Validation Report

This document will explain the testing procedure and methodology used to validate the Torrential Downpour v 1.22 investigative tool. Understanding the following terms[1] will be helpful when reading this document as many of them will be mentioned throughout.

A. GLOSSARY

availability
The number of complete copies of the torrent contents there are distributed in the part of the swarm you're connected to. The amount of the torrent contents you currently have is included in the availability count. A swarm with no seed and with an availability below 1.0 will likely be unable to finish transferring the complete torrent contents.

byte
A unit used for measuring the size of data on a computer storage device. Many people confuse "byte" for "bit" when referring to speeds. A byte is composed of 8 bits, so there is a clear distinction, and terminology should not be confused when referring to bytes.

client
The application a user is using when connected to a swarm. In this case, the application being used to connect to swarms is BitTorrent, so it is the client.

download
The act of transferring data from another computer onto your own.

firewall
A barrier (hardware and/or software) that prevents communication to and/or from certain computers, depending on the rules set in the firewall.

hash
A "fingerprint" of data assumed to be unique to the data. Because of the assumed uniqueness of the data, it is used to verify that a piece of data is indeed uncorrupted (since the corrupted data's hash would not match its expected hash).

hash check
The comparing of a piece of data's hash with a reference hash in order to verify the integrity of the piece of data.

hashfail
When a piece fails the hash check used to verify data integrity.

[1] This excerpt of definitions of terms was taken directly from the BitTorrent website. These definitions as well as a full list can be found at http://help.bittorrent.com/customer/portal/articles/179175-glossary.

interested

This word describes the state of a BitTorrent connection. When a peer is interested, it means the peer is interested in the data that the peer on the other end of the connection has and is willing to accept data from the other peer.

IP address

A number used to uniquely identify devices on a network.

LAN IP address (also referred to as a private IP)

The private, internal IP address that locates a computer on a LAN. A LAN IP address is not visible to users outside of the LAN. As described by RFC 1918, the following ranges are designated as reserved IP addresses for private LANs:

- 10.0.0.0 - 10.255.255.255
- 172.16.0.0 - 172.31.255.255
- 192.168.0.0 - 192.168.255.255

P2P (peer-to-peer)

The use of bandwidth of users using the same peer-to-peer service to perform the functions of the peer-to-peer service or software. Centralized servers are not what keeps P2P networks alive, but rather, the peers themselves.

payload

The actual data being transferred from sender to receiver, not counting overhead.

peer

A user/client connected to the swarm. People sometimes refer to peers as "leechers," though they also use the same word to refer to its more negative connotation. It's recommended that you use the word "leecher" to strictly refer to people who don't share so to keep the distinction clear and confusion to a minimum.

piece

The smallest appreciable unit of data in BitTorrent. The size of pieces can be different depending on the .torrent file in question.

protocol

A set of rules and description of how to do things. In the case of the BitTorrent protocol, it is a set of rules describing how BitTorrent clients should communicate and transfer data with each other.

seed

A peer with 100% of the data in the torrent contents.

seeding

The act of being connected to a swarm as a seed.

swarm
The collective group of peers (which includes seeds) that are connected by a common .torrent file.

torrent
A small file containing metadata from the files it is describing. In other contexts, it is sometimes used to refer to the swarm connected around that small file.

upload
The act of transferring data from your computer onto another.

B. SETUP: Note that all software was installed using default setting, except where otherwise noted.

1) Both the investigative system and the target system were created using VMWare Workstation Player 15 version 15 build-10952284. This free tool, downloaded from https://www.vmware.com, allows for the creation of virtual machines (VMs), which can easily be copied or transferred to other users or systems. As taken from the VMWare website:

"The isolation and sandbox capabilities of VMware Workstation Player make it the perfect tool to help you learn about operating systems, applications and how they work. Being able to run a server environment on a desktop PC also allows you to explore software and application development in a "real world" environment without interfering with the host desktop.' [2]

2) Windows 10 Pro (64 bit) build 1809 OS Build 17763.253 was chosen as the operating system for both VM's. Prior to conducting the validation test, Microsoft automatic updates was disabled on both machines. While Torrential Downpour will run on older versions of Windows, Windows 10 was selected as it is the most recent release of the Windows operating system. Torrential Downpour does not operate on other operating systems such as Linux or Mac iOS devices.

