THE JOURNAL OF DIGITAL FORENSICS, SECURITY AND LAW

Journal of Digital Forensics, Security and Law

Volume 10 | Number 4 Article 6

2015

Forensic Acquisition of IMVU: A Case Study

Robert van Voorst
National Police of the Netherlands

M-Tahar Kechadi
University College Dublin

Nhien-An Le-Khac
University College Dublin

Recommended Citation

van Voorst, Robert; Kechadi, M-Tahar; and Le-Khac, Nhien-An (2015) "Forensic Acquisition of IMVU: A Case Study," *Journal of Digital Forensics, Security and Law*: Vol. 10 : No. 4 , Article 6.
DOI: https://doi.org/10.15394/jdfsl.2015.1212
Available at: https://commons.erau.edu/jdfsl/vol10/iss4/6

FORENSIC ACQUISITION OF IMVU: A CASE STUDY

Robert van Voorst
National Police of the Netherlands
Rotterdam, Netherlands
rvanvoorst@politie-rijnmond.nl

M-Tahar Kechadi, Nhien-An Le-Khac
University College Dublin
Dublin 4, Ireland
{tahar.kechadi,an.lekhac}@ucd.ie

ABSTRACT

There are many applications available for personal computers and mobile devices that facilitate users in meeting potential partners. There is, however, a risk associated with the level of anonymity on using instant message applications, because there exists the potential for predators to attract and lure vulnerable users. Today Instant Messaging within a Virtual Universe (IMVU) combines custom avatars, chat or instant message (IM), community, content creation, commerce, and anonymity. IMVU is also being exploited by criminals to commit a wide variety of offenses. However, there are very few researches on digital forensic acquisition of IMVU applications. In this paper, we discuss first of all on challenges of IMVU forensics. We present a forensic acquisition of an IMVU 3D application as a case study. We also describe and analyse our experiments with this application.

Keywords: Instant messaging, forensic acquisition, Virtual Universe 3D, forensic process, forensic case study

1. INTRODUCTION

Instant messaging (IM) is one of the most popular digital communication technologies and is widely used today. In fact, IM belongs to the category of Internet based communication services. Traditionally the IM was designed to transfer text messages only, but now it has been enhanced with additional features such as voice/video message, file transfer, etc. Moreover, IM can be accessible through cell phones or other mobile devices, as well as on a computer. It is estimated that there are several millions of IM users who use IM for various purposes. There are numerous IM applications available for download today [1]. Besides, IM applications (apps) also allow users to create a detailed personal profile including name, email address, age, home address, phone number, school and hobbies. If users do not verify carefully during the sign-up process, they could reveal more information than they should. Easy accessible profiles can allow anyone to contact IM users. Indeed, some IM apps offer users the option of joining in chat with strangers. As a consequence, using of IM apps can encourage gossiping and bullying. Children could receive pornographic "spam" through IM apps or become victim of Sexual grooming of children [2]. In the recent years, police investigators have noticed that IM apps have been directly or indirectly involved in criminal activities. Due to their popularity, IM apps have the potential of being a rich source of evidential value in all kinds of criminal investigations. Indeed, Europol has identified the threat of misused IM communications by criminals to facilitate their illegal activities due to the fact that it is harder to monitor or to regulate these services [3].

On the other hand, a virtual universe is a computer-based simulated environment. In a 3D virtual universe, the users can take the form of 3D avatars (virtual character) visible to others. Today, Instant Messaging within a Virtual Universe (IMVU) combines custom avatars, chat/IM, community, content creation, commerce, and anonymity [4]. IMVU contains its own economy with a currency system based on IMVU "credits" and "promotional credits". IMVU members use the credits to purchase virtual items like fashion pieces (hair, clothes, skins, and accessories), pets, and 3D scenes such as homes, clubs and open landscapes. Except 'VIP users', everyone has "Guest_" in front of his/her avatar name. It is also possible to buy an 'Access Pass'. The Access Pass is available for those 18 years old and older. It grants access to more risqué clothing and slightly more intimate furniture and poses. It also allows chatting to adults only. IMVU was also mentioned as a common Virtual World addressing 'Crime and Policing in Virtual Worlds' (2010) by Marc Goodman [5]. He concludes that criminals can exploit this new technology to commit a wide variety of offenses. Almost any types of crime in the real world can also be found in virtual spaces from child abuse to terrorist attacks. Besides, the presence of child pornography in chat has also been denounced in 3D social networks like IMVU. Indeed, the emerging of 3D social networks is rapidly tainted with child pornography as stated in [6].

