



Abstract

Virtual currencies, such as Bitcoin, have recently dominated news headlines. The term virtual currency has been around since 2012. However, virtual currencies as we know them have existed before then. Most virtual currencies can be used to buy goods anonymously, which has made them popular among privacy concerned individuals. Bitcoin's recent spike in popularity has caused a proliferation and dissemination of information regarding its use and advantages to mainstream audiences. Adoption of virtual currencies by the public has increased, which has introduced a novel issue in digital forensic examinations.

This summer, MultiBit, Litecoin, and Darkcoin desktop wallet software was examined for valuable artifacts on Windows 7 and Ubuntu 14.04 operating systems. This was accomplished in a three-fold fashion: Hard drive, Memory, and Network evidence. In all cases, some form of application settings, user timeline information, and transaction logs were among the type of information able to be recovered. The MultiBit wallet contained the greatest amount of user timeline information including a file created upon uninstalling the application. Overall, the three wallets shared remarkable similarity in artifacts generated.

Introduction

Bitcoin's creation is credited to the alias Satoshi Nakamoto, whose identity is still unknown. The idea behind Bitcoin was initially proposed in October 2008 as a "purely peerto-peer version of electronic cash" that would cut out the middle man, i.e. financial institutions.

In the trust model, a trusted third party verifies that the money has not already been spent before allowing a transaction. This obstacle led to Bitcoin's greatest innovation, the blockchain. The blockchain is essentially a public ledger that is made up of blocks containing all previous transactions of the currency (Fig. 1). This blockchain prevents double spending by the verification process each transaction must undergo. This process is designed to take ten minutes to complete and is composed of these basic steps:

- 1. Assign the transaction to a block that is in a queue to be verified
- 2. Confirm coins were signed with the sender's address private key
- 3. Confirm coins have not already been spent by checking the blockchain
- 4. Repeat for all transactions in the block
- 5. Calculate a difficult SHA256 hash of the block plus "nonce"
- 6. Add the block to the blockchain

Materials & Methods

Materials used for this research project include:

One external hard drive

2. VM Workstation and ISO files for operating systems analyzed

3. Bitcoins, Litecoins, and Darkcoins for trading purposes

4. Access to forensic software programs

Cryptocurrency Artifact Analysis

Preston Miller, B.A.¹, Lance Nudd, J.D.², Chris Vance, B.S.¹, Terry Fenger, Ph.D.¹ ¹Marshall University Forensic Science Center, 1401 Forensic Science Dr, Huntington, WV 25701 ²Stroz Friedberg, 32 Avenue of Americas, New York, NY 10013

Results & Discussion



Figure 1. The blockchain is made up of blocks. Each block is made up of hundreds of transactions. The start of the chain, the Genesis block, is blue. The main chain is black and forks in the chain are purple, referred to as "orphan" blocks.

Disk Forensics

Figure 2. Demonstrates the process through which an address is generated. This process is primarily only one way. To protect the address used in this figure, the private and public key values were made up.