3) Private Internet Access (PIA) version .81 was used on both VMs. PIA allows for internet connectivity through a virtual private network (VPN):

"*PRIVATE INTERNET ACCESS provides state of the art, multi-layered security with advanced privacy protection using VPN tunneling. Scroll below to the Security Layers section to learn more about each individual layer.*

Our services have been designed from the ground up to be able to operate using built-in technology pre-existing in your computer or smartphone device.

[2] Quoted from the VMWare website at https://www.vmware.com/products/workstation-player.html

The services operate at the TCP/IP interface level, which means all of your applications will be secured, not just your web browser." [3]

4) In order to conduct packet captures to show network traffic, the software tool Wireshark version 2.6.6 was used on both VM's. This free tool is available for download from https://www.wireshark.org/ Wireshark is used to record network traffic (packet captures) on the target machine to show how a standard BitTorrent client, in this case BitTorrent, conducts multisource downloads to obtain a payload. When utilizing Torrential Downpour, Wireshark is used on both the investigative and target VMs to show that the software only conducts single source downloads (SSD).

"Wireshark is the world's foremost and widely-used network protocol analyzer. It lets you see what's happening on your network at a microscopic level and is the de facto (and often de jure) standard across many commercial and non-profit enterprises, government agencies, and educational institutions. Wireshark development thrives thanks to the volunteer contributions of networking experts around the globe and is the continuation of a project started by Gerald Combs in 1998.' [4]

Bram Cohen, the architect of the BitTorrent protocol, recommends the use of Wireshark to test BitTorrent applications:

"When developing a new implementation the Wireshark protocol analyzer and its dissectors for bittorrent can be useful to debug and compare with existing ones." [5]

5) To display hash values of created or downloaded files, the program Cyohash version 1.02 was utilized on both VM's. This tool adds right click functionality to the mouse that allows for hash values to be easily calculated and displayed for individual files. This tool will calculate both the MD5 and SHA1 hash values for a given file.

6) For this validation test, a .torrent file needed to be created to share non-copywritten files on the BitTorrent network. To create this .torrent and to download and share (seed) these files, BitTorrent 7.0 was used.

7) The images used in the creation of the validation torrent were downloaded from https://www.pexels.com/:

"It's hard to understand complex licenses that is why all photos on Pexels are licensed under the ***Creative Commons Zero (CC0) license****. This means the pictures are completely free to be used* ***for any legal purpose****.*

- *The pictures are* ***free for personal and even for commercial use****.*

[3] Quoted from the PIA website at https://www.privateinternetaccess.com/
[4] Quoted from the Wireshark website at https://www.wireshark.org/
[5] Bram Cohen, The BitTorrent Protocol Specification

- *You can modify, copy and distribute the photos.*
- *All without asking for permission or setting a link to the source. So,* ***attribution is not required****.*

The only restriction is that identifiable people may not appear in a bad light or in a way that they may find offensive, unless they give their consent. You should also make sure the depicted content (people, logos, private property, etc.) is suitable for your application and doesn't infringe any rights.

The CC0 license was released by the non-profit organization Creative Commons (CC). Get more information about Creative Commons images and the license on the official license page." [6]

Files of various sizes were used for the payload and were saved in a folder named "Validation stock photos". From there the pictures were arranged randomly in two separate sub directories named "1" and "2" respectively, with one file being left in the root directory. To create the .torrent file itself, the built-in creation tool incorporated into BitTorrent was pointed at the "Validation stock photos" folder. The piece size selected was 1024 kB and a number of trackers were added. No other options were changed.

8) For the investigative VM only, Roundup Torrential Downpour version 1.22 (TD) was used. This investigative software is available for law enforcement only and was developed by the University of Massachusetts, Amherst. TD follows the BitTorrent protocol with few exceptions. The first exception is that TD does not take advantage of what is referred to as file swarming. File swarming can speed up the download process by downloading from multiple BitTorrent programs simultaneously. Instead, TD only requests to download pieces from a single IP thereby insuring that any downloaded data came from a single sharing BitTorrent program. Although standard BitTorrent programs will download data from a single IP address if that is the only download candidate available at that moment, TD can *only* download from a single IP address regardless of the number of BitTorrent programs sharing the same data. Secondly, TD cannot share data back to the BitTorrent filesharing network. This is easily accomplished since every piece of data we become in possession of was downloaded from a single IP who would never need those pieces back.

