Cracking Cryptocurrency
Brainwallets

Ryan Castellucci
DISCLAIMER

- Don’t blame the victim
- Don’t be a jerk
Introduction

- Brainwallets aren’t a good idea - If you want something similar but actually secure, use WarpWallet with eight diceware words
- Don’t use brainwallets.
- Move your money out of any brainwallets you’re using. Please. Don’t get robbed.
  - Somebody just lost $14K last week. Here’s how.
What is a Cryptocurrency?

- Electronic money, secured with cryptography
- Bank and/or gov support not required
- Transfers are like checks, but signed cryptographically instead of with ink
- All transaction history is public and pseudonymous
- Bitcoin, Litecoin, Dogecoin, Defcoin, etc
- Control of private key == Control of money
What is a Brainwallet?

- Passphrase -> Private key & Address
- Knowledge of passphrase == Control of money
Why a Brainwallet?

● What people are thinking
  ○ “Plausible deniability”
  ○ “Possible 5th amendment protection against government seizure”
  ○ “Meat is a better random number generator than silicon because it can’t be backdoored”

● Philosophy can be admirable, but it’s technology that determines what works
  ○ This… doesn’t work.
Things you should remember:

- Cryptocurrency transactions are public
- Brainwallet addresses are in the transactions
- The same passphrase always results in the same address
- Guess-and-check cracking is possible
- A weak passphrase can be guessed
  - That $14K was sent to the passphrase of ""
  - Yes, an empty string
How a typical brainwallet tool works

"correct horse battery staple"   Passphrase
How a typical brainwallet tool works

"correct horse battery staple"  Passphrase
v  v  v  v  v  v  v  v  v  v  v  v
SHA256
c4bbcb1fbec99d65bf59d85c8cb62ee2  Private key
db963f0fe106f483d9afa73bd4e39a8a
How a typical brainwallet tool works

"correct horse battery staple" Passphrase
v v v v v v v v v v SHA256
c4bbcb1fbc99d65bf59d85c8cb62ee2 Private key
db963f0fe106f483d9afa73bd4e39a8a
v v v v v v v v v v v v v v v v v privateToPublic
  (UNCOMPRESSED)                          (COMPRESSED)
04 78d430274f8c5ec1321338151e9f27f4 -> 03 78d430274f8c5ec1321338151e9f27f4 Public key
  c676a008bdf8638d07c0b6be9ab35c71       c676a008bdf8638d07c0b6be9ab35c71
  a1518063243acd4dfe96b66e3f2ec801       a1518063243acd4dfe96b66e3f2ec801
  3c8e072cd09b3834a19f81f659cc3455
How a typical brainwallet tool works

"correct horse battery staple" Passphrase
v v v v v v v v v v SHA256
c4bbcb1fbec99d65bf59d85c8cb62ee2 Private key
db963f0fe106f483d9afa73bd4e39a8a
v v v v v v v v v v v v v v v v v v v v v v v v v v v v privateToPublic
(UNCOMPRESSED)                          (COMPRESSED)
04 78d430274f8c5ec1321338151e9f27f4 -> 03 78d430274f8c5ec1321338151e9f27f4 Public key
c676a008bdf8638d07c0b6be9ab35c71 c676a008bdf8638d07c0b6be9ab35c71
a1518063243acd4dfe96b66e3f2ec801 | | | | | | | | | SHA256
3c8e072cd09b3834a19f81f659cc3455 | | | | | | | | |
b57443645468e05a15302932b06b05e0 7c7c6f9e6b95780f7423ff9ccf0c552a
580fa00ba5f5e60499c5c7e7d9c7f50e 8a5a7f883bdb1ee6c22c05ce71c1f288
How a typical brainwallet tool works

