electronics

MDPI

Article

Forensic Analysis of File Exfiltrations Using AnyDesk, TeamViewer and Chrome Remote Desktop

Xabiel G. Pañeda, David Melendi *, Víctor Corcoba, Alejandro G. Pañeda, Roberto García and Dan García

Department for Informatics, University of Oviedo, 33203 Xixón, Spain; xabiel@uniovi.es (X.G.P.); corcobavictor@uniovi.es (V.C.); uo197450@uniovi.es (A.G.P.); garciaroberto@uniovi.es (R.G.); garciadan@uniovi.es (D.G.)

* Correspondence: melendi@uniovi.es

Abstract: The use of remote desktop applications has increased greatly in recent years, mainly because of the generalization of telecommuting due to the COVID-19 pandemic. This process has been carried out in a very controlled manner in some companies, but in other organizations it has been introduced in a more anarchic way. The direct use of on-premises company computers and resources from the internet without the necessary protection mechanisms, including VPNs, has increased the risk of data exfiltration. Apart from other types of data exfiltration, there are cases in which employees transfer files using encrypted communications, consciously or unconsciously, producing a leak of information undetected by data loss prevention systems. In this paper we analyse the question of whether a forensic investigation may answer questions about data exfiltrations; questions such as those regarding the when, what and who (or to whom) and the use of application logs and other available tools. The answers to these questions may form the basis of solid digital evidence for legal purposes, though they may only deliver a partial response to said questions. Other complementary sources are necessary to build a complete answer and accurate digital evidence. Nevertheless, we have identified and analysed several use cases that may help to raise an early alarm that can offer warning about certain behaviours in encrypted traffic that may be detected via network monitoring.

Keywords: remote desktop; TeamViewer; AnyDesk; Chrome RD; file exfiltration; evidence; forensic analysis

check for updates

Citation: G. Pañeda, X.; Melendi, D.; Corcoba, V.; G. Pañeda, A.; García, R.; García, D. Forensic Analysis of File Exfiltrations Using AnyDesk, TeamViewer and Chrome Remote Desktop. *Electronics* **2024**, *13*, 1429. https://doi.org/10.3390/electronics13081429

Academic Editor: Krzysztof Szczypiorski

Received: 11 March 2024
Revised: 2 April 2024
Accepted: 8 April 2024
Published: 10 April 2024

1. Introduction

In recent years, many company offices have undergone a rapid process of emptying. Telecommuting had already become frequent in certain sectors, but the COVID-19 pandemic caused a tremendous acceleration in the adoption of remote working. Telecommuting has brought many advantages, but it has also created some problems. In fact, it has caused a major paradigm shift in information security [1]. An increasing number of employees of all types of organizations perform tasks outside the office, often using their own devices. Thus, in cases in which its adoption has been carried out in a disorderly fashion, without clear regulation, restrictions or the deployment of access control and monitoring systems, telecommuting has become a major cybersecurity problem. A lack of control can make it easier for outside agents to break into corporate systems and for disloyal employees to exfiltrate sensitive information. While in the world of large companies the adoption of telecommuting has occurred in a very controlled way, in small and medium-sized enterprises it has been carried out in a substantially disordered manner. Previous work has described problems such as the poor competencies required to efficiently use IT solutions, a shortage of equipment, communication and coordination problems, the lack of methodology for organizing distance work, etc. [2,3].

Three applications that have become widely popular under telecommuting scenarios are TeamViewer, AnyDesk and Chrome Remote Desktop. Their adoption is a result of their ease of use and flexibility. Most internet access services combine both the usage of network address translation (NAT) and a dynamic IP address. This kind of configuration makes the remote access of a computer that is behind a similarly configured router problematic. For instance, special settings need to be used in the router and some dynamic DNS service needs to be used to map a fixed domain name to a variable IP address. To perform these tasks, users need to have some technical knowledge about networking and security. Nevertheless, the chosen applications have been designed to adapt to different network configurations and no changes need to be applied in the face of NAT or routing and security restrictions. They can even set up connections that trespass the usual firewalling controls thanks to the use of HTTPS tunnelling. Beyond their ease of use, all of these products have versions with free licenses.

In the event of an incident of information theft, it is paramount to be able to collect data regarding the activity of the employees. Therefore, in this paper we have analysed the forensic evidence produced when there is the exfiltration of a file from a Windows PC using TeamViewer, AnyDesk or Chrome Remote Desktop. We have chosen TeamViewer and AnyDesk because of their predominant position in the market [4], with TeamViewer having a 55.63% market share in remote support solutions and AnyDesk an 8.99% market share, according to recent reports [5]. Additionally, we have chosen Chrome Remote Desktop because it is a free solution available for most of the operating systems in the market and because it was notably popular during the COVID-19 lockdown [6]. The goal of the paper is twofold. Firstly, in order to be able to initiate a legal process with guarantees, the study aims to determine if the evidence produced by these applications and the underlying systems may answer the following questions: what has been exfiltrated, when has the exfiltration happened and who has exfiltrated the information (or to whom it has been carried out). Being able to respond to these questions is paramount for the said legal process. Secondly, this research seeks to identify aspects in the behaviour of these applications which may allow security officers and appliances to act proactively and stop, or even avoid, data exfiltrations performed with these systems.

The rest of the paper is organized as follows. Section 2 introduces the existing related research. Section 3 presents the base scenarios considered and the analysis methodology that was followed. Section 4 describes our study and the main outcomes of the research. Finally, Section 5 provides the conclusions and future work.

2. Related Work

Telecommuting is not a new concept. However, the COVID 19 pandemic has accelerated its adoption in multiple sectors, including those that are not directly linked with technology [7]. It is now common for companies to use software to access computers remotely, share files or hold meetings. Nevertheless, this software may also be used to illegally extract data and information from companies [8].