9) To view the contents of a .torrent file Roundup Torrent Viewer version 2.3, which is a standalone torrent viewing program, was used on both virtual machines. This program, which was written by the University of Massachusetts, Amherst, reads the data from a .torrent file and presents it in an easy to read format for the user. When directed at a .torrent file, the viewer will display information found within the torrent, which includes the following:

[6] Quoted from the Pexels website at https://www.pexels.com/photo-license/

- Info Hash
- Number of Pieces
- Files
- Creation Date (GMT)
- Publisher
- Public / Private
- Comment
- Piece Size
- Total Size
- Created By
- Publisher URL
- Files
    - File Name
    - Index Number
    - Size
    - Piece Range
    - Path
- Piece Hashes
- Announce / Announce List
- DHT Nodes

10) LibreOffice 6.1.3.2 was used for documentation purposes as it is also a free program which runs on multiple operating systems. It is available for download from https://www.libreoffice.org/

11) For consistent date and time documentation, the Atomic Clock application written by Timo Partl was downloaded through the Microsoft store and installed on both VMs. Information on Timo Partl can be found at https://timopartl.com/

12) To capture and record the entire validation process, Camtasia Studio 8 version 8.6.0 (paid version) was used https://www.techsmith.com/ . During the validation process the recording was conducted in real time and neither the recording of the investigative or target virtual machines was paused or stopped at any time.

C. <u>METHODOLOGY</u> This validation test was conducted on 01/23/2019. Times listed are Eastern Standard Time (EST).

1) Both the investigative and target virtual machines are started, and the atomic clock program run and placed in the bottom right corner of the screen. Both atomic clocks are compared to verify they are reporting the same time.
2) PIA started on both machines and connected to a location which allows for port forwarding.
3) **11:24 AM** Screen recording software started for target VM.
4) **11:24 AM** Private IP address of ethernet adapter displayed utilizing the Windows command prompt and ipconfig command. Private IP address is documented.

5) **11:24 AM** Public IP address for the target VM was displayed by going to the website www.IPChicken.com compared to the public IP reported by PIA. Public IP address and port is documented.
6) **11:25 AM** Validation .torrent file is opened in Roundup Torrent Viewer, and all information is displayed.
7) **11:25 AM** The Wireshark program is run and a packet capture recording of the network traffic is started. The Wireshark recording records all the communication in and out of the target VM. Analysis of this network traffic can be used to identify the communication, and confirm the download was available and conducted through connections with multiple sources as is typical when the data is available from multiple sources. To validate that TD conducts a download from a single BitTorrent program, rather than swarming, would be meaningless if the data was only available from a single BitTorrent program. This validation confirms that even though multiple sources for this data existed, TD will conduct a download from a single IP address.
8) **11:25 AM** BitTorrent is started, however no .torrent files are currently loaded into the program. The port number being used by BitTorrent is shown under the preferences of the program. This port number is the same as what is reported by PIA. The option to "Close button closes uT to tray" is disabled.
9) **11:26 AM** A standard download is initiated with BitTorrent by loading the .torrent file into the program. During the course of the download, the peers tab is displayed to show simultaneous active connections to multiple peers (swarming). This step documents that the data is available from multiple sources.
10) **11:28 AM** Once the payload for the validation torrent is completely downloaded and is displayed as seeding within BitTorrent, the Wireshark capture is terminated and saved onto the desktop of the VM. This packet capture is hashed and displayed using Cyohash. This step was conducted so that any expert will be able to confirm this is an accurate copy of the network traffic recording.
11) **11:28 AM** The .torrent file information is displayed from within the BitTorrent program by clicking on the bottom "General" and "Files" tabs. This data can be compared to the data previously displayed by the Torrent Viewer Program.
12) **11:28 AM** From within BitTorrent, the downloaded files referenced by the .torrent file are displayed. This is done by right clicking on the entry and selecting "Open Containing Folder". The names, sizes and hash values of each file are shown.
13) **11:30 AM** Screen recording the investigative VM is started.
14) **11:30 AM** Private IP address of ethernet adapter displayed utilizing the Windows command prompt and ipconfig command. Private IP address is documented.
15) **11:30 AM** Public IP address for the target VM was displayed by going to the website www.IPChicken.com compared to the public IP reported by PIA. Public IP address and port is documented.
16) **11:30 AM** Validation .torrent file is opened in Roundup Torrent Viewer, and all information is displayed.

