Science & Justice 59 (2019) 337–348

Contents lists available at ScienceDirect

Science & Justice

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

Forensic investigation of cross platform massively multiplayer online games: Minecraft as a case study

D.C. Paul J. Taylor[a], Henry Mwiki[a], Ali Dehghantanha[b,*], Alex Akibini[c], Kim Kwang Raymond Choo[d], Mohammad Hammoudeh[e], Reza Parizi[f]

[a] School of Computing, Science and Engineering, University of Salford, UK
[b] Cyber Science Lab, School of Computer Science, University of Geulph, Ontario, Canada
[c] Liverpool John Moores University, Liverpool, UK
[d] Department of Information Systems and Cyber Security, University of Texas at San Antonio, San Antonio, TX 78249, USA
[e] School of Computing, Mathematics and Digital Technology, Manchester Metropolitan University, UK
[f] Kennesaw State University, Marietta, GA, USA

ARTICLE INFO

Keywords:
Massively multiplayer online games (MMOG)
Minecraft forensics
MMOG forensics
Game forensics

ABSTRACT

Minecraft, a Massively Multiplayer Online Game (MMOG), has reportedly millions of players from different age groups worldwide. With Minecraft being so popular, particularly with younger audiences, it is no surprise that the interactive nature of Minecraft has facilitated the commission of criminal activities such as denial of service attacks against gamers, cyberbullying, swatting, sexual communication, and online child grooming. In this research, there is a simulated scenario of a typical Minecraft setting, using a Linux Ubuntu 16.04.3 machine (acting as the MMOG server) and Windows client devices running Minecraft. Server and client devices are then examined to reveal the type and extent of evidential artefacts that can be extracted.

1. Introduction

A Massively Multiplayer Online Game (MMOG) allows millions of individual users to interact together in one gaming environment. Their popularity with the gaming community lies with the ability to collaborate, challenge and communicate with other like-minded individuals from all over the world. However, there is also potential that the gaming platform can be abused by criminals to commit a variety of computer misuse such as cyberbullying, swatting, and sexual offences, including against children and young people [1] [2] [45]. For example, similar to other online venues, gaming platforms can be used as recruiting grounds for child sex tourism [3].

Minecraft is one such game, which is the focus of the research due to its popularity and cross platform nature. Specifically, Minecraft is a sandbox game that allows users to construct buildings and play in multiplayer mode to compete for resources. Minecraft was originally developed by Stockholm-based company Mojang [4] and in 2014 the company was acquired by Microsoft. Investment from Microsoft allowed Minecraft to proliferate across multiple platforms including Windows, Mac OS X, iOS, Android, Xbox and Playstation, and has reportedly 55 million monthly players [5]. The cross platform nature of Minecraft was further bolstered in September 2017 with the release of the “Better Together Update”, which allowed for players on Xbox, Windows 10, Virtual Reality and mobile devices to play together either in small groups or on massive online games that allow up to millions of players [6].

MMOG's like Minecraft attract the attention of online offenders who may seek to exploit the interactive nature of the game [7]. For example, they are able to communicate with users from different jurisdictional areas (e.g. different countries and continents) and have the ability to maintain a level of anonymity that may hamper the efforts of cross-border law enforcement investigations [8].

It is not uncommon for offenders to target other users for purely malicious reasons, an act referred to in the community as ‘griefing’ [9] when it concerns players setting out to steal or destroy objects built by other players, or ‘swatting’ [2] [10] when the swatter calls an emergency service (e.g. 911 or 999) to report a fictitious serious crime involving another user (victim) that results in law enforcement actions be undertaken against the user. Distributed denial of service (DDoS) attacks can also be launched against the server hosting the game as well as against other user(s). In the latter, perpetrators can obtain the IP address of a user/victim through other online conversations [11]. It has

[*] Corresponding author.
E-mail addresses: Paul.Taylor_Titan@titan.police.uk (D.C.P.J. Taylor), Ali@CyberScienceLab.org (A. Dehghantanha), o.a.akinbi@ljmu.ac.uk (A. Akibini), raymond.choo@fulbrightmail.org (K.K.R. Choo), M.Hammoudeh@mmu.ac.uk (M. Hammoudeh), rparizi1@kennesaw.edu (R. Parizi).

https://doi.org/10.1016/j.scijus.2019.01.005
Received 9 May 2018; Received in revised form 29 December 2018; Accepted 27 January 2019

even been suggested that the use of 'booters' to facilitate DDoS attacks against Minecraft users is part of the very culture of the game [12] despite the act constituting an offence in the UK under s3 Computer Misuse Act 1990. Of most concern is the abuse of the chat facility by adult offenders in order to communicate with children and young people in a sexually explicit manner, inciting them to commit sexual acts or even grooming them with the aim of arranging a meeting in real life [13,14].

Owing to its popularity, in more recent times Minecraft has become a vehicle to assist in the delivery of malware to users who chose to apply 'skins' that modify their avatars [46]; the malware invokes malicious Powershell scripts that can disrupt computers by deleting backup data and programs. The script pushes text output to the user in a similar way to typical gamer communication and this research identifies opportunities to identify such infection.

There are mechanisms in place to assist in the prevention of online offending and protection of young gamers. One such development is the introduction of 'Realms' by Microsoft, which restricts multiplayer games to a maximum of 10 invite-only players [15]. There is also Minecraft specific advice focussed at parents to assist in education and crime prevention, such as that provided by the National Society for the Prevention of Cruelty to Children (NSPCC) [7]. Where offences are committed it is important to be cognisant of the evidential opportunities available on victim devices and the servers they use for online multiplayer games. Digital forensic investigation of victim and suspect devices is often critical in such cases and there is a need to know what data is available on the various Minecraft platforms, and where to find such data.

1.1. Problem statement

This research focuses on identifying, collecting and analysing digital artefacts in a forensically sound manner on Windows and Linux operating systems, two widely used platforms in gaming. The research aims to identify evidential artefacts on client devices that may be in the hands of victims, witnesses or offenders. An additional aim is to determine what evidential products are held on a Linux based Minecraft server should law enforcement identify such a server being either the target of computer misuse offences (e.g. under the Computer Misuse Act 1990 when considering DDoS attacks or Sexual Offences Act 2003 and Malicious Communications Act 1998 when considering offences relating to illicit communications in UK; 18 U.S. Code § 1030, 18 U.S. Code § 2701, 18 U.S. Code § 2251, etc. in the U.S.) or a repository for evidence of offending (e.g. under 18 U.S.C. § 2252A- certain activities relating to material constituting or containing child pornography in the U.S.).

The initial scoping exercise led the researchers to believe that the official Minecraft server, as provided by Microsoft and used in this research, would incorporate the 'Better Together' update advertised by Microsoft. This would have allowed Minecraft users on all device platforms to connect to the same server; however, it was discovered that cross-platform access is dependent on a number of factors. The Minecraft server provided by Microsoft allows for MMOGs to be conducted over the internet. However, this is restricted to desktop environments including Windows, Mac and Linux. The reason for this is that the Better Together update released in September 2017 became the primary version on iOS, Android, Console and Windows 10. The new version allows for massively multiplayer gaming but through a small selection of Microsoft vetted partner server providers. Becoming a Microsoft partner for the purposes of hosting such a server is beyond the scope of this research and for that reason this paper will not explore evidential opportunities on iOS and Android devices.

Thus, the server software that is made available by Microsoft and used in this research is actually referred to as *Minecraft: Java Edition* and only compatible client software available on Desktop will be able to connect to this server.

The Android Google PlayStore does have an app that allows connection to a Java Edition server. This app is called *Boardwalk* and is offered by Zhuowei Zhang [16]. The description states that Boardwalk "[a]llows you to run the PC version of Minecraft on your device", but it also warns of unreliability. Although Boardwalk initially ran, it did not progress past the initial splash screen and there was no option to connect to custom servers. There is also no comparative app on the Apple App Store, at the time of this research.

Research has been conducted by other scholars with Minecraft installations on Xbox, PlayStation 4 gaming consoles [17] and on a Windows Minecraft server [18]. However, there is no study focusing on the forensic analysis of Minecraft installed on a Linux Ubuntu server, which is the recommended Operating System for deployments on Microsoft Azure cloud platforms [19] and a popular choice in user forums [20]. The outcome of this research will be to expand upon previous findings, with a particular focus on live memory and traffic data artefacts.

1.2. Research questions

The overall goal will be to provide a framework to forensically recover evidential artefacts from the multiplayer online game Minecraft. The purpose being the discovery of evidence of offending against Minecraft end users and Linux based Minecraft server providers.

Similar to the approach undertaken by Quick and Choo [21] [22] in this research there is an attempt to answer the following questions:

1. What artefacts of evidential interest are retained on the Linux Minecraft server after installation of the Minecraft server software and the running of the game with multiple cross platform users?
2. What evidence is there on the server and client devices of communication between users running Minecraft?
3. What information can be seen from server and client logs that can assist in the identification and profiling of a user?
4. What can be seen from network traffic between clients and the server?

1.3. Outline

The structure of this paper is as follows. Section 2 contains a review of the current literature on the topic of digital forensics and MMOG's. Section 3 presents the framework used to guide the forensic investigation, followed by the experimental setup. Section 4 covers the evidential collection phase and seeks to answer the first research question. Section 5 continues with analysing the data found on the hard disk of the client. Section 6 looks at the memory, which will address the second and third questions. Section 7 looks into what can be seen the network traffic with the aim of addressing the fourth research questions. Finally, Section 8 provides a conclusion to the paper and outlines the potential for future studies.

2. Literature review

Since the introduction of MMOGs to the gaming community, it has gained significant fanfare [23]. Specifically, Minecraft has increased in popularity, including among autistic children [24] who seek to meet like-minded friends their own age. However, it is known that the platform can be abused by child sexual and other offenders [25]. The Leahy Center for Digital Investigation [23], for example, recognized such a risk and examined chat function logs from several MMOG's. They identified that chat logs were stored in plain text and different vendors took entirely different approaches to storage conventions and this highlighted a need to conduct further research with other MMOG's, such as Minecraft.

Minecraft began its existence as a Java based application and then evolved to incorporate a Windows 10 edition due to Microsoft compatibility issues relating to Java applications [26]. Players of Minecraft explore a virtual world and interact with other characters that spawn in and out of existence as gameplay progresses and users across the internet connect to multiplayer game hosting servers. There is not one clear objective for all users. Players can enter a game and choose to explore the virtual world around them, create buildings and structures, or play a survival game whereby bots and other characters have the aim of destroying one another.

Due to the acknowledgement that MMOG's can be a platform for online offending, research continues into the real-time identification of offenders through behavioural analysis [27]. The researchers look at the evidence available in the aftermath of a crime being committed. In their study, Ki, Woo and Kim [28] suggested that some individual deeds such as malicious social interactions and behaviors can be observed in MMOG's and automated bots are also used for such offending.