There are very few forensic research works analysing remote desktop solutions. Hornyák [9] describes how one may obtain traces of attempts at remote access in Windows using the RDP protocol. Manson [10] analyses the artifacts obtained using TeamViewer and Windows. The author states that it is possible to determine if a machine has used a remote desktop service or has been accessed remotely, if files have been transferred or if that computer has been rebooted remotely. However, this analysis was carried out with old software versions and the study is not focused on file exfiltration. Other common remote desktop applications, such as AnyDesk or Chrome Remote Desktop, were not tested. Kerai and Vekariya [11] conducted a study on evidence of the usage of several remote desktop and teleconferencing applications in Windows. The applications analysed were RealVNC, TightVNC, Cisco WebEx, GoToMeeting and LogMeIn. The authors examined the entries in the Windows registry and the specific log files created by these applications. They found that it is possible to obtain the clients IP addresses and information about transferred files. However, these programs have suffered many changes in recent years and the study is currently outdated. Computers may also be accessed remotely using mobile devices such as tablets or smartphones. Kerai [12] analysed whether it is possible to obtain evidence of these services on Windows mobile phones. The author found that many of the applications used in this environment did not encrypt connection passwords. Furthermore, it was possible to obtain details such as domain names, IP addresses, transport ports and log-in and log-out times. However, to the best of our knowledge, there are no similar works on current mobile operating systems such as IOS or Android.

Other studies have focused on analysing whether, using side-channel information such as packet lengths and packet arrival times, it is possible to identify the activity being performed by a user accessing a computer via a remote desktop. Altschaffel et al. [13] proposed a method to identify actions made by the user, based on statistical pattern recognition. In their work they used the TeamViewer application. The authors were able to detect file transfers, video conferencing, audio conferencing, text chats and remote sessions. Although the results are interesting, the program version is very old (from 2013) and has undergone many changes. Similarly, Jiang et al. [14] have analysed the possibility of distinguishing between five common activities using machine learning and side-channel information provided by remote desktop software. They analysed AnyDesk, ConnectWise, MicroRDS, RealVNC, TeamViewer and Zoho Assist. The analysed activities included editing documents, reading documents, web browsing, watching videos and installing software. In this study, it was observed that traffic encryption mechanisms are not sufficient to totally hide the activity of the users and that it is possible to identify what they do with great accuracy.

In addition to remote desktop applications, there are other applications that permit file transfers, even though their main use is for meetings or collaborative work. These applications include products such as Skype, Zoom, Slack and Microsoft Teams. Yang et al. [15] analysed the release of Skype available in the official Windows app store. The researchers found valuable information, such as the names of the files transferred and conversation logs. They also observed that traces of these activities remained after the program was uninstalled. Nicoletti and Bernaschi [16] focused their research on the business version of Skype. They analysed the protocols used by the application, its architecture and the information available in Windows logs and the event viewer.

Microsoft Teams is another software that is widely used by companies and educational institutions for telecommuting and e-learning. Paligu and Varol [17] examine the evidence obtained from the IndexedDB storage of Microsoft Teams. In this research, the authors used time-frame analyses and managed to obtain messages from chats, voice mail and several Teams extensions. Additionally, Khalid et al. [18] performed a forensic analysis of the memory, disk and network traffic produced by Microsoft Teams. The authors were able to extract critical information such as email addresses, exchanged messages, deleted messages and transferred files. Google has a comparable tool named Google Meet. Iqbal et al. [19] carried out an exhaustive memory forensic analysis to determine the evidence that Google Meet leaves after it has been used. The authors perform experiments with different browsers (Chrome, Firefox and Mozilla) and different amounts of RAM memory. In their study, they obtained traces of information about the meetings, such as the email addresses and profiles of the participants.

In the literature, we found few studies that examine the traces that may be obtained from the most common applications used for professional communications and to work remotely. However, many of these studies are outdated, do not provide a detailed explanation of how they have obtained the information presented or are not focused on file exfiltrations. In our study, we have analysed artifacts from application logs, operating systems and network traffic to determine if we can answer questions regarding the when, what and who/to whom.

3. Base Scenarios and Methodology

To perform this study, we have analysed two types of situations: (1) when an employee connects from an on-premises computer to a computer outside the company and moves a file to that external computer (push scenario) and (2) the situation in which a user outside the company connects to an on-premises computer and extracts a file (pull scenario). Figure 1 shows both situations.

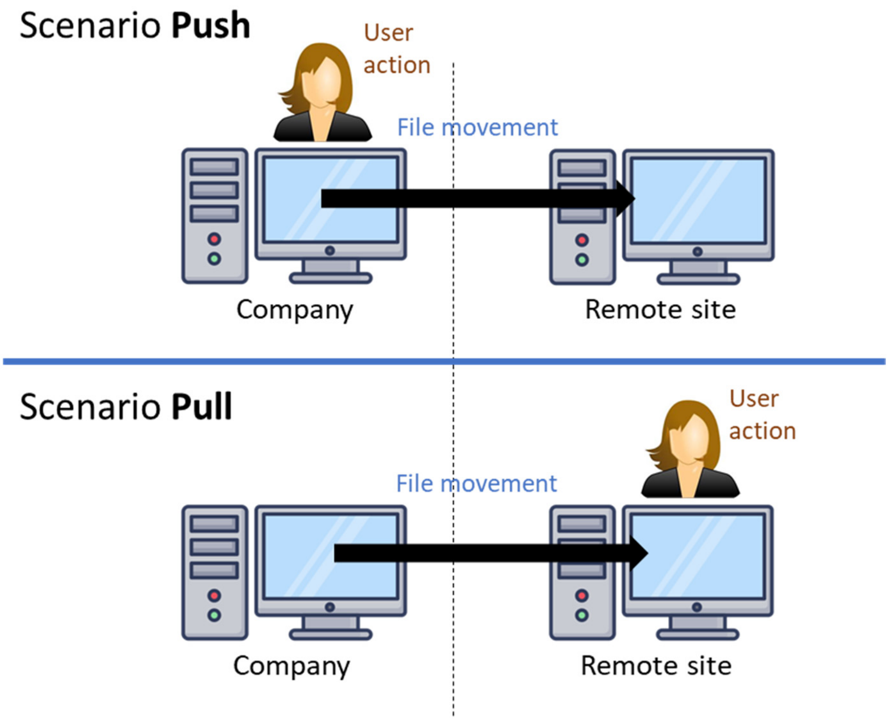


Figure 1. Scenarios considered.

We have created the scenarios using two 64-bit Windows 10 virtual machines. In these machines we have installed AnyDesk version 7.1.6, TeamViewer version 15.31.5 and Chrome Remote Desktop version 2.1. To capture network traffic, we have installed the sniffer WireShark. Additionally, we have created a folder where some image files have been stored. We have activated the Windows Object audit service on the folder to have maximum control over the file actions provided by the Windows operating system. In the virtualization environment, each machine has been connected to a different virtual network. Furthermore, a router allows the communication between both machines and the communication of those machines with the internet by using a static NAT configuration.