Multibit_2014-08-01 - Notepad		🗄 Entries 🛈 Bookmarks 🔍 Search Hits 🥰 Records 🖾 👀	Name	Description	File File Type Ex	File Created	Entry Modified				
File Edit Format View Help	🔁 Home	🕫 File Extents 🧯 Permissions 👸 References 📝 Hash 🕩	12 11 InstallRemover754993591611	9 File, Archive, Not Indexed	Log log	07/17/14 08:53:09AM	07/17/14 08:53:11AM				
L5:05:53.906 [AWT-EventQueue-0] DEBUG org.multibit.network.MultiBitService - MultiBitService#sendCoins - Sent coins has completed L5:05:53.906 [AWT-EventQueue-0] DEBUG org.multibit.network.MultiBitService - MultiBitService#sendCoins - Sent coins. Transaction hash L5:05:53.906 [AWT-EventQueue-0] DEBUG org.multibit.network.MultiBitService - Added bitcoinController 20511156 as listener to tx = 9ad L5:05:53.906 [AWT-EventQueue-0] DEBUG org.multibit.network.MultiBitService - Added bitcoinController 20511156 as listener to tx = 9ad	n is 9ad3169a 3169acc99e22	Temp Temp Temp Temp Temporary Internet Files	13 izpack7771606733487857202 14 izpack7771606733487857202	d Folder, Deleted, Overwritten log File, Archive, Not Indexed	d Log log	07/17/14 08:52:27AM 07/17/14 08:52:27AM	07/17/14 08:53:10AM 07/17/14 08:53:10AM				
Transaction Hash			•								
TRANSACTION Received on Time: 1699	🔚 Text 🐙] Hex 🎧 Doc 🧐 Transcript 🔝 Picture 📃 Report 💽 Consol	e 🔞 Details 🗐 Output 🛛 Lock 🔽 Co	depage 🗌 0/544697							
<pre>L5:19:11.161 [NioClientManager] INFO com.google.bitcoin.core.MemoryPool - [83.40.36.156]:8333: Peer announced new transaction [1] 41. L5:19:12.081 [NioClientManager] DEBUG com.google.bitcoin.core.Wallet - Saw relevant pending transaction 4146a0b6fdebb2235d05de99e5d in [30440220421d7f587a9f8d6e7a8cc286a39bf1d5dced3a3b5b2bb1ea0bff56bae76dcffa02204cec588a4e3c22caf6ce15abdf0f7c5aa875a62b9f98b05f685 Dutpoint:9ad3169acc99e22e2f8f9f26d5b349e1b986985723e1d762dc3ceefe7696788c:0 Dut DUP HASH160 [c9a074d45ac907c1d861d3855602e272ceb2b5e0] EQUALVERIFY CHECKSIG 0.0095 BTC Dut DUP HASH160 [fde6c54b05f81ab98e8f808bf2c60eb6791f4782] EQUALVERIFY CHECKSIG 0.0004 BTC</pre>	46a0b6fdebb2 00002014-0 0150ed 149 0300iles (0450.mod.b 0600er.jar 0750pack.u	00002014-07-17T08:52:27.514 Phase 1: JarFile: C:\Program Files (x86)\MultiBit-0.5.18\Uninstaller\uninstaller.jar 2014-07-17T08:52:27.841 Phase 1: Extract 0150 ed 149 files into C:\Users\									
L5:19:12.081 [NioClientManager] INFO com.google.bitcoin.core.Peer - [83.40.36.156]:8333: Downloading dependencies of 4146a0b6fdebb22	235d05de99e5d 0900\jre7\ 1050ers\	<pre>bin\java.exe -Xmx64m -XX:MaxPermSize=16m - \AppData\Local\Temp\izpack777160673348</pre>	classpath C:\Users\ 7857202 -Dself.mod.jar=C:\Pro	AppData\Local\Temp\izpac gram Files (x86)\ <mark>MultiBi</mark>	k7771606733	487857202.d -Dsel	f.mod.base=C:\Us ler.jar -Dself.				
TRANSACTION Sent on line: 1773	1200mod.cl	ass=com.izforge.izpack.uninstaller.Uninstalle	r -Dself.mod.method=uninstall	-Dself.mod.phase=3 -D	self.memory	=64 com.izforge.i	zpack.uninstalle				
L5:27:19.785 [AWT-EventQueue-0] DEBUG o.m.v.s.a.SendBitcoinConfirmAction - Just about to complete the tx (and calculate the fee) L5:27:19.785 [AWT-EventQueue-0] INFO com.google.bitcoin.core.Wallet - Completing send tx with 1 outputs totalling 1000000 satoshis (L5:27:19.785 [AWT-EventQueue-0] INFO com.google.bitcoin.core.Wallet - with 0.0134103 coins change L5:27:19.785 [AWT-EventQueue-0] INFO com.google.bitcoin.core.Wallet - with a fee of 0.0001 L5:27:19.785 [AWT-EventQueue-0] INFO com.google.bitcoin.core.Wallet - completed: 2f495764adff13ee641eb055f943542198e0aa5dbc1c421 in	not includin alc5b22982eb	EModifier 2014-07-17T08:52:29.277 Phase 3: In: 10 Phase 3: Method returned, waiting for othe return value = 0 C\Users\\AppData\Local\Temp\zpad	voking method: com.izforge.izg r threads 2014-07-17T08:53:10 	ack.uninstaller.Uninstal .098 Phase 2: deleteing LS 34425216 CL 4303152 SO 000	FO 0 LE 1)	11(String[] args) 14-07-17T08:53:10.	2014-07-17T08:5 301 Phase 2: Pha				