In fact, there are many tools available for collecting artefacts, some of them specific for chat/IM such as Paraben chat examiner [7] or Belkasoft Evidence Centre 2015 [8]. However, at the time of writing this paper, none of them supports the investigation of IMVU apps. Despite that IMVU exists a while, has millions of users worldwide and it is associated with child abuse (material), to the best of our knowledge there is no technique or software tool for forensic acquisition and analysis of IMVU artefacts. Therefore, the objective of this paper is to investigate whether it is possible to forensically acquire and analyse artefacts on a computer after the installation and use of the IMVU application. The rest of this paper is structured as follows: Section 2 shows the background of this research including related work in this domain. We present IMVU apps and forensic process in Section 3. We describe our case study in Section 4 and discuss on experiment results in Section 5 and 6. We conclude and discuss on future work in Section 7.

2. RELATED WORK

In fact, we have tried to find in literature about this specific subject, but to the best of our knowledge, there is no published research regarding forensic artefacts of IMVU. Hence, we present some literature survey on the forensic acquisition and analysis of IM applications in general. A more traditional investigation method of IM has been described in [9]. Kiley et al [10] conducted some research on IM forensic artefacts and they also pointed out that IM is being exploited by criminals due to its popularity and privacy features. One of the conclusions from this work is that forensic evidence is recoverable after these programs have been used, but investigators must know certain elements of the conversations in order to perform string searches.

Lun [11] describes the popularity of IMs and the challenges by extracting information from these IMs by Law Enforcement. His finding indicated that the forensic tool was able to aid the forensic examiner to extract digital evidence from instant messenger, but it was not comprehensive enough at that forensic examiners could not rely on one tool during an investigation.

Dickson [12] describes the examination of AOL Instant Messenger (AIM) 5.5. This research focuses on the contact information (identification) and the ability to trace back from a known suspect's computer system to identify contact with his alleged victims. This is specifically interesting when an allegation of grooming is made. According to the author, one of the first things an examiner should do is to determine the suspect's computer system that was used to contact the informant's account in the first place.

Husain et al. [13] describe the Forensic Analysis of Instant Messaging on Smart Phones. Their results have shown that various useful artefacts, related to IMs, can be recovered, including username, password, buddy list, last log-in time, and conversation timestamp as well as conversation details.

In [14], authors observed that the relevant research on the topic of evidence collection from IM services was limited. Authors presented the forensic acquisition and analysis of WhatsApp, Viber, Skype and Tango IMs and VoIPs for both iOS and Android platforms and authors tried to answer on how evidence can be collected when IM communications are used. This work provides useful information that assists us in developing our case study in this paper. More precisely the authors have provided a very useful taxonomy of target artefacts.

3. IMVU AND FORENSIC PROCESS

In this section, we describe our forensic process for investigating IMVU apps for Windows. The objective of this paper is to forensically examine significant artefacts present on a computer after the installation and use of the IMVU app. Therefore, the approach chosen is empirical or experimental research and differential forensic analysis. While there is no available technique or software tool to evaluate and there is very little known about IMVU artefacts, our approach includes the collection of data during experiments and try to discover significant forensic artefacts by observing and analysing the data. In fact, with an IMVU app, apart from the typical conversation log showing on screen, and possible (log) files saved on the logical drive, evidence of conversation could also be found in page files and unallocated hard disk space, the windows registry or volatile data in the internal memory (RAM). There are two main steps in our forensic process: data collection and data analysis.