To perform the experiments, we have followed a simple methodology. We activated the monitoring tools, performed several exfiltration processes and revised the evidence generated by the different monitoring systems and tools. The exfiltration consists in the transference of three pictures of different sizes to a remote computer: “lapiquera.jpg” (92 KB), “lateyera.jpg” (4.5 MB) and “xixun.jpg” (5 MB). Additionally, to better analyse the traffic generated by the applications, we also performed experiments with large files of sizes greater than 180 MB. Finally, we compared the results with a baseline in which no file transfers were produced.

4. Analyses Performed

4.1. TeamViewer

Distributed and implemented by TeamViewer AG, TeamViewer is an application that provides remote functionalities to computers running a host application. Once a connection has been established, the application provides remote control, file transfer and software deployment capabilities, among others. It is easy to use and does not require end users to perform complex tasks such as modifying firewall rules or configure port forwarding on routers. TeamViewer uses primarily TCP and UDP transmissions over port

5938. Alternatively, it can use ports 443 or 80. It secures data transferences using encryption, based on 4096-bit RSA key exchanges and 256-bit AES session encoding.

TeamViewer offers two different possibilities: to install the software in the computer or to use a portable version. These two possibilities produce different log files in different paths, with interesting information about the activity of the users. The files are either in the "Program Files\TeamViewer" or in the "<user>\AppData\Roaming\TeamViewer" folders. In the "TeamViewer15Logfile.log" file (in "Program Files\TeamViewer") TeamViewer registers details about file transferences. However, this file is not generated by the portable version of the application. Instead, we can find similar information in a file with the same name, located in the "<user>\AppData\Roaming\TeamViewer" folder. In both files we can observe interesting events. We have details about the establishment of the connection with the remote site. Additionally, we can observe the IDs of both the local and remote devices and the timestamp (local time) of the connection. These IDs cannot be directly mapped to a specific IP address. A sample of the records registered in these files follows:

```
2023/01/09 12:18:01.584672 5040 G2               Client connection to
1308133946, server version is 15.37.3, OS = 19 Participant =
[1308133946, 1296448264]
```

File transfers are also detailed in the log file in the on-premises computer. When the connection corresponds to a push scenario, apart from the name of the transferred file, we also have access to the origin and destination paths and to the volume of bytes transmitted (the size of the file), as shown in the following record:

```
Upload from ''C:\fileorigin\lapiquera.jpg'' to
''C:\Users\smiot\Documents\lapiquera.jpg'' (92.36 kB)
```

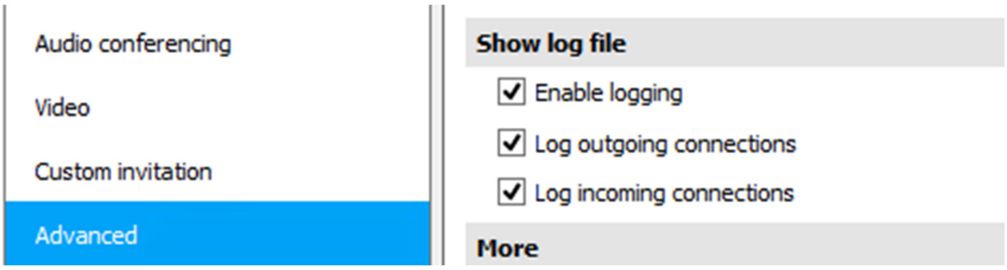
On the other hand, when the connection corresponds to a pull scenario, TeamViewer registers the file transfer with the following line:

```
Send file C:\fileorigin\lapiquera.jpg
```

Moreover, in this scenario the application produces an entry in the "connections_incoming.txt" log file, with the ID of the remote PC and the connection timestamp (GMT). An example of one of these entries follows:

```
1308133946      DESKTOP-HQDP856 09-01-2023 12:43:42 09-01-2023 12:45:31
smiot    RemoteControl (e9bef4d9-8fcf-4c78-a956-fd29e8c0856a)
```

However, TeamViewer allows users to configure what is registered in log files, as shown in Figure 2 (in the main menu under "Options → Advanced"). Thus, all of the details may be reduced depending on the configuration available for each user.



Show log file

☑ Enable logging

☑ Log outgoing connections

☑ Log incoming connections

Figure 2. Log options in TeamViewer.

The Windows operating system is also able to register relevant information about the usage of the application. With the default audit, the firewall produces several events (types 5154, 5156 and 5158) that report permitted connections with remote devices. The object audit may complement this information with specific event details (types 4656, 4658 and 4663). This set of events will provide the name of the exfiltrated files.

Network monitoring can also provide interesting information. For instance, although Jiang et al. [14] do not analyse file transfers performed with TeamViewer, the authors have already stated that the traffic encryption mechanisms used by this application cannot completely hide the activity of the users. Remote sessions are usually established with end-to-end full-duplex connections using a proprietary protocol implemented over UDP and variable port numbers. These end-to-end connections are encrypted using 256-bit AES encoding. We have reverse engineered this protocol and have detected that there are different types of messages, each of them with a different type code sent in the 13th byte of the payload of UDP datagrams. Messages containing data have hexadecimal type codes 0x70 and 0x6b. Additionally, they carry a 4-byte sequence number in little endian format in bytes 1 to 4 of the UDP payload, starting in 1. For each direction of the communication, the protocol uses a sliding window controlled with frequent acknowledgement messages identified by type code 0x6f. These acknowledgements carry the number of the data message acknowledged in bytes 5 to 8 of the UDP payload. The frequency of the acknowledgments and the number of acknowledged messages depends on the volume of information delivered in the opposite direction, as shown in Figure 3.

