20 00 48 00 69 00-2C 00 20 00 64 00 6F 00 ]· ·H·i·,· ·d·o·
18dfccf0 20 00 79 00 6F 00 75 00-20 00 68 00 61 00 76 00 ·y·o·u· ·h·a·v·
18dfcd00 65 00 20 00 6D 00 79 00-20 00 74 00 68 00 69 00 e· ·m·y· ·t·h·i·
18dfcd10 6E 00 67 00 73 00 3F 00-0A 00 28 00 46 00 61 00 n·g·s·?···(·F·a·
18dfcd20 6D 00 69 00 6C 00 79 00-29 00 5B 00 54 00 54 00 m·i·l·y·)·[·T·T·
18dfcd30 55 00 50 00 6C 00 61 00-79 00 65 00 72 00 5D 00 U·P·l·a·y·e·r·]··
18dfcd40 20 00 59 00 65 00 73 00-2C 00 20 00 49 00 20 00 ·Y·e·s·,· ·I··
18dfcd50 64 00 6F 00 2E 00 0A 00-28 00 46 00 61 00 6D 00 d·o····(·F·a·m·
18dfcd60 69 00 6C 00 79 00 29 00-5B 00 54 00 54 00 55 00 i·l·y·)·[·T·T·U·
18dfcd70 50 00 6C 00 61 00 79 00-65 00 72 00 5D 00 20 00 P·l·a·y·e·r·]··
18dfcd80 4C 00 65 00 74 00 27 00-73 00 20 00 70 00 6C 00 L·e·t·'·s· ·p·l·
18dfcd90 61 00 79 00 0A 00 28 00-46 00 61 00 6D 00 69 00 a·y····(·F·a·m·i·
18dfcda0 6C 00 79 00 29 00 5B 00-55 00 43 00 33 00 4D 00 l·y·)·[·U·C·3·M·
18dfcdb0 50 00 6C 00 61 00 79 00-65 00 72 00 5D 00 20 00 P·l·a·y·e·r·]··
18dfcdc0 6F 00 6B 00 61 00 6E 00-BD 7E 35 53 00 00 00 94 o·k·|·a·n·~5S····
18dfcdd0 02 3F 00 00 01 00 00 00-23 20 01 00 00 00 00 00 ·?·····#·····

```

Figure 14. Lobby Chat after volatile analysis

iff4f0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
iff500	00 00 00 00 02 00 5B 00-41 00 6C 00 6C 00 5D 00[·A·l·l·]
iff510	20 00 54 00 54 00 55 00-50 00 6C 00 61 00 79 00	·T·T·U·P·l·a·y·
iff520	65 00 72 00 20 00 3A 00-20 00 20 00 48 00 69 00	e·r· :· :· ·H·i·
iff530	0A 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
iff540	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
iff550	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
iff560	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
iff570	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
iff580	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
iff590	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
iff5a0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
iff5b0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
iff5c0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
iff5d0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
iff5e0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
iff5f0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
iff600	00 00 00 00 5B 00 46 00-61 00 6D 00 69 00 6C 00[·F·a·m·i·l·
iff610	79 00 5D 00 20 00 55 00-43 00 33 00 4D 00 50 00	y·]· ·U·C·3·M·P·
iff620	6C 00 61 00 79 00 65 00-72 00 3A 00 20 00 49 00	l·a·y·e·r·:· ·I·
iff630	20 00 61 00 6D 00 20 00-55 00 43 00 33 00 4D 00	·a·m· ·U·C·3·M·
iff640	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00

Figure 15. In-game chat after volatile analysis

32f2b510	37 D4 FB 01 00 00 00 8E-5B 00 59 00 4F 00 55 00	70â....[·Y·O·U·
32f2b520	54 00 55 00 42 00 45 00-5D 00 20 00 4A 00 75 00	T·U·B·E·]· ·J·u·
32f2b530	61 00 6E 00 6D 00 61 00-3A 00 20 00 48 00 65 00	a·n·m·a·:· ·H·e·
32f2b540	6C 00 6C 00 6F 00 20 00-49 00 20 00 61 00 6D 00	l·l·o· ·I· ·a·m·
32f2b550	20 00 54 00 54 00 55 00-20 00 50 00 6C 00 61 00	·T·T·U· ·P·l·a·
32f2b560	79 00 65 00 72 00 20 00-32 00 0A 00 5B 00 59 00	y·e·r· ·2·...[·Y·
32f2b570	4F 00 55 00 54 00 55 00-42 00 45 00 5D 00 20 00	O·U·T·U·B·E·]· ·
32f2b580	54 00 54 00 55 00 20 00-54 00 68 00 65 00 73 00	T·T·U· ·T·h·e·s·
32f2b590	69 00 73 00 3A 00 20 00-44 00 6F 00 20 00 79 00	i·s·:· ·D·o· ·y·
32f2b5a0	6F 00 75 00 20 00 68 00-61 00 76 00 65 00 20 00	o·u· ·h·a·v·e· ·
32f2b5b0	6D 00 79 00 20 00 74 00-68 00 69 00 6E 00 67 00	m·y· ·t·h·i·n·g·
32f2b5c0	3F 00 0A 00 5B 00 59 00-4F 00 55 00 54 00 55 00	?·...[·Y·O·U·T·U·
32f2b5d0	42 00 45 00 5D 00 20 00-4A 00 75 00 61 00 6E 00	B·E·]· ·J·u·a·n·
32f2b5e0	6D 00 61 00 3A 00 20 00-59 00 65 00 73 00 2C 00	m·a·:· ·Y·e·s·,
32f2b5f0	20 00 49 00 20 00 64 00-6F 00 0A 00 5B 00 59 00	·I· ·d·o·...[·Y·
32f2b600	4F 00 55 00 54 00 55 00-42 00 45 00 5D 00 20 00	O·U·T·U·B·E·]· ·
32f2b610	54 00 54 00 55 00 20 00-54 00 68 00 65 00 73 00	T·T·U· ·T·h·e·s·
32f2b620	69 00 73 00 3A 00 20 00-4F 00 6B 00 2C 00 20 00	i·s·:· ·O·k·,
32f2b630	62 00 79 00 65 00 0A 00-5B 00 59 00 4F 00 55 00	b·y·e·...[·Y·O·U·
32f2b640	54 00 55 00 42 00 45 00-5D 00 20 00 4A 00 75 00	T·U·B·E·]· ·J·u·
32f2b650	61 00 6E 00 6D 00 61 00-3A 00 20 00 62 00 79 00	a·n·m·a·:· ·b·y·
32f2b660	65 00 6B 00 2C 00 65 00-58 D4 FB 01 78 00 00 88	e·k·, ·e·X0â·x·...

Figure 16. YouTube chat after volatile analysis

- **Friends who are online**

CSNZ shows when a member of your family (friends list) is online with a message of:

<Username> has logged in.

This information is also stored in volatile memory. The way to find it is by using the keywords: has logged in

fff100	00 00 00 00 55 00 43 00-33 00 4D 00 50 00 6C 00	...U-C-3-M-P-l-
fff110	61 00 79 00 65 00 72 00-20 00 68 00 61 00 73 00	a-y-e-r- h-a-s-
fff120	20 00 6C 00 6F 00 67 00-67 00 65 00 64 00 20 00	-l-o-g-g-e-d-
fff130	69 00 6E 00 2E 00 00 00-00 00 00 00 00 00 00	i-n-.....
fff140	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00

Figure 17. Online friends after volatile analysis

▪ Players involved in the game

The players that were playing the in-side game rounds are shown in CSNZ with the message: <Username> has joined <type of terrorist>

This information is also stored in volatile memory. The way to find it is by using the keywords: has joined

fff1f0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
fff200	00 00 00 00 55 00 43 00-33 00 4D 00 50 00 6C 00	...U-C-3-M-P-l-
fff210	61 00 79 00 65 00 72 00-20 00 68 00 61 00 73 00	a-y-e-r- h-a-s-
fff220	20 00 6A 00 6F 00 69 00-6E 00 65 00 64 00 20 00	-j-o-i-n-e-d-
fff230	74 00 68 00 65 00 20 00-63 00 6F 00 75 00 6E 00	t-h-e- c-o-u-n-
fff240	74 00 65 00 72 00 2D 00-74 00 65 00 72 00 72 00	t-e-r--t-e-r-r-
fff250	6F 00 72 00 69 00 73 00-74 00 73 00 2E 00 00 00	o-r-i-s-t-s-....
fff260	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00

Figure 18. Players in the game after volatile analysis

▪ Room number and password

If the players want to make in-game rounds private, they can create a private room. Additionally, the player who creates the room can send an invitation for playing to the members of their family who are online, and also to create a password for that room. The room is identified by a unique number. It is also possible to obtain this number and the password from a memory dump.

The room number appears in volatile memory in the following way: (#No.room)

080fb070	A0 5B B8 34 A0 5B B8 34-A0 5B B8 34 A0 5B B8 34	[,4 [,4 [,4 [,4
080fb080	53 00 74 00 75 00 64 00-69 00 6F 00 20 00 3E 00	S-t-u-d-i-o- ->-
080fb090	20 00 28 00 23 00 35 00-35 00 33 00 34 00 39 00	-(-#-5-5-3-4-9-
080fb0a0	29 00 20 00 53 00 74 00-75 00 64 00 69 00 6F 00)- S-t-u-d-i-o-
080fb0b0	20 00 43 00 61 00 73 00-65 00 20 00 32 00 00 00	-C-a-s-e- -2---

Figure 19. Room number after volatile analysis

The password was also found in memory:

```

3dc7eb20|00 00 00 00 00 00 00 00-00 00 00 00 00 00 00|.....
3dc7eb30|32 00 74 00 75 00 64 00-69 00 6F 00 00 00 00|2-t-u-d-i-o-
3dc7eb40|06 00 00 00 07 00 00 00-00 00 00 00 00 00 00|.....
3dc7eb50|00 00 00 00 02 00 00 00-00 00 00 00 00 00 00|.....

```

Figure 20. Room password after volatile analysis

▪ YouTube Live Streaming traces

There are some references found in volatile memory that shows that the user was doing a YouTube Live Streaming, like the link to the video, the YouTube channel and the streaming key. The best way to find it is by using as a keyword: `channel_id` (youtube channel), `youtube` (streamed link), `rtmp.youtube` (streamed key) and `codec` (video encoder).

```

07f3eea0|70 61 72 61 6D 73 22 3A-5B 7B 22 6B 65 79 22 3A|params":[{"key":
07f3eeb0|22 63 72 65 61 74 6F 72-5F 63 68 61 6E 6E 65 6C|"creator_channel
07f3eec0|5F 69 64 22 2C 22 76 61-6C 75 65 22 3A 22 55 43|_id","value":"UC
07f3eed0|2D 70 50 48 38 52 49 56-6D 6C 46 42 4D 62 4A 6A|-pPH8RIVmlFBMbJj
07f3eee0|55 63 50 5A 7A 77 22 7D-2C 7B 22 6B 65 79 22 3A|UcPZzw"}, {"key":
07f3eef0|22 6C 6F 67 67 65 64 5F-69 6E 22 2C 22 76 61 6C|"logged_in","val

```

Figure 21. YouTube Channel after volatile analysis

```

0204f060|68 74 74 70 73 3A 2F 2F-72 32 2D 2D 2D 73 6E 2D|https://r2---sn-
0204f070|32 35 67 6C 65 6E 37 6C-2E 63 2E 79 6F 75 74 75|25glen7l.c.youtu
0204f080|62 65 2E 63 6F 6D 2F 76-69 64 65 6F 70 6C 61 79|be.com/videoplay
0204f090|62 61 63 6B 3F 69 64 3D-51 64 34 4F 4C 30 74 33|back?id=Qd40L0t3
0204f0a0|62 42 77 2E 30 26 69 74-61 67 3D 31 34 30 26 73|bBw.0&itag=140&s

```

Figure 22. YouTube streaming link after volatile analysis

```

69651da0|72 74 6D 70 3A 2F 2F 61-2E 72 74 6D 70 2E 79 6F|rtmp://a.rtmp.yo
69651db0|75 74 75 62 65 2E 63 6F-6D 2F 6C 69 76 65 32 2F|utube.com/live2/
69651dc0|32 70 70 77 2D 35 78 32-6A 2D 63 7A 30 7A 2D 36|2ppw-5x2j-cz0z-6
69651dd0|72 72 76 00 00 65 00 2F-00 00 00 00 0F 00 00 00|rrv-e-.....

```

Figure 23. YouTube streaming key after volatile analysis

The encoder of the video can also be obtained from volatile memory.

3f5255a0	78 32 36 34 20 2D 20 63-6F 72 65 20 25 64 25 73	x264 - core %d%s
3f5255b0	20 2D 20 48 2E 32 36 34-2F 4D 50 45 47 2D 34 20	- H.264/MPEG-4
3f5255c0	41 56 43 20 63 6F 64 65-63 20 2D 20 43 6F 70 79	AVC codec - Copy
3f5255d0	25 73 20 32 30 30 33 2D-32 30 31 37 20 2D 20 68	%s 2003-2017 - h
3f5255e0	74 74 70 3A 2F 2F 77 77-77 2E 76 69 64 65 6F 6C	ttp://www.videol
3f5255f0	61 6E 2E 6F 72 67 2F 78-32 36 34 2E 68 74 6D 6C	an.org/x264.html
3f525600	20 2D 20 6F 70 74 69 6F-6E 73 3A 20 25 73 00 6C	- options: %s-l
3f525610	65 66 74 00 20 72 32 37-36 32 20 39 30 61 36 31	eft- r2762 90a61
3f525620	65 63 00 00 41 56 43 2D-49 6E 74 72 61 20 53 45	ec]-AVC-Intra SE

Figure 24. YouTube streaming encoder after volatile analysis

▪ CSNZ extra references

Apart from the ones explained above, there are also some additional traces that show that the user was playing the game such as the acronym of the game, the name of the game and the game mode. The keywords associated with them are, respectively: CSNZ, Counter Strike and Studio/Zombie.

0078edd0	43 00 4F 00 4D 00 4D 00-4F 00 4E 00 5C 00 43 00	C·O·M·M·O·N·\·C·
0078ede0	53 00 4E 00 5A 00 5C 00-42 00 49 00 4E 00 5C 00	S·N·Z·\·B·I·N·\·

Figure 25. CSNZ acronym after volatile analysis

35f13cb0	65 78 6F 6E 3A 20 5A 6F-6D 62 69 65 73 20 45 55	exon: Zombies EU
35f13cc0	4C 41 00 00 00 00 00 00-43 6F 75 6E 74 65 72 20	LA·-·-·-·-·Counter
35f13cd0	53 74 72 69 6B 65 20 4E-65 78 6F 6E 3A 20 5A 6F	Strike Nexon: Zo
35f13ce0	6D 62 69 65 73 20 43 6F-6E 74 65 6E 74 00 00 00	mbies Content·-·-

Figure 26. GameName after volatile analysis

42133c40	43 4F 4E 00 00 00 06 00-00 00 4F 74 68 65 72 54	CON·-·-·-·-·OtherT
42133c50	49 54 32 00 00 00 1F 00-00 00 43 6F 75 6E 74 65	IT2·-·-·-·-·Counte
42133c60	72 20 53 74 72 69 6B 65-20 4F 6E 6C 69 6E 65 20	r Strike Online
42133c70	2D 20 53 74 75 64 69 6F-54 50 45 31 00 00 00 10	- StudioTPE1·-·-

Figure 27. CSNZ mode after volatile analysis

4.2.2. Volatile memory analysis with Volatility

The network connections, as well as the processes used, can be obtained with volatility. First of all, to retrieve information with volatility, we have to select the proper profile. The command to find the profile is:

```
volatility_2.6_win64_standalone.exe imageinfo -f <memoryimage>
```

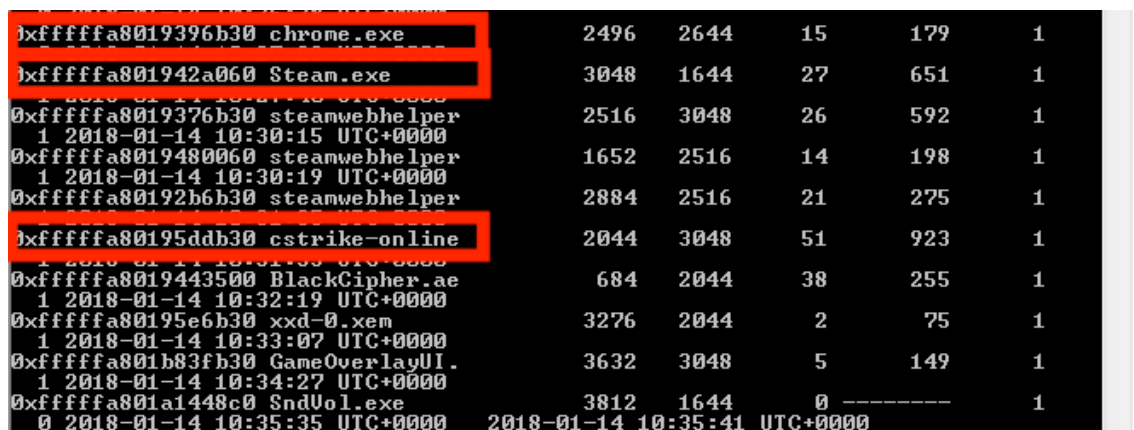
In our study, the profile was Windows7SP1x64.

▪ List of processes

The command for finding the processes executed are:

```
volatility_2.6_win64_standalone.exe pslist --profile=<profile> -f
<memoryimage>
```

To assure that the game was played, it should appear a reference of Steam and then of Counter Strike in the output. Additionally, if a web browser was used, it could indicate that the user was doing a live chat on YouTube.



0xfffffa8019396b30	chrome.exe	2496	2644	15	179	1
0xfffffa801942a060	Steam.exe	3048	1644	27	651	1
0xfffffa8019376b30	steamwebhelper	2516	3048	26	592	1
0xfffffa8019480060	steamwebhelper	1652	2516	14	198	1
0xfffffa80192b6b30	steamwebhelper	2884	2516	21	275	1
0xfffffa80195ddb30	cstrike-online	2044	3048	51	923	1
0xfffffa8019443500	BlackCipher.ae	684	2044	38	255	1
0xfffffa80195e6b30	xxd-0.xem	3276	2044	2	75	1
0xfffffa801b83fb30	GameOverlayUI.	3632	3048	5	149	1
0xfffffa801a1448c0	SndVol.exe	3812	1644	0	-----	1

Figure 28. List of processes after volatile analysis

▪ Network connections

The network connections can be found as well in volatile memory, apart from the network capture. The command for finding the processes executed are:

```
volatility_2.6_win64_standalone.exe netscan --profile=<profile> -f
<memoryimage>
```

To assure that the game was played, it should appear networks connections with respect to Steam and then to Counter Strike in the output. Additionally, if a web browser was used, it could indicate that the user was doing a chat of YouTube Live Streaming.

0x7f9d6520	804	Socket.exe	2018-01-14 11:09:37	010+0000
LISTENING	TCPv4	127.0.0.1:27060		0.0.0.0:0
0x7f839cf0	3048	Steam.exe		
CLOSED	TCPv4	127.0.0.1:49422		56.219.93.25:0
0x7f8908b0	3048	Steam.exe		
CLOSED	TCPv4	127.0.0.1:49422		127.0.0.1:6672
0x7f8b8010	336	ManagementAgen		
CLOSED	TCPv4	127.0.0.1:49422		104.160.66.25:0
0x7f8d1670	2644	chrome.exe		
CLOSED	TCPv4	127.0.0.1:49422		56.251.131.27:0
0x7f8d4a90	2044	cstrike-online		
CLOSE_WAIT	TCPv4	127.0.0.1:49422		216.58.210.138:443
0x7f8ed550	2044	cstrike-online		
CLOSED	TCPv4	127.0.0.1:49422		104.160.66.25:0
	336	ManagementAgen		

Figure 29. Network connections after volatile analysis

4.3. Disk analysis

We performed a post-mortem acquisition with FTK Imager installed in an external USB. This process is explained in Annex 4. The images were saved as E01 instead of RAW. The reason why we took that decision was to avoid any problems in court as with raw, it is a copy byte per byte, so the metadata can be manipulated, with E01 it uses compression and also it contains a separate metadata file. The later analysis of the disk of both cases leads us to the methodology that we describe below. The images were added as evidences in FTK Imager and the files analyzed were chosen based on other academic studies like [32].

4.3.1. Windows Registry

In NTUSER.DAT and SOFTWARE hives we found some references to the game that shows that it was installed in the system, like the path where it was stored. Besides, from a forensic point of view, it is relevant the date times as they are useful for building a timeline of the case.

In order to find this information, extract those hives with FTK Imager and open them with Access Data Registry Viewer. NTUSER.DAT is stored in `\Users\<username>` while SOFTWARE is stored in `SystemRoot\System32\config`.

- **NTUSER.DAT**

In this hive, there must be a folder called *Software*. The path to the CSNZ game should be `NTUSER.DAT\Software\Valve\Steam\Apps\273310`. The description of it will show that it corresponds to Counter Strike Nexon Zombies. The last written time corresponds to the last time the game was played.

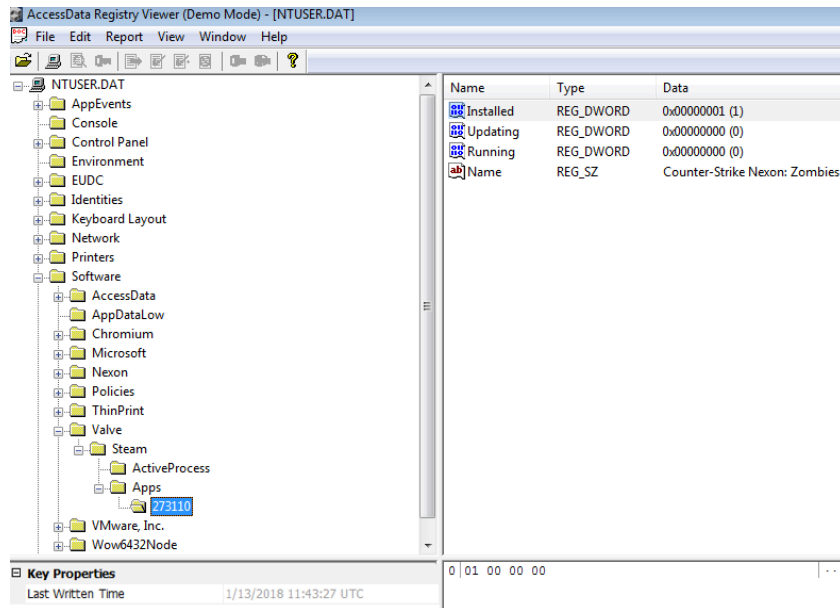


Figure 30. CSNZ (I) in NTUSER.DAT

Additionally, in the path *NTUSER.DAT\Software\Nexon\CStrike-Online* it also appears the last written time when the game was played.

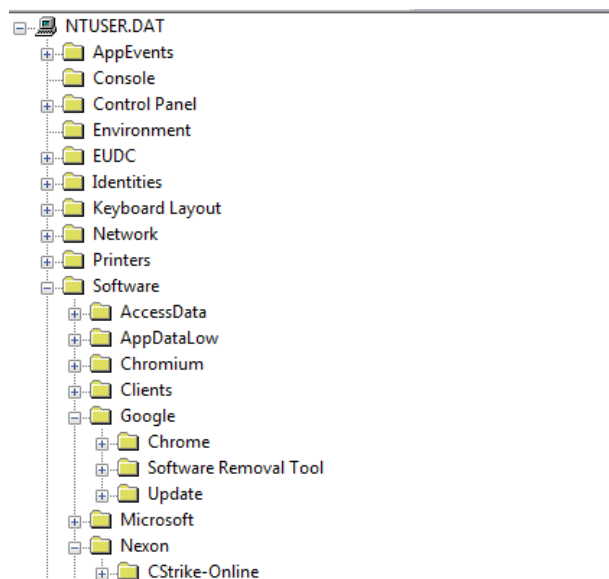


Figure 31. CSNZ (II) in NTUSER.DAT

Regarding the chat of YouTube Live Streaming, it is needed that the user has installed a web browser on the system. Consequently, there should be an entity in the hive. For example, in our case, the path to Google Chrome Browser is:

NTUSER.DAT\Software\Google\Chrome

▪ SOFTWARE

In this hive, there is an important reference to the game that shows when it was installed in the system. The way to find this information is by looking at the last written time of the Uninstalling CSNZ software, as this program was stored in the system at the same time the game was installed. The path of the uninstalling software is:

SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstalled\Steam App 273110

Additionally, the path to the game also appears in this hive in the description of the following entity:

SOFTWARE\Wow6432Node\Valve\Steam\Apps\273110

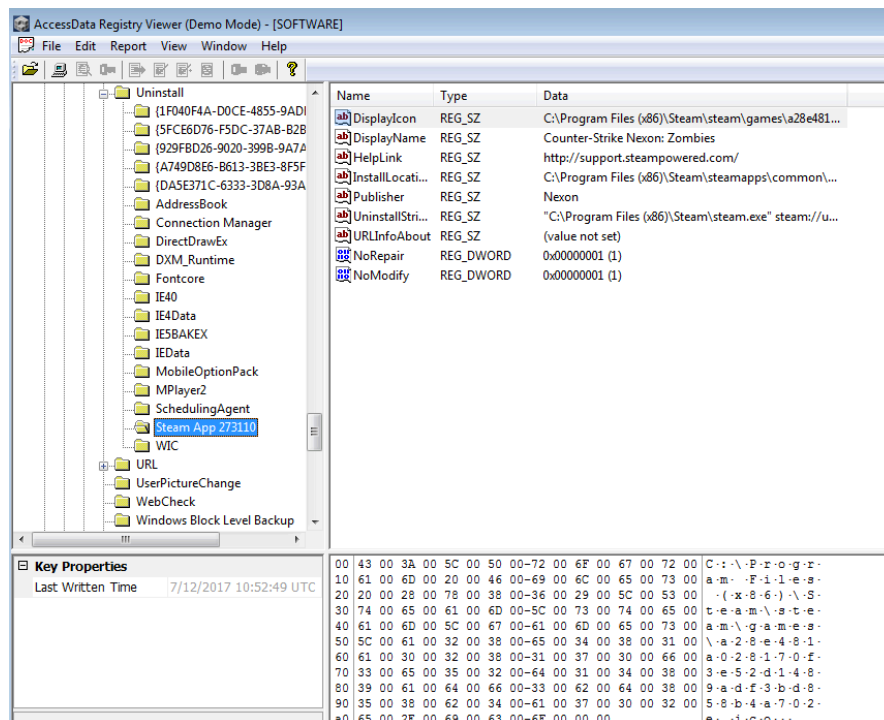


Figure 32. CSNZ (I) in SOFTWARE

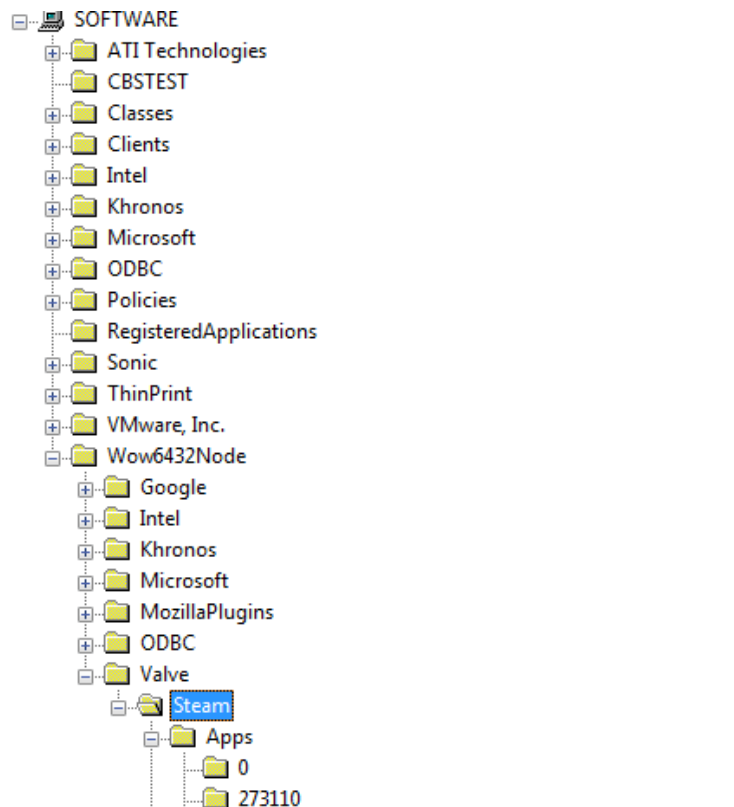


Figure 33. CSNZ (II) in SOFTWARE

4.3.2. Shortcuts

By default, there should be a shortcut to Steam on the Desktop. Steam is needed for executing CSNZ. Besides, there could be as well a shortcut of the game itself. However, the user has to execute first Steam and then CSNZ. It can be found directly with FTK Imager under the path *Users\<Username>\Desktop*:

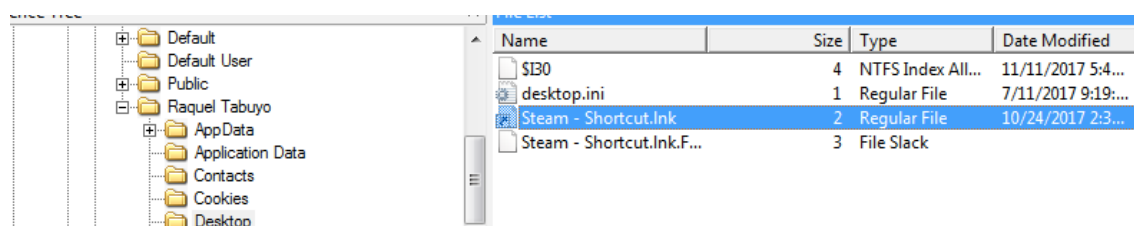


Figure 34. Steam shortcut

4.3.3. Prefetch

The Prefetch folder is stored in *\Windows\Prefetch* and it can be extracted with FTK Imager. After its extraction, open it with WinPrefetchView. There will be references to

Steam and CSNZ video-game. From a forensic point of view, it is relevant as it shows all the affected programs and the date-times. Consequently, we can assure that the game depends on Steam and we correlate facts between them. Besides, we can see that Counter Strike Nexon Zombies video-game is related to NAR files. The analysis of these files is explained in section 4.3.7.

CSTRIKE-ONLINE.EXE...	11/20/2017 4:30...	1/13/2018 12:32...	106,044	CSTRIKE-ONLINE...	C:\PROGRAM FILES (x86)\STEAM\STEAMA...	30	1/13/2018 12:32:14 PM	Yes
CVTRES.EXE-6280F3A...	7/12/2017 12:05...	11/17/2017 12:10...	14,358	CVTRES.EXE	C:\Windows\MICROSOFT.NET\FRAMEWO...	10	11/17/2017 12:10:04 PM	No
DEFRAG.EXE-738093E...	7/12/2017 12:55...	1/13/2018 12:01...	17,522	DEFRAG.EXE	C:\Windows\System32\Defrag.exe	6	1/13/2018 12:00:53 PM	No
DEVICESDISPLAYOBJE...	1/13/2018 1:18:0...	1/13/2018 1:18:0...	36,486	DEVICESDISPLAYOB...	C:\Windows\System32\DEVICESDISPLAYOB...	1	1/13/2018 1:17:59 PM	No
DFSVC.EXE-CC3A03FB...	1/10/2018 8:15:5...	1/10/2018 8:15:5...	162,872	DFSVC.EXE	C:\Windows\MICROSOFT.NET\FRAMEWO...	1	1/10/2018 8:15:43 PM	No
DINOTIFY.EXE-06EB7...	1/10/2018 7:46:0...	1/10/2018 7:46:0...	15,196	DINOTIFY.EXE	C:\Windows\System32\dinotify.exe	1	1/10/2018 7:45:56 PM	No
DLLHOST.EXE-6202E8...	12/9/2017 1:27:2...	12/9/2017 1:27:2...	89,028	DLLHOST.EXE	C:\Windows\System32\dlhost.exe	1	12/9/2017 1:27:15 PM	No
DLLHOST.EXE-893DD...	7/11/2017 11:14...	1/13/2018 1:21:3...	17,788	DLLHOST.EXE	C:\Windows\System32\dlhost.exe	164	1/13/2018 1:21:29 PM	No
DLLHOST.EXE-98FD0...	1/10/2018 7:45:5...	1/12/2018 5:53:4...	38,476	DLLHOST.EXE	C:\Windows\System32\dlhost.exe	3	1/12/2018 5:53:30 PM	No
DLLHOST.EXE-F1E57A...	1/10/2018 7:45:5...	1/10/2018 7:45:5...	16,766	DLLHOST.EXE	C:\Windows\System32\dlhost.exe	1	1/10/2018 7:45:36 PM	No
Filename	Full Path	Device Path	Index					
SMFT	C:\Windows\SysWOW64\apphelp.dll	\DEVICE\HARDDISKVOLUME1\SMFT	107					
ADVAPI32.DLL	C:\Windows\SysWOW64\advapi32.dll	\DEVICE\HARDDISKVOLUME1\WIND...	24					
APISETSCHEMA.DLL	C:\Windows\System32\APISETSCHE...	\DEVICE\HARDDISKVOLUME1\WIND...	8					
APPHELP.DLL	C:\Windows\SysWOW64\apphelp.dll	\DEVICE\HARDDISKVOLUME1\WIND...	106					
CFGMR32.DLL	C:\Windows\SysWOW64\cfgmgr32.dll	\DEVICE\HARDDISKVOLUME1\WIND...	49					
CLBCATQ.DLL	C:\Windows\SysWOW64\clbcatq.dll	\DEVICE\HARDDISKVOLUME1\WIND...	32					
CONFIG.BC	C:\PROGRAM FILES (x86)\STEAM\ST...	\DEVICE\HARDDISKVOLUME1\PROG...	105					
CRYPT32.DLL	C:\Windows\SysWOW64\crypt32.dll	\DEVICE\HARDDISKVOLUME1\WIND...	42					
CRYPTBASE.DLL	C:\Windows\SysWOW64\CRYPTBASE...	\DEVICE\HARDDISKVOLUME1\WIND...	17					
CRYPTSP.DLL	C:\Windows\SysWOW64\cryptsp.dll	\DEVICE\HARDDISKVOLUME1\WIND...	103					
CSTRIKE-ONLINE.EXE	C:\PROGRAM FILES (x86)\STEAM\ST...	\DEVICE\HARDDISKVOLUME1\PROG...	11					
CSTRIKE.NAR	C:\PROGRAM FILES (x86)\STEAM\ST...	\DEVICE\HARDDISKVOLUME1\PROG...	108					
CSTRIKE_NA.NAR	C:\PROGRAM FILES (x86)\STEAM\ST...	\DEVICE\HARDDISKVOLUME1\PROG...	90					
CSTRIKE_NA_BR.NAR	C:\PROGRAM FILES (x86)\STEAM\ST...	\DEVICE\HARDDISKVOLUME1\PROG...	100					
CSTRIKE_NA_DE.NAR	C:\PROGRAM FILES (x86)\STEAM\ST...	\DEVICE\HARDDISKVOLUME1\PROG...	94					
CSTRIKE_NA_EN.NAR	C:\PROGRAM FILES (x86)\STEAM\ST...	\DEVICE\HARDDISKVOLUME1\PROG...	89					
CSTRIKE_NA_ES.NAR	C:\PROGRAM FILES (x86)\STEAM\ST...	\DEVICE\HARDDISKVOLUME1\PROG...	99					