IMVU data collection: in this phase data is collected during the running time of an IMVU app. The contents of the IMVU application and data directory, unallocated disk space, the Windows registry, volatile data from the internal memory and network traffic are acquired. We attempt moreover to infer the behaviour of the IMVU application through the reverse engineering process. The objective is to trace user activities, like the history of performed activities such as instant messages sent or received (chat logs), date and time stamp of the communication, sender, receiver and password. All experiments were carried out several times to ensure that results are repeatable.

IMVU data analysis: This phase gives meaning to the data that have been collected so that the research question can be answered. Differential forensic analysis [15] is carried out by comparing the collected data (snapshot) before and after the experiments with IMVU to observe the differences between them such as report the files that have been added, deleted, renamed, and altered, the difference between Windows registry hive files. The objective of the analysis is to understand the changes and to determine if it contains any relevant information or not. A taxonomy of target artefacts is defined in order to guide and structure subsequent forensic analysis. The taxonomy used in our analysis is described in [14].

4. IMVU FORENSICS: A CASE STUDY

In our case study, we used computers with network access (using a static IP address). The testing was conducted using Windows 7 Enterprise, SP1. Besides, the monitoring tool SysTracer was installed onto the system to record file activity and changes made to the Windows registry. Another computer was setup as a man-in-the-middle proxy using MITMproxy, running on the Ubuntu 14.04.1 LTS, this computer is used to monitor and intercept network traffic as shown in Figure 1. Next, an IMVU client program, version 514.0, was installed. In order to use IMVU, firstly a user has to create an account. For this case study, we create three IMVU accounts (*InspectorAlgar, LadyElizabethSmall* and *missycanaryyellow*). We conduct different conversations to create test data. The conversations were in public rooms but also limited by two participants. We created moreover an AutoIt [16] script that simulated an IMVU user. On beforehand a text file was created containing the conversation text of one of the discussion partners. A chat message was send to the IMVU application within a fixed (pre-adjustable) interval. By starting two scripts on different machines in appropriate time, we can simulate a chat conversation. The chat messages contained the chat text and a UNIX timestamp. An example from the chat messages is shown in Figure 2.

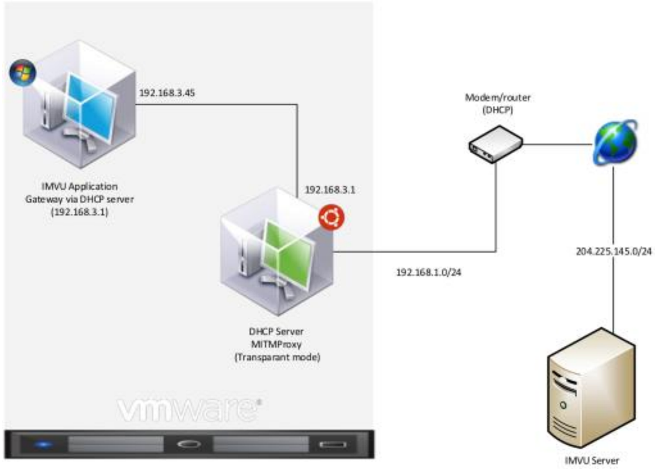
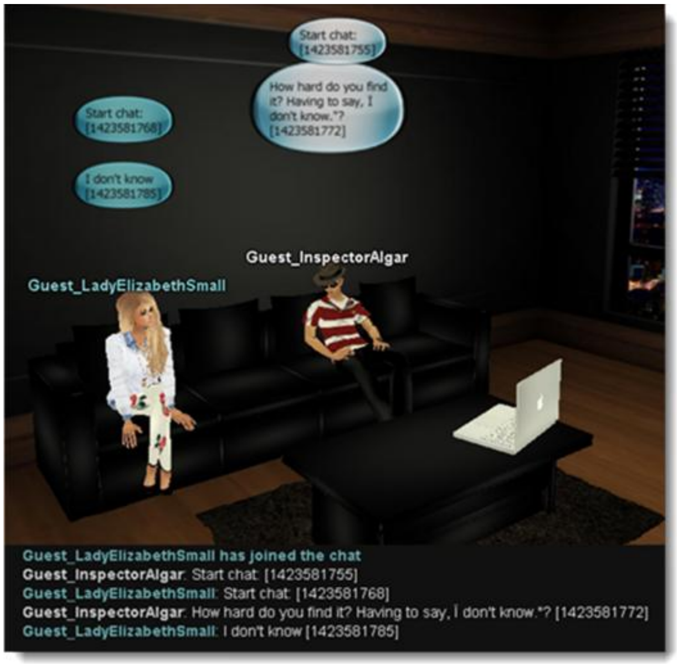