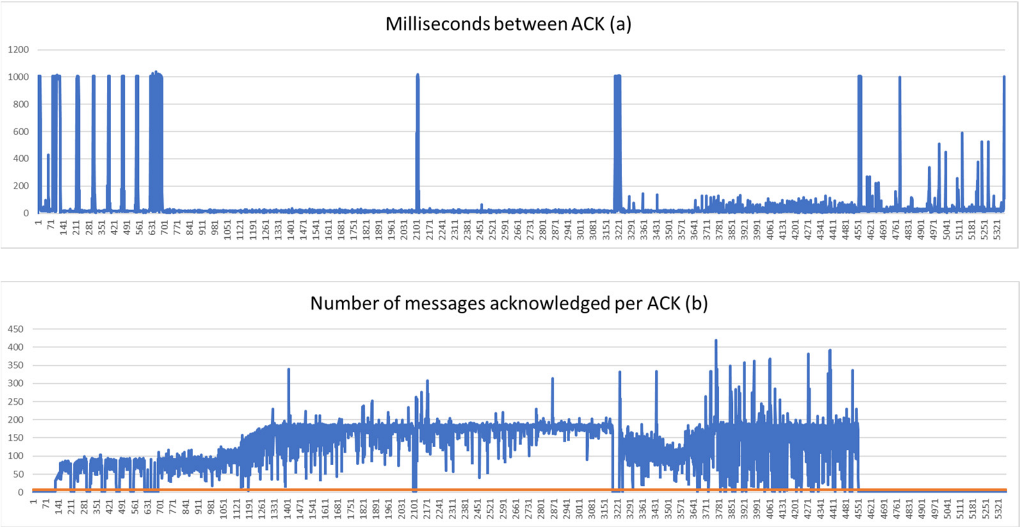


Figure 3. Acknowledgements sent from the remote computer in a push scenario. Milliseconds between acknowledgements (**a**) and number of messages acknowledged in each acknowledgement message (**b**).

Figure 3a shows the frequency of the acknowledgements sent by the remote computer in a push scenario. Low values correspond with acknowledgements sent during a file exfiltration whereas peak (high) values correspond with periods in which no file transfers are being carried out. Moreover, Figure 3b shows the number of messages acknowledged by each acknowledgement message. Values over the orange line correspond to file exfiltrations. We have observed that a network device inspecting this traffic may detect file exfiltrations and the direction they are occurring, even if the information is delivered using encrypted communications.

In a push scenario, the typical traffic of the application produces acknowledgements in the opposite direction of a potential exfiltration every 176 ms on average. Nevertheless, these periods are reduced to approximately 17 ms on average when a file exfiltration is occurring. Additionally, there are differences in the number of messages being acknowledged in those acknowledgements. Acknowledgements of ordinary traffic will be equal to or lower than 6, as shown with the orange line in Figure 3b. When a file is being transferred, these acknowledgements are greater than 6. On the other hand, in a pull scenario the behaviour of acknowledgments changes because file transfers are carried out in the same direction as the rest of the multimedia elements (e.g., screen information). In this scenario, file exfiltrations can be distinguished from other session activities because the average time between acknowledgements in the opposite direction will decrease. Normal traffic produces acknowledgements every 45 ms on average, but when a file is being transferred, these periods decrease to approximately 17 ms. Additionally, there are differences in the amount of information being acknowledged. Acknowledgements of standard traffic are equal to or lower than 40 and acknowledgements of traffic when there is a file transference are greater than 40.

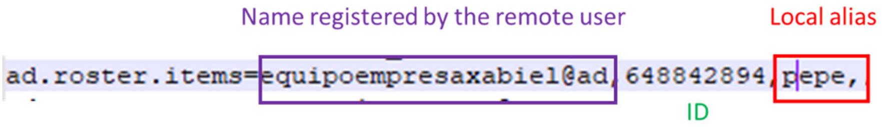
4.2. AnyDesk

AnyDesk is a remote desktop application implemented and distributed by AnyDesk Software GmbH. It provides a remote access service to personal computers running its host application. The functionalities of this product include remote access, file transfer and remote software installation. Just like its competitors, AnyDesk can run in complex networking scenarios without the need for specific routing rules or firewall changes. AnyDesk uses TCP ports 80, 443 and 6568 to establish connections and TLS 1.2 to secure communications.

AnyDesk has three operating modes. Two of these modes allow users to run the application without installation. The first mode is the portable version and does not require administrative privileges to run. The second mode is called "with elevation" and allows users to interact with Windows user account control (UAC). The third option is to install the program, which permits the application to remain in the background even after all of the program windows have been closed.

From the forensics point of view, AnyDesk has two interesting files in the folder "<user>\AppData\Roaming\AnyDesk", a log file named "ad.trace" and the user configuration file "user.conf". Additionally, in the "<user>\ProgramData\AnyDesk" folder there are another two interesting files named "connection_trace.txt" and "ad_svc.trace".

The "user.conf" file has important information related to the remote PC, including the ID of the remote device, the alias defined by the local user for that remote device and the name of that device if it has been registered by the remote user. Figure 4 shows all of these elements identified in the configuration item.



```
ad.roster.items=equipoempresaxabiel@ad,648842894,pepe,
```

Figure 4. Details of the remote device stored in the "user.conf" file.

The default paths for both sides of the session are also registered in the "user.conf" file, in the "local_browser_start_path" and the "remote_browser_start_path" variables, as shown on the following sample:

```
ad.session.local_browser_start_path=908158965:C*\\origen
ad.session.local_file_sort_order=908158965:33
ad.session.remote_browser_start_path=908158965:d*\\
ad.session.remote_file_sort_order=908158965:33
```

The "ad.trace" file registers the different events that are produced during the sessions. When a remote computer connects with the local computer (pull scenario) the following entry is stored in the file:

```
app.backend_session - Incoming session request:  xabiel (908158965)
```

This log entry includes the ID of the remote PC, in brackets, and the username used to start the connection. In cases when the remote PC has a registered name, the log entry includes this name instead of the ID of the computer as shown, in brackets, in the following sample:

```
Incoming session request:  Alumno (equipoempresaxabiel@ad)
```

On the other hand, when the local computer connects with a remote PC (push scenario), the log entry in the "ad.trace" file includes the ID of the remote PC, as shown in the following sample:

```
win_app.frontend - Sending a connection request for address 908158965.
```

Additionally, both the operating system of the remote device and the version of AnyDesk are registered in the "ad.trace" file, as shown in the following sample:

```
app.frontend_session - Remote OS: Windows
app.backend_session - Remote version:  7.0.14
```

A file transfer would produce several log lines in the "ad.trace" file in the on-premises computer. In one of the lines, we can observe the name of the origin folder in quotation marks. However, the name of the transferred file does not appear. A sample follows:

```
app.prepare_task - Preparing files in 'C:\origen'.
app.local_file_transfer - Preparation of 1 files completed (io_ok).
```

Moreover, the "connection_trace.txt" file registers incoming connections. These log file entries provide the same information in "ad.trace" but in a different format, time and date of the incoming connection and ID or registered names of the involved devices, as shown in the following sample:

```
Incoming 2022-11-15, 08:49   User   equipoempresaxabiel@ad    648842894
```

Furthermore, the "ad_svc.trace" file registers incoming connections. However, apart from the device ID, this log file includes the IP address of the remote device. This is very relevant because the destination of the transmission can be discovered. A sample follows:

```
anynet.any_socket - Connecting to 908158965.
anynet.any_socket - Retrieving client information.
anynet.any_socket - Client-ID: 908,158,965 (FPR: 95c686d63e15).
anynet.any_socket - Logged in from 156.35.171.129:63370 on relay
cc24d14a.
net.connection_mgr - making a new connection to client
95c686d63e15355eb7cb
```

Additionally, sessions may have been recorded and the videos would be stored in "<user>\Videos\AnyDesk".