CSTRIKE_NA_FR.NAR	C:\PROGRAM FILES (x86)\STEAM\ST...	\DEVICE\HARDDISKVOLUME1\PROG...	97					
CSTRIKE_NA_PL.NAR	C:\PROGRAM FILES (x86)\STEAM\ST...	\DEVICE\HARDDISKVOLUME1\PROG...	96					
CSTRIKE_NA_RU.NAR	C:\PROGRAM FILES (x86)\STEAM\ST...	\DEVICE\HARDDISKVOLUME1\PROG...	95					
CSTRIKE_NA_TR.NAR	C:\PROGRAM FILES (x86)\STEAM\ST...	\DEVICE\HARDDISKVOLUME1\PROG...	98					

Figure 35. CSNZ in Prefetch

4.3.4. Jump Lists

Extract Jump Lists with FTK Imager and opens them with JumpListView. It is stored inside:

%APPDATA%\Microsoft\Windows\Recent\AutomaticDestinations\[AppID].automaticDestinations-ms

Thanks to this Jump Lists you can see the recent documents. Forensically, it is useful as it could lead to interesting files that are not considered. In our case, after checking them, we realized that Counter Strike Nexon Zombies stores by default screenshots taken during the game rounds and videos, the default folder is “Documents”. Analyzing Jump Lists are useful to find those locations, as the user could change the default ones. These screenshots and video files are studied in the following section.

File Name	Full Path	Record Time	Created Time	Modified Time	Accessed Time	File Attributes	File Size	Entry ID	Application ID	Application Name	File Extension	Computer Name
[7BF3955E-3B9...	...D6EE0668-A00A-4407-9371-BE804C386831\31\7BF3955E-3B90-4184-BD14-5307C15F1FC	11/4/2017 10:28:37...						5	7e4dca80246863a3	Control Panel (I)		
[8E98FC9-8EC...	...D6EE0668-A00A-4407-9371-BE804C386831\31\8E98FC9-8EC-40F6-9158-FACAD07000D0	12/9/2017 5:30:19...						3	7e4dca80246863a3	Control Panel (I)		
[8B96C84-C29...	...D6EE0668-A00A-4407-9371-BE804C386831\31\8B96C84-C295-4F75-8A99-C8058477E1E1	11/4/2017 9:28:34...						1	7e4dca80246863a3	Control Panel (I)		
BlackCallLog	C:\Program Files (x86)\Steam\steamapps\common\CSNZ\Bin\BlackCIPHER\BlackCallLog	18/6/2017 11:00:54...	7/12/2017 3:53:29...	7/12/2017 4:35:23...	7/12/2017 3:53:29...	A	4,277	1	9d4c4f85c234c2b	Notepad (8-4)	log	win-6gat3714
BlackCIPHER	C:\Program Files (x86)\Steam\steamapps\common\CSNZ\Bin\BlackCIPHER	18/6/2017 11:00:54...	7/12/2017 3:53:29...	7/12/2017 3:53:30...	7/12/2017 3:53:30...	D	4,096	5	1b4d6729cb3362	Windows Explorer Pinne...		win-6gat3714
Desktop	C:\Users\Raquel Tabuyo\Desktop	11/11/2017 9:25:28...	7/11/2017 11:14:09...	11/11/2017 9:25:28...	11/11/2017 9:25:28...	RD		7	1b4d6729cb3362	Windows Explorer Pinne...		win-6gat3714
Devin Sakhu Info	C:\Users\Raquel Tabuyo\Desktop\Windows 7 Ultimate by Devin Sakhu\Devin Sakhu Info	10/24/2017 12:02:4...	10/24/2017 12:02:4...	8/12/2009 9:43:41 PM	10/24/2017 12:02:4...	A	1,518	1	1b4d6729cb3362	Windows Explorer Pinne...	info	win-6gat3714
Documents Library	C:\Users\Raquel Tabuyo\AppData\Roaming\Microsoft\Windows\Libraries\Documents Library	7/11/2017 11:20:22...	7/11/2017 11:19:02...	7/11/2017 11:19:33...	7/11/2017 11:19:33...	A	3,583	1	1b4d6729cb3362	Windows Explorer Pinne...	library-ms	win-6gat3714
Music Library	C:\Users\Raquel Tabuyo\AppData\Roaming\Microsoft\Windows\Libraries\Music Library	7/11/2017 11:20:22...	7/11/2017 11:19:02...	7/11/2017 11:19:33...	7/11/2017 11:19:33...	A	3,540	3	1b4d6729cb3362	Windows Explorer Pinne...	library-ms	win-6gat3714
Reaming\Shots	C:\Users\Raquel Tabuyo\AppData\Roaming\Microsoft\Windows\Libraries\Reaming\Shots	7/11/2017 11:20:22...	7/11/2017 11:19:02...	7/11/2017 11:19:33...	7/11/2017 11:19:33...	A	3,540	3	1b4d6729cb3362	Windows Explorer Pinne...	library-ms	win-6gat3714
Screen Shot	C:\Users\Raquel Tabuyo\Documents\Counter-Strike Nexon Zombies\Screen Shot	12/6/2017 12:29:36...	10/24/2017 2:36:48...	12/7/2017 12:29:36...	12/7/2017 12:29:36...	D	4,096	5	1b4d6729cb3362	Windows Explorer Pinne...		win-6gat3714
Videos Capture	C:\Users\Raquel Tabuyo\Documents\Counter-Strike Nexon Zombies\Videos Capture	12/9/2017 1:28:00...	10/24/2017 2:36:48...	12/9/2017 11:57:21...	12/9/2017 11:57:21...	D	4,096	8	1b4d6729cb3362	Windows Explorer Pinne...		win-6gat3714

Figure 36. CSNZ in Jump Lists

4.3.5. Documents folder

Counter Strike Nexon Zombies stores by default the screenshots taken by the user during an in-game round. Besides, it makes screenshots automatically when the round finishes.

Concerning the YouTube Live Streaming, it stores the streamed video in MP4 format once the user finishes doing the streaming. From a forensic point of view, it is a focal point of valuable information as the chat messages can be seen and voice messages can be listened directly from this file.

Inside Documents folder, there are two directories: one for Zombie mode and the other for Studio mode (Counter Strike Online). The directories with the screenshots and the videos are stored in those folders. The screenshots can be seen directly with FTK Imager but also can be exported and open with an Image Viewer program.

Name	Size	Type	Date Modified
SI30	8	NTFS Index All...	1/12/2018 6:37:...
2017_1030_2018_33_0.j...	91	Regular File	10/30/2017 6:1...
2017_1030_2018_33_0.j...	2	File Slack	
2017_1030_2353_28_0.j...	13	Regular File	10/30/2017 9:5...
2017_1030_2353_28_0.j...	4	File Slack	
2017_1125_1917_02_0.j...	139	Regular File	11/25/2017 5:1...
2017_1125_1917_02_0.j...	2	File Slack	
2017_1207_0025_23_0.j...	149	Regular File	12/6/2017 10:2...
2017_1207_0025_23_0.j...	4	File Slack	
2017_1207_0037_19_0.j...	102	Regular File	12/6/2017 10:3...
2017_1207_0037_19_0.j...	3	File Slack	
2017_1207_0037_20_0.j...	103	Regular File	12/6/2017 10:3...
2017_1207_0037_20_0.j...	2	File Slack	
2017_1207_0114_20_0.j...	111	Regular File	12/6/2017 11:1...
2017_1207_0114_20_0.j...	2	File Slack	

Figure 37. CSNZ in Documents



Figure 38. Screenshot in Documents

The video files have to be exported from the *Video Capture* folder in order to be watched. From the network analysis part, we found that the encoder was from VLC; therefore, we opened the video recorded of the YouTube Live Streaming with VLC media player and we can see the whole YouTube chat. The first 9 characters of the name of the files are the dates when the videos were recorded.

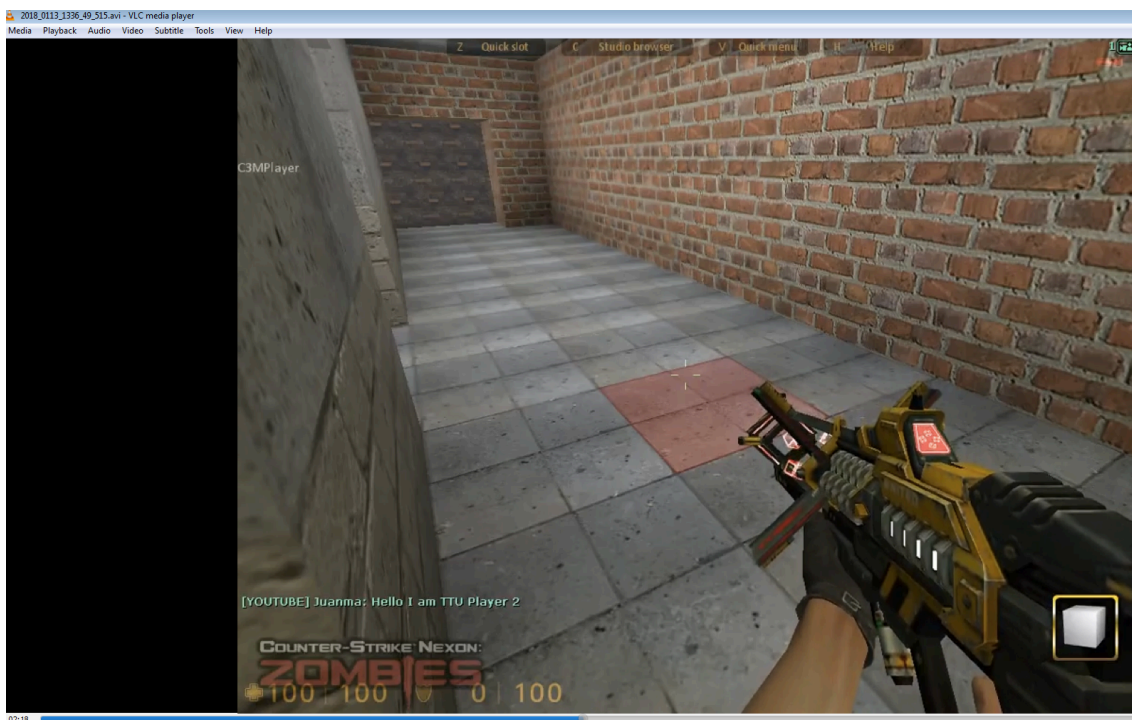
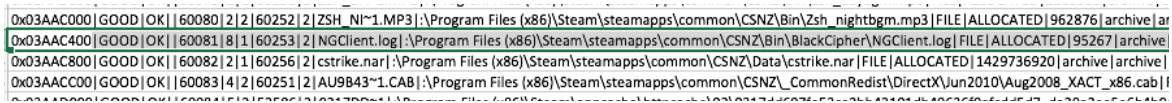


Figure 39. Streamed video in Documents

4.3.6. MFT

We can extract the MFT with FTK Imager and convert it to CSV with Mft2csv tool. With the CSV file, we can search for references about Counter Strike and find where the data is stored.



0x03AAC000	GOOD	OK	60080	2	2	60252	2	ZSH_NI~1.MP3	:\Program Files (x86)\Steam\steamapps\common\CSNZ\Bin\Zsh_nightbgm.mp3	FILE	ALLOCATED	962876	archive	ar
0x03AAC400	GOOD	OK	60081	8	1	60253	2	NGClient.log	:\Program Files (x86)\Steam\steamapps\common\CSNZ\Bin\BlackCipher\NGClient.log	FILE	ALLOCATED	95267	archive	ar
0x03AAC800	GOOD	OK	60082	2	1	60256	2	cstrike.nar	:\Program Files (x86)\Steam\steamapps\common\CSNZ\Data\cstrike.nar	FILE	ALLOCATED	1429736920	archive	archive
0x03AAC000	GOOD	OK	60083	4	2	60251	2	AU9B43~1.CAB	:\Program Files (x86)\Steam\steamapps\common\CSNZ\CommonRedist\DirectX\Jun2010\Aug2008_XACT_x86.cab	FILE	ALLOCATED	1429736920	archive	archive

Figure 40. CSNZ in MFT

Thanks to the analysis of the MFT, we found that by default CSNZ folder is stored in the system in the following path: *\Program Files (x86)\Steam\steamapps\common\CSNZ*

In addition, there are also two directories related to the game in the following paths:

\Program Files (x86)\Steam

\Program Data\Nexon\Common

4.3.7. CSNZ & Steam dedicated folders

CSNZ NAR Files

From the Prefetch analysis, we found that CSNZ uses the NAR files (specific CSNZ file format). We analyzed them with NAR extractor tool, after obtaining them from *\Program Files (x86)\Steam\steamapps\common\CSNZ\Data\cstrike.nar*. However, those files are used for modeling the game but they do not give any information about the user or that could be useful in the forensic case.

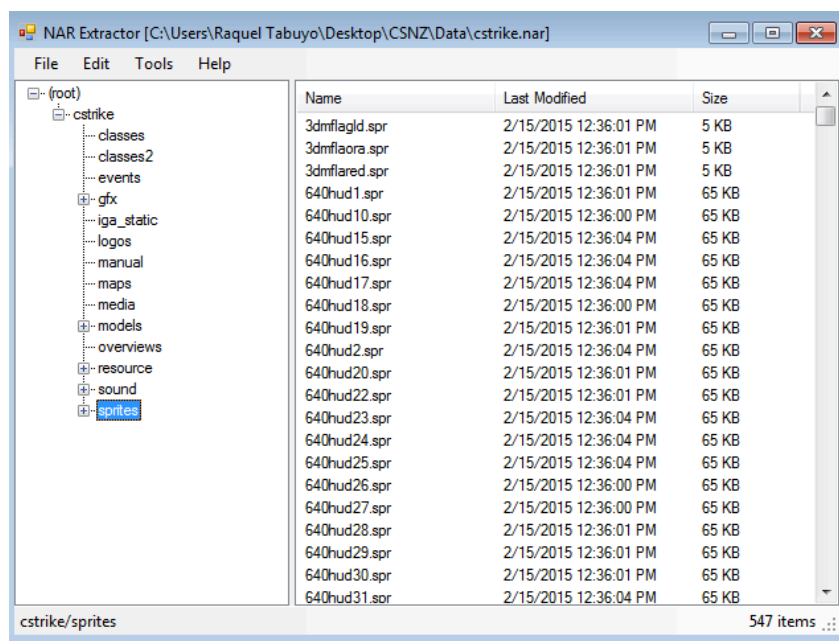


Figure 41. CSNZ dedicated folder

There is a configuration file that shows the GameID in Steam; it is important to know this number as some artifacts can only be found with this identification number. The path to this file is *\Program Files (x86)\Steam\steamapps\common\CSNZ\Bin\steam_appid.txt*

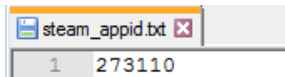


Figure 42. GameID log file

Nexon folder

The analysis of the Nexon folder gives handy forensic information as there are stored all the time-stamps with their corresponding activities in a log file, some of them are related with the messenger feature of the game, so we can assure that the user was chatting at a specific time period. CSNZ loads a messenger module when a player uses the chatting feature of it. In the log, there are references to this module and its associated time-stamps.

This file is stored in *\Program Data\Nexon\Common\nmcogame.log*.