Figure 1. MITM proxy setup



```
Guest_LadyElizabethSmall has joined the chat
Guest_InspectorAlgar: Start chat: [1423581755]
Guest_LadyElizabethSmall: Start chat: [1423581768]
Guest_InspectorAlgar: How hard do you find it? Having to say, I don't know."? [1423581772]
Guest_LadyElizabethSmall: I don't know [1423581785]
```

Figure 2. Chat script usage example

5. IMVU: FILE SYSTEM FORENSICS

In this section, we describe our forensic acquisition results of IMVU file systems including application file, log files, profile data and location data.

5.1 Application Files

IMVU application files are stored in: *C:\Users\%USERNAME%\AppData\Roaming\IMVUClient\* folder. This path is fixed during the installation. During the installation a shortcut to *IMVUQualityAgent.exe* is added into the Windows Start-up folder as well on the user's desktop. We also found that IMVU uses Mozilla's Gecko rendering engine and the XML User Interface Language (XUL) for building its user interface [17].

Besides, a configuration file *imvu.cfg* is added into the user Flash Player Trust folder: *C:\Users\%USERNAME%\AppData\Roaming\Macromedia\FlashPlayer\#Security\FlashPlayerTrust\*. Configuration files inside this *FlashPlayerTrust* folder normally contain a list of directory paths. We also observe that application shortcuts are added to IMVU folder: *C:\Users\%USERNAME%\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\IMVU\*. We found moreover the IMVU application is written in Python and the compiled script files are stored in *library.zip*: *C:\Users\%USERNAME%\AppData\Roaming\IMVUClient\library.zip*. We also decompiled these files using Easy Python Decompiler and to explore the IMVU application through reverse engineering.

5.2 Log Files

IMVU log files are stored in *C:\Users\%USERNAME%\AppData\Roaming\IMVU\* folder including *cpp.log, IMVUQualityAgent.log, IMVULog.log,* etc. The *cpp.log* contains information related to user interface and this file is created every time the application starts. The *IMVUQualityAgent.log* file is also created every time the application starts. It contains 'application validation' information and local time stamp. Regarding to *IMVULog.log* file, from the digital forensics examination point of view, this file contains the most valuable information. In fact, this log file is generated by a round-robin process based on session and file size. A new log file is created every time the application starts. When the size of the log file is greater than 2MB (2097152 bytes), this process successively creates new files with the same pathname as the first file, but with different extensions started with ".1", ".2" and so on. If it is greater than ".5", the extension restarts from ".1", older files will be deleted. Each record in IMVULog.log file contains the following attributes (Figure 3):

1. **The number of seconds since start of application:** Floating point decimal value. The length is 6 characters minimum including the dot and 3 decimals (milliseconds rounded). If the length is less than 6, a leading 0 is added.
2. **The thread id:** String value, minimum length 5 characters, padded by spaces if necessary.
3. **File name of the source file where the logging call was issued:** String value, minimum length 30 characters, padded by spaces if necessary
4. **Source line number where the logging call was issued:** Signed integer decimal value, minimum length 4 digits, padded by spaces if necessary.
5. **The logging level for the message:** String value, minimum length 7 characters, padded by spaces if necessary. Possible values are 'DEBUG', 'INFO', 'WARNING', 'E RROR', 'CRITICAL'.
6. **The log message**: String value.

Moreover, we notice that the maximum length of an entire log record (attributes 1-6) is 512 characters. If the length exceeds 512 characters, it is truncated to the first 509 characters and three dots ('...') are added to the end of the record. Besides, if the length of the log message (attribute 6) is greater than a defined length this log message is truncated and three dots are also added to the end. We have also observed that the log message regarding to logging level 'ERROR', and 'INFO' consists of multiple lines, while other log messages only contain a single line.