The Windows operating system also registers relevant information. With the default audit configuration, the Windows firewall produces several events (event types 5154 and 5158) indicating some allowed connections with remote devices. The object audit may complement this information with a specific event (event types 4656, 4658 and 4663). This set of events provides the names of exfiltrated files.

Network monitoring may also produce interesting information. AnyDesk transfers all of the multimedia elements over a single end-to-end bidirectional TCP connection using

TLS. In a push scenario there is little information to send from the local computer to the remote device, so most TCP segments will be empty when they are sent and they will only be used to acknowledge the data being received in the opposite direction. Therefore, in this scenario we can clearly identify a file exfiltration because of an increase in the number of TCP segments with payload being sent by the local computer and by the behaviour of the acknowledgements sent back from the remote computer. A sequence of TCP segments with payload sizes equal to the maximum segment size (MSS) in that network segment indicates a file exfiltration. Moreover, a file exfiltration, or the usage of the file transfer tool, will make the remote computer send TCP acknowledgement segments without payload continuously, as piggybacking will not be sufficient to acknowledge all of the data segments being received. On the other hand, in a pull scenario, the delivery of the exfiltrated files takes place in the same direction as the delivery of all of the multimedia elements of the session. Thus, it is more difficult to identify an exfiltration. The delivery of the multimedia elements of the session generates traffic with notably burst-like behaviour. Under a regular use of the system, a sliding window of 1 s used to calculate average times between data segments will obtain values ranging from less than 1 ms to more than one hundred milliseconds. If an exfiltration is performed, this method to calculate average times between data segments will produce sustained average times between data segments below 1 ms. If the connection is monitored in the opposite direction, the same burst-like behaviour is observed for TCP acknowledgements. When an exfiltration is performed, a sliding window of 1 s used to calculate average times between acknowledgement segments will obtain sustained values below 1 ms, whereas under a regular use of the system this value ranges between 1 and more than 100 ms.

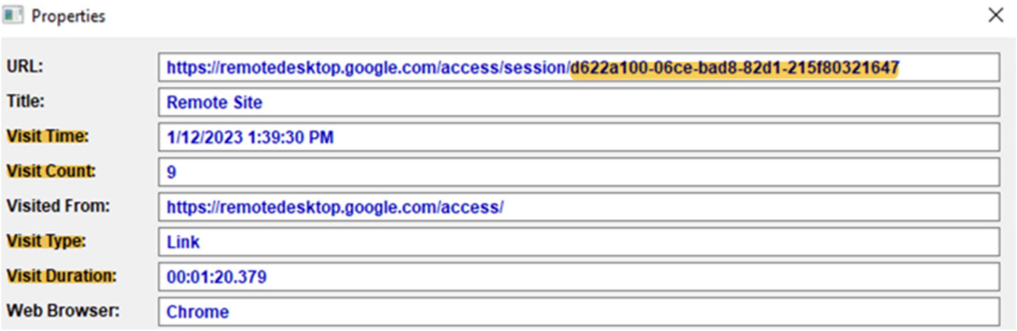
In summary, we can ascertain whether a connection with a remote device has taken place and when. Additionally, we can discover if a file transmission has been produced, when it has been produced, the number of transmitted files, the source folder and the ID and IP address of the destination host. If the Windows object audit is activated, we can also determine the names of the exfiltrated files.

4.3. Chrome Remote Desktop

Chrome Remote Desktop (Chrome RD) is an application developed by Google LLC (Menlo Park, CA, USA) that employs a proprietary protocol called Chromoting. This application is compatible with Linux, Windows, MacOS and ChromeOS. Additionally, there is a software client available for mobile devices running Android or IOS. Chrome RD is based on QUIC and the webRTC protocol stack and uses the VP8 and VP9 codecs.

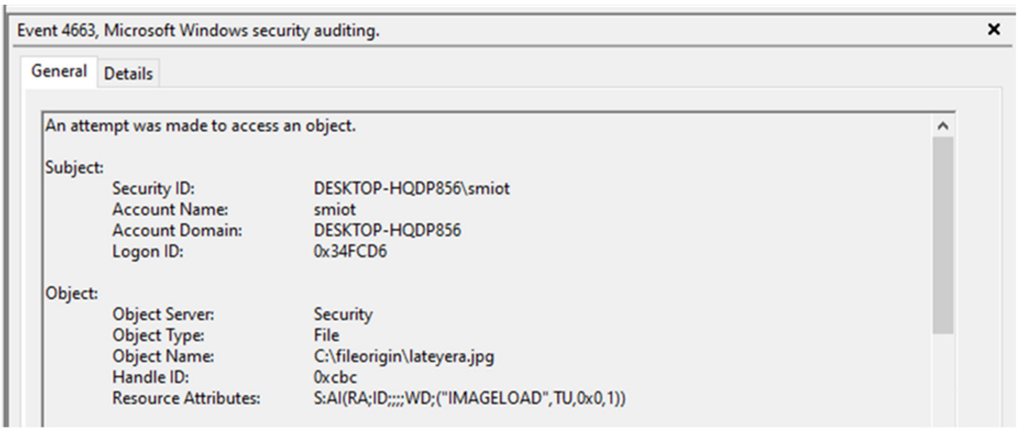
Chrome RD allows two types of connections. The first type is named “Remote Access”. It is a pre-authorized and permanent connection that requires both computers to use the same Google account. The second type is named “Remote Support”. This type has been designed for occasional connections in which one person provides support to a remote user. In this connection mode, the remote user must generate an access code that should be given to the support person to allow the connection between their devices. Connections are blocked automatically after a certain period, forcing remote users to confirm that they would like to continue sharing their desktop by pressing a “Continue Sharing” button. Our analysis was carried out using Chrome RD version 2.1 for Windows 10, which is currently the latest stable version of the system.