```

2018/01/13 13:32:44, NMCO_SetLocaleAndRegion( 768, 600 ): ID & Code are not valid!
2018/01/13 13:32:44, Fail to load messenger module!
    Version file URL: http://platform.nexon.com/Messenger/version.xml
    Patch option: 0
    Use NGM: 0
    Messenger module path: C:\Program Files (x86)\Steam\steamapps\common\CSNZ\bin\nmconew.dll
2018/01/13 13:32:57, Fail to load messenger module!
    Version file URL: http://platform.nexon.com/Messenger/version.xml
    Patch option: 0
    Use NGM: 0
    Messenger module path: C:\Program Files (x86)\Steam\steamapps\common\CSNZ\bin\nmconew.dll
2018/01/13 13:32:58, Fail to load messenger module!
    Version file URL: http://platform.nexon.com/Messenger/version.xml
    Patch option: 0
    Use NGM: 0
    Messenger module path: C:\Program Files (x86)\Steam\steamapps\common\CSNZ\bin\nmconew.dll
2018/01/13 13:32:58, Fail to load messenger module!
    Version file URL: http://platform.nexon.com/Messenger/version.xml
    Patch option: 0
    Use NGM: 0
    Messenger module path: C:\Program Files (x86)\Steam\steamapps\common\CSNZ\bin\nmconew.dll
2018/01/13 13:32:59, Fail to load messenger module!
    Version file URL: http://platform.nexon.com/Messenger/version.xml
    Patch option: 0
    Use NGM: 0
    Messenger module path: C:\Program Files (x86)\Steam\steamapps\common\CSNZ\bin\nmconew.dll
2018/01/13 13:43:22, Fail to load messenger module!
    Version file URL: http://platform.nexon.com/Messenger/version.xml
    Patch option: 0
    Use NGM: 0
    Messenger module path: C:\Program Files (x86)\Steam\steamapps\common\CSNZ\bin\nmconew.dll
2018/01/13 13:43:22, Fail to load messenger module!
    Version file URL: http://platform.nexon.com/Messenger/version.xml
    Patch option: 0
    Use NGM: 0
    Messenger module path: C:\Program Files (x86)\Steam\steamapps\common\CSNZ\bin\nmconew.dll
2018/01/13 13:43:22, Fail to load messenger module!
    Version file URL: http://platform.nexon.com/Messenger/version.xml
    Patch option: 0

```

Figure 43. Nexon log file

Steam folder

Inside the Steam folder, there are some configuration and log files that provide information such as the Steam UserID, the Steam Username, the ID of the game, etc. The way how to obtain this data is explained below.

The **configuration files** are stored in `\Program Files (x86)\Steam\config\`, they have an extension of *VDF* and can be opened with Notepad++:

- *config.vdf*: contains the userID and the username.
- *loginusers.vdf*: contains the users who logged into Steam with the system acquired. It shows the username, the nickname, userID and the last time accessed, which has to be transformed from EPOCH to UTC time.

```

"Accounts"
{
  "thttu"
  {
    "SteamID"      "76561198404618625"
  }
}

```

Figure 44. Configuration file in Steam folder

```

loginusers.vdf
{
  "users"
  {
    "76561198404618625"
    {
      "AccountName"      "thttu"
      "PersonaName"      "TTU - Thesis"
      "RememberPassword" "0"
      "mostrecent"       "1"
      "Timestamp"        "1515843123"
      "WantsOfflineMode" "0"
    }
  }
}

```

Figure 45. Logged users in Steam folder

With the userID, visit <http://steamcommunity.com/profiles/<Steam User ID>>, as it is described in the network analysis section of this chapter, and it is possible to obtain more forensic information such as the last time played of the game, online status, etc.

Additionally, there is another configuration file stored in `\Program Files (x86)\Steam\userdata\<localIDnumber>\localconfig.vdf`: that links undoubtedly the user with the game as it contains the GameID and the username. Next to the GameID, it appears the last played time. However, it appears as the EPOCH time, so it should be transformed to UCT time.

```

"444352897"
{
  "NameHistory"
  {
    "0" "TTU - Thesis"
  }
  "name" "TTU - Thesis"
  "PersonaName" "TTU - Thesis"
}

"Software"
{
  "Valve"
  {
    "Steam"
    {
      "Apps"
      {
        "273110"
        {
          "LastPlayed" "1510506224"
          "ViewedLaunchEULA" "1"
          "BadgeData" "020000000800"
        }
      }
      "LastPlayedTimesSyncTime" "1510506224"
      "PlayerLevel" "0"
    }
  }
}

```

Figure 46. Player and Game log

The **remote connection file** is stored in `\Program Files (x86)\Steam\logs\remote connections.txt`, it is a TXT file that contains all the remote connections with the PC used by the user, it is possible to see with whom the user is playing (machine name and the IP).

```
[2018-01-12 20:55:18] Loaded client id: 16888911976095687970
[2018-01-12 20:55:18] Listening for broadcast on: 27036
[2018-01-12 20:55:18] Received broadcast message from client 7108249717000450312 (Juanma-PC): 10.10.1.23:27036
[2018-01-12 20:55:21] Received broadcast message from client 7108249717000450312 (Juanma-PC): 10.10.1.23:27036
[2018-01-12 20:55:41] Received broadcast message from client 7108249717000450312 (Juanma-PC): 10.10.1.23:27036
[2018-01-12 20:56:12] Received offline message from client 7108249717000450312
[2018-01-12 20:57:24] Received broadcast message from client 7108249717000450312 (Juanma-PC): 10.10.1.23:27036
[2018-01-12 20:57:24] Received broadcast message from client 7108249717000450312 (Juanma-PC): 10.10.1.23:27036
[2018-01-12 20:57:24] Received discovery message from client 7108249717000450312
[2018-01-12 20:57:26] Received broadcast message from client 7108249717000450312 (Juanma-PC): 10.10.1.23:27036
[2018-01-12 20:57:27] Received discovery message from client 7108249717000450312
[2018-01-12 20:57:36] Received discovery message from client 7108249717000450312
[2018-01-12 20:57:42] Received broadcast message from client 7108249717000450312 (Juanma-PC): 10.10.1.23:27036

[2018-01-13 13:10:48] Loaded client id: 16888911976095687970
[2018-01-13 13:10:48] Listening for broadcast on: 27036

[2018-01-13 13:31:56] Loaded client id: 16888911976095687970
[2018-01-13 13:31:56] Listening for broadcast on: 27036
[2018-01-13 13:31:56] Received broadcast message from client 7108249717000450312 (Juanma-PC): 10.10.1.22:27036
[2018-01-13 13:32:01] Received broadcast message from client 7108249717000450312 (Juanma-PC): 10.10.1.22:27036
[2018-01-13 13:32:25] Received broadcast message from client 7108249717000450312 (Juanma-PC): 10.10.1.22:27036
[2018-01-13 13:59:41] Received broadcast message from client 7108249717000450312 (Juanma-PC): 10.10.1.22:27036
[2018-01-13 13:59:41] Received discovery message from client 7108249717000450312
[2018-01-13 13:59:43] Received discovery message from client 7108249717000450312
[2018-01-13 13:59:46] Received broadcast message from client 7108249717000450312 (Juanma-PC): 10.10.1.22:27036
[2018-01-13 14:00:09] Received broadcast message from client 7108249717000450312 (Juanma-PC): 10.10.1.22:27036
```

Figure 47. Connection file in Steam folder

Additionally, with respect to network connections, there is a database file with some cookies stored in the following path:

`Users\<user>\AppData\Local\Steam\htmlcache\Cookies`

It is an SQLite database that can be extracted with FTK Imager and open with DB Browser for SQLite. There is a table with some sessionIDs and time references in the EPOCH time. This time should be transformed to UCT time. As we saw in the network part, it is possible to do a session cloning if the user is still logged into Steam with these cookies.

Database Structure Browse Data Edit Pragmas Execute SQL										
Table: cookies										
	creation_utc	host_key	name	value	path	expires_utc	secure	httponly	last_access_utc	has
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	13153314182...	store.steamp...	browserid	11544708117...	/	13184850182...	0	0	13155240056...	1
2	13153845354...	www.csnzom...	_icl_current_l...	en	/	13153931754...	0	0	13153845354...	1
3	13154362311...	store.steamp...	recentapps	%7B%222366...	/	13162138311...	0	0	13155240056...	1
4	13154895946...	help.steampo...	timezoneOffset	7200,0	/	13186431946...	0	0	13154896028...	1
5	13155240057...	store.steamp...	timezoneOffset	7200,0	/	13186776057...	0	0	13155240057...	1
6	13155240058...	.steampower...	_ga	GA1.2.694143...	/	13218312058...	0	0	13155240058...	1
7	13155240058...	.steampower...	_gid	GA1.2.197378...	/	13155326458...	0	0	13155240058...	1
8	13155240075...	steamcommu...	timezoneOffset	7200,0	/	13186776075...	0	0	13155240075...	1
9	13155240075...	.steamcommu...	_ga	GA1.2.109112...	/	13218312075...	0	0	13155240075...	1
10	13155240075...	.steamcommu...	_gid	GA1.2.138833...	/	13155326475...	0	0	13155240075...	1

Figure 48. SessionIDs database

4.3.8. Recent files

The Recent folder can show the recent activity of the user related to the game. It can provide information such as the paths to specific files that are stored in the system and that the user could modify if they are the default ones. The Recent folder can be exported with FTK Imager and open with RecentFilesView tool from the path: *Users\<user>\AppData\Roaming\Microsoft\Windows\Recent* . In our case, it provided us information about the screenshots taken and the folder where they are stored.

Filename	Modified Time	Created Time	Execute Time	Missing ...	Stored In	Extension	File Only
C:\Users\Raquel Tabuyo\Documents\Counter-Strike Nexon Zombies\Screen Shot	12/7/2017 12:22:08...	10/30/2017 7:18:33...	12/9/2017 1:27:37 ...	Yes	Recent Folder	Screen Shot	
C:\Users\Raquel Tabuyo\Documents\Counter-Strike Nexon Zombies\Screen Shot\2017_1207_0115_04_0.jpg	12/7/2017 12:15:04...	12/7/2017 12:15:04...	12/9/2017 1:27:37 ...	Yes	Recent Folder	jpg	2017_1207_0115_04_0.jpg
C:\Users\Raquel Tabuyo\Documents\Counter-Strike Nexon Zombies\Screen Shot\2017_1207_0116_24_0.jpg	12/7/2017 12:16:24...	12/7/2017 12:16:24...	12/9/2017 1:27:31 ...	Yes	Recent Folder	jpg	2017_1207_0116_24_0.jpg
C:\Users\Raquel Tabuyo\Documents\Counter-Strike Nexon Zombies\Screen Shot\2017_1207_0116_24_1.jpg	12/7/2017 12:16:24...	12/7/2017 12:16:24...	12/9/2017 1:27:27 ...	Yes	Recent Folder	jpg	2017_1207_0116_24_1.jpg
C:\Users\Raquel Tabuyo\Documents\Counter-Strike Nexon Zombies\Screen Shot\2017_1207_0122_08_0.jpg	12/7/2017 12:22:08...	12/7/2017 12:22:08...	12/9/2017 1:27:23 ...	Yes	Recent Folder	jpg	2017_1207_0122_08_0.jpg
C:\Users\Raquel Tabuyo\Documents\Counter-Strike Nexon Zombies\Screen Shot\2017_1207_0122_08_1.jpg	12/7/2017 12:22:08...	12/7/2017 12:22:08...	12/9/2017 1:27:16 ...	Yes	Recent Folder	jpg	2017_1207_0122_08_1.jpg

Figure 49. CSNZ in Recent folder

4.3.9. Thumbnails

The Thumbcache stored in the path below can be open with ThumbCache Viewer tool:

Users\<user>\AppData\Roaming\Microsoft\Windows\Explorer\thumbcache_.db*

It provides information about the game: where it is stored and the OS under it was executed, in our case it was Windows 7 OS.

#	Filename	Cache Entry Offset	Cache Entry S...	Data Offset	Data Size	Data Checksum	Header Checksum	Cache Entry Hash	System
55	26bc39509ac0ae5	633433 B	0 KB	633513 B	0 KB	0000000000000000	47f5f7a90cd16e10	26bc39509ac0ae5	Windows 7
56	7cdede3a631ca2c6	633513 B	0 KB	633593 B	0 KB	0000000000000000	cf0abd39278b808	7cdede3a631ca2c6	Windows 7
57	4db777912ee82917.png	633593 B	41 KB	633673 B	41 KB	1cf98ee09696cc3	cd4ee30a045cd54	4db777912ee82917	Windows 7
58	d382abdb16b08b0d	675919 B	0 KB	675999 B	0 KB	0000000000000000	0c68e1a440d8c5f9	d382abdb16b08b0d	Windows 7
59	ed1f25521b4df589	675999 B	0 KB	676079 B	0 KB	0000000000000000	9a7cee0ae766ed1	ed1f25521b4df589	Windows 7
60	80928b3a753a89f8	676079 B	0 KB	676159 B	0 KB	0000000000000000	193b62067aa204ea	80928b3a753a89f8	Windows 7
61	442360ac70678185	676159 B	0 KB	676239 B	0 KB	0000000000000000	02297dd0ebb17941a	442360ac70678185	Windows 7
62	1bab49908bb9f2c3	676239 B	0 KB	676319 B	0 KB	0000000000000000	e33a45030fcbfd1a	1bab49908bb9f2c3	Windows 7
63	ce4120a979c3d387	676319 B	0 KB	676399 B	0 KB	0000000000000000	9bdd3a9fa266c6e1	ce4120a979c3d387	Windows 7
64	90989530078a033e.png	676399 B	32 KB	676479 B	32 KB	26e844436203387f	d93be41671e12ac0	90989530078a033e	Windows 7
65	ddef40b19135b11d	710105 B	0 KB	710185 B	0 KB	0000000000000000	3d56496b97c815ce	ddef40b19135b11d	Windows 7
66	a59bb62255f72299	710185 B	0 KB	710265 B	0 KB	0000000000000000	f206a77e9934461	a59bb62255f72299	Windows 7
67	70baa41f4a5126b9	710265 B	0 KB	710345 B	0 KB	0000000000000000	1de3549e7d4f729c	70baa41f4a5126b9	Windows 7
68	4423910b2939d2ee.png	710345 B	38 KB	710425 B	38 KB	976bfa3d5ae8c06b	c3483902d0977fc6	4423910b2939d2ee	Windows 7
69	bf2eaeffec079226	749860 B	0 KB	749940 B	0 KB	0000000000000000	fb2fc1d2fcl8d374	bf2eaeffec079226	Windows 7
70	2d54aae240f82456	749940 B	0 KB	750020 B	0 KB	0000000000000000	3a3146e4cae4c41	2d54aae240f82456	Windows 7
71	::(645FF040-5081-101...	750020 B	0 KB	750148 B	0 KB	0000000000000000	4d8b4199f128fe1d	0924bc51f9b84ee8	Windows 7
72	bf2eaeffec079226	750148 B	0 KB	750228 B	0 KB	0000000000000000	fb2fc1d2fcl8d374	bf2eaeffec079226	Windows 7
73	::(645FF040-5081-101...	750228 B	0 KB	750356 B	0 KB	0000000000000000	4d8b4199f128fe1d	0924bc51f9b84ee8	Windows 7
74	bf2eaeffec079226	750356 B	0 KB	750436 B	0 KB	0000000000000000	fb2fc1d2fcl8d374	bf2eaeffec079226	Windows 7

Figure 50. CSNZ in Thumbnails cache

4.3.10. LogFile

The LogFile contains information about the logged users. It is the same as the one stored in *loginusers.vdf*. However, as it is a file from Windows system, it is more difficult to be modified by the user. *loginusers.vdf* could be edited by the user with a Text Editor so comparing both files will assure the integrity of the data.