When analysing the log files, we noticed that the IMVULog file also stores information related to text communications. To analyse an IMVU chat, it is necessary to understand the following important aspects of a chat conversation: (i) Who says what to whom; (ii) In which chat session the conversation takes place, note that an user can be present in multiple rooms (chat sessions) at the 'same' time; (iii) Timestamp of the messages.

The text of a single message appears more than once in different log records. Besides, in a chat room, it is possible to have a private conversation with just a single person in that room (at least one of users is a VIP). Whisper is a special feature available exclusively for VIPs. A VIP member can whisper to anyone and vice-versa, anyone can whisper to a VIP. Two non-VIP users however cannot use this feature to chat with each other. Moreover, the log message contains a list of several elements. The first element is the member id (*'userId'*), the second one is a chat id (*'chatId'*) and the last one contains dictionary data with keys and values. The value '0' of the key '*to*' indicates that the message was sent to the participant(s) of a room. There are one or more participants in a room. The value of the key '*message*' contains the actual chat message. We also note that the value sometimes contains an asterisk followed by a chat command. Besides, if the message exceeds the maximum length, as mentioned previously, it is truncated and three dots are added to the end.

The value of the key *'userId'* contains the member id of the sender of the text message. The value of the key *'chatId'* contains a unique id for each chat session. Concurrent chat sessions in different rooms therefore have different IDs. This log record does not contain a timestamp. To determine the date and time of the message for these log records, there are two possibilities:

1. Calculate the approximate timestamp using the number of seconds since the start of the application. This number is listed at the beginning of the log record.
2. Combine it with the log records derived from the file *SessionDispatcher.pyo*.

5.3 User profile data

The *IMVULog* file only contains user profile data when the status is 'Running' (i.e. the application is running and the user is logged in). The following information can be retrieved from the user profile data: the *user's avatarname* associated with the IMVU account, eight (8) asterisks of the user's password. Note that the real user password is always replaced by eight asterisks. It is independent on the actual length of the password. Indeed, the most comprehensive information can be retrieved from the multi-line log message started with self.userInfo_ and followed by multiple lines of dictionary (pairs of <*key, value*>) data.

The above information is based on the data entered by the user (Account Settings) and data allocated by IMVU. Besides, objects in IMVU are identified by a unique id such as, for example, Member ID, Room(Instance) ID, Session ID, Product ID or Chat ID.

5.4 User authentication data

By investigating the Windows registry, we found that IMVU authenticate users by using username and password. The password is stored in the registry if the option 'Save Password' is selected. Indeed, the relevant password can be retrieved from the registry where it is stored as a hexadecimal Unicode character notation. The user's avatar name associated with the IMVU account is also present in the *IMVULog* file. The *IMVULog* file also contains information about contacts such as 'Friends", "Buddies" or "Fans". Besides, we also found that the *buddystate.pyo* script implements the logging process.

5.5 Location data

There is limited information on location in the user profile data as has been described in Section 5.3, where the *'country'* key referrers to the geo-location of the IP address used and *'country_code'* refers to the location specified by the user in the profile settings. In the *IMVULog* file the client and server IP addresses are listed. When the system is directly connected to the Internet and receives an external Wide Area Network (WAN) public IP address, this IP address is listed in the log file.