17) **11:31 AM** The Wireshark program is run, and a packet capture recording of the network traffic is started on the investigative VM. The Wireshark recording records all the communication in and out of the investigative VM. Analysis of this network traffic can be used to identify the communication, and confirm the download was available and conducted through a connection with only a single source, even though the download was available from multiple sources as seen above in step 9.
18) **11:31 AM** A second Wireshark recording of the network traffic is started on the target VM. This second recording serves to document the investigative download made by TD. Analysis of this Wireshark recording can be used to confirm that all the data being referenced by this .torrent was shared to the investigative computer.
19) **11:32 AM** From within BitTorrent on the target VM, the .torrent being seeded[7] is highlighted and the bottom "Peers" tab selected. This is done to display any connections between this BitTorrent client and other BitTorrent clients which are communicating about and /or downloading pieces of this .torrent.
20) **11:32 AM** TD program is run, and an investigative download is initiated by loading the .torrent and specifying the IP address and port to connect to. Although this method of initiating an investigation is somewhat manual, TD has the ability to load these investigations into the program and conduct downloads automatically. This can be achieved by specifying an IP address, a range of IP addresses, or a geographic region. To determine the approximate location, TD utilizes the free geolocation database provided by www.maxmind.com. Regardless of the method used to initiate the investigation, only three pieces of information is used by TD: the .torrent identifier (infohash), the IP address and the port.
21) **11:32 AM** The date and time of the single source download is documented for both the target and UC VMs.
22) **11:32 – 11:33 AM** Once the single source download has completed, the Wireshark captures are stopped on both machines and saved to their respective desktops. Cyohash is used to display the hash values of these captures on both machines. This step was conducted so that any expert will be able to confirm this is an accurate copy of the network traffic recordings.
23) **11:33 AM – 11:43 AM** The validation worksheet on the investigative VM is completed. While completing the worksheet, the various log files are displayed as well as the specific file information such as file names, sizes and hash values. These can be compared to the data displayed in the Torrent Viewer Program as well as the information that was displayed on the target VM.
24) **11:40 AM** On the target VM the downloaded Validation stock photos folder is copied onto the desktop. A new .torrent file named "New Torrent" is created using this copied folder. This new .torrent is viewed in the torrent viewer program and verified as having the same infohash as the original torrent.
25) **11:42 AM** On the target VM, the copied "Validation stock photos" folder located on the desktop is renamed to "Validation stock".

[7] Peers possessing all the pieces of a .torrent that continue sharing that content are referred to as a seed. As a seed, the BitTorrent application will typically connect to other peers in order to share pieces of a .torrent.