If the “Remote Access” option is used, it is not necessary to install the Chrome RD extension in the on-premises computer in a push scenario. However, traces of the usage of this software may be found in the history of the browser, as shown in Figure 5. The browser records the time and day of the connection, its duration and a counter with the number of times the remote device was accessed. This counter is updated every time a new connection is performed. In addition, an ID is assigned to every new remote computer, to be later used to report future connections. Regarding the transference of files, the Windows object audit permits one to determine the directory and name of exfiltrated files, as shown in Figure 6. There are three events registered for this type of action (event types 4656, 4658 and 4663).


Properties

URL:	https://remotedesktop.google.com/access/session/d622a100-06ce-bad8-82d1-215f80321647
Title:	Remote Site
Visit Time:	1/12/2023 1:39:30 PM
Visit Count:	9
Visited From:	https://remotedesktop.google.com/access/
Visit Type:	Link
Visit Duration:	00:01:20.379
Web Browser:	Chrome

Figure 5. Relevant information found in the browsing history of an on-premises computer in a push scenario, when the "Remote Access" connection mode is used.



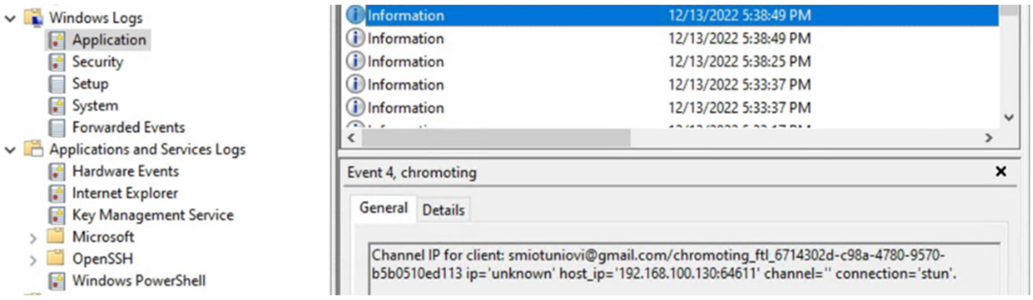
```
An attempt was made to access an object.

Subject:
          Security ID:            DESKTOP-HQDP856\smiot
          Account Name:           smiot
          Account Domain:         DESKTOP-HQDP856
          Logon ID:               0x34FCD6

Object:
          Object Server:          Security
          Object Type:            File
          Object Name:            C:\fileorigin\lateyera.jpg
          Handle ID:              0xcbc
          Resource Attributes:    S:AI(RA;ID;;;;WD;("IMAGELOAD",TU,0x0,1))
```

Figure 6. Directory and file names of exfiltrations registered by the Windows object audit.

In a pull scenario, the application running in "Remote Access" mode also produces several artifacts. Two Windows informative events (event types 1 and 4) are generated by Chrome RD when a client connects. Analysing these events, we can obtain the connection date and time, together with the Google email address used for the remote access as shown in Figure 7. The Windows firewall also produces an event (event type 5158). Furthermore, the Windows object audit allows us to determine the name of exfiltrated files (event types 4656, 4658 and 4663).



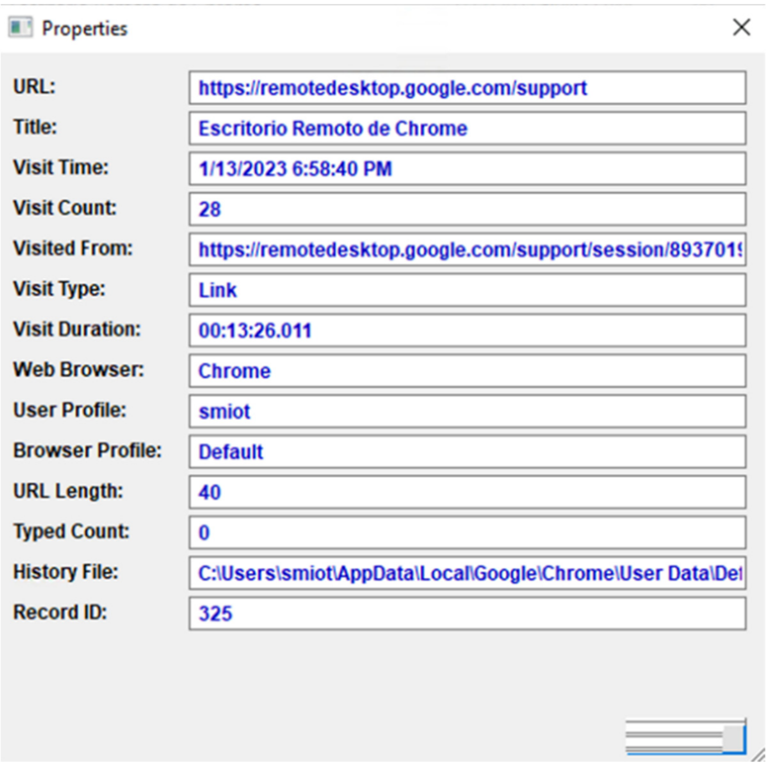
```
Information        12/13/2022 5:38:49 PM
Information        12/13/2022 5:38:49 PM
Information        12/13/2022 5:38:25 PM
Information        12/13/2022 5:33:37 PM
Information        12/13/2022 5:33:37 PM
```

```
Channel IP for client: smiotuniovi@gmail.com/chromoting_ftl_6714302d-c98a-4780-9570-
b5b0510ed113 ip='unknown' host_ip='192.168.100.130:64611' channel='' connection='stun'.
```

Figure 7. Windows informative event log entries generated by Chrome RD.

Once an exfiltration has been performed, if the perpetrator uninstalls the Chrome RD extension from the on-premises computer to remove evidence, there are several variables related to this software which remain in the Windows registry. Moreover, several artifacts generated by the usage of Chrome RD remain in the cache of Google Chrome, although they are only generated when the program is executed.