6. IMVU INTERNAL MEMORY FORENSICS

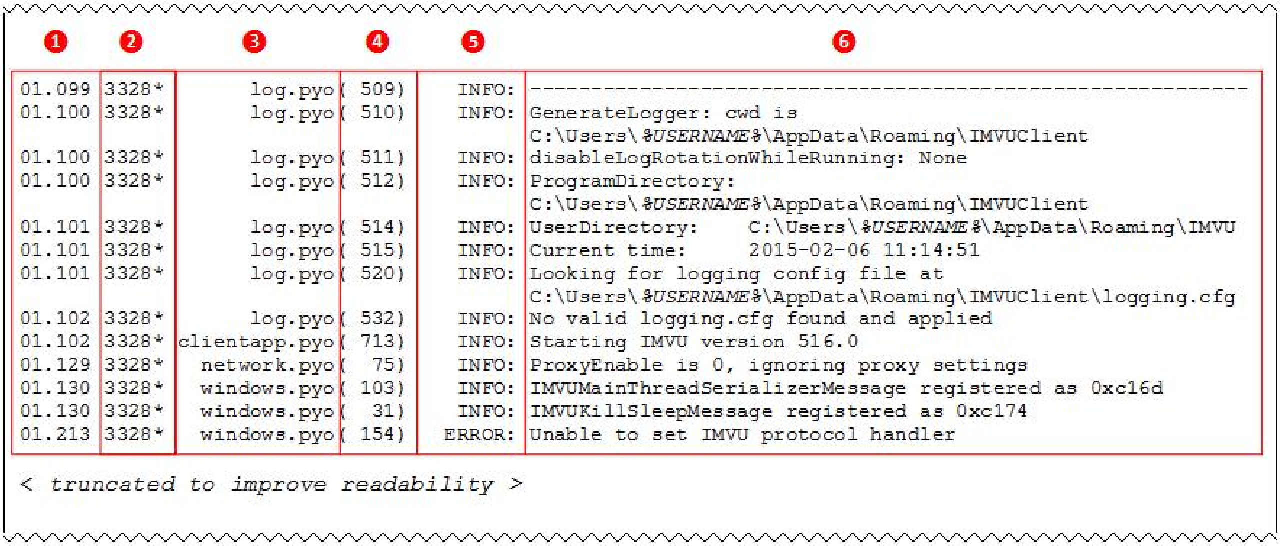
In this test, authors have explored the Python source code to understand how log records are buffered in memory. Whenever a record is added into the buffer, a check is made to see whether the buffer should be flushed to file or not. In fact, the buffer is periodically flushed whenever the buffer is full, or when an event of a certain severity happened. After making a memory dump of a running system, authors were able to retrieve log record data as previously described above. Besides, because a record consists of isolated fragments, it is difficult to link the messages to a date and time, a chat ID or the participants. IMVU also creates a SQLite database in memory. Authors were able to modify the source code and save this database to hard disk. The database consists of only a single table and contains information about the HTTP cached files. IMVU (log) files are nonvolatile and normally available for investigation.

Authors were also able to retrieve the IMVU password in clear text from memory, even when the option 'Save Password' not was used and therefore the password not could be retrieved from the registry.

7. CONCLUSION

In this paper, we present a forensic acquisition process of IMVU apps for Windows. IM is one of the most popular digital communication technologies and widely used, but also being exploited by criminals due to its popularity and privacy features. So it is important that we are able to collect artefacts of IM apps. While most IM apps use a well-known formatted history file (also known as IM log or chat log) and there are tools available for digital evidence collection, this is not the case with IMVU. IMVU has a set of fixed number log files that contain not only chat messages, but also identifiable data. IMVU app also deletes old log files when the number of log files reaches a limit number. Forensic artefacts can also be found in user profile data, authenticated data and location data. Indeed, we can also retrieve IMVU artifact from the internal memory by explore its SQLite databases.