26) **11:42 AM** A new .torrent file named “Name Change” is created using the newly renamed “Validation stock” folder. Once created, this new torrent is viewed in the torrent viewer program and shown to calculate a completely different infohash. This is done to show that any changes made to the file names, directories, data etc. will create a completely new .torrent. BitTorrent programs can only communicate and / or share with each other when both programs are communicating about a .torrent with the se infohash.
27) **11:43 AM** The validation worksheet is saved and closed and cyohash is used to display the hash values of the worksheet.
28) **11:44 AM** A third Wireshark capture of the network traffic is started on the target VM.
29) **11:44 AM** A second Wireshark capture of the network traffic is started on the UC VM
30) **11:44 AM** On the target VM, the original containing folder for the seeding .torrent (found in the “Downloads” directory) is opened and the top-level directory “Validation stock photos” is renamed to “Validation stock” and BitTorrent is shown to still be displaying the “seeding” message.
31) **11:45 AM** Another investigative download of the .torrent from the target VM is attempted.
32) **11:45 – 11:46 AM** BitTorrent seeding failure message displayed on target machine. Upon failure, the Wireshark captures on both VMs are terminated and saved to desktop. Both captures are hashed using Cyohash. This step was conducted so that any expert will be able to confirm this is an accurate copy of the network traffic recordings. The purpose of this step is to illustrate that when changes are made to any data referenced by the .torrent at the location where it is shared from, the BitTorrent program quickly recognized the change and therefore stops the seeding (sharing) process.
33) **11:46 AM** The top-level directory “Validation stock” is renamed back to its original name of “Validation stock photos”. BitTorrent is shown to still be displaying the error message from step 34. The Validation stock photos torrent is selected within the program and the option to “Start” is selected. The error message is shown to change to “Seeding”.
34) **11:46 AM** A fourth Wireshark recording of the network traffic is started on the target VM.
35) **11:47 AM** While BitTorrent is sharing the data, the folder containing the data is moved to a different location (desktop) and BitTorrent is shown to still be “seeding” the files. The purpose of this step is to illustrate that when any data referenced by the .torrent at the location where it is shared from is moved from that shared location, the BitTorrent program quickly recognized the change and therefore stops the seeding (sharing) process.
36) **11:48 AM** A third Wireshark recording of the network traffic is started on the UC VM.
37) **11:48 AM** Another investigative download of the .torrent from the target VM is attempted.
38) **11:48 – 11:49 AM** BitTorrent seeding failure message displayed on target machine. Upon failure, the Wireshark recordings on both VMs are terminated and saved to desktop. Both captures are hashed using Cyohash. This step illustrates what was described in step 35 above
39) **11:49 – 11:50 AM** The recordings for both VMs are terminated.

D. CONCLUSIONS

1. TD properly performed an investigative download from a single sharing BitTorrent program, which can be verified through the Wireshark recordings.
2. TD did not share any file data with any other BitTorrent program on the BitTorrent file sharing network.
3. Downloads can only occur when the data remains available in the location where it is being shared from.
4. Data can only be shared when the file(s) and/or directory(s) remained unchanged.
5. Downloads can only occur when two BitTorrent programs are communicating about the same .torrent (having identical infohashes).
6. Understanding the method by which BitTorrent shares data, that being that a .torrent file is a requirement to download any data, the download of unshared files is impossible.
7. All communications to and from the investigative computer were documented with a Wireshark recording (packet capture). Any nefarious activity where Torrential Downpour would send non- standard BitTorrent protocol messages would be exposed in the review of these packet captures.
8. Dates and times are properly recorded in the log files created by the software.
9. The infohash is properly recorded in the log files created by the software.
10. The IP address and port being investigated is properly recorded in the log files created by the software.
11. The IP address of the investigating computer is properly recorded in the log files created by the software
12. Files(s) and paths are properly recorded in the log files created by the software and match what is defined by the .torrent.
13. The data downloaded matched the data being shared on the target computer.
14. Data can only be downloaded while it is being shared by the BitTorrent program. If the data is moved or deleted it immediately ceases.
15. When a change is made to the shared data, even something as minor as renaming a file, the sharing BitTorrent application quickly recognizes the change and stops sharing the data.

E. POST TEST

1) At the conclusion of the validation video, Cyohash was used to hash the original recording files for both the investigative VM (Image 1) and the target VM (Image 2).