The personal information and virtual properties of the people who subscribe and participate in such games draws a lot of attention from attackers [29]. Other criminal activities include compromising of user accounts. A popular reason for account compromise is to achieve easy profits and in extreme scenarios this can even lead to gold farming, which is the stock piling of virtual game assets for eventual profitable sale in the real world [29].

With the rise of MMOG's being exploited for cyber crime, forensic investigators are facing challenges during the collection of evidence, which can be essential to prove that the crime did really happen [30]. In MMOG forensics, there are situations with which forensic tools can be fooled by an attacker, whereby they remove data traces so as to make sure that they cannot be easily be tracked [31]. It has been suggested that if users are engaging in criminal activity and there is no in-depth knowledge of where or how to find evidence, only the most apparent incidents of misconduct can be discerned, prevented or eradicated [18].

There has been research conducted on Xbox and PlayStation 4 platforms running Minecraft, which identified that established tools alone could not be relied upon for the retrieval of all relevant artefacts [17]. Similarly, research has been done on a Windows Minecraft server and client, which highlighted the availability of chat logs but identified that further work was required in relation to the analysis of network traffic [18]. The research aims to fill the gap by exploring evidential opportunities on a Linux server and in the cross-platform network traffic.

Forensic investigation of a cross-platform massively multiplayer online game like Minecraft is important as different users make use of different platforms and the research will help to identify what artefacts of evidential value can be found. The footprints detected by the forensic analysis from these different platforms can be used in criminal proceedings, in particular where the investigation is needed to fulfill further rigid restrictions dictated by the strict procedures agreed in court [31].

Aside from a Linux server and a Windows client, the authors considered mobile device clients. Rajendran and Gopalan [32] argued the very important challenges in mobile device forensics are variations from design to design, model to model, time to time, manufacturer to manufacturer, and the adaptation of the technology [32].

It is acknowledged that users of MMOG's like Minecraft will utilise third-party applications to communicate with other gamers whilst in play. Sound research has been conducted into the evidential retrieval of data from popular VoIP applications on mobile platforms [33] so it is now necessary to also explore the in-built chat facilities between various operating system platforms.

Open source intelligence following a report of crime initiated or facilitated on an MMOG may lead investigators to a repository of information, suspect machine or even assist in the identification of victims through attribution of their devices. The use of this intelligence could be combined with application specific forensic methods, as outlined in this paper, to counter the challenge of the growing volume of disparate data with consideration for the future use of a Digital Forensic Data Reduction Framework [34].

3. Research methodology

3.1. Forensic framework

This is an outline of the digital forensic framework and the experimental setup relied upon to conduct the research. It is necessary for the digital forensic method to adhere to the well-established guidelines and policies published internationally.

This research aims to primarily assist those involved in criminal investigations and as such it should be noted that there have been recent updates concerning the provision of digital forensic evidence in UK courts. The UK's Association of Chief Police Officers (ACPO) Good Practice Guide for Digital Evidence [35] contains four overarching principles; prevent the original data from being changed, where data has to be accessed this should be done by competent examiners, they must maintain accurate, reproducible records and the decisions being made must be the decision of the person in charge of the investigation. These principles are complemented with the practical advice available from the US National Institute for Justice's Guide for First Responders [36], which outlines guidance for handling a wide variety of digital media at crime scenes. These long standing guidance documents continue to remain but more recently, the UK's Forensic Science Regulator has mandated that anyone reporting scientific or technical work to the courts must comply with the Forensic Code of Conduct [37]. This stipulates that public, police and commercial digital forensic providers have to be accredited to International Organisation for Standardisation (ISO) and International Electrotechnical Commission (IEC) standard 17025. The National Institute of Science and Technology (NIST) suggested an outline for a digital forensic process of data collection, examination, analysis and reporting [38]. Efforts continue to be made internationally to find consensus in the approach taken by digital forensic examiners [39] and consideration was given to the above policies and guidance when conducting this research.

A decision was made to choose to utilise the framework set out by Martini and Choo [40] as shown in Fig. 1. This shares similarities with other established frameworks but allows for iterations between the examination and analysis step through to the evidence identification and preservation step, which assists in the preservation of evidence found in cloud platforms.

3.1.1. Phase 1: evidence source identification and preservation

In this first stage, the hardware had to be subjected to analysis, which differed depending on the platform. The Minecraft server and Windows client were contained within a virtual machine and the files of interest were VMWare virtual disk files (VMDK) and virtual memory

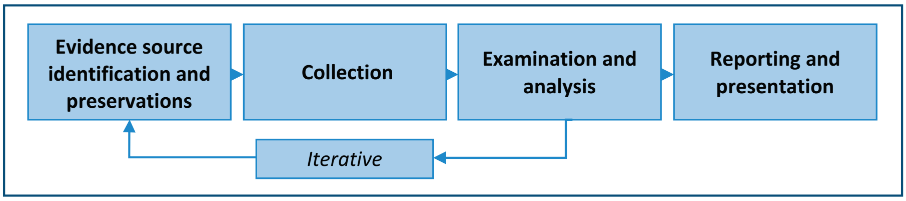

Evidence source identification and preservations
Collection
Examination and analysis
Reporting and presentation
Iterative

Fig. 1. Digital forensics framework of Martini and Choo.

files (VMEM). These were extracted and a bit-for-bit working copy was produced of each.

3.1.2. Phase 2: collection

For this phase files from both the server and client machines were isolated and collected. The files that contained the items of interest for forensic analysis, keyword searching and network traffic data were isolated by determining their state or presence on the machine before and after a particular process was run. MD5 and SHA1 hash values corresponding to the files of interest were collected and retained for subsequent comparison to ensure that working file copies were true to the original.

3.1.3. Phase 3: analysis

Firstly, in this phase there was an examination of the data contained within the forensic captures of the hard disks and memory from the Linux server and Windows clients. Secondly, traffic data captured during live gameplay was examined. The sets of data were subjected to keyword searching for known usernames, passwords and extracts of conversation exchanged during gameplay. Log files stored on the machines were analysed and carved to establish the existence of evidential artefacts.

3.1.4. Phase 4: reporting and presentation

For this final phase, a short-form template report was recommended that could be used as an evidential exhibit for production with accompanying witness testimony in court.

3.2. Experimental setup

For this research, there was a simulation of the scenario of an individual running a Minecraft server on their Linux Ubuntu 16.04.3 Server machine and hosting a multiplayer game. The other gamers were connected to the server from a Windows 7 Enterprise Service Pack 1 PC. The Linux server and Windows client were hosted in their own Virtual Machines (VM) on the same network and used the snapshot tool of the VMWare software to create 8 snapshots representing different environments based on the real-life use of the server and client. Base VM snapshots were used as a control to compare against snapshots taken after some activity. Table 1 provides a list of the different VM snapshots captured and the description of the activity taken prior to the snapshot. Fig. 2 shows a flow diagram of the snapshots taken.

The Windows version used was a free developer edition from Microsoft. A Windows 7 Enterprise Service Pack 1 virtual machine file was downloaded from *https://developer.microsoft.com/en-us/microsoft-edge/tools/vms*/.

A network adaptor was added and set to allow a bridged connection from the virtual machine. A snapshot was taken in this state (1.0). VM snapshots were used as opposed to physical hardware for practical time considerations and convenience. This was aligned to the points made by Teing et al. [41] and Quick and Choo [21] in relation to the ease in which a machine state can be reverted to a restore point for repeated experimental activity and to allow for the file space to be kept to a minimum for collection of evidence and analytical efficiency.

During each snapshot, VMWare creates multiple *.vmdk* files so in order to combine these into one file to allow for forensic acquisition the tool *vmware-diskmanager* was used with the following command: -r "Virtual Disk-000001.vmdk" –t 0 [new file name].vmdk.

Live RAM captures were represented by the corresponding VMEM files generated by VMWare upon the initialisation of a snapshot. In order to simulate a real-life situation, the snapshots (and hence the RAM captures) were taken following a particular activity and whilst the machine was powered on. Guidance to UK police officers is to isolate the power immediately from a machine rather than initiate a shutdown procedure [35] and hence snapshots from live machines were most representative of the scenario.

In order to prepare the installation snapshot on Windows (1.1), Minecraft was downloaded from the official site *https://minecraft.net/en-us/download*/ and installed to the default directory. A snapshot was taken at this stage and an attempt was made to 'Play', however it crashed due to a graphics adaptor issue. It was necessary to install VMWare Tools and to disable Windows Update to prevent changes to the system outside of the Minecraft installation.

Following the installation of VMWare tools the system had to be restarted. A further snapshot was taken. A test was then conducted to see if Minecraft could be played and it could so the latter snapshot was taken as the install snapshot.

The activity taken to create the 'Activity VM' snapshots involved connecting to the server, starting a new game and communicating through the chat facility. It was decided that no particular avatar interaction was required due to previous research by Alstad, Duncan, Detlor et al. [42] finding that network activity levels between active and idling players were similar.

The capturing of network traffic on desktop operating systems is relatively straightforward when compared to the difficulties encountered with capturing traffic from mobile devices [34]. The approach taken was to capture the network traffic by running Wireshark 2.4.2 in the host environment, with the option set to only capture traffic passing through the virtual network adaptor. Table 2 lists the tools used for the collection and analysis phases of the research.

In order to capture the traffic exclusively between the Windows client and the Linux server the following steps were taken:

1. Open Wireshark on the host machine and capture all traffic on the default network adaptor.
2. This was tested first with ping commands between machines.
3. Apply a filter for the two relevant IP addresses; *(ip.addr = = 192.168.0.29 and ip.addr = = 192.168.0.82)*

Table 1
Configurations of virtual machines on Linux (server) and Windows (client).

VM Snapshot	Description
Base VM 1.0, 2.0	The Base VM snapshots were prepared for the following operating systems: ● Windows 7 Enterprise Service Pack 1 (Build 7601) with 1GB RAM (1.0). ● Linux Ubuntu 16.04.3 Server with 2GB RAM and 20GB hard disk (2.0).
Install VM 1.1, 2.1	Minecraft software was installed on the operating systems and further snapshots were taken. ● On the Windows machine; Minecraft version 1.12.2 (1.1). ● On the Linux machine; Minecraft Server version 1.12.2 (2.1).
Activity VM 1.2, 2.2	The Minecraft server was initiated on the Linux machine and the Minecraft client on the Windows machine then connected over the network. Snapshots were taken on the Windows machine (1.2) and the Linux machine (2.2).
Uninstall VM 1.3, 2.3	Further snapshots were created following the standard uninstallation of Minecraft from the operating systems. Standard documented uninstall procedures were conducted on the Windows (1.3) and Linux (2.3) machines.

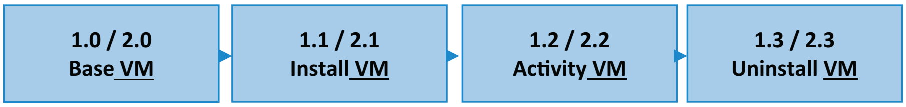


Fig. 2. VM snapshots created on Linux and Windows machine running Minecraft server and client respectively.