If the "Remote Support" connection mode is used, it is only possible to transfer files in the push scenario. Chrome RD allows users to copy files to the machine being assisted, but it does not allow users to extract files from that device. Evidence from the remote assistance session can be found in the history of the browser, as shown in Figure 8. The information available is the same as that which may be found when the system is used in the “Remote Access” mode.


Properties

URL:	https://remotedesktop.google.com/support
Title:	Escritorio Remoto de Chrome
Visit Time:	1/13/2023 6:58:40 PM
Visit Count:	28
Visited From:	https://remotedesktop.google.com/support/session/893701
Visit Type:	Link
Visit Duration:	00:13:26.011
Web Browser:	Chrome
User Profile:	smiot
Browser Profile:	Default
URL Length:	40
Typed Count:	0
History File:	C:\Users\smiot\AppData\Local\Google\Chrome\User Data\De
Record ID:	325

Figure 8. Relevant information found in the browsing history of an on-premises computer in a push scenario, when the “Remote Support” connection mode is used.

Network monitoring may also produce interesting information to detect when a remote session starts and ends using both modes of operation of Chrome RD. During the remote desktop session, QUIC messages are exchanged between the computer and Google servers. However, the exchange of data produced between the involved computers is performed using webRTC data channels. This implies that multimedia data are multiplexed and sent encrypted using DTLS 1.2 over UDP, including the exfiltrated files. As with the rest of the applications, in a push scenario most data are delivered from the remote device to the local computer, so in the opposite direction the majority of UDP datagrams carry control information only, with payloads smaller than 400 bytes and rates under 50 datagrams per second. However, when a file exfiltration occurs, there is a considerable increase in the delivery of UDP datagrams containing DTLS messages. In this scenario, rates over 50 UDP datagrams per second with a DTLS payload from the local computer to the remote computer correspond with file exfiltration operations. Moreover, we also see an increase in the number of DTLS + UDP datagrams being sent in the opposite direction. In a pull scenario, exfiltrated files are sent in the same direction as the rest of the multimedia elements of the session. Though exfiltrations are more difficult to detect in these scenarios, we have seen that rates over 70 UDP datagrams per second with a DTLS payload from the local computer to the remote computer correspond with file exfiltrations.

In summary, we can determine when a connection has been made to a remote computer and its duration by using either the “Remote Access” or the “Remote Support” modes. In addition, it is possible to obtain the Google email account associated with the established session in the pull scenario when the user runs the system in the “Remote Access” mode. If the Windows object audit is activated, we can also determine the name of the files transmitted.

5. Conclusions and Future Work

In this paper we have analysed the behaviour of three applications used for remote desktop operations. The main goal of this research is to verify whether it is possible to detect file exfiltrations using these applications or not. Additionally, we sought to analyse the evidence which may be found in end-systems and the behaviour of their communications to be able to answer a set of interesting questions from a forensics point of view. The answers to these questions may form the basis of solid digital evidence for legal purposes. Nevertheless, our findings only deliver a partial response to said questions, thus, other complementary sources are necessary in order to build a complete answer and strong digital evidence. This includes inventory information, details of connections performed from a certain location or authentication logs in corporate services.

The study has been focused on three popular solutions: TeamViewer, AnyDesk and Chrome Remote Desktop. Moreover, we have studied two types of situations: a push scenario, in which users connect from an on-premises computer to a computer outside the company to exfiltrate files and a pull scenario, in which the user is connecting from a remote location to a on-premises computer. Table 1 shows a summary of the relevant evidence identified in the study.

Table 1. Summary of the relevant evidence identified in the study.

	TeamViewer	**AnyDesk**	**Chrome RD**
When	Remote session initiation time, file transfer time	Remote session initiation time, file transfer time	Remote session initiation time and end time
Who/to whom	ID of remote device	IP address and ID of remote device	Google account
What	Number of files, names, sizes (push scenario), source and destination folders	Number of files, source folder and destination folder	
Additional information	Connection scenario (push, pull)	Connection scenario (push and pull), remote OS and application version, remote device name	Google email, only in pull scenario using remote access
Windows audit	File name	File name	File name
network monitoring	Exfiltration produced, destination IP	Exfiltration produced, destination IP	Exfiltration produced, destination IP

The use of TeamViewer or AnyDesk to exfiltrate files has similarities and differences from the forensics point of view. Both technologies register connections and file transmissions. However, AnyDesk does not include the names of exfiltrated files in log entries, only the source folder. On the other hand, TeamViewer includes not only the name, but also the origin and destination paths. Moreover, AnyDesk registers in one of its log files the IP addresses of remote computers, thus, new investigation lines may be opened thanks to this information (provider, geolocation, etc.).

Chrome Remote Desktop has totally different behaviour. This application is based on a browser plugin and does not generate log files. It has two modes of operation: “Remote Access” and “Remote Support”. The latter is more limited and only allows users to transfer files in a push scenario. The activity details need to be extracted from the Chrome history. However, these details do not include relevant information from an investigation point of view. Furthermore, the browser history may be deleted without admin privileges, so evidence may be easily cleaned.

The Windows event log in its default configuration produces limited information for the three analysed applications. Connection events indicate that the application is in use and that connections with master servers have been established. The object audit provides more information about the files transmitted. This is very positive for an investigation, but it is important to consider that the use of this type of audit is extremely resource consuming and, thus, is rarely activated.

From the point of view of the network traffic generated by the applications, all of them show specific behaviours which may be used to identify an exfiltration, even when encryption is in use. The files being transferred cannot be identified, but a device monitoring the activity in the network may be able to detect file exfiltrations and trigger an alarm. In the case of TeamViewer, a proprietary protocol implemented over UDP is in use. Although contents are sent using encryption, file exfiltrations may be identified by monitoring both the rate at which acknowledgements are being sent and the number of fragments being acknowledged in these messages. In the case of AnyDesk, TLS and TCP are used to deliver multimedia information. File exfiltrations may be detected by monitoring the size of the payload carried in TCP segments and the rate at which segments carrying data are being sent. Finally, Chrome Remote Desktop is based on the webRTC protocol stack. File transferences are performed using DTLS. An exfiltration may be detected by monitoring the rate at which UDP datagrams carrying a DTLS payload are being generated.

If the user behind the remote device would like to be identified, this needs the cooperation of manufacturers in the case of TeamViewer and AnyDesk. Nevertheless, these companies are reluctant to provide this type of information without a warrant, because of personal data protection policies in force in several regions of the world. Thus, it is necessary to produce enough evidence to support a request for this type of information. In the case of Chrome RD, it is possible to obtain the Google account associated with a file exfiltration, when a user from outside the organization connects to an on-premises computer using the "Remote Access" connection mode.