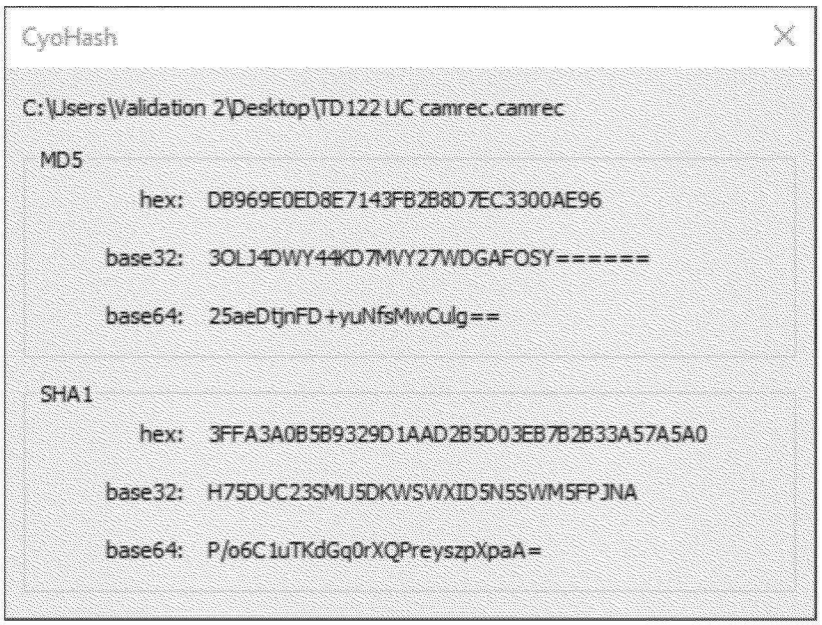

Image 1 Hash values of recording file of investigative VM

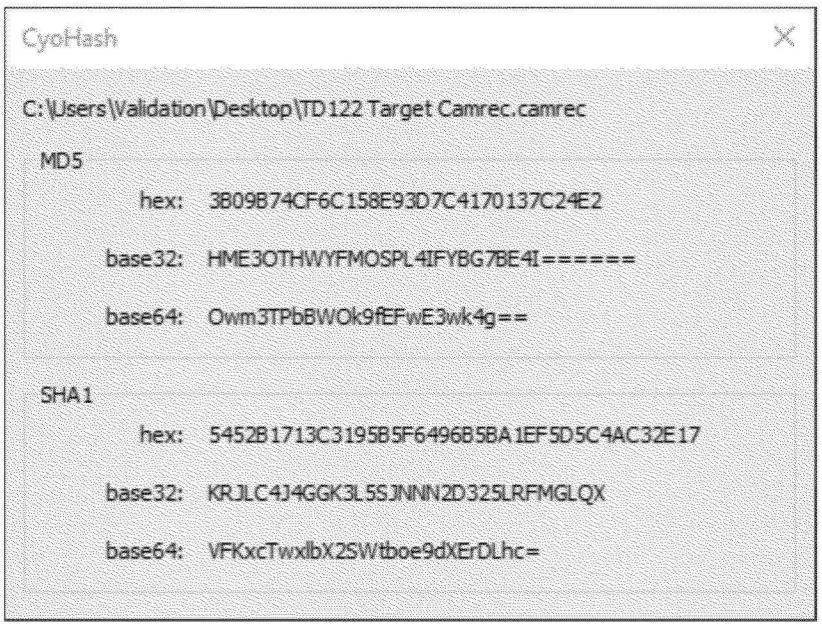

Image 2 Hash values of recording file of target VM

2) Both recording files were then compressed into a .zip format using 7-Zip, and hash values were calculated and documented for these .zip files (images 3 and 4).