4. Export the displayed data to a new file *capture1*.pcap (to remove background noise from the host machine).

4. Server analysis

In this analysis there was substantial data involved so it was decided that only data which would contain potential information of interest for Minecraft forensics would be collected. Examples of the data collected are chat log artefacts, artefacts related to users, artefacts related to the game server and client machine and network artefacts [18]. These data can be found in files stored in different location such as Minecraft log files, Window system files, and unallocated partitions.

For control purposes examination of base snapshots were confirmed to have no Minecraft artefacts and as such were suitable for comparison against subsequent snapshots.

The image files were examined for changes between the base Linux server and the server following installation of the Minecraft software and game activity.

The */etc/hostname* directory contains the *hostname* file and this showed that the server had been titled ‘minecraft-server’. The *passwd* file within/*etc*/shows the user account, which for this research was ‘taylor’.

The Date Modified attribute had been updated for 3 sub-directories within/*var/lib*:

var/lib/update-notifier; The file named updates-available residing within this directory had been changed to reflect an increase of 2 updatable packages and 6 security updates, however nothing identifying Minecraft was present.

var/lib/lxd; The contents of this directory remained empty and there had been no change other than the Date Modified attribute being updated to reflect the time of installation.

var/lib/snapd; The file named *state.json* file contained a date string that reflected the time of last access, under a heading ‘last-refresh’. This was accurate when compared to the time record in the experiment for last access to the server.

The *var/tmp* directories all displayed an updated Date Modified attribute and a new directory had been created; */hsperfdata_taylor*. Contained within was a file named *1363* that contained file paths for Minecraft's Java installation and contained no other evidential material.

There were no changes to the */opt* directory but significant logs available within */var/log*. The *syslog* file recorded the Minecraft server's IP address of 192.168.0.23 as per the DHCP requests made following initialisation. There were also records following each initialisation indicating that the system time had been synchronised over the internet with the time server at ntp.ubuntu.com.

The *auth.log* file detailed the time and dates of successful logins to the server under the root user, taylor.

There were significant items of interest contained with the user directory */home/taylor*. The file *.bash_history* listed all the recent commands input by the server operator, including the download command necessary to obtain the Minecraft software; *wgethttps://s3.amazonaws.com/Minecraft.Download/versions/1.12.2/minecraft_server.1.12.2.jar*

The downloaded software *minecraft_server.1.12.2.jar* was saved to the same directory with a Date Modified value not representative of any user action and likely the date and time it was uploaded to the Amazon server by developers.

Following installation of the software a number of files were created in the same directory. Of note is the file *server.properties*, which allows the server operator to configure a number of settings prior to launching a game. The contents of *server.properties* is in plain text and useful information like time stamp, port number and ‘Message of the Day’ can be seen and compared to defaults, which are shown in Fig. 3; the values changed from default are displayed in bold and underlined. Difficulty was set to 0 to prevent spawned characters from attacking the user and preventing the necessary activity required to conduct the research. Online-mode was set to 0 to prevent the server from attempting to verify connecting users with Microsoft, which was necessary due to running the applications on a LAN.

The file *usercache.json* provides a list of previously connected users. The user name is recorded, in this case ‘taylor_salford’, a Unique User Identification Number is assigned and an expiry date is set for one month after their initial joining of the server.

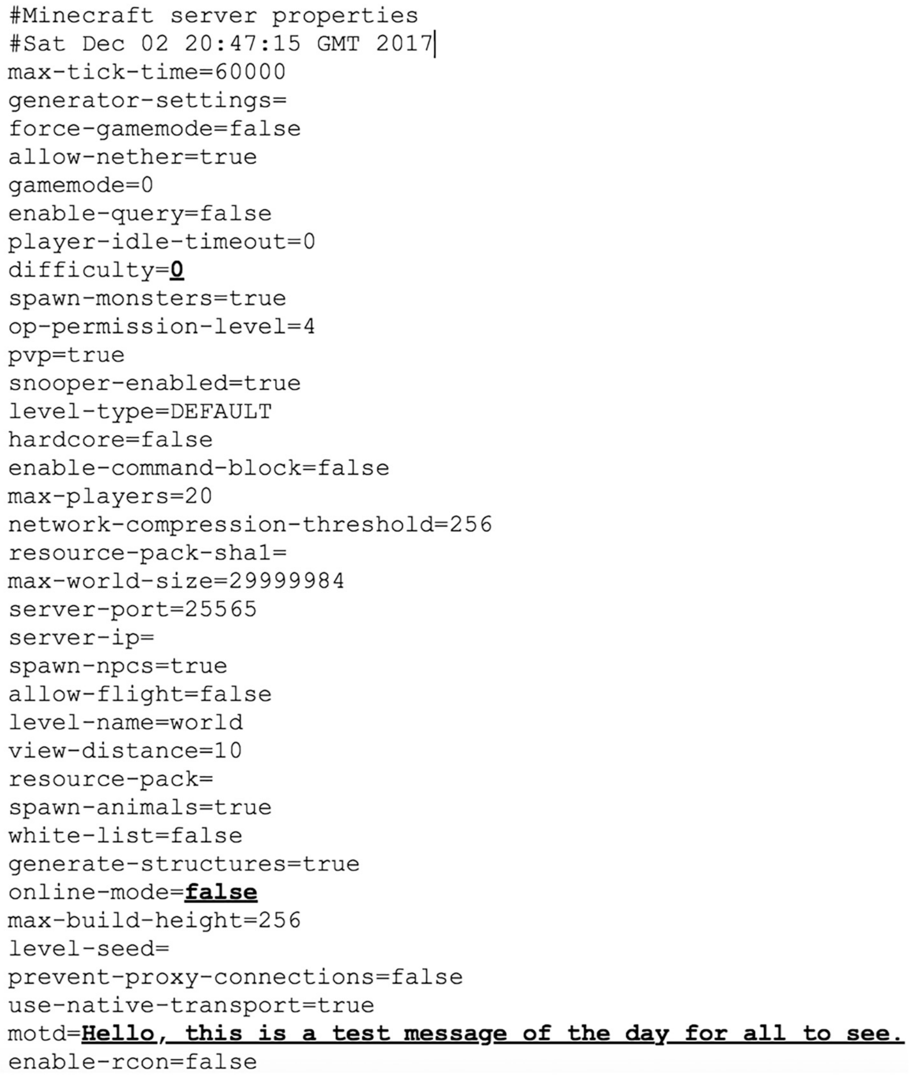
Within */home/taylor/*logs there were a number of files of the name format *yyyy-mm-dd-[sequential number from 1].log.gz*. Each of these files contained complete outputs of the server user's screen, including timestamps for when commands were typed after the initialisation of the Minecraft server software. The most recent file contained within the */home/taylor/logs* directory was *latest.log*, the contents of which are shown in Fig. 4.

Fig. 4 shows that the contents of *latest.log* replicate the screen output made to the Minecraft server operator. The following information could be useful to investigators:

- “Starting Minecraft server on *:25565”; This is the port that the server will be available on.
- “Server is running in offline/insecure mode!”; This is a change from the default option and allows for users to connect to the server with any username they chose; there will be no verification of the username with Microsoft. The feature can be enabled to assist with blacklisting of offending users.
- The entry at time [20:48:27] shows username taylor_salford connecting from 192.168.0.82, which is in fact the Windows client used in the research.

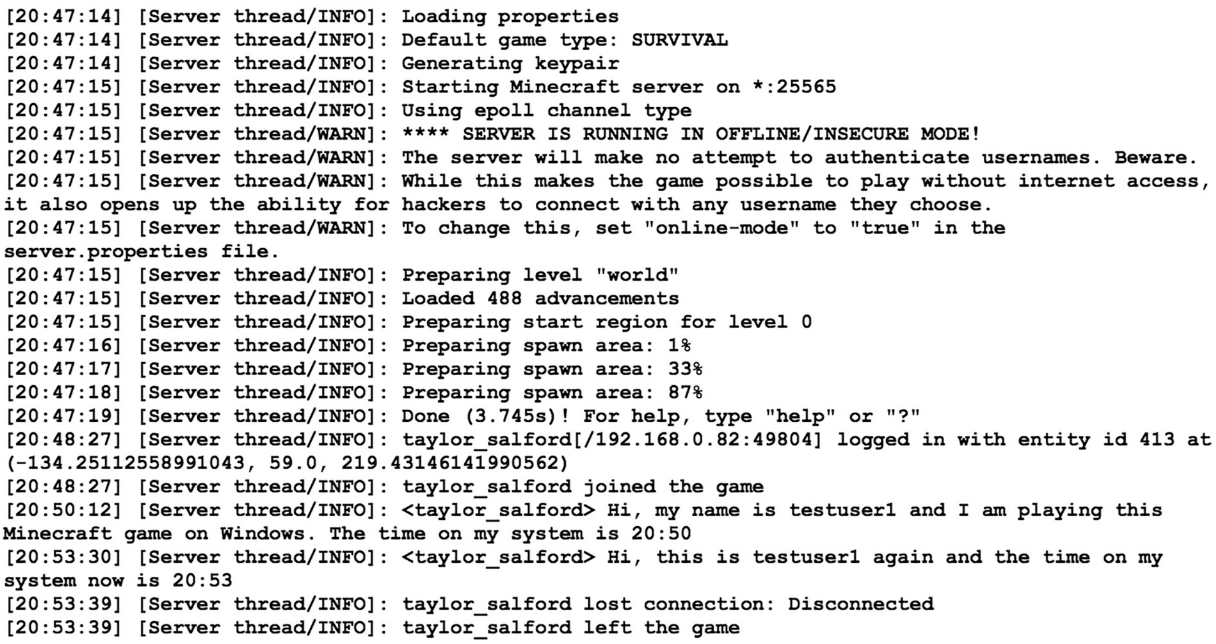
Table 2
Research tools and their usage.

Tool	Usage
VMWare Fusion version 8.5.8 (Mac)	To create the Linux server and Windows client and allow for the capture of system snapshots.
FTK Imager version 3.2.0 (Windows)	To acquire forensic images for .VMDK and .VMEM files from the Linux and Windows machine snapshots.
Autopsy version 4.5.0 (Windows)	To view and analyse directories, files, Windows registry data, page and swap files.
HxD version 1.7.7.0 (Windows)	To keyword search live memory captures and carve relevant data.
Volatility version 2.6 (Linux)	For automated carving of evidential artefacts from captured live memory.
Wireshark version 2.4.2 (Mac)	To capture network traffic passing through the virtual adaptor on the host machine.
Network Miner version 2.2 (Mac)	To parse out relevant information from the network traffic capture files.
Windows Event Viewer (Windows)	To view relevant events recorded in the Windows system.