27:19.785 [AWT-EventQueue-0] INFO com.goo 27:19.785 [AWT-EventQueue-0] INFO com.goo 27:19.785 [AWT-EventQueue-0] INFO com.goo 27:19.785 [AWT-EventQueue-0] INFO com.goo cc99e22e2f8f9f26d5b349e1b986985723e1d762dc3ceefe7696788c:1 hash160:49d5d227d6143d21def6ca5e165df97f3cedffb0 Input Address
[1e64eb9041df47cb0186b31e5b6be1c825447cb9] EQUALVERIFY CHECKSIG 0.01 BTC
[c9a074d45ac907c1d861d3855602e272ceb2b5e0] EQUALVERIFY CHECKSIG 0.0134103 BTC EventQueue-0] DEBUG o.m.v.s.a.SendBitcoinConfirmAction - The fee after completing the transaction was 10000 Queue-0] DEBUG o.m.v.s.action.SendBitcoinNowAction - Sending from wallet C:\Users\Stroz LLC\AppData\Roaming\MultiBit\mult acc99e22e2f8f9f26d5b349e1b986985723e1d762dc3ceefe7696788c:1 hash160:49d5d227d6143d21def6ca5e165df97f3cedffb0 [1e64eb9041df47cb0186b31e5b6be1c825447cb9] EQUALVERIFY CHECKSIG 0.01 BTC [c9a074d45ac907c1d861d3855602e272ceb2b5e0] EQUALVERIFY CHECKSIG 0.0134103 BTC 27:25.401 [AWT-EventQueue-0] DEBUG org.multibit.network.MultiBitService - Ping: [98.186.167.15]:8333 27:25.526 [AWT-EventQueue-0] DEBUG org.multibit.network.MultiBitService - MultiBitService#sendCoins fust about to send coins 27:25.900 [AWT-EventQueue-0] INFO com.google.bitcoin.core.Wallet - commitTx of 2456f14833634389d12380be0215ba55002ac18f67cf88c154ba6b7f46a688 27:25.900 [AWT-EventQueue-0] INFO com.google.bitcoin.core.Wallet - marked 9ad3169acc99e22e2f8f9f26d5b349e1b986985723e1d762dc3ceefe7696788c; 27:25.900 [AWT-EventQueue-0] INFO com.google.bitcoin.core.Wallet - 9ad3169acc99e22e2f8f9f26d5b349e1b986985723e1d762dc3ceefe7696788c prevtx 27:25.900 [AWT-EventQueue-0] INFO com.google.bitcoin.core.Wallet - >pending: 2456f14833634389d12380be0215ba55002ac18f67cf88c154ba6b7f46a6883

Figure 3. Transactions sent from the wallet appear after a "SendBitcoinConfirmAction" entry. Transactions to the wallet appear after a "Peer announced new transaction" entry. These transactions contain the time, transaction hash, address inputs and outputs, and amount traded. This can also be compared with the start and stop times to determine transactions in particular sessions of the application.



Figure 6. A more detailed version of Fig. 2 starting with the generated ESDCA public key, which is made up of its (x, y) position on the curve. The Script value from Wireshark is already the 20-byte product from the RIPEM160 and SHA256 hash of the public key. After removing the first 3-bytes and last 2-bytes from the Wireshark Script value the remaining 20-bytes must be processed. First a checksum must be calculated by sequential SHA-256 hashing of the 20-byte value with a prepended 00. The first 4 bytes of this value is appended to the original 20-bytes and base58 encoded. Then a Network ID must be prepended to arrive at the actual address. For Bitcoin, the Network ID is often 1 or 3. It is recommended that a script is created to automate this process.