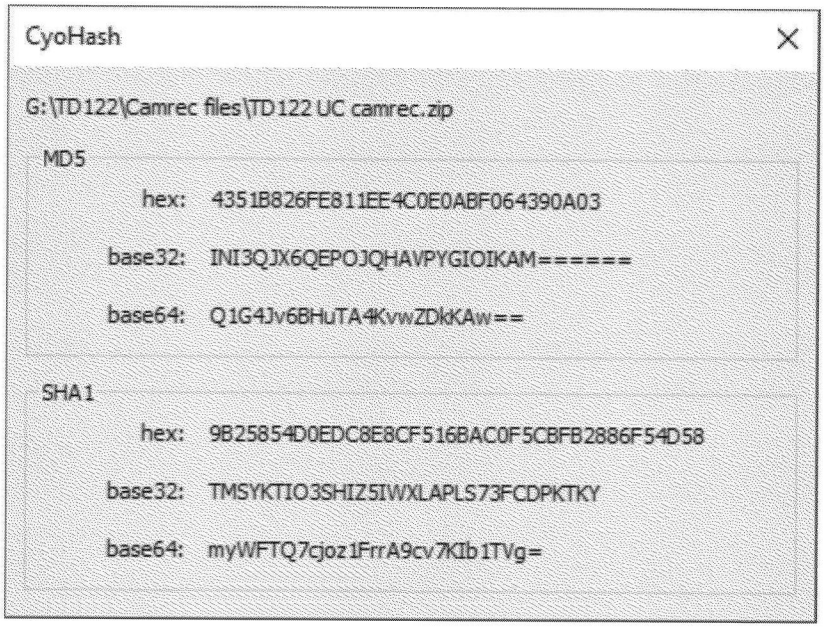

Image 3 Hash values of compressed investigative VM recording

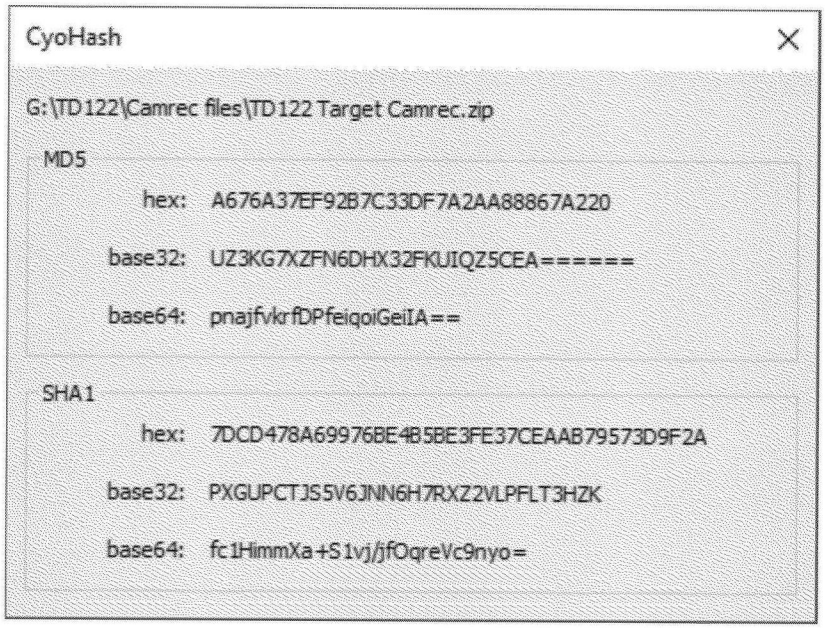

Image 4 Hash values of compressed target VM recording

3) The validation .torrent file, Validation stock pictures folder, and the validation worksheet were transferred from both the investigative VM and target VM to a containing folder outside of the VM's. This containing folder was compressed into a .zip format using 7-Zip and a hash value calculated and documented for the .zip file using Cyohash (Image 5).

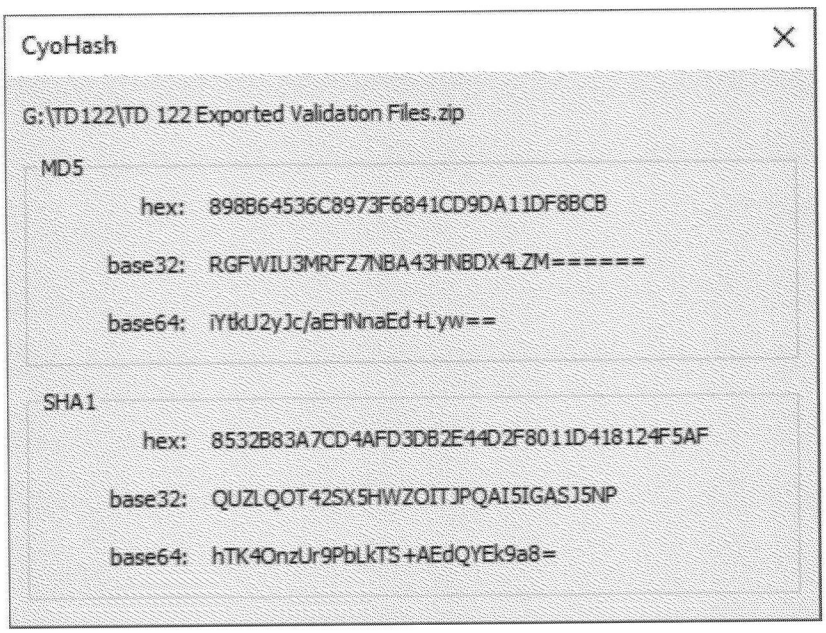

Image 5 Hash values of the exported validation files from both UC and target VMs

4) The Wireshark captures were transferred from both the investigative VM and the target VM into a containing folder outside of the VMs. Once all files were successfully copied the folder was compressed into a password protected .zip file using 7-zip and a hash value was calculated and documented (Images 6). The password used for this .zip file will be included in a separate document.