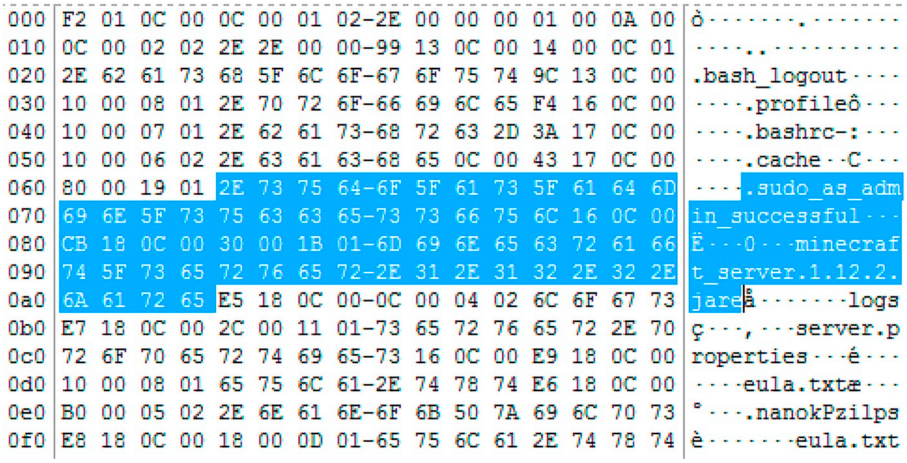
```
#Minecraft server properties
#Sat Dec 02 20:47:15 GMT 2017
max-tick-time=60000
generator-settings=
force-gamemode=false
allow-nether=true
gamemode=0
enable-query=false
player-idle-timeout=0
difficulty=0
spawn-monsters=true
op-permission-level=4
pvp=true
snooper-enabled=true
level-type=DEFAULT
hardcore=false
enable-command-block=false
max-players=20
network-compression-threshold=256
resource-pack-sha1=
max-world-size=29999984
server-port=25565
server-ip=
spawn-npcs=true
allow-flight=false
level-name=world
view-distance=10
resource-pack=
spawn-animals=true
white-list=false
generate-structures=true
online-mode=false
max-build-height=256
level-seed=
prevent-proxy-connections=false
use-native-transport=true
motd=Hello, this is a test message of the day for all to see.
enable-rcon=false
```

Fig. 3. Contents of server.properties.



```
[20:47:14] [Server thread/INFO]: Loading properties
[20:47:14] [Server thread/INFO]: Default game type: SURVIVAL
[20:47:14] [Server thread/INFO]: Generating keypair
[20:47:15] [Server thread/INFO]: Starting Minecraft server on *:25565
[20:47:15] [Server thread/INFO]: Using epoll channel type
[20:47:15] [Server thread/WARN]: **** SERVER IS RUNNING IN OFFLINE/INSECURE MODE!
[20:47:15] [Server thread/WARN]: The server will make no attempt to authenticate usernames. Beware.
[20:47:15] [Server thread/WARN]: While this makes the game possible to play without internet access,
it also opens up the ability for hackers to connect with any username they choose.
[20:47:15] [Server thread/WARN]: To change this, set "online-mode" to "true" in the
server.properties file.
[20:47:15] [Server thread/INFO]: Preparing level "world"
[20:47:15] [Server thread/INFO]: Loaded 488 advancements
[20:47:15] [Server thread/INFO]: Preparing start region for level 0
[20:47:16] [Server thread/INFO]: Preparing spawn area: 1%
[20:47:17] [Server thread/INFO]: Preparing spawn area: 33%
[20:47:18] [Server thread/INFO]: Preparing spawn area: 87%
[20:47:19] [Server thread/INFO]: Done (3.745s)! For help, type "help" or "?"
[20:48:27] [Server thread/INFO]: taylor_salford[/192.168.0.82:49804] logged in with entity id 413 at
(-134.25112558991043, 59.0, 219.43146141990562)
[20:48:27] [Server thread/INFO]: taylor_salford joined the game
[20:50:12] [Server thread/INFO]: <taylor_salford> Hi, my name is testuser1 and I am playing this
Minecraft game on Windows. The time on my system is 20:50
[20:53:30] [Server thread/INFO]: <taylor_salford> Hi, this is testuser1 again and the time on my
system now is 20:53
[20:53:39] [Server thread/INFO]: taylor_salford lost connection: Disconnected
[20:53:39] [Server thread/INFO]: taylor_salford left the game
```

Fig. 4. Contents of /home/taylor/logs/latest.log.



```
000 F2 01 0C 00 0C 00 01 02-2E 00 00 00 01 00 0A 00 ò·······.·······
010 0C 00 02 02 2E 2E 00 00-99 13 0C 00 14 00 0C 01 ····..··········
020 2E 62 61 73 68 5F 6C 6F-67 6F 75 74 9C 13 0C 00 .bash_logout····
030 10 00 08 01 2E 70 72 6F-66 69 6C 65 F4 16 0C 00 ····.profileô···
040 10 00 07 01 2E 62 61 73-68 72 63 2D 3A 17 0C 00 ····.bashrc-:···
050 10 00 06 02 2E 63 61 63-68 65 0C 00 43 17 0C 00 ····.cache··C···
060 80 00 19 01 2E 73 75 64-6F 5F 61 73 5F 61 64 6D ····.sudo_as_adm
070 69 6E 5F 73 75 63 63 65-73 73 66 75 6C 16 0C 00 in_successful···
080 CB 18 0C 00 30 00 1B 01-6D 69 6E 65 63 72 61 66 Ë···0···minecraf
090 74 5F 73 65 72 76 65 72-2E 31 2E 31 32 2E 32 2E t_server.1.12.2.
0a0 6A 61 72 65 E5 18 0C 00-0C 00 04 02 6C 6F 67 73 jareå·······logs
0b0 E7 18 0C 00 2C 00 11 01-73 65 72 76 65 72 2E 70 ç···,···server.p
0c0 72 6F 70 65 72 74 69 65-73 16 0C 00 E9 18 0C 00 roperties···é···
0d0 10 00 08 01 65 75 6C 61-2E 74 78 74 E6 18 0C 00 ····eula.txtæ···
0e0 B0 00 05 02 2E 6E 61 6E-6F 6B 50 7A 69 6C 70 73 °···.nanokPzilps
0f0 E8 18 0C 00 18 00 0D 01-65 75 6C 61 2E 74 78 74 è·······eula.txt
```

Fig. 5. Data contained within /home/taylor showing Minecraft server software.

- Any chat typed out by the connected user taylor_salford is output to the screen on the server. The test message shows that the time on the Windows client machine was in sync with that of the Linux server.

Following the uninstallation process, evidence was still available to suggest the machine was running the Minecraft sever software.

The *.bash_history* that resides in */home/taylor* shows the command used to uninstall Minecraft; rm –vr ./*

Further to this, hex data exists for the directory*/home/taylor* and indicates that the software was once present on the machine, as per Fig. 5.

5. Client analysis

Similar to other massively multiplayer online games such as Second Life and World of Warcraft, the main starting point of Minecraft investigation on the Windows client machine is in:

Users/[username]/AppData/Roaming/.minecraft.

The files of particular interest within this directory are:

*Users/[username]/AppData/Roaming/.minecraft/*logs/*latest.log.*

Users/[username]/AppData/Roaming/.minecraft/launcher_profiles.json.

Users/[username]/AppData/Roaming/.minecraft/servers.dat.

The examination began by identifying artefacts related to the user. Firstly, the user setting file was found, which is:

Users/[username]/AppData/Roaming/.minecraft/logs/2017-11-27-1.log.gz/2017-11-27-1.log.

It was discovered that the name of the file itself reflected the date the user settings were made. The file also provides timestamp of the user setting, as can be seen in Fig. 6.

Knowing user details is important as they can be used by an investigator to reveal identity of the person who was playing the game. It was possible to find the information related to users in this file:

Users/[username]/AppData/Roaming/.minecraft/launcher_profiles.json.

The file provided details of the user such as Universal Unique Identifier (UUID) which is used to identify player accounts (prefixed with *'profiles:'*), email address of the player (prefixed with *'username:'*), display name of the player in the game (prefixed *with 'displayName:'*), user account number (prefixed with *'selectedUser:'*), access token (prefixed with *'accessToken:'*), analytics token (prefixed with *'analyticsToken:'*) and client token (prefixed with *'clientToken:'*). Fig. 7 shows the contents of this file.

Chat logs were stored in:

*Users/[username]/AppData/Roaming/.minecraft/*logs/*latest.log.*

It was possible to obtain the full chat logs. They were not encrypted and were saved as plain text as such were easily accessible. Apart from chat logs, *latest.log* file also provided details such as an IP address to which the Windows client connected, port number and the display name of the player. This is fixed and will only change when user opens another account according to Minecraft [18]. Timestamps of activity were present but dependent on the local machine time zone settings, when the player started and stopped the game. Fig. 8 shows the discrepancy in time between the player's local and machine time.

Logs were also recorded in *latest.log* when attempts were made to log into the game using the same accounts details. An error would be output as there is a duplicate login and the player had already logged in from another location. This logging this can help the investigator know if the account was compromised and someone else used the account to log into the game. Fig. 9 shows an example one the message output to the log file.

Artefacts related to server were identified on the Windows client. The file

Users/[username]/AppData/Roaming/.minecraft/servers.dat contained information about the server, the information is stored in plain text and easily readable. This file provides information such as server name and IP address of the Minecraft server.

Like any other application, Minecraft can crash and it was discovered that the file *Users/[username]/AppData/Roaming/.minecraft/crash-reports* provided details of the crash, which included the source of the crash and associated timestamp.

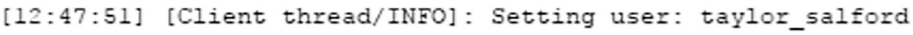
After uninstallation of the Minecraft software on the Windows machine it could be seen that the folder *Users/[username]/AppData/Roaming/.minecraft/* remained on the hard drive, however it was empty.

Log files are important artefacts to extract as they allow for operating system events to be determined [43]. Log analysis was conducted by searching for the term *Minecraft* and this involved going through the entries identifying events which were relevant to Minecraft.

It was possible to locate event entries referencing the installation and uninstallation of the Minecraft application. This provided timestamps in Windows event files such as *Application.evtx*, as shown in Fig. 10.