Our study also highlights that Chrome RD is the most dangerous remote access application as the evidence produced for a forensic investigation is very limited and may be easily erased. AnyDesk and TeamViewer produce much more information, but some investigation questions cannot be answered if the information registered in their log files is not combined with other log mechanisms of the operating system in use. Nevertheless, in AnyDesk we do not have the name of the files transferred and in TeamViewer we do not have the destination IP address. Although communications are encrypted for all of the applications, network traffic analysis may be useful as an early warning mechanism, because exfiltrations can be detected. Finally, we must consider that Chrome RD is not a complete application and may be easily blocked. It requires logging into a Google account to function, something that AnyDesk or TeamViewer do not require.

Though our study has produced interesting results and the analyses performed were focused on some of the main solutions that are currently available, technology is constantly evolving. Current applications continuously change and new versions and upgrades are released frequently. Furthermore, new solutions occasionally appear that may use different protocols or communication techniques. Thus, the manner in which exfiltrations may be detected and the registered evidence also change continuously, so this type of study must be constantly repeated.

Author Contributions: Conceptualization, X.G.P. and D.M.; methodology, R.G.; software, D.G.; validation, A.G.P.; formal analysis, R.G.; investigation, X.G.P., D.M. and V.C.; resources, D.M.; data curation, D.G.; writing—original draft preparation, X.G.P., D.M. and V.C.; writing—review and editing, A.G.P.; supervision, X.G.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Dataset available on request from the authors.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Georgiadou, A.; Mouzakitis, S.; Askounis, D. Working from home during COVID-19 crisis: A cyber security culture assessment survey. *Secur. J.* **2022**, *35*, 486–505. [CrossRef]
2. Strakšienė, G.; Ruginė, H.; Šaltytė-Vaisiauskė, L. Characteristics of Distance Work Organization in SMEs During the COVID-19 Lockdown: Case of Western Lithuania Region. *Entrep. Sustain. Issues* **2021**, *8*, 2010–2225. [CrossRef] [PubMed]
3. Islam, M.A.; Igwe, P.A.; Rahman, M.; Saif, A.N.M. Remote working challenges and solutions: Insights from SMEs in Bangladesh during the COVID-19 pandemic. *Int. J. Qual. Innov.* **2021**, *5*, 119–140. [CrossRef]
4. Fortune Business Insights. "Remote Desktop Software Market". Available online: https://www.fortunebusinessinsights.com/remote-desktop-software-market-104278 (accessed on 26 March 2024).
5. 6Sense. "Remote Support", Technographics. 2024. Available online: https://6sense.com/tech/remote-support (accessed on 26 March 2024).
6. TrustRadius. "Remote Desktop Software Statistics and Trends". Available online: https://solutions.trustradius.com/vendor-blog/remote-desktop-buyer-statistics-and-trends (accessed on 26 March 2024).
7. Haider, M.; Anwar, A.I. The prevalence of telework under COVID-19 in Canada. *Inf. Technol. People* **2023**, *36*, 196–223. [CrossRef]
8. Azhar, M.A.H.B.; Timms, J.; Tilley, B. Forensic Investigations of Google Meet and Microsoft Teams—Two Popular Conferencing Tools in the Pandemic. In *Digital Forensics and Cyber Crime*; ICDF2C 2021. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering; Gladyshev, P., Goel, S., James, J., Eds.; Springer: Cham, Switzerland, 2022; Volume 441.
9. Hornyák, O. Protection against remote desktop attacks. *Prod. Syst. Inf. Eng.* **2022**, *10*, 3. [CrossRef]
10. Manson, J. Remote Desktop Software as a forensic resource. *J. Cyber Secur. Technol.* **2022**, *6*, 1–26. [CrossRef]
11. Kerai, P.; Vekariya, V. An exploration of artefacts of remote desktop applications on Windows. In Proceedings of the 14th Australian Digital Forensics Conference, Edith Cowan University, Perth, Australia, 5–6 December 2016; pp. 42–49.
12. Kerai, P. Tracing VNC And RDP Protocol Artefacts on Windows Mobile and Windows Smartphone for Forensic Purpose. In Proceedings of the 1st International Cyber Resilience Conference, Edith Cowan University, Perth, Australia, 23 August 2010; pp. 58–68.
13. Altschaffel, R.; Clausing, R.; Kraetzer, C.; Hoppe, T.; Kiltz, S.; Dittmann, J. Statistical Pattern Recognition Based Content Analysis on Encrypted Network: Traffic for the TeamViewer Application. In Proceedings of the 2013 7th International Conference on IT Security Incident Management and IT Forensics, Nuremberg, Germany, 12–14 March 2013; pp. 113–121.
14. Jiang, M.; Gou, G.; Shi, J.; Xiong, G. I Know What You Are Doing with Remote Desktop. In Proceedings of the 2019 IEEE 38th International Performance Computing and Communications Conference, London, UK, 29–31 October 2010; pp. 1–7.
15. Yang, T.Y.; Dehghantanha, A.; Choo, K.K.R.; Muda, Z. Windows Instant Messaging App Forensics: Facebook and Skype as Case Studies. *PLoS ONE* **2016**, *11*, e0150300. [CrossRef] [PubMed]
16. Nicoletti, M.; Bernaschi, M. Forensic analysis of Microsoft Skype for Business. *Digit. Investig.* **2019**, *29*, 159–179. [CrossRef]
17. Paligu, F.; Varol, C. Microsoft Teams desktop application forensic investigations utilizing IndexedDB storage. *J. Forensic Sci.* **2022**, *67*, 1513–1533. [CrossRef] [PubMed]
18. Khalid, Z.; Iqbal, F.; Al-Hussaeni, K.; MacDermott, A.; Hussain, M. Forensic Analysis of Microsoft Teams: Investigating Memory, Disk and Network. In *Science and Technologies for Smart Cities*; Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering; Springer: Cham, Switzerland, 2022; Volume 442, pp. 583–601.
19. Iqbal, F.; Khalid, Z.; Marrington, A.; Shah, B.; Hung, P.C. Forensic investigation of Google Meet for memory and browser artifacts. *Forensic Sci. Int. Digit. Investig.* **2022**, *43*, 301448. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.