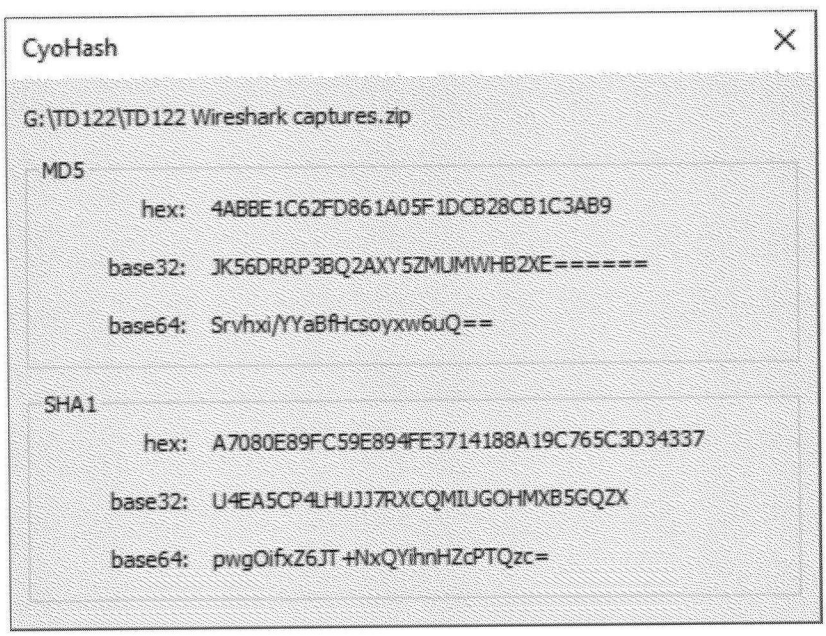

Image 6 Hash values of compressed Wireshark captures from UC and target VM

5) Once all files were copied from both the investigative VM and target VM, they were both shut down, compressed into password protected .zip files using 7-zip and a hash value was calculated and documented for both .zip files (Images 7 and 8). The password used for both .zip files will be included in a separate document.

Image 7 Hash values of compressed investigative VM

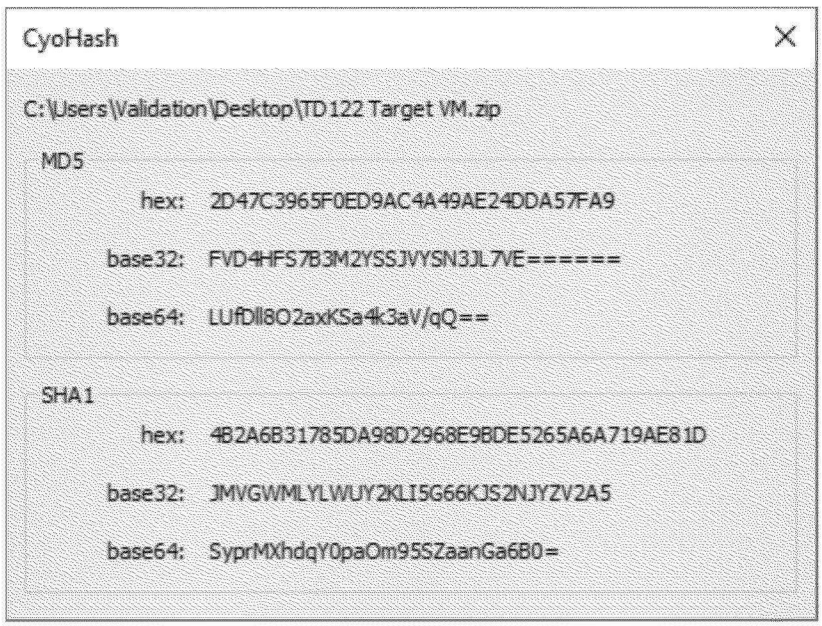

Image 8 Hash values of compressed target VM

F. ATTACHMENTS

i. The BitTorrent Protocol Specification – written by Brian Cohen
ii. Validation worksheet
iii. Validation stock photos folder
iv. Validation .torrent file

v. Wireshark captures
vi. Recording files for investigative VM
vii. Recording files for target VM
viii. Investigative VM
ix. Target VM