We are working on the acquisition and analysis more forensic artefacts of an IMVU app such as registry information, network traffic data and unallocated disk space. In the case of analysing large amount of forensics data, we also look at using data mining and knowledge map techniques [18][19].



```
01.099 3328*        log.pyo( 509)    INFO: ------------------------------------------------------------
01.100 3328*        log.pyo( 510)    INFO: GenerateLogger: cwd is
                                           C:\Users\$USERNAME$\AppData\Roaming\IMVUClient
01.100 3328*        log.pyo( 511)    INFO: disableLogRotationWhileRunning: None
01.100 3328*        log.pyo( 512)    INFO: ProgramDirectory:
                                           C:\Users\$USERNAME$\AppData\Roaming\IMVUClient
01.101 3328*        log.pyo( 514)    INFO: UserDirectory:    C:\Users\$USERNAME$\AppData\Roaming\IMVU
01.101 3328*        log.pyo( 515)    INFO: Current time:     2015-02-06 11:14:51
01.101 3328*        log.pyo( 520)    INFO: Looking for logging config file at
                                           C:\Users\$USERNAME$\AppData\Roaming\IMVUClient\logging.cfg
01.102 3328*        log.pyo( 532)    INFO: No valid logging.cfg found and applied
01.102 3328* clientapp.pyo( 713)    INFO: Starting IMVU version 516.0
01.129 3328*   network.pyo(  75)    INFO: ProxyEnable is 0, ignoring proxy settings
01.130 3328*   windows.pyo( 103)    INFO: IMVUMainThreadSerializerMessage registered as 0xc16d
01.130 3328*   windows.pyo(  31)    INFO: IMVUKillSleepMessage registered as 0xc174
01.213 3328*   windows.pyo( 154)   ERROR: Unable to set IMVU protocol handler

< truncated to improve readability >
```

Figure 3. Partial IMVU.log file

REFERENCES

Wikimedia Foundation, Inc. (2014) Comparison of instant messaging clients. Retrieved on December 17 from http://en.wikipedia.org/wiki /Comparison_of_instant_messaging_clients

Wikimedia Foundation, Inc., (2013) Child grooming. Retrieved on December 3 from http: //simple.wikipedia.org/wiki/Child_grooming

Europol (2013) Threat Assessment - Italian Organised Crime. Retrieved on January 19 from https://www.europol.europa.eu/sites/ default/files/publications/italian_organised _crime_threat_assessment_0.pdf

IMVU (2014) About IMVU. Retrieved on December 17 from http://nl.imvu.com/about/

M. Goodman (2014) Crime and Policing in Virtual Worlds, Retrieved on December 17 from http://f3magazine.unicri.it/? p=360

M. Garcia-Ruiz et al. (2011) An Overview of Child Abuses in 3D Social Networks and Online Video Games. *Investigating Cyber Law and Cyber Ethics, Issues, Impacts and Practices, Hershey, Information Science Reference (IGI Global), p. 321*

Paraben Corporation (2014) Paraben Chat Examiner. Retrieved on December 18 from: https://www.paraben.com/chat-examiner.html.

Belkasoft (2014): http://belkasoft.com/

National Institute of Justice (2007) Investigations Involving the Internet, Retrieved from https://www.ncjrs.gov/pdffiles1/nij/

M. Kiley, S. Dankner and M. Rogers (2008) Forensic Analysis of Volatile Instant Messaging *Advances in digital forensics IV, New York, Springer,* 2008

B. Lun (2012) Forensic Investigation for Instant Messenger, *School of Computing and Mathematical Sciences, Auckland, New Zealand*, 2012

M. Dickson (2006), “An examination into AOL Instant Messenger 5.5 contact identification,” Elsevier Ltd., 2006

M. I. Husain and R. Sridhar (2009) iForensics: Forensic Analysis of Instant Messaging on Smart Phones *Digital Forensics and Cyber Crime, Albany, NY, USA, Springer*, p. 182

C. Sgaras, M.-T. Kechadi and N.-A. Le-Khac (2014) Forensics Acquisition and Analysis of instant messaging and VoIP applications Springer Verlag LNCS 3850, 2011

S. Garfinkel, A. J. Nelson and J. Young (2012) A general strategy for differential forensic analysis *Elsevier Ltd.*

AutoIt Consulting Ltd (2014) AutoIt, Retrieved on December 20 from https://www.autoitscript.com/site/autoit/

Mozilla Developer Network, (2015) “Retrieved on February 5 from https://developer.mozilla.org/en-US/

L. M. Aouad, N-A. Le-Khac and M-T. Kechadi, *"Lightweight Clustering Technique for Distributed Data Mining Applications"*, Springer LNAI 4597, 2007,

N-A. Le-Khac, L. M. Aouad and M-T. Kechadi *"A new approach for Distributed Density Based Clustering on Grid platform"*, Springer LNCS 4587, 2007.