```

"users"
    "76561198404618625"
        "AccountName"
        "PersonaName"
        "RememberPassword"
        "mostrecent"
        "Timestamp"
        "WantsOfflineMode"
        "thttu"
        "TTU - Thesis"
        "1"
        "1"
        "1509888855"
        "0"

```

Figure 51. CSNZ in LogFile

4.3.11. Web browsing information

For the YouTube Live Streaming, we used Chrome Browser for chatting. It is important to consider that the browser used could be other rather than Chrome. We found some references about YouTube, but we could not retrieve the chat messages. However, some time's references help to create a timeline of the case. Those files are stored in *Users\<user>\AppData\Google\Chrome\Default* that can be extracted with FTK Imager:

- **Cookies:** open it with DB Browser for SQLite. It will contain sessionIDs.

	creation_utc	host_key	name	value	path	expires_utc	secure	httponly	last_access_utc	has_expires	persistent	priority	encrypted_value
1	13160248164...	.google.com	1P_JAR		/	13162841992...	0	0	13160317442...	1	1	1	BLOB
2	13160248181...	.youtube.com	VISITOR_INF...		/	13181286161...	0	1	13160317424...	1	1	1	BLOB
3	13160248210...	.myaccount.google.com	__utma		/	13223320210...	0	0	13160248210...	1	1	1	BLOB

Figure 52. Cookies in Chrome

- **Cache:** open it with ChromeCacheView, there will be traces about the YouTube Live Streaming activity.

Filename	URL	Content Type	File Size	Last Accessed	Server Time	Server Last Modified	Expire Time	Server Name	Server Response	Content En...	Cache Name
sqps-oaymwE...	https://i.ytimg.com/vi/ovX1HicGdA/hqdefault.jpg?sqps=...	image/jpeg	10,606	1/13/2018 12:44:06...	1/12/2018 6:07:44 ...		1/12/2018 8:07:44 ...	sfte	HTTP/1.1 200		data_3 [4677632]
sqps-oaymwE...	https://i.ytimg.com/vi/U624W1r-8qw/hqdefault.jpg?sqps=...	image/jpeg	13,350	1/13/2018 12:44:04...	1/13/2018 12:24:16...		1/13/2018 2:24:16 ...	sfte	HTTP/1.1 200		data_3 [10854400]
sqps-oaymwE...	https://i.ytimg.com/vi/GmH9FW6U/hqdefault.jpg?sqps=...	image/jpeg	18,162	1/13/2018 12:44:04...	1/13/2018 12:21:59...		1/13/2018 2:21:59 ...	sfte	HTTP/1.1 200		1,000011

Figure 53. YouTube cache in Chrome

- **History:** open it ChromeHistoryView, there are references to the notifications of the streaming, which corresponds to the messages received.

URL	Title	Visit Count	Last Visited Time	Referring Page
http://www.google.es/accounts/Logout?service=youtube&ilo=1&ils=sEE%2Cs.E5&ilc=2&continue=https%3A%2F%2Fwww.youtube.com&sz=1987306324	YouTube	1	1/13/2018 12:24:43...	
http://www.google.com/accounts/Logout?service=youtube&ilo=1&ils=drilts%2Co.notifications.google.com%2Cs.EE%2Cs.E5&ilc=1&continue=https%3A%2F%2Fwww.youtube.com&sz=1835168888	YouTube	1	1/13/2018 12:24:43...	
https://www.google.com/accounts/Logout?service=youtube&ilo=1&ils=drilts%2Co.notifications.google.com%2Cs.EE%2Cs.E5&ilc=1&continue=https%3A%2F%2Fwww.youtube.com&sz=1835168888	YouTube	1	1/13/2018 12:24:43...	

Figure 54. YouTube History in Chrome

Web browsing information with Autopsy

It is easier to find Web information with Autopsy as it divides automatically data related to the web searches. After creating two different cases for each example of study, we found more information with respect to YouTube Live Streaming. Besides, we made a relevant discovery that we decided to include it as a separate section.

The most relevant findings concerning YouTube Live Streaming with Autopsy are:

- **Link to the streamed YouTube video.** After performing the live streaming, the video is automatically uploaded to YouTube. It will be kept in the user's channel unless he manually deletes it. It was found in the *History* file (mentioned

previously). From this file, we can see the YouTube channel name and the links to the videos that he has uploaded to it.

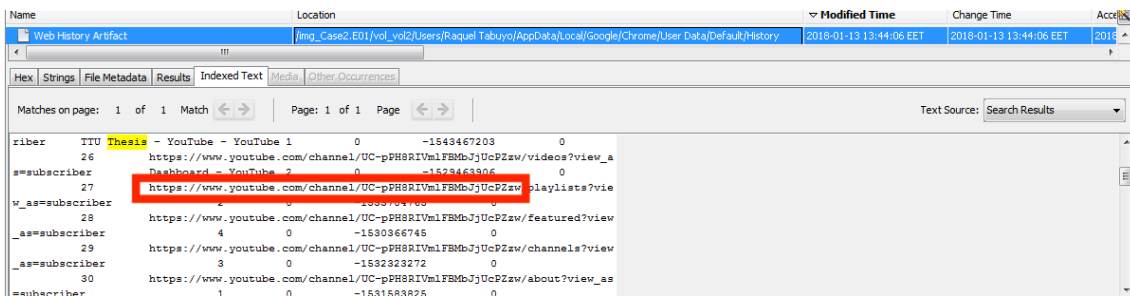


Figure 55. YouTube Channel + link with autopsy

- The **email** used for the Google account to access to YouTube. It is found in the *Login Data* file of the Default folder, commented above.

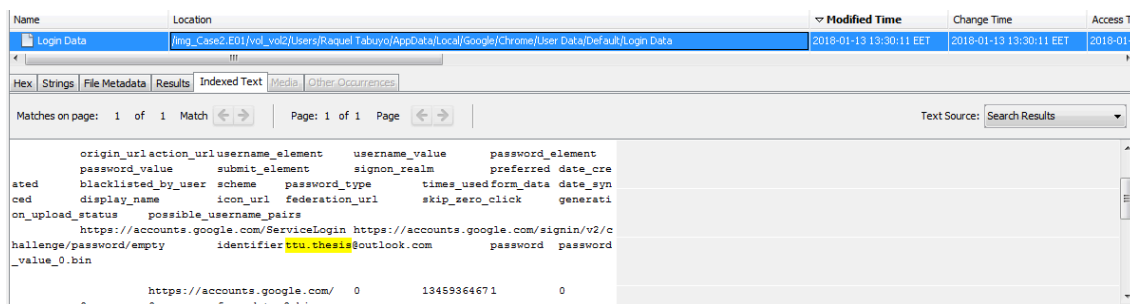


Figure 56. YouTube associated email with autopsy

4.3.12. Analysis with Autopsy

In the previous section, we commented that we found the YouTube Live Streaming link with respect to our user under suspicion, his channel and his associated email. We decided to create another section because it was not related to YouTube but it is a relevant finding that affects both cases under study. We used the same procedure used in [40], based on the keyword search feature that Autopsy provides.

We found the password associated with the Steam account of our suspect. The way to obtain this password is by using as keyword: **password**. Autopsy will search inside the file system data, carved files and unallocated space.

From the picture below, we can see that the password is found in unallocated space and also as a carved file. Recovering the password from those locations can be considered as a pinch of luck. However, there is also a file that must be used to obtain this password, as it is stored locally in the system, even though the user didn't check the option of "remember password". The path to this file is:

Users\<user>\AppData\Local\Steam\htmlcache\Cache\data_1

It can be seen directly with Autopsy in cleartext, as we can see from the picture below. Additionally, there are some links of CSNZ, so we can assure that the user was playing the game at that specific time, as he logged with his credentials.

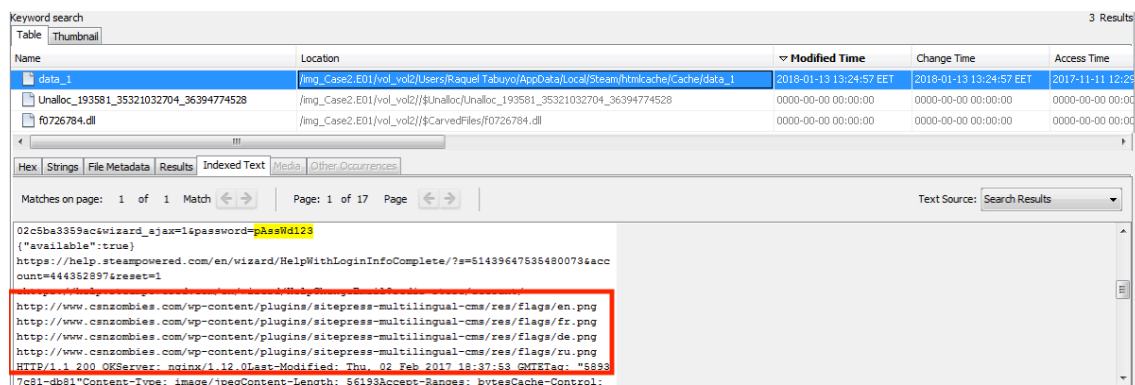


Figure 57. Steam password with Autopsy

Furthermore, Autopsy performs data carving by default and shows the results obtained. However, after analyzing the carved files, we didn't find any more relevant information apart from the password mentioned previously. Besides, as we commented before, we also examined the unallocated space by keyword searching without any important finding rather than the password too.

5. Analysis Results

After applying the methodology developed in this thesis in the two cases of study, we were able to find a set of artifacts that contributes to build a time-line of the case and relate the player with the usage of the chat in an accurate way. Therefore, the causality of our found procedures can be defined as finding enough information about the existence of a communication between players done through the game, with well-established methods and tools. Those procedures can be generalized to other games with similar characteristics (same category) or that need Steam to be run, as the login process is the same for all the games in Steam. The validation was done through these two case scenarios which really reflect how the game can be used for communication in real life.

5.1. Primary artifacts

We found some common artifacts that reflect accurately that it is possible to find specific information from the game. The primary artifacts that should be considered when facing to a real case scenario with respect to a Steam video-game, more specifically to CSNZ, and its chatting features are:

Table 2. Primary artifacts

	Network analysis	Volatile memory analysis (keywords)	Disk analysis
UserID	steamLogin cookie	steamID	\Program Files(x86)\Steam\config\config.vdf