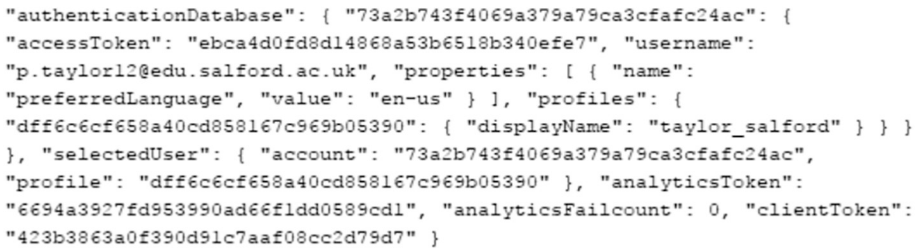
6. Memory analysis

The Volatility tool was utilised in order to analyse the live memory



```
[12:47:51] [Client thread/INFO]: Setting user: taylor_salford
```

Fig. 6. Contents of 2017-11-27-1.log.



```
"authenticationDatabase": { "73a2b743f4069a379a79ca3cfafc24ac": {
"accessToken": "ebca4d0fd8d14868a53b6518b340efe7", "username":
"p.taylor12@edu.salford.ac.uk", "properties": [ { "name":
"preferredLanguage", "value": "en-us" } ], "profiles": {
"dff6c6cf658a40cd858167c969b05390": { "displayName": "taylor_salford" } } }
}, "selectedUser": { "account": "73a2b743f4069a379a79ca3cfafc24ac",
"profile": "dff6c6cf658a40cd858167c969b05390" }, "analyticsToken":
"6694a3927fd953990ad66f1dd0589cd1", "analyticsFailcount": 0, "clientToken":
"423b3863a0f390d91c7aaf08cc2d79d7" }
```

Fig. 7. Contents of launcher_profiles.json file.

captures from the Windows machine following Minecraft activity. There were difficulties with adding a custom profile for the Linux machine in Volatility; Linux Ubuntu 16.04.3 Server does not have a pre-configured profile available for download and so the memory carving was completed manually.

On the Windows client the *netscan* Volatility plugin failed to identify an IP address for the Linux server.

The *pscan* Volatility plugin was used to compare processes in memory on the Windows machine before and after the game being started and ended. It could be seen that the process named *MinecraftLaunc* was present in memory from the moment of boot up. Following the software being launched and a game being played it could be seen that a process called *javaw.exe* had been spawned from *MinecraftLaunc* and both had exited upon the software being closed.

The *memdump* plugin was run against the *MinecraftLaunc* process causing a 254.7 MB .dmp file to be isolated, which contained the contents of all the memory concerned with the running process. Fig. 11 shows the data available in the memory of the dumped process. The IP address and name of the Linux server was recorded, as was the server owner's message of the day and the last communication sent from the Windows client over the Minecraft chat facility. If the Windows machine was in the possession of a victim, this information could assist in the identification of a server that hosted data relating to other users responsible for committing illicit acts against the victim. Similarly, if the Windows machine was in the hands of a suspect, then identification of the server it was connected to may provide investigators with opportunities to identify the server owner and establish the nature of the relationship to the suspect and assess further opportunities for victim safeguarding, for example.

The live memory capture from the Windows machine was manually searched for known information, to establish what was retained in memory during active game play.

The IP address and name of the server was recorded in memory in several locations. Additionally, the following data provides time stamps for the connection to the server, the IP address and port number of the server and all chat sent from the client machine.

[12:48:22] [Client thread/INFO]: Connecting to 192.168.0.29, 25,565.

[12:48:28] [Client thread/INFO]: Loaded 16 advancements

[12:50:12] [Client thread/INFO]: [CHAT] < taylor_salford > Hi, my name is testuser1 and I am playing this Minecraft game on Windows. The time on my system is 20:50

[12:53:30] [Client thread/INFO]: [CHAT] < taylor_salford > Hi, this is testuser1 again and the time on my system now is 20:53

The same chat information was stored in one other location in memory and could be found with a slightly adapted keyword search of "info [CHAT]".

The Linux live memory was analysed and further useful information was identified. If the local server IP address was not already known then it could be searched for by performing a keyword search in a hex viewer for "DHCPOFFER" and reading the adjacent timestamp and IP address displayed.

The Windows client IP address was only stored in the Linux memory in one location, which was in an area of strings that replicated the contents of the log file from Fig. 4. Similarly, this area of memory provided the clearest and most chronologically sound evidence of the chat sent from the Windows client.

Artefacts of chat were found elsewhere in memory, however they were often fragmented and incoherent, as is shown in Fig. 12.

The full contents of the *server.properties* file was stored in memory, which included the message of the day.

Fig. 13 shows that the Unique User Identification Number (UUID) assigned by the server was stored in memory alongside the connected username.

The username and password for logging into the server existed in memory adjacent to the bash command *ifconfig*, as shown in Fig. 13.

Although the server username and password does not reside in memory alongside any data that could easily be located to identify this information, the username is stored many times in memory following the string, "/home/".

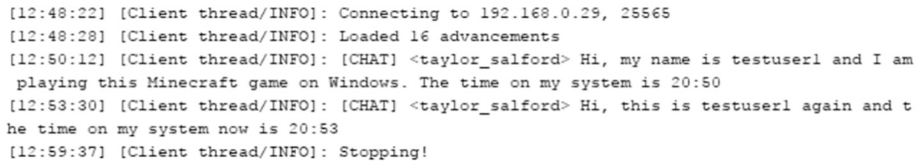
7. Network analysis

A Minecraft game begins when the server operator logs in, starts up the Minecraft software and waits for incoming connections. A client can then choose to connect to either a recommended Microsoft server or manually enter an IP address and connect to a private server (desktop Java Edition only). Upon connection, a welcome message is displayed to the client in the form of a 'message of the day' and the user has a character/avatar immediately spawned into an expansive landscape ready for gameplay.

Upon successful connection, the Windows client immediately passes its Minecraft username to the server and the unique user identification number is passed back in return. Then follows a rapid succession of packets being passed back and forth that contain seemingly encoded data.

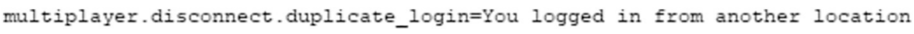
Over the period of activity of approximately 5 min a significant volume of traffic was observed between the Linux server and Windows client. The average flow of data was 21 KB/s.

The open ports on the connected machines could be determined using the Wireshark *Statistics* > *Endpoints* feature. Fig. 14 shows that the Linux machine (192.168.0.29) communicated through port 25,565



```
[12:48:22] [Client thread/INFO]: Connecting to 192.168.0.29, 25565
[12:48:28] [Client thread/INFO]: Loaded 16 advancements
[12:50:12] [Client thread/INFO]: [CHAT] <taylor_salford> Hi, my name is testuser1 and I am
 playing this Minecraft game on Windows. The time on my system is 20:50
[12:53:30] [Client thread/INFO]: [CHAT] <taylor_salford> Hi, this is testuser1 again and t
he time on my system now is 20:53
[12:59:37] [Client thread/INFO]: Stopping!
```

Fig. 8. Contents of latest.log file.



```
multiplayer.disconnect.duplicate_login=You logged in from another location
```

Fig. 9. Duplicate username log.

(as set in the *server.properties* file) yet the Windows machine (192.168.0.82) incrementally changed the port number, albeit the majority of communication was on port 49804. The Windows port number was not pre-determined prior to running the software.

The contents of the conversation generated by the Windows client utilising the chat feature was sent via TCP in plaintext, as can be seen in Fig. 15.

Also sent in plaintext was the server operator's message of the day, which can be located by searching for the identifiers that enclose the message. Fig. 16 shows the message in its entirety and a ‘tcp contains’ search for *{“description”* will allow for an examiner to locate the message of the day in captured traffic. The message could include any information deemed relevant by the server operator, which could include conditions of use that may support an investigation concerned with misuse of the gaming platform by a client.

TCP packets conveying information about actual gameplay are not in a readable form as they appear to be encoded. At times of player idling patterns can be seen in the TCP stream, which include a large number of full stops and number 6's. An example of concatenated TCP packets sent from the Windows client to the Linux server during idling is:

```
.........&...........&...........&...........&...........&...........&...........&...........&...........&...........&...........&...
........&...........&...........&...........&...........&...........&...........&...........&......./...&...........&...........&....
.......6..`..&...........&...........&...........&...........6..f..&...........&...........&...........&...........&...........'....[.
...B....6...A..'...........        .<.........6......'....4.j.N.....6......
```

There is no evidence of the use of encryption for any of the game's identifiers, user, communications or operations. This is all information that could assist investigators in evidencing malicious communications and identifying users; the capturing of such traffic may be a tactical consideration for live, reactive investigations.

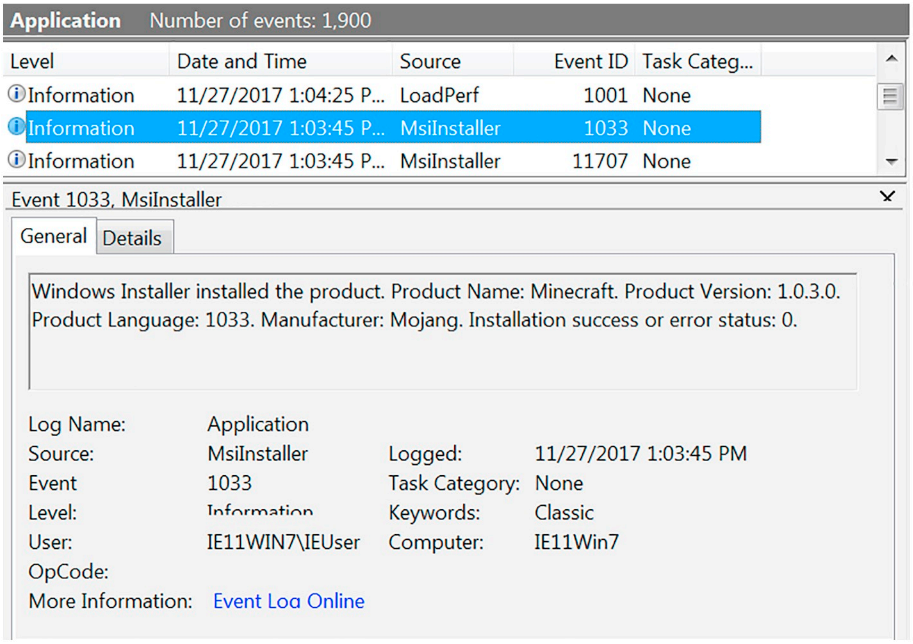
8. Conclusion

The research was able to determine that communications, player identifiers and IP addresses were all available on both the Linux server and the Windows client. There are several scenarios that could present themselves and allow for this research to apply in a live investigation;

- The Minecraft server is identified as being administered by an offender; evidence will exist on the machine of communication to connected users, which could include victims and associates, and identifying usernames and IP addresses that will allow for potential tracing and safeguarding of victims.
- The Minecraft server is a repository of evidence; any offending clients will have their username, IP address, port number and communications stored on the server to be made available to law enforcement.
- The Windows Minecraft client is operated by a victim of some crime; communications sent and received with others will be available, albeit it could be difficult to determine how much of the communication was unaltered from its original transmission. As will identifying information about other users and importantly the server operator, who will be able to provide more extensive data regarding the offending connected user.