Figure 5. The breakdown of the contents of a Litecoin tx message. This process can be repeated for any cryptocurrency using the Bitcoin protocol. Please note that the length of the "Script Signature" (10) is variable and the "Output Count" (12) will determine how many "Transaction Output" fields there are. Additionally, the "Transaction Output Script" value must be processed to obtain the address in a similar fashion described in the Bitcoin section (note the Network ID for Litecoin is typically



Figure 7. The packet details of the three main messages in the Bitcoin protocol are detailed. The fields within Transaction Input and Output are not shown in this figure. Note that these packet details have the same structure as Litecoin network traffic or any other cryptocurrency using the Bitcoin protocol.



Figure 4. The izpack log appears to be generated when the user uninstalls the MultiBit program from their system.

Network Forensics

recievetradefromUbuntu.pcap [Wireshark 1.10.0 (SVN Rev 49790 from /trunk-1.10)]							
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>G</u> o <u>C</u> apture <u>A</u> nalyze <u>S</u> tatistics Telephony <u>T</u> ools <u>I</u> nternals <u>H</u> elp							
◉ ◉ ∡ ■ ∡ ⊨ ≞ ≌ ≍ ₴ ᆃ ᆃ ⊋ 중 뫞 🗐 🖬 ! ਦ ਦ ਦ 🗉							
Filter: tcp.port == 9333 Expression Clear Apply Save							
No. Time Source Destination Protocol Ler	ngth Info						
68 11:58:33.941885 109.228.164.67 192.168.139.144 TCP	304 9333 > 36235 [PSH, ACK] Seq=117 Ack=62 win=64240 Len=250						
69 11:58:33.957882 192.168.139.144 188.138.197.50 TCP	109 42993 > 9333 [PSH, ACK] Seq=1 Ack=1 Win=64952 Len=55						
	•						
Frame 68: 304 bytes on wire (2432 bits), 304 bytes captured (2432 bits)							
Ethernet II, Src: Vmware_f6:cf:90 (00:50:56:f6:cf:90), Dst: Vmware_df:b6:	03 (00:0c:29:dT:b6:03) 02 168 130 144 (102 168 130 144)						
Transmission Control Protocol. Src Port: 9333 (9333). Dst Port: 36235 (36	235). Sea: 117. Ack: 62. Len: 250						
Data (250 bytes)							
Data: fbc0b6db747800000000000000000000000000000000000							
[Length: 250]							
0000 00 0c 29 df b6 03 00 50 56 f6 cf 90 08 00 45 00)P VE.	Alternating color scheme to differentiate items listed:						
0010 01 22 C0 00 00 00 80 06 1a as 6d e4 a4 43 C0 a8							
0030 fa f0 11 7b 00 00 fb c0 b6 db 74 78 00 00 00 00 {	1. Packet Magic 12. Output Count						
0040 00 00 00 00 00 00 00 e2 00 00 00 5C a6 a0 eC 01 00br .#(.@.F.	2. Command Name						
0060 d6 0f 27 24 ca 98 14 f8 8d ec a8 5d 80 a6 2e ed'\$]	3. Payload Length Transaction Output						
00/0 e4 d9 C5 00 00 00 00 bb 48 30 45 02 20 66 T6 0b	4. Payload Checksum 13. Value						
0090 09 54 79 b5 0c 0d 2e 9c d5 0e 47 bf 94 02 21 00 .TyG!	5. Transaction Version 14. Script Length						
00a0 d5 54 e6 d0 8b 86 ca b5 71 b8 58 e7 f6 1b e7 ba .T q.X	6. Input Count 15. Script						
00c0 01 21 03 52 0a 77 ee 8e f6 90 80 8e c0 35 a7 69 .!.R.w5.i							
00d0 d5 d8 07 2c 0d 5c 79 0e e3 62 d3 c8 3e c0 b8 94, y . b>	Transaction Input Transaction Output						
00f0 19 76 a9 14 ad ab e6 1b 2e 0a 43 46 86 36 e6 49 .v	7. Previous Output - Hash 16. Value						
$\begin{bmatrix} 0100 & bc & 5d & b7 & 24 & 38 & 50 & c8 & 82 & 88 & ac & 92 & 79 & b6 & 30 & 00 & 00 & 0 & . \end{bmatrix}$, \$8Py.0.	8. Previous Output - Index 17. Script Length						
0120 4d 77 76 7e a8 da 19 b4 b5 1e 88 ac 00 00 00 00 Mwv~	9. Script Length 18. Script						
	10. Script Signature						
	11. Sequence 19. Block Lock Time or Block ID						
🛑 💅 File: "G:\Summer Project\Windows 7\Windo 🛛 Packets: 1382 · Displayed: 117 🛛 Profile: Default							