User name	http://steamcommunity.com/profiles/<Steam User ID>	SteamUser	\$LogFile \Program Files(x86)\Steam\config\loginusers.vdf
User password		password=	Users\<user>\AppData\Local\Steam\htmlcache\Cache\data_1
Nickname	http://steamcommunity.com/profiles/<Steam User ID>	PersonaName	\$LogFile \Program Files(x86)\Steam\config\loginusers.vdf \Program Files(x86)\Steam\userdata\<LocalIDnumber>\Localconfig.vdf

List of friends	Session cloning	has logged in	
Chat information		<p><u>Lobby chat:</u> (Type of receiver) [Nickname of the sender] : message</p> <p><u>In-game chat:</u> [Type of receiver] Nickname of the sender : message</p> <p><u>YouTube chat:</u> [YOUTUBE] Nickname of the sender : message</p>	<p><u>Messenger module:</u> \ProgramData\Nexon\Common\nmcogame.Log</p> <p><u>Remote connections:</u> \ProgramFiles(x86)\Steam\Logs\remoteconnections.txt</p>
YouTube Live Streaming traces	<p>connect('live2')</p> <p>rtmp://a.rtmp.youtube.com/live2</p> <p>releaseStream('<key>')</p>	<p>youtube</p> <p>channel_id</p> <p>rtmp.youtube</p> <p>codec</p>	Users\<user>\AppData\Google\Chrome\Default\ History
GameID	http://steamcommunity.com/profiles/<Steam User ID>	steamGameID	\ProgramFiles(x86)\Steam\steamapps\CSNZ\Bin\steam_appid.txt
<p>User time-stamps:</p> <p>- Game installed</p> <p>- Last time accessed</p> <p>- Last time played</p>	<p>http://steamcommunity.com/profiles/<Steam User ID></p>	<p>volatility_2.6_win64_standalone.exe pslist --profile=<profile> -f <memoryimage></p>	<p>SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstalled\Steam App <GameID></p> <p>NTUSER.DAT\Software\Valve\Steam\Apps\<GameID></p> <p>\ProgramFiles(x86)\Steam\config\Loginusers.vdf</p> <p>NTUSER.DAT\Software\Nexon\CStrike-Online</p> <p>\ProgramFiles(x86)\Steam\userdata\<LocalIDnumber>\Localconfig.vdf</p>
History of the game	http://steamcommunity.com/profiles/<Steam User ID>		<p>\ProgramFiles(x86)\Steam\Logs\remoteconnections.txt</p> <p>Documents folder</p>

From the table above, we can see that there are some evidences that proves that the user was using the chatting feature of the game: messenger module logs or remote connections logs (disk analysis) or chat messages sent/received (volatile memory analysis).

5.2. Tables of Results

The results obtained, with respect to each case, after applying the procedures developed are summarized in the following tables. The table in the previous section shows the main artifacts in a more general way; however, these tables show if specific data was found while applying the procedures developed for each case:

Table 3. Case 1 Results

	CASE 1					
	Network	Steam Session cloning	Volatile memory	Disk		
				Steam & CSNZ dedicated folders	Windows registry	Windows system files & folders**
CSNZ Game name	✓	✓	✓	✓	✓	
CSNZ Game mode			✓			
Game acronym			✓	✓		
UserName Steam account		✓	✓	✓		✓
Steam User ID	✓	✓	✓	✓		✓
NickName Steam account	✓	✓	✓	✓		✓
Password Steam account			✓	✓		
Associated email in Steam account		✓	✓			
CSNZ username			✓	✓		
CSNZ Game ID			✓	✓	✓	
Chat messages sent in lobby			✓*			
Chat messages received in lobby			✓*			
CSNZ Player 2			✓	✓		
Zombie in-game room number			✓			
CSNZ family members (list of friends)			✓			

Chat messages sent in Zombie game			✓*			
Chat messages received in Zombie game			✓*			
Game execution logs (last time accessed and last played time)	✓	✓	✓	✓	✓	✓
Network Session IDs	✓	✓		✓		
Network connections	✓		✓	✓		
User status (online/offline)	✓	✓	✓			

Table 4. Case 2 Results

	CASE 2					
	Network	Steam Session cloning	Volatile memory	Disk		
				Steam & CSNZ dedicated folders	Windows registry	Windows system files & folders**
CSNZ Game name	✓	✓	✓	✓	✓	
CSNZ Game mode			✓			
Game acronym			✓	✓		
UserName Steam account		✓	✓	✓		✓
Steam User ID	✓	✓	✓	✓		✓
NickName Steam account	✓	✓	✓	✓		✓
Password Steam account			✓	✓		
Associated email in Steam account		✓	✓			
CSNZ username			✓	✓		
CSNZ Game ID			✓	✓	✓	
Google account email						✓
Google account username			✓			✓
Google account password						

YouTube Live Streaming main server URL	✓		✓			
YouTube Live Streaming encoder	✓		✓			
YouTube Channel name			✓			✓
YouTube Live Streaming key	✓		✓			
YouTube Live Streaming video			✓			✓
YouTube Live Streaming viewer			✓			
Studio in-game room number			✓			
Password Studio room			✓			
YouTube Live Streaming chat messages sent			✓*	✓***		
YouTube Live Streaming chat messages received			✓*	✓***		
CSNZ Player 2			✓	✓		
Game execution logs (last time accessed and last played time)	✓	✓	✓	✓	✓	✓
Network Session IDs	✓	✓		✓		✓
Network connections	✓	✓	✓	✓		✓
YouTube Live Streaming execution logs	✓					✓
User status (online/offline)		✓	✓			

*Chats can only be obtained if the acquisition of volatile memory is done before the user logs out.

**The Windows files and folders considered are: Shortcuts, Prefetch, Jump Lists, LogFile, MFT, Thumbnails, Recent files, Google Chrome Default folder (Cookies, Cache, History and Login Data files).

***Chats can only be obtained if the video is still uploaded to YouTube (the user didn't delete it manually) or if the user didn't disable the option of saving the video in the system.

5.3. Evidences

As we mentioned in previous chapters, we followed the McKemmish forensic framework [42]. In order to assure the integrity of the evidences, we made copies of all of them. Those copies are the ones which were analyzed.

As we want to make a research in the most realistic way possible, we provide in the following table the hashes of all the evidences taken:

Table 5. Evidences

Case name	MD5 hash	Description
Case1.pcap	ec451d2fb12890c4b6cea7dbf3992233	Network capture Case 1
Case1-1.mem	257249692e38dd3d0b52a774beb2f00e	Memory dump Case 1 before the user logs out
Case1-2.mem	1d5b55b11bd250603d4a0742e32cda07	Memory dump Case 1 before the user logs out
Case1.e01	40750e123186f1a8b7bcfaa7aded7b1d	Disk copy of Case 1
Case2.pcap	dc56888f8ba320bdbfbc3b9a8ea624e0	Network capture Case 2
Case2-1.mem	59d7d18a8b472113c96c335a85aab2d7	Memory dump Case 2 before the user logs out
Case2-2.mem	bf41cce069f78cda38247ac0ec9a1d8d	Memory dump Case 2 before the user logs out
Case2.e01	ea2d48011ae501968683727b4bf4269b	Disk copy of Case 2

6. Conclusion and Future Work

In this thesis, we performed an in-depth forensic analysis of Counter Strike Nexon Zombies video-game, more precisely about the online chatting features that this game provides. This game is one of the most played nowadays [1] and allows users to play cooperatively or to fight between them with the possibility to communicate while playing with text or audio messages; but also, players can chat without being in an in-game round. Messaging features offered in online gaming is being used lately for criminal purposes: money laundering, hidden channel communication for terrorists, etc. Additionally, a great number of players are making money thanks to YouTube Live Streaming, by recording themselves while playing a video-game and interacting at the same time with their viewers with the live chat.