One of the configurable *server.properties* settings is the port number. As Minecraft is a popular platform with many opportunities to create custom gaming opportunities, server hosting is popular in the community of gamers. Consideration should be given to server hosting from a home address, whereby port forwarded must be configured on a typical home broadband internet access point. Therefore, evidence of Minecraft server hosting could be located within the home broadband access point settings.

The memory dump of the Windows client provided the IP address



Application Number of events: 1,900

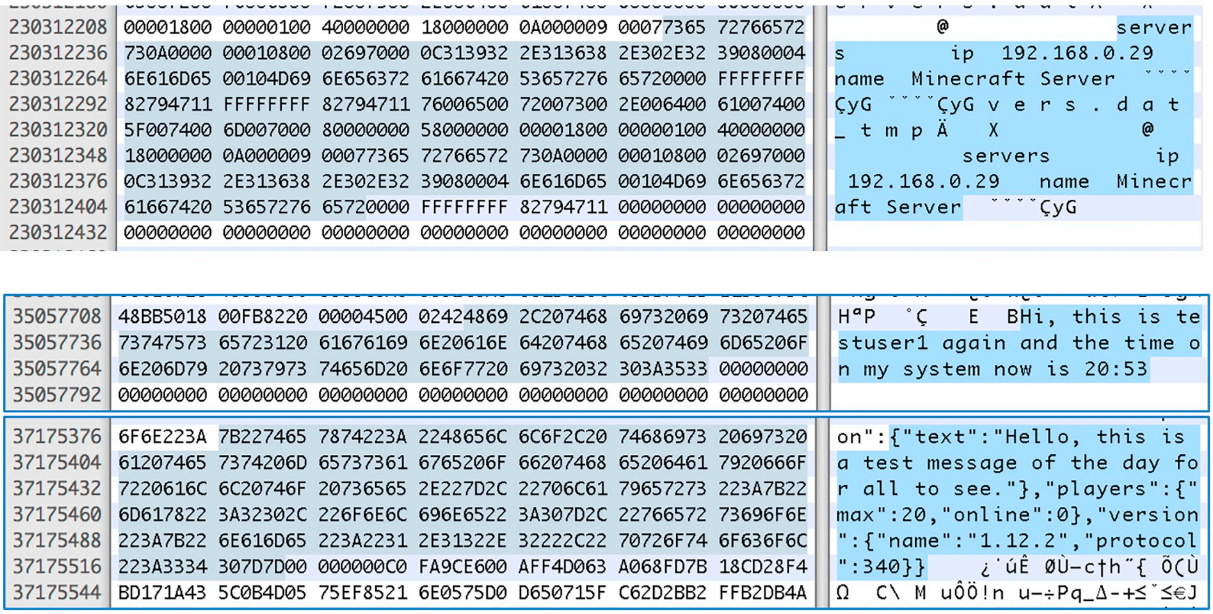
Level	Date and Time	Source	Event ID	Task Categ...
Information	11/27/2017 1:04:25 P...	LoadPerf	1001	None
Information	11/27/2017 1:03:45 P...	MsiInstaller	1033	None
Information	11/27/2017 1:03:45 P...	MsiInstaller	11707	None

Event 1033, MsiInstaller

Windows Installer installed the product. Product Name: Minecraft. Product Version: 1.0.3.0. Product Language: 1033. Manufacturer: Mojang. Installation success or error status: 0.

Log Name:	Application		
Source:	MsiInstaller	Logged:	11/27/2017 1:03:45 PM
Event	1033	Task Category:	None
Level:	Information	Keywords:	Classic
User:	IE11WIN7\IEUser	Computer:	IE11Win7
OpCode:			
More Information:	Event Log Online		

Fig. 10. Windows event log entry for Minecraft installation.

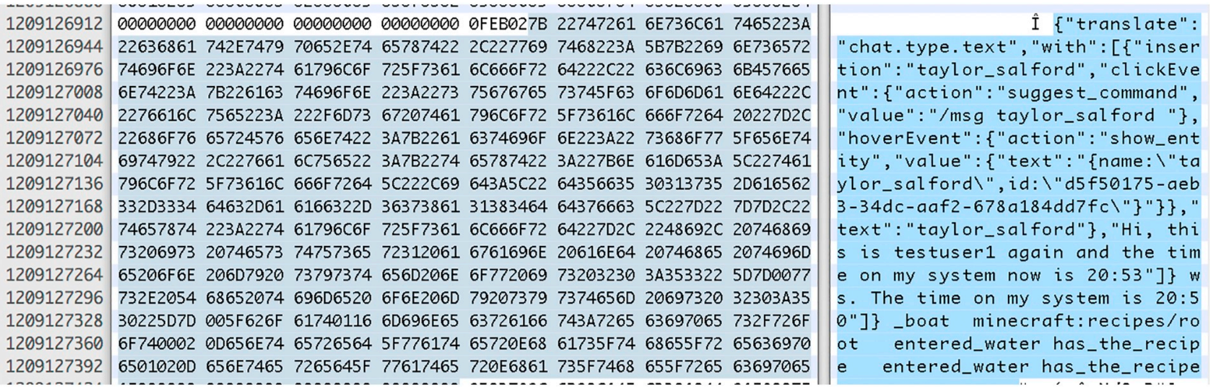


```
230312208  00001800 00000100 40000000 18000000 0A000009 00077365 72766572         @          server
230312236  730A0000 00010800 02697000 0C313932 2E313638 2E302E32 39080004  s        ip  192.168.0.29
230312264  6E616D65 00104D69 6E656372 61667420 53657276 65720000 FFFFFFFF  name  Minecraft Server  ˘˘˘˘
230312292  82794711 FFFFFFFF 82794711 76006500 72007300 2E006400 61007400  ÇyG ˘˘˘˘ÇyG v e r s . d a t
230312320  5F007400 6D007000 80000000 58000000 00001800 00000100 40000000  _ t m p Ä   X            @
230312348  18000000 0A000009 00077365 72766572 730A0000 00010800 02697000           servers         ip
230312376  0C313932 2E313638 2E302E32 39080004 6E616D65 00104D69 6E656372   192.168.0.29   name  Minecr
230312404  61667420 53657276 65720000 FFFFFFFF 82794711 00000000 00000000  aft Server   ˘˘˘˘ÇyG
230312432  00000000 00000000 00000000 00000000 00000000 00000000 00000000
```

```
35057708  48BB5018 00FB8220 00004500 02424869 2C207468 69732069 73207465  HªP  ˚Ç   E  BHi, this is te
35057736  73747573 65723120 61676169 6E20616E 64207468 65207469 6D65206F  stuser1 again and the time o
35057764  6E206D79 20737973 74656D20 6E6F7720 69732032 303A3533 00000000  n my system now is 20:53
35057792  00000000 00000000 00000000 00000000 00000000 00000000 00000000
```

```
37175376  6F6E223A 7B227465 7874223A 2248656C 6C6F2C20 74686973 20697320  on":{"text":"Hello, this is
37175404  61207465 7374206D 65737361 6765206F 66207468 65206461 7920666F  a test message of the day fo
37175432  7220616C 6C20746F 20736565 2E227D2C 22706C61 79657273 223A7B22  r all to see."},"players":{"
37175460  6D617822 3A32302C 226F6E6C 696E6522 3A307D2C 22766572 73696F6E  max":20,"online":0},"version
37175488  223A7B22 6E616D65 223A2231 2E31322E 32222C22 70726F74 6F636F6C  ":{"name":"1.12.2","protocol
37175516  223A3334 307D7D00 000000C0 FA9CE600 AFF4D063 A068FD7B 18CD28F4  ":340}}    ¿˙úÊ ØÙ–c†h˝{ Õ(Ù
37175544  BD171A43 5C0B4D05 75EF8521 6E0575D0 D650715F C62D2BB2 FFB2DB4A  Ω  C\ M uÔÖ!n u–÷Pq_∆-+≤˘≤€J
```

Fig. 11. Data stored in memory of MinecraftLaun process.



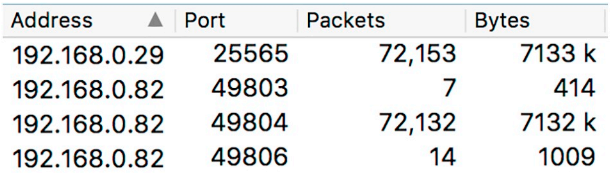
```
1209126912  00000000 00000000 00000000 00000000 0FEB027B 22747261 6E736C61 7465223A             Î {"translate":
1209126944  22636861 742E7479 70652E74 65787422 2C227769 7468223A 5B7B2269 6E736572  "chat.type.text","with":[{"inser
1209126976  74696F6E 223A2274 61796C6F 725F7361 6C666F72 64222C22 636C6963 6B457665  tion":"taylor_salford","clickEve
1209127008  6E74223A 7B226163 74696F6E 223A2273 75676765 73745F63 6F6D6D61 6E64222C  nt":{"action":"suggest_command",
1209127040  2276616C 7565223A 222F6D73 67207461 796C6F72 5F73616C 666F7264 20227D2C  "value":"/msg taylor_salford "},
1209127072  22686F76 65724576 656E7422 3A7B2261 6374696F 6E223A22 73686F77 5F656E74  "hoverEvent":{"action":"show_ent
1209127104  69747922 2C227661 6C756522 3A7B2274 65787422 3A227B6E 616D653A 5C227461  ity","value":{"text":"{name:\"ta
1209127136  796C6F72 5F73616C 666F7264 5C222C69 643A5C22 64356635 30313735 2D616562  ylor_salford\",id:\"d5f50175-aeb
1209127168  332D3334 64632D61 6166322D 36373861 31383464 64376663 5C227D22 7D7D2C22  3-34dc-aaf2-678a184dd7fc\"}"}},"
1209127200  74657874 223A2274 61796C6F 725F7361 6C666F72 64227D2C 2248692C 20746869  text":"taylor_salford"},"Hi, thi
1209127232  73206973 20746573 74757365 72312061 6761696E 20616E64 20746865 2074696D  s is testuser1 again and the tim
1209127264  65206F6E 206D7920 73797374 656D206E 6F772069 73203230 3A353322 5D7D0077  e on my system now is 20:53"]} w
1209127296  732E2054 68652074 696D6520 6F6E206D 79207379 7374656D 20697320 32303A35  s. The time on my system is 20:5
1209127328  30225D7D 005F626F 61740116 6D696E65 63726166 743A7265 63697065 732F726F  0"]} _boat  minecraft:recipes/ro
1209127360  6F740002 0D656E74 65726564 5F776174 65720E68 61735F74 68655F72 65636970  ot   entered_water has_the_recip
1209127392  6501020D 656E7465 7265645F 77617465 720E6861 735F7468 655F7265 63697065  e   entered_water has_the_recipe
```

Fig. 12. Remnants of chat from Windows client stored in Linux memory.

and name of the Linux server.

The message of the day is preserved in plain text in the memory of the Windows client and in several areas of memory in the Linux server. The entire server properties configuration settings are not only found within the */home/[user]* folder but also in their entirety along with timestamp in memory.