inv	getdata				tx	
Item	Byte offset	Item		Byte offset	Item	
Packet Structure	0-53	Packet Structure	1	0-53	Packet Structure	
Data	54-	Data	1	54-	Data	
Packet Magic	54-57	Packet Magic	1	54-57	Packet Magic	
Command Name (69:6e:76)	58-69	Command Name		58-69	Command Name (74:78)	
Payload Length	(67:65:74:64:61:74:61)			70-73	Payload Length	
Payload checksum	70-73	3 Payload Length		74 77	Devide and a backgroup	
Count	74-77	Payload checksum		14-11	Payload checksum	
Type (01 - TX, 02 - Block, 03 -	78	Count		78-81	Transaction Version	
Uknown)	Uknown) 79-82 Type (01 - TX, 02 - Block, 03 -			82	Input Count	
. Data hash	Data hash 83-114 Data hash			83-262*	Transaction Input *Variable length - 180 bytes in this example	
				263	Output Count	
				264-297	Transaction Output *Amount of tx output fields dependent on output count	
				297-301	Block lock time or Block ID	





Conclusions

Examiners must be aware of common artifacts and the extent of information that is obtainable from evidence. In this study, the best method for observing transactions were network captures in combination with a blockchain lookup utility. However, it is important to find remnants of those transactions on a user's computer. When examining a wallet application, it is vital to determine if there is a corresponding log file because they contain the most user activity and transaction information, outside of a network capture. The cryptocurrency wallet software examined shared remarkable similarity in the amount of information stored on the user's machine for both Windows 7 and Ubuntu 14.04. Network traffic was also similar across the cryptocurrencies studied. The process of parsing Bitcoin's network traffic, such as a tx message, was identical in Litecoin and Darkcoin. Consumers and investors have demonstrated high interest in Bitcoin and virtual currencies as a whole and examiners should be prepared to analyze cryptocurrencies effectively as the likelihood of cases involving them continues to increase.

References

Barber, Simon, et al. "Bitter to Better - How to Make Bitcoin a Better Currency." Financial Cryptography and Data Security (2012).

Duffield, Evan and Kyle Hagan. "Darkcoin: Peer-to-Peer Crypto-Currency with Anonymous Blockchain Transactions and an Improved Proof-of-Work System." (2014).

Katz, J and Y Lindell. Introduction to Modern Cryptography. CRC Press, 2007.

Litecoin Wiki. Comparison between Litecoin and Bitcoin. 22 January 2014. <https://litecoin.info/User:Iddo/ Comparison_between_Litecoin_and_Bitcoin>.

Nakamoto, Satoshi. Bitcoin: A Peer-to-Peer Electronic Cash System. 31 October 2008. < http://nakamotoinstitute.org/ bitcoin/>

Percival, Colin. "Stronger Key Derivation Via Sequential Memory-Hard Functions.: (2009).

Perry, David. Bitcoin Mining in Plain English. 6 September 2012. <http://codinginmysleep.com/bitcoin-mining-inplain-english/>.

Shirriff, Ken. Bitcoins the hard way: Using the raw Bitcoin protocol. < http://www.righto.com/2014/02/bitcoins- hardway-using-raw-bitcoin.html>.

Southurst, Jon. Blockchain's SharedCoin Users Can Be Identified, Says Security Expert. 10 June 2014. < http:// www.coindesk.com/blockchains-sharedcoin-users-canidentified-says-security-expert/>.

Acknowledgments

- Dr. Terry Fenger, MUFSC
- Lance Nudd, Stroz Friedberg
- Michael Younger, Stroz Friedberg
- John Shumway, Stroz Friedberg
- Dan Blank, Stroz Friedberg
- Chris Vance, MUFSC
- Dr. Pamela Staton, MUFSC
- Stroz Friedberg
- Marshall University