For all the reasons explained above, the methodology that we have developed about a current game serves as a guideline to a need from the forensic community to this new form of communication and interaction between criminals. This game allows multiple ways of messaging (chatting inside/outside the game and live chat when doing YouTube Live Streaming). We provided a full analysis of the game from a network, volatile memory and hard disk perspective so that forensic experts can use it when facing a real case with this game installed in the system (or a similar one).

This game runs under Steam platform, which supplies the biggest quantity of video-games in the market [7]. Therefore, for executing the game, we need to login with Steam first. When doing the previous analysis, we dealt with some features of Steam that led us to a vulnerability finding (session hijacking) that was reported to the developers as soon as it was found and that we were allowed to explain in this thesis.

We managed to overcome the encryption issues of the network data by performing session cloning. We were able to find information about the user such as username, passwords and chat messages from the volatile analysis; session IDs and user ID from the network examination; connections time-stamps, user and chat information, logs and the video of the YouTube Live Streaming performed from the disk analysis. All this information was validated through two well-stated case scenarios that cover all the main issues with respect to communication: chatting inside and outside an in-game round and also through YouTube Live chat. Consequently, these cases reflect how the game can be used for

communication in real life. Therefore, as a conclusion, we can state that our methodology will help the forensic community when dealing with the new threat of online-gaming communication, from all the forensic angles: network analysis, live acquisition and post-mortem acquisition.

Regarding the Future Work, some video-games offer server hosting features, that were not contemplated in this thesis as the game always runs under official servers; so, it is also a characteristic to be considered by the forensic community in coming academic studies.

References

- [1] K. Lofgren. (April 5th, 2017). *2017 Video Game Trends and Statistics – Who's Playing What and Why?*. Accessed: 20/01/2018. Available: <http://www.bigfishgames.com/blog/2017-video-game-trends-and-statistics-whos-playing-what-and-why/>
- [2] D. Schofield, "Playing with evidence: Using video games in the courtroom," *Entertainment Computing*, vol. 2, no. 1, pp. 47-58, 2011/01/01/ 2011.
- [3] B. Mastroianni. (November 18th, 2015). *How terrorists could use video games to communicate undetected*. Accessed: 20/01/2018. Available: <https://www.cbsnews.com/news/how-terrorists-could-use-video-games-to-communicate-undetected/>
- [4] Editor. (December 24th, 2015). *Why online gaming is the new frontier for cybercrime*. Accessed: 20/01/2018. Available: <https://www.welivesecurity.com/2015/12/24/online-gaming-new-frontier-cybercriminals/>
- [5] G. M. Graff. (December 23rd, 2017). *How a dorm room Minecraft scam brought down the Internet*. Accessed: 20/01/2018. Available: <https://www.wired.com/story/mirai-botnet-minecraft-scam-brought-down-the-internet/>
- [6] C. Hope. (November 04th, 2017). *What is the most popular operating system?*. Accessed: 20/01/2018. Available: <https://www.computerhope.com/issues/ch001777.htm>
- [7] C. Smith. (August 12th, 2017). *26 Interesting Steam Statistics (August 2017) | By the Numbers*. Accessed: 20/01/2018. Available: <https://expandedramblings.com/index.php/steam-statistics/- .Wd4cU9srzBI>
- [8] J. Gaudiosi. (February 26th, 2016). *Twitch Ups Its Game to Compete with YouTube Gaming*. Accessed: 20/01/2018. Available: <http://fortune.com/2016/02/26/twitch-ups-its-game-to-compete-with-youtube-gaming/>
- [9] N. Grayson. (August 7th, 2016). *The Counter-Strike Gambling Scandal, Explained*. Accessed: 20/01/2018. Available: <https://steamed.kotaku.com/why-people-are-flipping-out-over-the-counter-strike-gam-1783369102>
- [10] A. Leon Mare. (April 24th, 2014). *Windows Forensics and Security*. Accessed: 21/01/2018. Available: <https://articles.forensicfocus.com/2014/04/14/windows-forensics-and-security/>
- [11] M. Rouse. *Definition - Network forensics*. Accessed: 21/01/2018. Available: <http://searchsecurity.techtarget.com/definition/network-forensics>
- [12] S. Garfinkel. (April 24th, 2002). *Network Forensics: Tapping the Internet*. Accessed: 21/01/2018. Available: <http://archive.oreilly.com/pub/a/network/2002/04/26/nettap.html>
- [13] J. Zaaïman and L. Leenen, *ICCWS 2015 - Proceedings of the 10th International Conference on Cyber Warfare and Security*. March 2015.
- [14] C. Castrejon. (September 28th, 2017). *What is the difference between a logical and physical image when it comes to digital forensics? Which of the Paraben tools supports these types of images?*. Accessed: 22/01/2018. Available: <http://support.paraben.com/Knowledgebase/Article/View/36/6/what-is-the-difference->

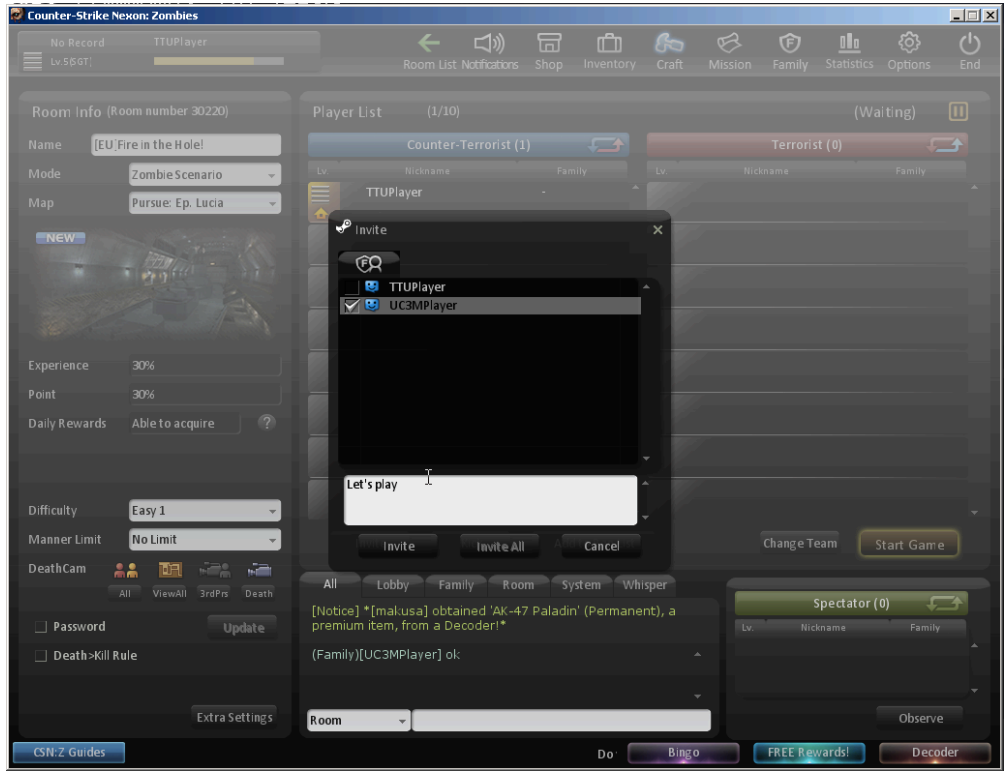
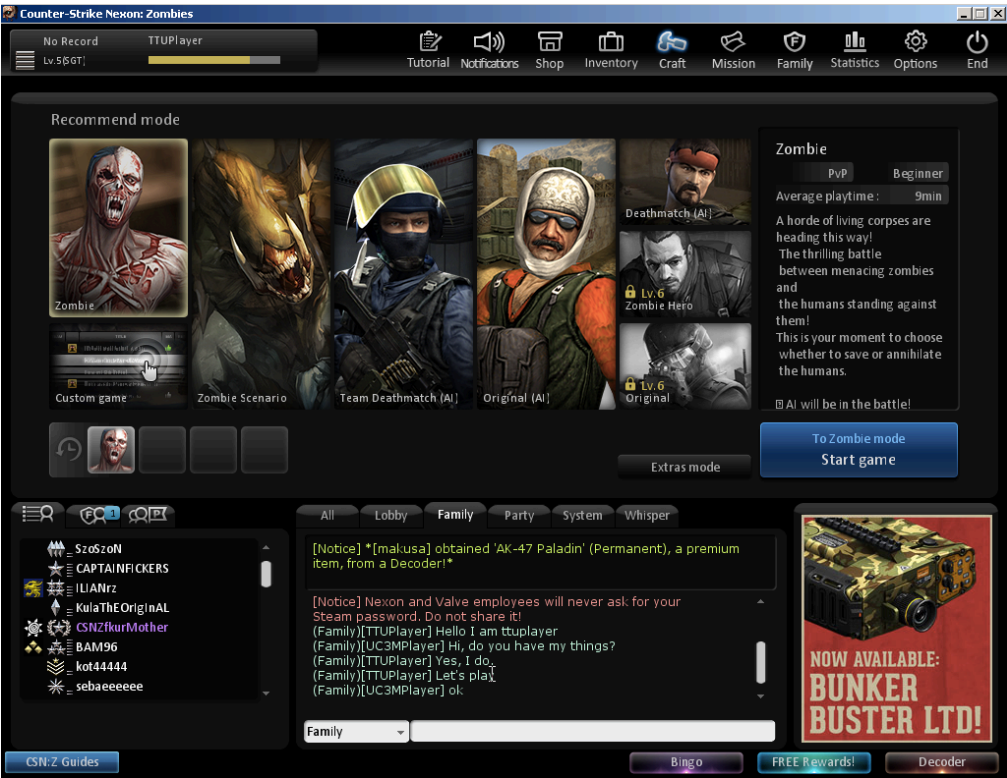
[between-a-logical-and-physical-image-when-it-comes-to-digital-forensics-which-of-the-paraben-tools-supports-these-types-of-images](#)

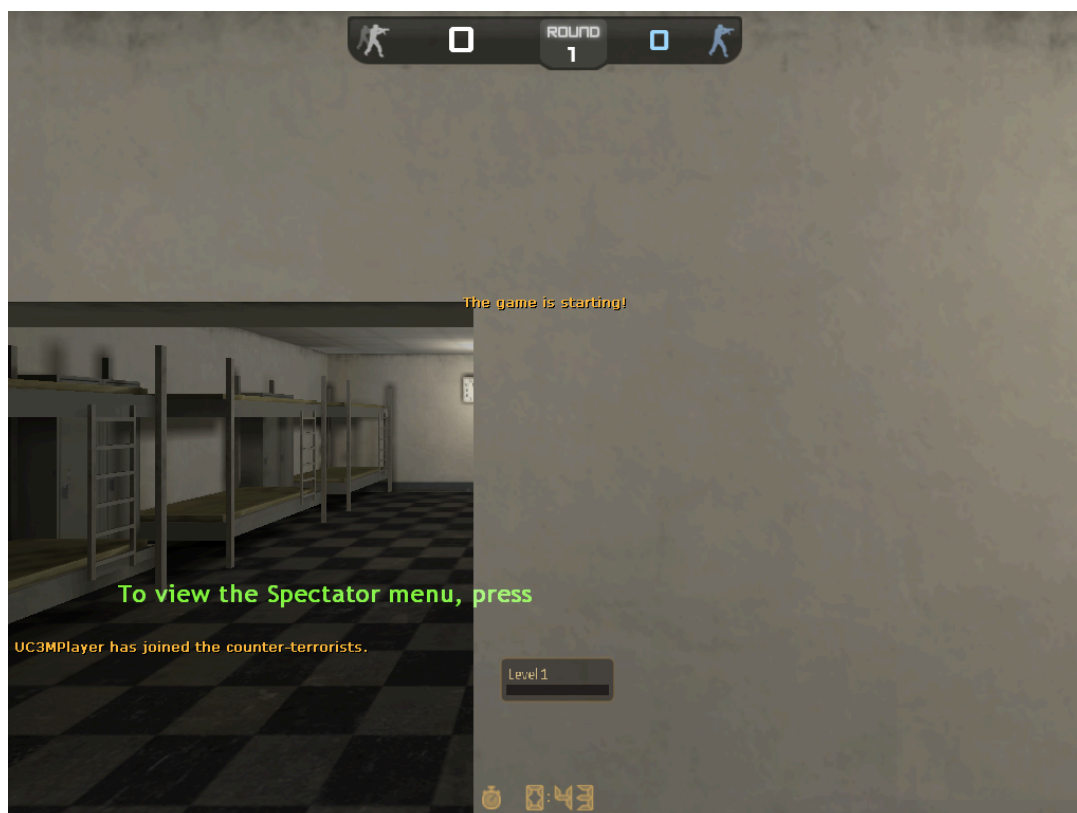
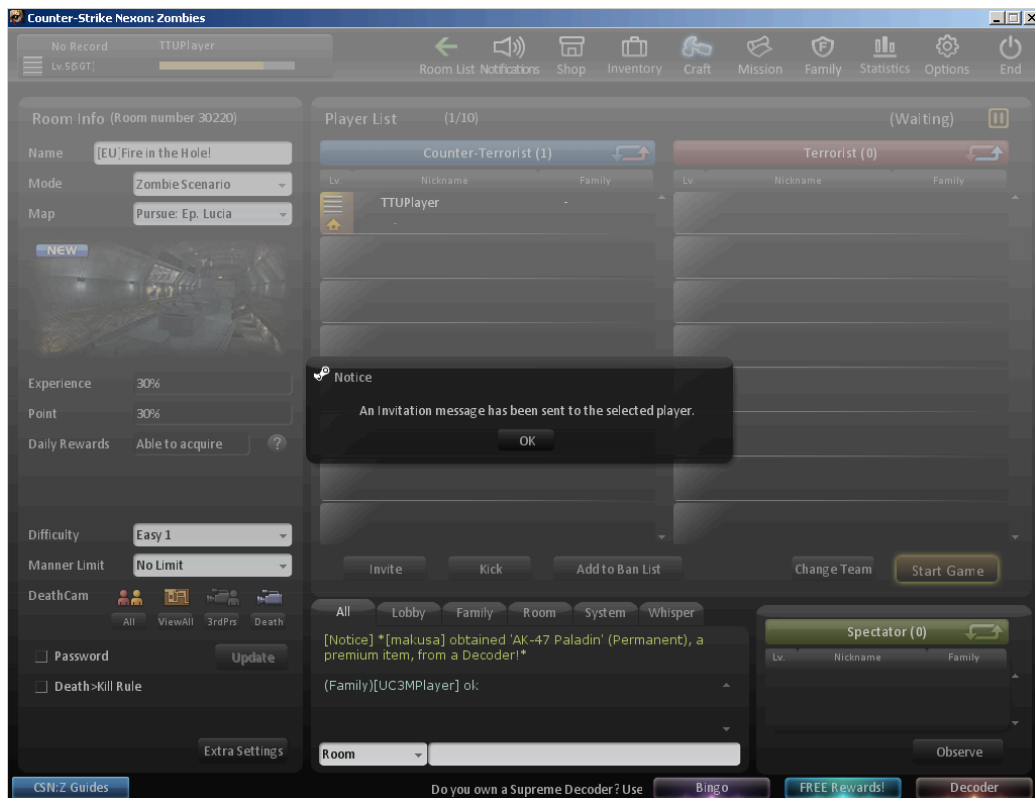
- [15] Wikipedia. *Glossary of digital forensics terms*. Accessed: 22/01/2018. Available: https://en.wikipedia.org/wiki/Glossary_of_digital_forensics_terms
- [16] A. Reyes, K. O'Shea, J. Steele, J. R. Hansen, B. R. Jean, and T. Ralph, "Chapter 5 - Incident Response: Live Forensics and Investigations," *Cyber Crime Investigations*. Burlington: Syngress, pp. 89-109, 2007.
- [17] SANS. *FOR500: Windows Forensic Analysis*. Accessed: 22/01/2018. Available: <https://www.sans.org/course/windows-forensic-analysis>
- [18] Microsoft-Documentation. *Registry Hives*. Accessed: 22/01/2018. Available: [https://msdn.microsoft.com/es-es/library/windows/desktop/ms724877\(v=vs.85\).aspx](https://msdn.microsoft.com/es-es/library/windows/desktop/ms724877(v=vs.85).aspx)
- [19] ComputerHope. (October 17th, 2017). *Shortcut*. Accessed: 22/01/2018. Available: <https://www.computerhope.com/jargon/s/shortcut.htm>
- [20] ComputerHope. (April 26th, 2017). *Prefetch*. Accessed: 24/01/2018. Available: <https://www.computerhope.com/jargon/p/prefetch.htm>
- [21] ComputerHope. (October 11th, 2017). *Log*. Accessed: 24/01/2018. Available: <https://www.computerhope.com/jargon/l/log.htm>
- [22] NTFS.com. *NTFS Master File Table (MFT)*. Accessed: 24/01/2018. Available: <http://www.ntfs.com/ntfs-mft.htm>
- [23] PCMag.com. *Definition of: thumbnail*. Accessed: 24/01/2018. Available: <https://www.pcmag.com/encyclopedia/term/52873/thumbnail>
- [24] A. Merola, "Data Carving Concepts," *SANS Institute InfoSec Reading Room*, November 10th, 2008.
- [25] C. Anglano, "Forensic analysis of WhatsApp Messenger on Android smartphones," *Digital Investigation*, vol. 11, no. 3, pp. 201-213, 2014.
- [26] F. Karpisek, I. Baggili, and F. Breitingner, "WhatsApp network forensics: Decrypting and understanding the WhatsApp call signaling messages," *Digital Investigation*, vol. 15, pp. 110-118, 2015.
- [27] G. . Jhala KY, "WhatsApp Forensics: Decryption of Encrypted WhatsApp Databases on Non Rooted Android Devices", *Journal of Information Technology & Software Engineering*, vol. 05, no. 02, 2015.