The Linux server assigns a unique user identification number to each user, provides the user with a lease of this number of one month and stores this data in *usercache.json* file within */home/[user]*. The data also resides in Linux memory after the game has ceased and the file on disk has been erased. When investigations concern allegations of impersonation of a Minecraft gamer, consideration should be given to the importance of such unique identification numbers and it should be noted that the expiry time will always be one calendar month ahead of the date the user connected to the server.



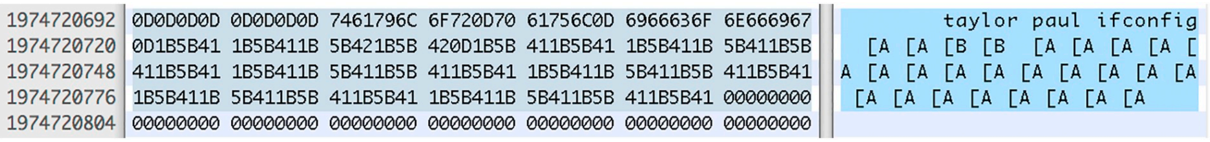
Address ▲	Port	Packets	Bytes
192.168.0.29	25565	72,153	7133 k
192.168.0.82	49803	7	414
192.168.0.82	49804	72,132	7132 k
192.168.0.82	49806	14	1009

Fig. 14. Ports utilised by server and client.

The password to login to the Linux server and begin typing the commands necessary to run the server software was available in plaintext in memory alongside the user name. Even if the username for the server was not known it could easily be identified by conducting a search for "/home/" which quickly identified the server user account.

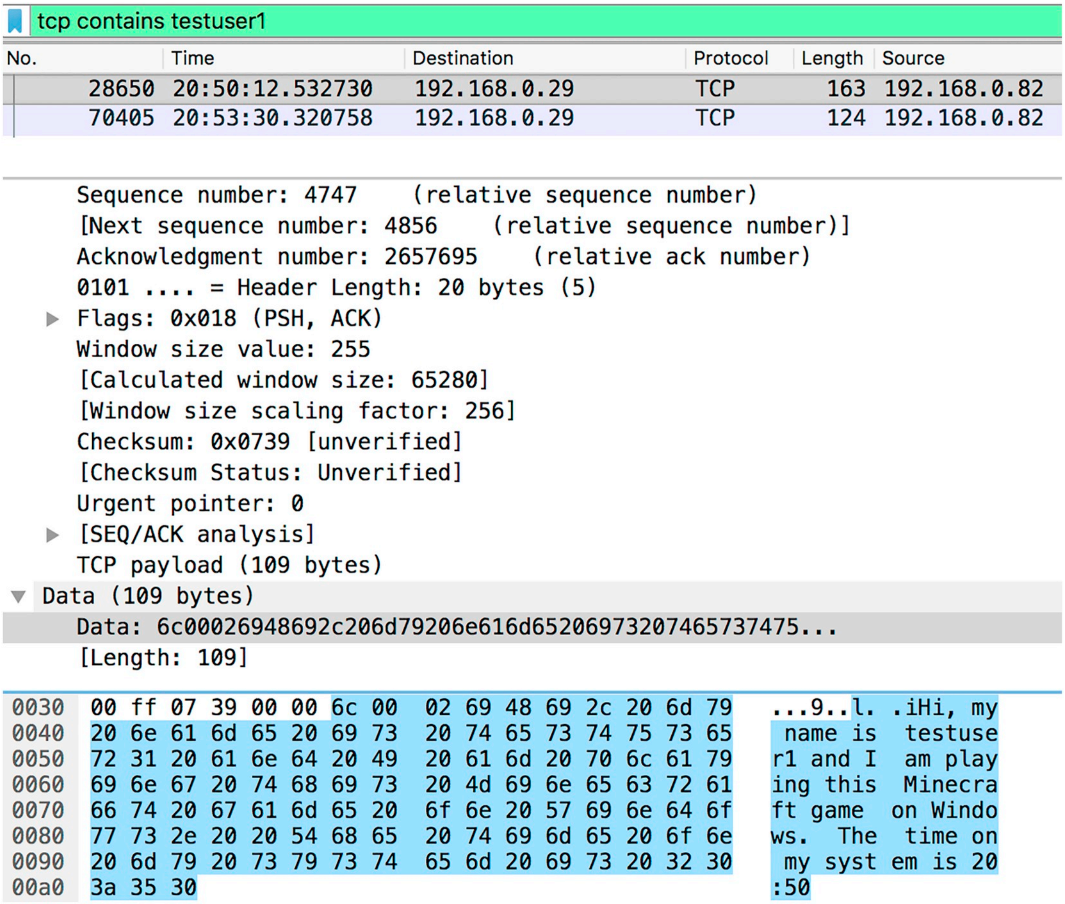
By acquiring the above results, the objectives of this research were met as follows:

1. Artefacts of evidential interest were retained on the Linux Minecraft server after installation of the software and activity during gameplay



```
1974720692  0D0D0D0D 0D0D0D0D 7461796C 6F720D70 61756C0D 6966636F 6E666967          taylor paul ifconfig
1974720720  0D1B5B41 1B5B411B 5B421B5B 420D1B5B 411B5B41 1B5B411B 5B411B5B    [A [A [B [B   [A [A [A [A [
1974720748  411B5B41 1B5B411B 5B411B5B 411B5B41 1B5B411B 5B411B5B 411B5B41  A [A [A [A [A [A [A [A [A [A
1974720776  1B5B411B 5B411B5B 411B5B41 1B5B411B 5B411B5B 411B5B41 00000000   [A [A [A [A [A [A [A [A
1974720804  00000000 00000000 00000000 00000000 00000000 00000000 00000000
```

Fig. 13. Server username and password.



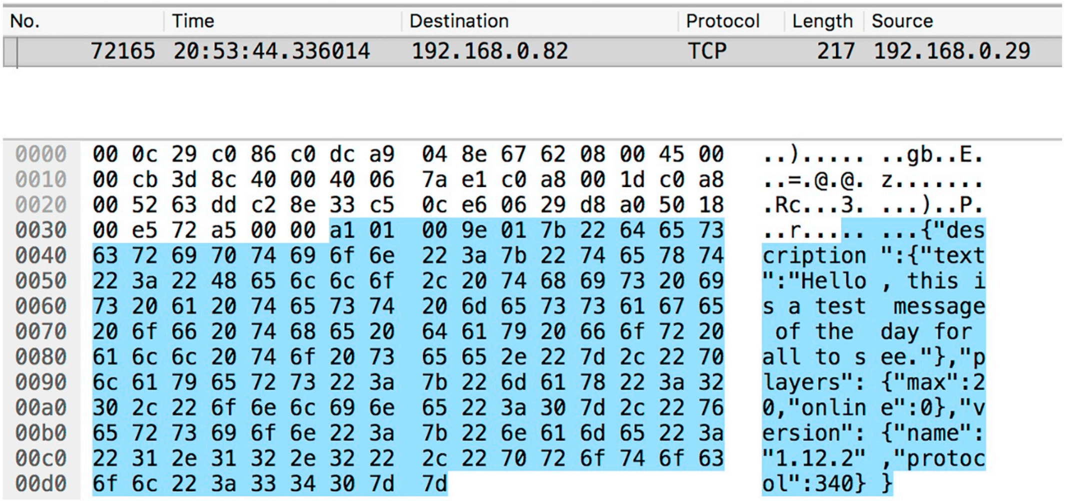
```
tcp contains testuser1

No.      Time              Destination     Protocol  Length  Source
  28650  20:50:12.532730   192.168.0.29    TCP          163  192.168.0.82
  70405  20:53:30.320758   192.168.0.29    TCP          124  192.168.0.82

    Sequence number: 4747    (relative sequence number)
    [Next sequence number: 4856    (relative sequence number)]
    Acknowledgment number: 2657695    (relative ack number)
    0101 .... = Header Length: 20 bytes (5)
  ▶ Flags: 0x018 (PSH, ACK)
    Window size value: 255
    [Calculated window size: 65280]
    [Window size scaling factor: 256]
    Checksum: 0x0739 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  ▶ [SEQ/ACK analysis]
    TCP payload (109 bytes)
▼ Data (109 bytes)
    Data: 6c00026948692c206d79206e616d6520697320746573747475...
    [Length: 109]

0030  00 ff 07 39 00 00 6c 00  02 69 48 69 2c 20 6d 79   ...9..l. .iHi, my
0040  20 6e 61 6d 65 20 69 73  20 74 65 73 74 75 73 65    name is  testuse
0050  72 31 20 61 6e 64 20 49  20 61 6d 20 70 6c 61 79   r1 and I  am play
0060  69 6e 67 20 74 68 69 73  20 4d 69 6e 65 63 72 61   ing this  Minecra
0070  66 74 20 67 61 6d 65 20  6f 6e 20 57 69 6e 64 6f   ft game  on Windo
0080  77 73 2e 20 20 54 68 65  20 74 69 6d 65 20 6f 6e   ws.  The  time on
0090  20 6d 79 20 73 79 73 74  65 6d 20 69 73 20 32 30    my syst em is 20
00a0  3a 35 30                                           :50
```

Fig. 15. Plaintext communication sent from Windows client.



```
No.      Time              Destination     Protocol  Length  Source
  72165  20:53:44.336014   192.168.0.82    TCP          217  192.168.0.29

0000  00 0c 29 c0 86 c0 dc a9  04 8e 67 62 08 00 45 00   ..)..... ..gb..E.
0010  00 cb 3d 8c 40 00 40 06  7a e1 c0 a8 00 1d c0 a8   ..=.@.@. z.......
0020  00 52 63 dd c2 8e 33 c5  0c e6 06 29 d8 a0 50 18   .Rc...3. ...)..P.
0030  00 e5 72 a5 00 00 a1 01  00 9e 01 7b 22 64 65 73   ..r..... ...{"des
0040  63 72 69 70 74 69 6f 6e  22 3a 7b 22 74 65 78 74   cription ":{"text
0050  22 3a 22 48 65 6c 6c 6f  2c 20 74 68 69 73 20 69   ":"Hello , this i
0060  73 20 61 20 74 65 73 74  20 6d 65 73 73 61 67 65   s a test  message
0070  20 6f 66 20 74 68 65 20  64 61 79 20 66 6f 72 20    of the  day for 
0080  61 6c 6c 20 74 6f 20 73  65 65 2e 22 7d 2c 22 70   all to s ee."},"p
0090  6c 61 79 65 72 73 22 3a  7b 22 6d 61 78 22 3a 32   layers": {"max":2
00a0  30 2c 22 6f 6e 6c 69 6e  65 22 3a 30 7d 2c 22 76   0,"onlin e":0},"v
00b0  65 72 73 69 6f 6e 22 3a  7b 22 6e 61 6d 65 22 3a   ersion": {"name":
00c0  22 31 2e 31 32 2e 32 22  2c 22 70 72 6f 74 6f 63   "1.12.2" ,"protoc
00d0  6f 6c 22 3a 33 34 30 7d  7d                        ol":340} }
```

Fig. 16. Plaintext message of the day.

with a Windows client. Client IP addresses, chat logs, player data and server credentials were all available in unencrypted form.