- [28] A. Iqbal, H. Alobaidli, A. Almarzooqi, and A. Jones, "LINE IM app Forensic Analysis", *ResearchGate*, 2015.
- [29] J. McQuaid, "Skype Forensics: Analyzing Call and Chat Data From Computers and Mobile", *MAGNET Forensics*, 2014.
- [30] K. Wong, A. C. T. Lai, J. C. K. Yeung, W. L. Lee, and P. H. Chan, "Facebook Forensics.", *Valkyrie-X Security Research Group*.
- [31] G. Satrya, P. Daely, and M. Arif Nugroho, "Digital Forensic Analysis of Telegram Messenger on Android Devices", *Digital Investigation*, pp. 1-7, 2016.

- [32] C. Sgaras, T. Kechadi, and N.-A. Le-Khac, *Forensics Acquisition and Analysis of Instant Messaging and VoIP Applications*. 2014.
- [33] C. Anglano, M. Canonico, and M. Guazzone, "Forensic analysis of the ChatSecure instant messaging application on android smartphones," *Digital Investigation*, vol. 19, no. Supplement C, pp. 44-59, 2016.
- [34] S. Wu, Y. Zhang, X. Wang, X. Xiong, and L. Du, "Forensic analysis of WeChat on Android smartphones," *Digital Investigation*, vol. 21, pp. 3-10, 2017.
- [35] G. B. Satrya, P. T. Daely, and S. Y. Shin, "Android forensics analysis: Private chat on social messenger," in *2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN)*, pp. 430-435, 2016.
- [36] T. Y. Yang, A. Dehghantanha, K.-K. R. Choo, and Z. Muda, "Windows Instant Messaging App Forensics: Facebook and Skype as Case Studies," *PLOS ONE*, vol. 11, no. 3, p. e0150300, 2016.
- [37] N. Barghuthi and H. Said, "Social Networks IM Forensics: Encryption Analysis", *Journal of Communications*, vol. 8, no. 11, pp. 708-715, 2013.
- [38] M. Davies, H. Read, K. Xynos, and I. Sutherland, "Forensic analysis of a Sony PlayStation 4: A first look," *Digital Investigation*, vol. 12, pp. S81-S89, 2015.
- [39] S. Khanji, R. Jabir, F. Iqbal, and A. Marrington, "Forensic analysis of xbox one and playstation 4 gaming consoles", *Digital Investigation*. vol. 12, pp. S81-S89, 2016.
- [40] J. Moore, I. Baggili, A. Marrington, and A. Rodrigues, "Preliminary forensic analysis of the Xbox One", *Digital investigation*, vol. 11, pp. S57-S65, 2014.
- [41] L. E. Daniel. (2010). *Multiplayer Game Forensics*. Accessed: 02/02/2018. Available: <https://www.forensicsmag.com/article/2010/05/multiplayer-game-forensics>
- [42] R. McKemmish, "What is Forensic Computing?", *Australian Institute of Criminology*, Art. no. 118, 1999.
- [43] Wireshark. *About Wireshark*. Accessed: 02/02/2018. Available: <https://www.wireshark.org/>
- [44] Wikipedia. *Wireshark*. Accessed: 02/02/2018. Available: <https://en.wikipedia.org/wiki/Wireshark>
- [45] NETRESEC. *NetworkMiner*. Accessed: 02/02/2018. Available: <http://www.netresec.com/?page=NetworkMiner>
- [46] AccessData. *FTK Imager User Guide*. Accessed: 04/02/2018. Available: https://ad-pdf.s3.amazonaws.com/Imager/3_4_3/FTKImager_UG.pdf
- [47] VolatilityFoundation. *Volatility*. Accessed: 04/02/2018. Available: <http://www.volatilityfoundation.org/26>
- [48] SleuthKit. *Autopsy*. Accessed: 04/02/2018. Available: <https://www.sleuthkit.org/autopsy/>
- [49] GitHub. *jschicht/Mft2Csv*. Accessed: 04/02/2018. Available: <https://github.com/jschicht/Mft2Csv>

- [50] GameBanana. *Nar Extractor CSO 2.0*. Accessed: 04/02/2018. Available: <https://gamebanana.com/tools/5235>
- [51] NirSoft. *JumpListView*. Accessed: 05/02/2018. Available: https://www.nirsoft.net/utils/jump_lists_view.html
- [52] NirSoft. *WinPrefetchView*. Accessed: 05/02/2018. Available: http://www.nirsoft.net/utils/win_prefetch_view.html
- [53] NirSoft. *RecentFilesView*. Accessed: 05/02/2018. Available: http://www.nirsoft.net/utils/recent_files_view.html
- [54] AccessData. *Registry Viewer*. Accessed: 08/02/2018. Available: https://ad-pdf.s3.amazonaws.com/RegistryViewer_UG.pdf
- [55] GitHub. *keydet89/RegRipper2.8*. Accessed: 08/02/2018. Available: <https://github.com/keydet89/RegRipper2.8>
- [56] NotepadPlusPlus.org. *NotePad++*. Accessed: 08/02/2018. Available: <https://notepad-plus-plus.org/>
- [57] ThumbCacheViewer. *About*. Accessed: 08/02/2018. Available: <https://thumbcacheviewer.github.io/>
- [58] VideoLAN. *VLC media player*. Accessed: 08/02/2018. Available: <https://www.videolan.org/vlc/index.html>
- [59] NirSoft. *ChromeCacheView*. Accessed: 08/02/2018. Available: https://www.nirsoft.net/utils/chrome_cache_view.html
- [60] NirSoft. *ChromeHistoryView*. Accessed: 08/02/2018. Available: https://www.nirsoft.net/utils/chrome_history_view.html
- [61] NirSoft. *ChromeCookiesView*. Accessed: 08/02/2018. Available: https://www.nirsoft.net/utils/chrome_cookies_view.html
- [62] SQLiteBrowser.org. *DB Browser for SQLite*. Accessed: 08/02/2018. Available: <http://sqlitebrowser.org/>
- [63] NETMARKETSHARE. *Desktop Operating System Market Share*. Accessed: 12/02/2018 Available: <https://www.netmarketshare.com/operating-system-market-share.aspx?qprid=10&qpcustomd=0>
- [64] Wikipedia. *Steam (Software)*. Accessed: 12/02/2018. Available: [https://en.wikipedia.org/wiki/Steam_\(software\)](https://en.wikipedia.org/wiki/Steam_(software))
- [65] Wikipedia. *Counter-Strike*. Accessed: 12/02/2018. Available: <https://en.wikipedia.org/wiki/Counter-Strike>

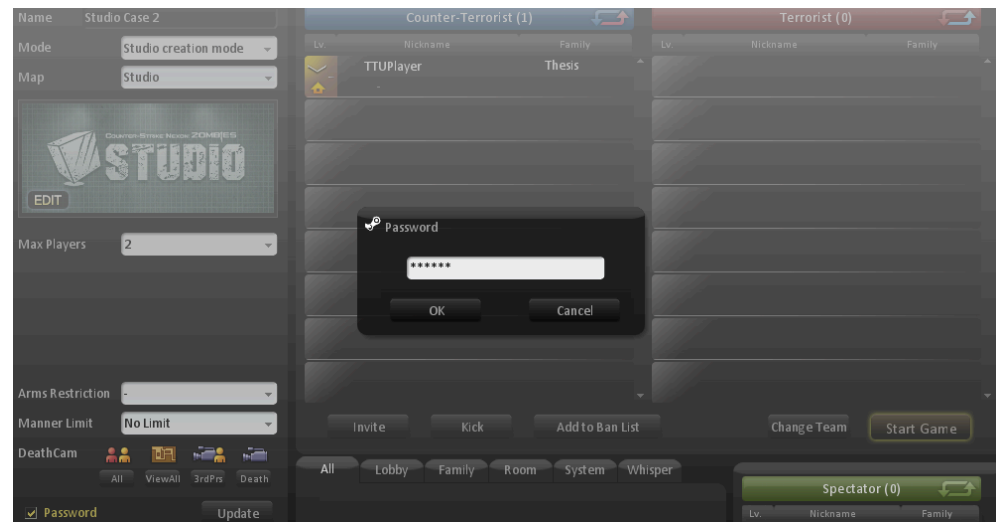
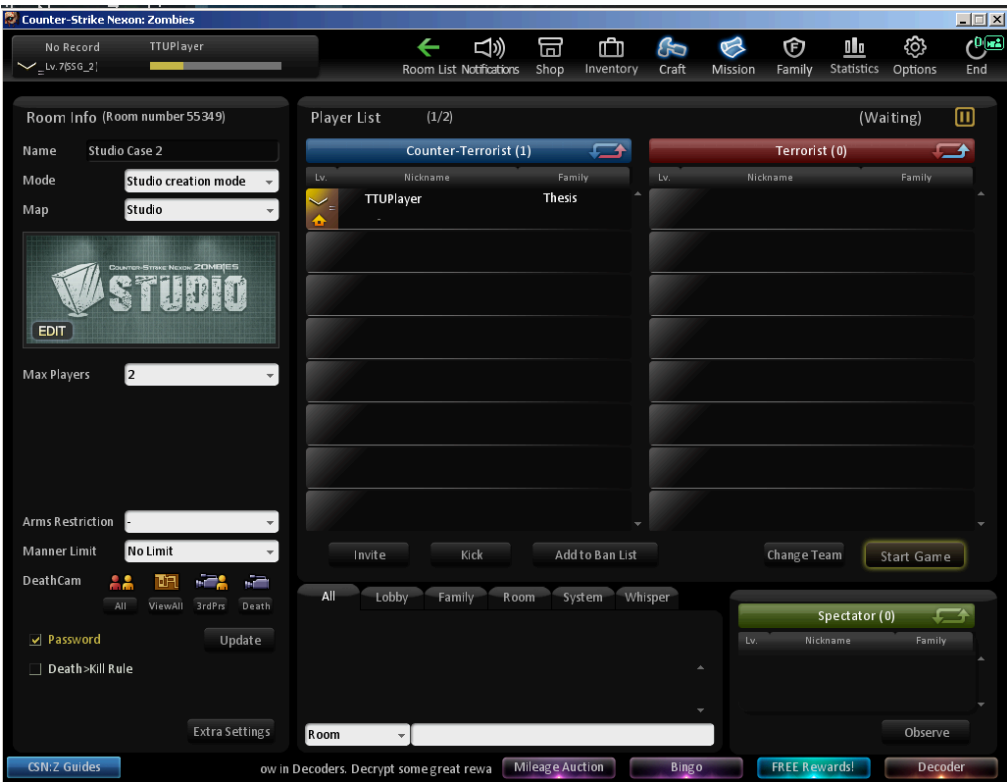
Appendix 1 – Game Screenshots Case 1

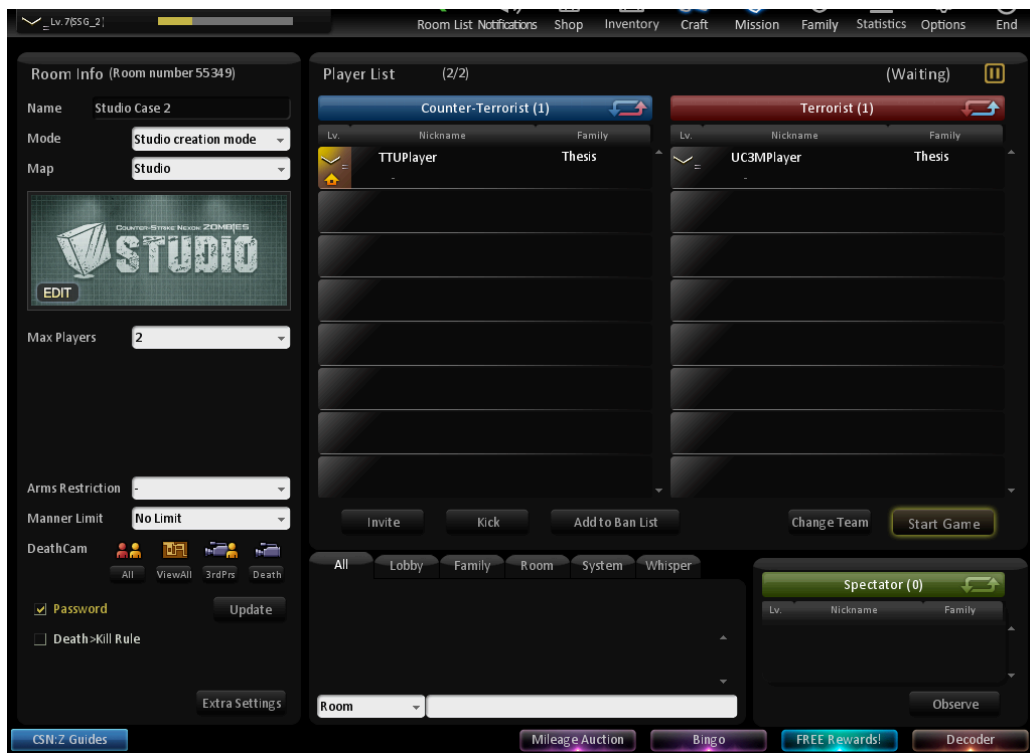


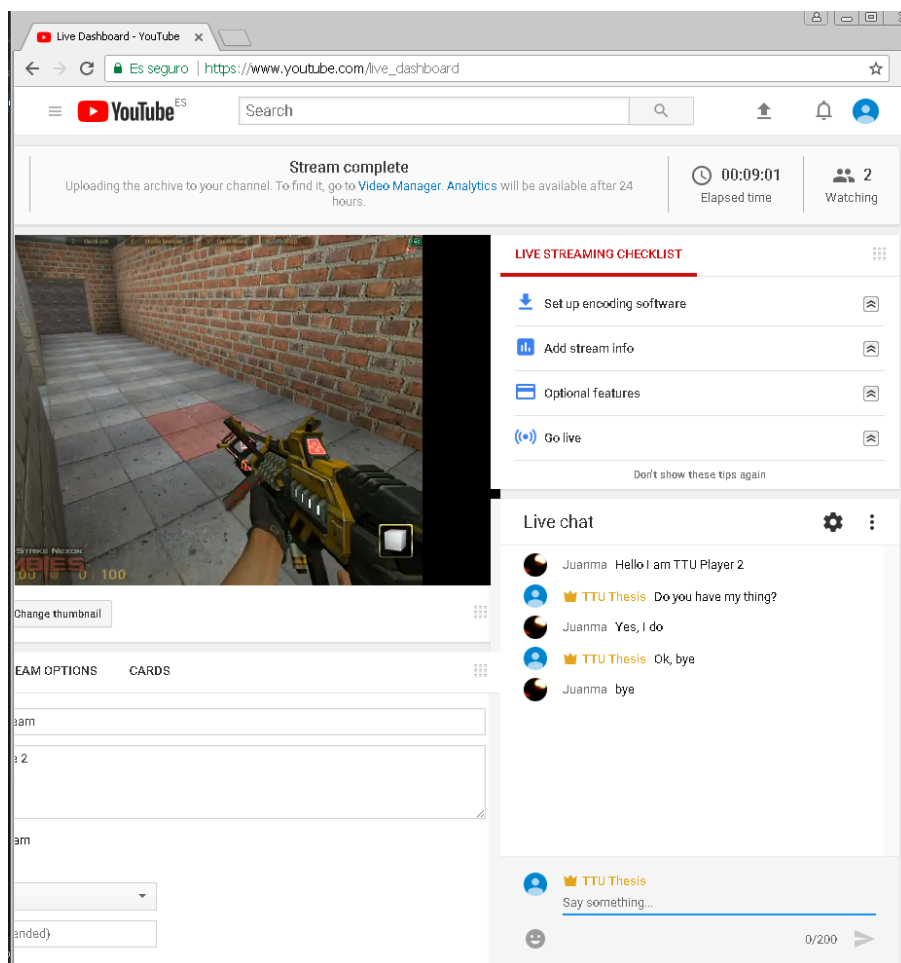
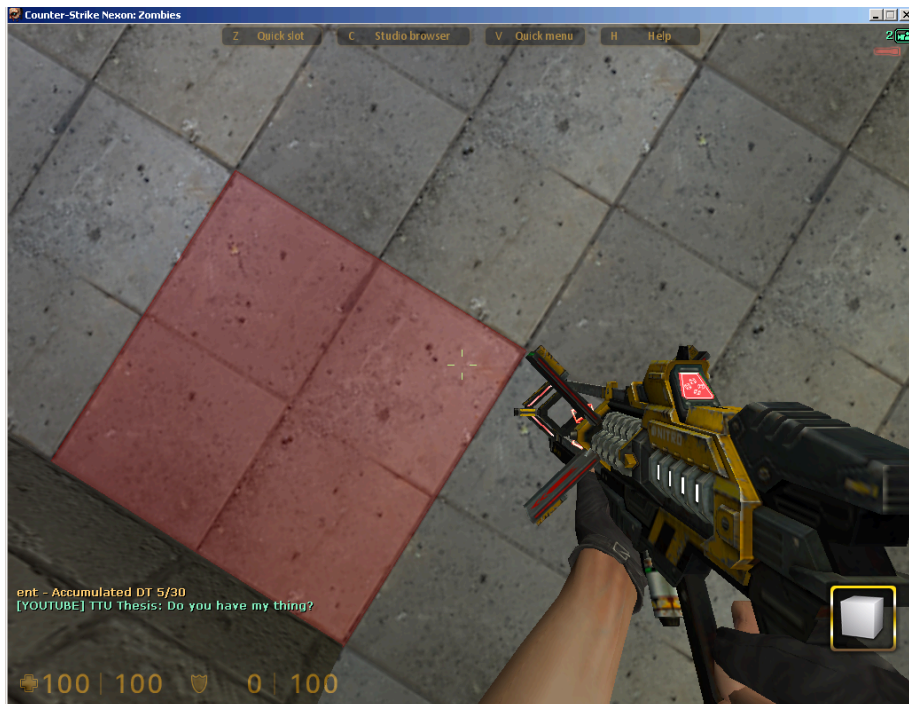


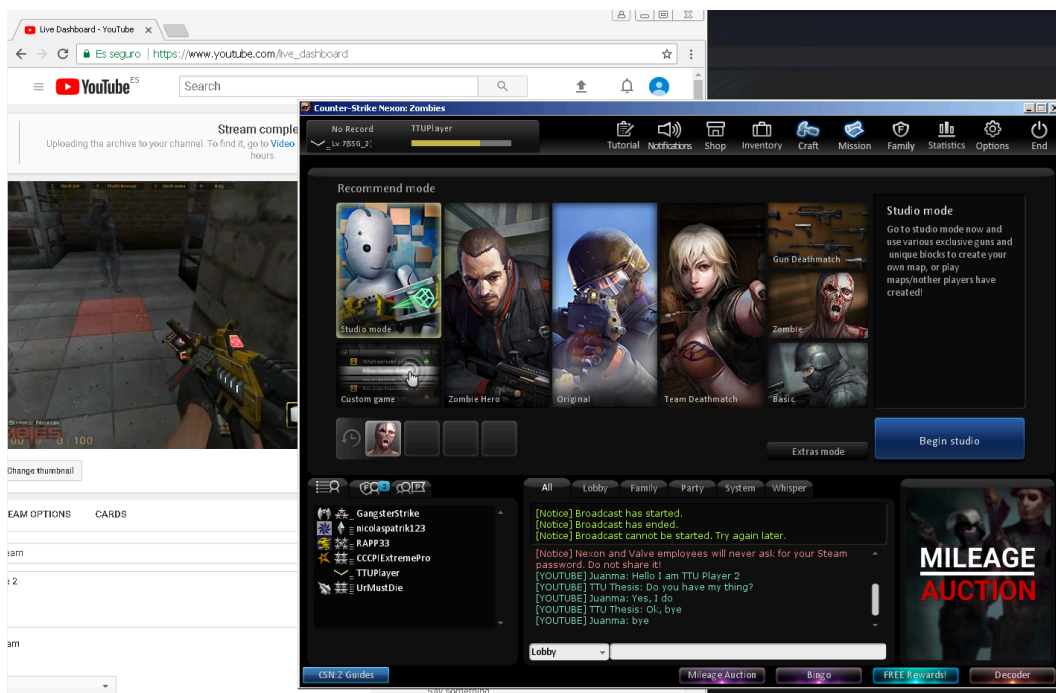
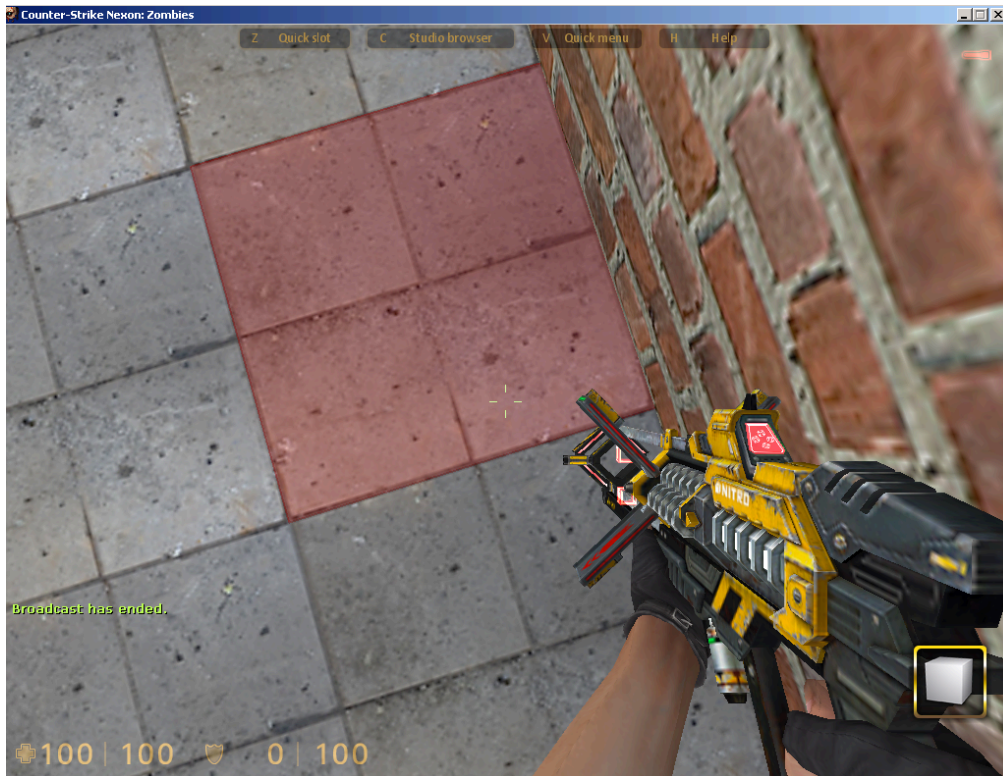


Appendix 2 – Game Screenshots Case 2

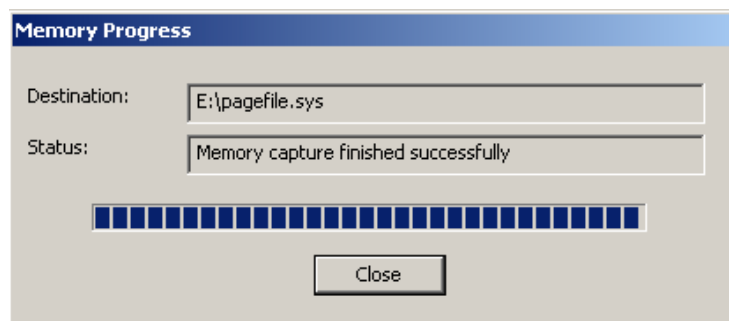
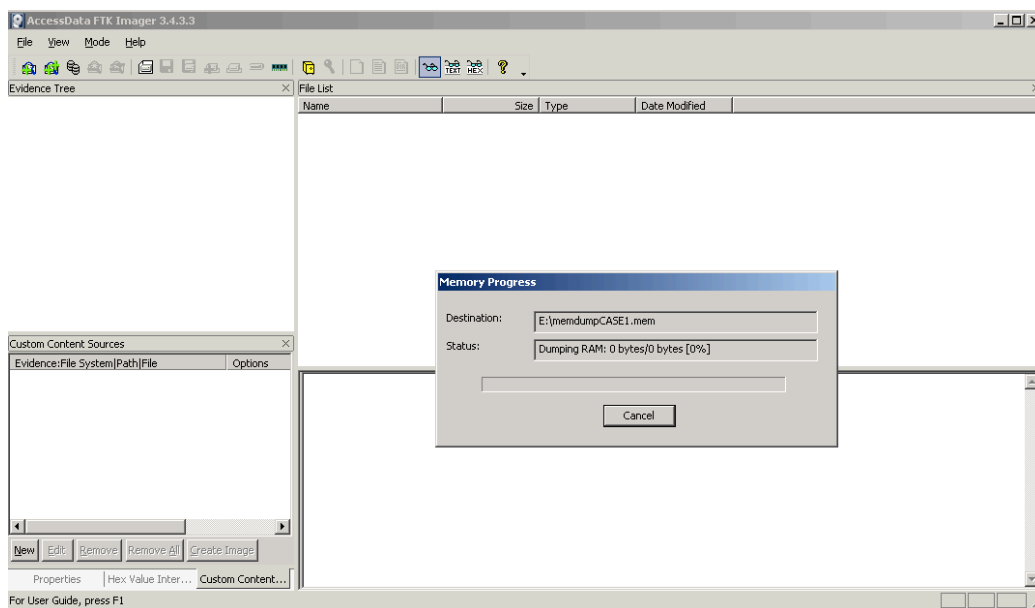
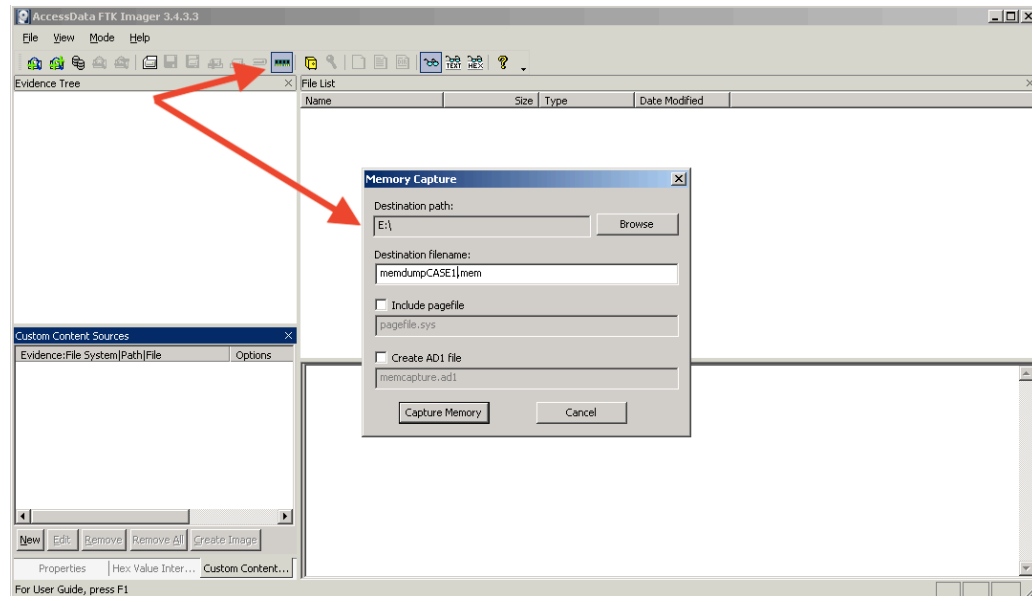




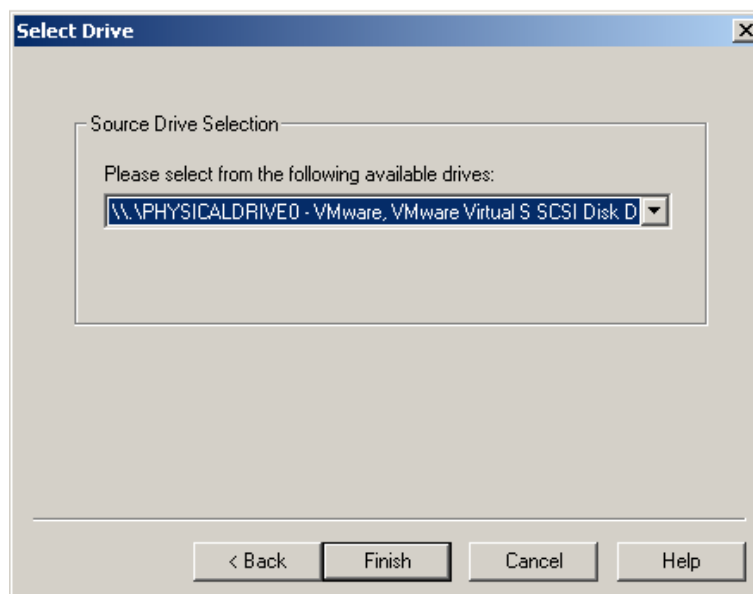
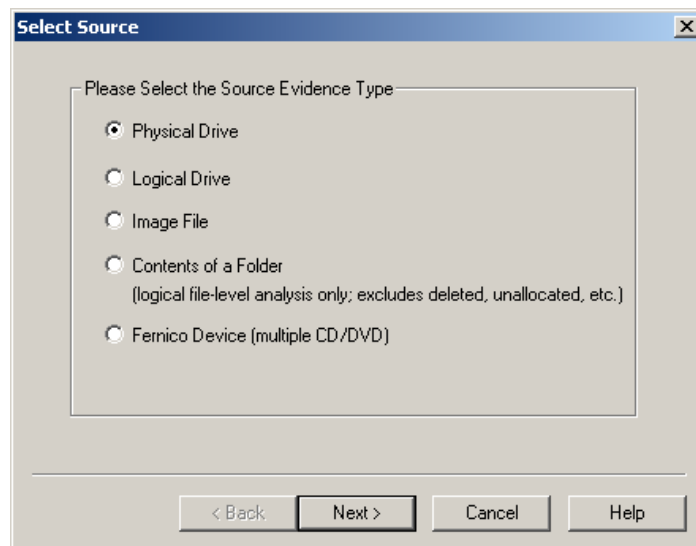
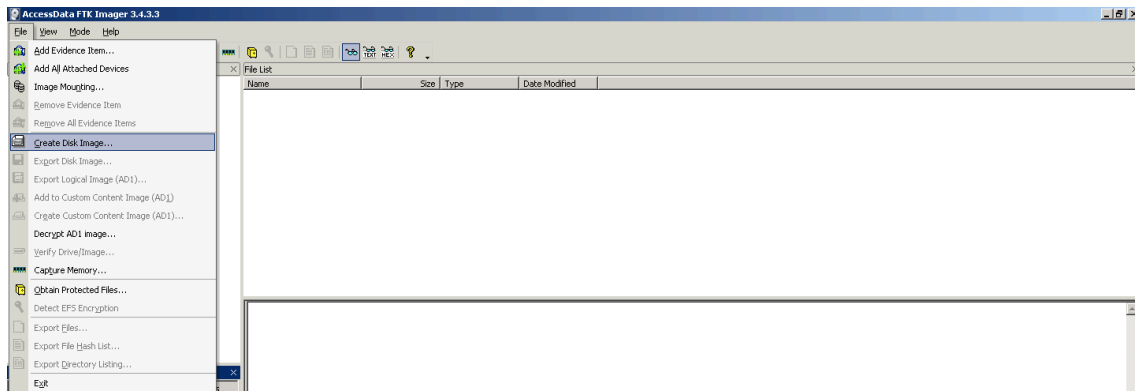




Appendix 3 – Live acquisition with FTK Imager



Appendix 4 – Physical disk acquisition with FTK Imager



Create Image

Image Source:

Starting Evidence Number:

Image Destination(s):

☒ Verify images after they are created ☐ Precalculate Progress Statistics

☐ Create directory listings of all files in the image after they are created

Create Image

Select Image Type

Please Select the Destination Image Type

☐ Raw (dd)

☐ SMART

☒ E01

☐ AFF

Create Image

Evidence Item Information

Case Number:

Evidence Number:

Unique Description:

Examiner:

Notes:

Create Image [X]

Select Image Destination [X]

Image Destination Folder

Image Filename (Excluding Extension)

Image Fragment Size (MB)
 For Raw, E01, and AFF formats: 0 = do not fragment

Compression (0=None, 1=Fastest, ..., 9=Smallest)

Use AD Encryption ☐

< Back Finish Cancel Help

Start Cancel

Create Image [X]

Select Image Destination [X]

Image Destination Folder

Image Filename (Excluding Extension)

Image Fragment Size (MB)
 For Raw, E01, and AFF formats: 0 = do not fragment

Compression (0=None, 1=Fastest, ..., 9=Smallest)

Use AD Encryption ☐

< Back Finish Cancel Help

Start Cancel

Create Image [X]

Image Source

Starting Evidence Number:

Image Destination(s)

Add... Edit... Remove

Add Overflow Location

☒ Verify images after they are created ☐ Precalculate Progress Statistics

☐ Create directory listings of all files in the image after they are created

Start Cancel