2. Evidence of communication between client devices and the server was identified on all connected devices; communications over Minecraft's chat feature were stored with timestamps on disk and in live memory.
3. In order to assist in identifying and profiling users, IP addresses were available along with Unique User Identification numbers that remained the same for one month and the Windows client stored the e-mail address associated to the registered Minecraft user.
4. The network traffic revealed communications, the 'message of the day' IP addresses, port numbers, user names and operational commands, albeit in a seemingly encoded format.

Consideration for future work must be given to mobile platforms, with appreciation for the fact that unrooted devices can only make use of the official Minecraft client app and can only connect to verified Microsoft servers. There are third party applications available, such as PocketMine [44] that allow users of mobile devices to run a modified and unofficial version of Minecraft in order to connect to multiplayer servers that do not have to be subjected to verification by Microsoft. Research extended into these areas would enlighten investigators and give an appreciation for the popularity of unofficial and customisable Minecraft server and client platforms.

In addition, it would be beneficial to perform more tests and increase the volume of communication between server and client devices in order to establish the longevity of recorded chat messages.

References

[1] K.R. Choo, Online child grooming: a literature review on the misuse of social networking sites for grooming children for sexual offences, Aust. Inst. Criminol. (2009) 132.

[2] L.-K. Bernstein, Investigating and Prosecuting Swatting Crimes, United States Atty. Bull. 64 (3) (2016) 51–56.

[3] A. Carpinteri, B. Bang, K. Klimley, R.A. Black, V.B. Van Hasselt, Commercial sexual exploitation of children: an assessment of offender characteristics, J. Police Crim. Psychol. 33 (2) (Jun 2018) 150, https://doi.org/10.1007/s11896-017-9242-0.

[4] Minecraft to join Microsoft - News Center, [Online]. Available https://news.microsoft.com/2014/09/15/minecraft-to-join-microsoft [Accessed: 20-Oct-2017].

[5] Minecraft has 55 Million Monthly Players, 122 Million Sales - IGN, [Online]. Available http://uk.ign.com/articles/2017/02/27/minecraft-has-55-million-monthly-players-122-million-sales [Accessed: 20-Oct-2017].

[6] The Better Together Update is here!, Minecraft, [Online]. Available https://minecraft.net/en-us/article/better-together-update-here [Accessed: 23-Oct-2017].

[7] Minecraft: A parent's guide, NSPCC, [Online]. Available https://www.nspcc.org.uk/preventing-abuse/keeping-children-safe/online-safety/minecraft-a-parents-guide [Accessed: 24-Oct-2017].

[8] H. Hillman, C. Hooper, K.-K.R. Choo, Online child exploitation: challenges and future research directions, Comput. Law Secur. Rev. 30 (6) (2014) 687–698 Dec.

[9] L. Achternbosch, C. Miller, C. Turville, P. Vamplew, Griefers versus the Griefed - what motivates them to play massively multiplayer online role-playing games? Comput. Games J. Ltd 3 (1) (2014).

[10] E.M. Jaffe, Swatting: the new cyberbullying frontier after Elonis V. United States, Drake Law Rev. (2016) 455–483.

[11] A. Choo, A. May, Maintaining long distance togetherness synchronous communication with minecraft and skype, Games Innovation Conference (IGIC), 2013 IEEE International, 2013.

[12] A. Noroozian, M. Korczyński, C.H. Gañan, D. Makita, K. Yoshioka, M. van Eeten, Who gets the boot? Analyzing victimization by DDoS-as-a-service, in: F. Monrose, M. Dacier, G. Blanc, J. Garcia-Alfaro (Eds.), Research in Attacks, Intrusions, and Defenses: 19th International Symposium, RAID 2016, Paris, France, September 19–21, 2016, Proceedings, Springer International Publishing, Cham, 2016, pp. 368–389.

[13] Welsh gamer jailed for grooming two boys on Minecraft | UK news The Guardian, [Online]. Available https://www.theguardian.com/uk-news/2017/jan/20/welsh-gamer-jailed-for-grooming-two-boys-on-minecraft (Accessed: 24-Oct-2017).

[14] J. Taylor, Online investigations: protection for child victims by raising awareness, ERA Forum 16 (3) (2015) 349–358.

[15] Mojang, Minecraft Realms, [Online]. Available: https://help.mojang.com/customer/en/portal/articles/1018151-minecraft-realms (Accessed: 24-Oct-2017).

[16] Z. Zhang, Github - zhuowei/Boardwalk, [Online]. Available https://github.com/zhuowei/Boardwalk (Accessed: 20-Dec-2017).

[17] S. Khanji, R. Jabir, F. Iqbal, A. Marrington, Forensic analysis of xbox one and playstation 4 gaming consoles, 8th IEEE Int. Work. Inf. Forensics Secur. 2017 WIFS, 2016.

[18] M. Cheah, L. Wyndham-Birch, B. Bird, What Artifacts of Evidentiary Value Can Be Found when Investigating Multi-User Virtual Environments, (2015).

[19] Minecraft server, Microsoft Azure Marketplace, [Online]. Available https://azuremarketplace.microsoft.com/en-us/marketplace/apps/msftstack.minecraft-server?tab=Overview (Accessed: 24-Oct-2017).

[20] What OS for server? - Server Administration - Server Support - Support - Minecraft Forum - Minecraft Forum, [Online]. Available http://www.minecraftforum.net/forums/support/server-support/server-administration/2801103-what-os-for-server (Accessed: 24-Oct-2017).

[21] D. Quick, K.-K.R. Choo, Google Drive: Forensic analysis of data remnants, J. Netw. Comput. Appl. 40 (Apr. 2014) 179–193.

[22] D. Quick, K.-K.R. Choo, Dropbox analysis: Data remnants on user machines, Digit. Investig. 10 (1) (Jun. 2013) 3–18.

[23] L. C. for D. Investigation, "1/21/2016 175"Leahy Center for Digital Investigation, no. 802. 2016.

[24] A. Rutkin, Your place or Minecraft? New Sci. 230 (3071) (2016) 22–23.

[25] E. Lough, E. Flynn, D.M. Riby, Mapping real-world to online vulnerability in young people with developmental disorders: illustrations from Autism and Williams Syndrome, Rev. J. Autism Dev. Disord. 2 (1) (2015) 1–7 Mar.

[26] "Mojang - Minecon 2015 - Day Two - Twitch.", [Online]. Available https://www.twitch.tv/videos/6949826 (Accessed: 05-Jan-2018).

[27] Z. Zhang, H. Anada, J. Kawamoto, K. Sakurai, Detection of illegal players in massively multiplayer online role playing game by classification algorithms, Proc. - Int. Conf. Adv. Inf. Netw. Appl, vol. 2015, AINA, 2015, pp. 406–413. April.

[28] Y. Ki, J. Woo, H.K. Kim, Identifying Spreaders of malicious behaviors in online games, Proceedings of the 23rd International Conference on World Wide Web, 2014, pp. 315–316.

[29] J. Oh, Z.H. Borbora, J. Srivastava, Automatic detection of compromised accounts in MMORPGs, 2012 International Conference on Social Informatics, 2012, pp. 222–227.

[30] A.S.V. Nair, B.A.S. Ajeena, A log based strategy for fingerprinting and forensic investigation of online cyber crimes, Proceedings of the 2014 International Conference on Interdisciplinary Advances in Applied Computing, 2014, pp. 7:1–7:5.

[31] M. Barni, B. Tondi, Threat models and games for adversarial multimedia forensics, Proceedings of the 2nd International Workshop on Multimedia Forensics and Security, 2017, pp. 11–15.

[32] S. Rajendran, N.P. Gopalan, Mobile forensic investigation (MFI) life cycle process for digital data discovery (DDD), in: L.P. Suresh, B.K. Panigrahi (Eds.), Proceedings of the International Conference on Soft Computing Systems: ICSCS 2015, vol. 2, Springer India, New Delhi, 2016, pp. 393–403.

[33] T. Dargahi, A. Dehghantanha, M. Conti, Chapter 2 - Forensics Analysis of Android Mobile VoIP Apps, in: K.-K.R. Choo, A. Dehghantanha (Eds.), Contemporary Digital Forensic Investigations of Cloud and Mobile Applications, Syngress, 2017, pp. 7–20.

[34] K.K.R. Choo, A. Dehghantanha, Contemporary Forensic Investigation of Cloud and Mobile Applications, (2017).

[35] ACPO, ACPO Good Practice Guide for Digital Evidence, Association of Chief Police Officers, 2012.

[36] N. C. J. U.S. Department of Justice, Electronic crime scene investigation: a guide for first responders, NIJ Res. Rep. (2001) 96 no. NCJ 187736.

[37] F.S. Regulator, Codes of Practice and Conduct Issue 4, (2017).

[38] K. Kent, S. Chevalier, T. Grance, H. Dang, Guide to Integrating Forensic Techniques into Incident Response, (2006).

[39] A. Antwi-Boasiako, H. Venter, A model for digital evidence admissibility assessment, in: G. Peterson, S. Shenoi (Eds.), Advances in Digital Forensics XIII: 13th IFIP WG 11.9 International Conference, Orlando, FL, USA, January 30–February 1, 2017, Revised Selected Papers, Springer International Publishing, Cham, 2017, pp. 23–38.

[40] B. Martini, K.K.R. Choo, An integrated conceptual digital forensic framework for cloud computing, Digit. Investig. 9 (2) (2012) 71–80.

[41] Y. Teing, A. Dehghantanha, K.R. Choo, Forensic Investigation of Cooperative Storage Cloud Service: Symform as a Case Study, no. May (2016).

[42] T. Alstad, et al., Minecraft computer game performance analysis and network traffic emulation by a custom bot, Proc. 2015 Sci. Inf. Conf. SAI, 2015, pp. 227–236.

[43] Y. Gubanov, Retrieving Digital Evidence: Methods, Techniques and Issues, (2012).

[44] Get PocketMine-MP, [Online]. Available http://www.pocketmine.net (Accessed: 04-Jan-2018).

[45] NJ Police Officer Among 24 Arrested on Charges of Luring Underage Children for Sex, [Online]. Available https://eu.northjersey.com/story/news/new-jersey/2018/09/18/nj-police-officer-among-24-arrested-child-luring-sting/1346054002 (Accessed: 04-Oct-2018).

[46] Close to 50,000 Minecraft Accounts Infected with Malware Designed to Reformat Hard-Drives and More, Alexej Savčin, 17 April 2018." [Online]. Available: https://blog.avast.com/minecraft-players-exposed-to-malicious-code-in-modified-skins (Accessed 27-Dec-2018).