"correct horse battery staple" Passphrase
v v v v v v v v SHA256
c4bbcb1fbec99d65bf59d85c8cb62ee2 Private key
db963f0fe106f483d9afa73bd4e39a8a
v v v v v v v v v v v v v v v v
privateToPublic (UNCOMPRESSED) (COMPRESSED)
04 78d430274f8c5ec1321338151e9f27f4 -> 03 78d430274f8c5ec1321338151e9f27f4 Public key
c676a008bdf8638d07c0b6be9ab35c71 c676a008bdf8638d07c0b6be9ab35c71
a1518063243acd4dfe96b66e3f2ec801 | | | | | | | | SHA256
3c8e072cd09b3834a19f81f659cc3455 | | | | | | | |
b57443645468e05a15302932b06b05e0 7c7c6fae6b95780f7423ff9ccf0c552a
580fa00ba5f5e60499c5c7e7d9c7f50e 8a5a7f883db1ee6c22c05ce71c1f288
v v v v v v v v v v v v v v v v RIPEMD160
c4c5d791fc4654a1ef5 79fbfc3f34e7745860d7 Hash160
e03fe0ad3d9c598f9827 6137da68f362380c606c (used for tx)
How a typical brainwallet tool works

"correct horse battery staple" Passphrase
v v v v v v v v v v v v SHA256
c4bbcb1fbc99d65bf59d85c8cb62ee2 Private key
db963f0fe106f483d9afa73bd4e39a8a
v v v v v v v v v v v v v v v v v v v v v privateToPublic

(UNCOMPRESSED) (COMPRESSED)
04 78d430274f8c5ec1321338151e9f27f4 -> 03 78d430274f8c5ec1321338151e9f27f4 Public key
c676a008bdf8638d07c0b6be9ab35c71 c676a008bdf8638d07c0b6be9ab35c71
a1518063243acd4de96b66e3f2ec801 | | | | | | | | SHA256
3c8e072cd09b3834a19f81f659cc3455 | | | | | | | |
c7c6fae6b95780f7423ff9ccf0c552a
b57443645468e05a15302932b06b05e0 8a5a7f883b0e6c22c05ce71c1f288
580fa00ba5f5e60499c5c7e7d9c7f50e 8a5a7f883b0e6c22c05ce71c1f288
v v v v v v v v v v v v v v v v v v v v v RIPEMD160
c4c5d791fcb4654a1ef5 79fbfc3f34e7745860d7 Hash160
e03fe0ad3d9c598f9827 6137da68f362380c606c (used for tx)
v v v v v v v v v v v v v v v v v v v v v Base58Check
1JwSSubhmg6iPtRjtyqhUYYH7bZg3LfY1T 1C7zdTfnkzmr13HfA2vNm5SJJYRK6nEKyq8 Address
Brainwallets make the Blockchain a public password hash database
(What questions do we ask when a password database leaks?)
Brainwallets make the Blockchain a public password hash database (What questions do we ask when a password database leaks?)

● Were the passwords hashed?
Brainwallets make the Blockchain a public password hash database
(What questions do we ask when a password database leaks?)

- Were the passwords hashed? Yes
Brainwallets make the Blockchain a public password hash database (What questions do we ask when a password database leaks?)

● Were the passwords hashed? Yes
● Were the passwords salted?
Brainwallets make the Blockchain a public password hash database
(What questions do we ask when a password database leaks?)

● Were the passwords hashed? Yes
● Were the passwords salted? No
Brainwallets make the Blockchain a public password hash database
(What questions do we ask when a password database leaks?)

• Were the passwords hashed? Yes
• Were the passwords salted? No
• Is the hash slow to crack?
Brainwallets make the Blockchain a public password hash database
(What questions do we ask when a password database leaks?)

- Were the passwords hashed? Yes
- Were the passwords salted? No
- Is the hash slow to crack? Kinda
Brainwallets make the Blockchain a public password hash database
(What questions do we ask when a password database leaks?)

- Were the passwords hashed? Yes
- Were the passwords salted? No
- Is the hash slow to crack? Kinda
- Bonus: Cracking yields money
My original cracker

- C + OpenSSL
- Reads a file of hash160s, looks for passphrases passed on STDIN
- ~10,000 guesses/second
- I’ll feed it a bunch of wordlists I have - It’ll be fun!
My original cracker

- C + OpenSSL
- Reads a file of hash160s, looks for passphrases passed on STDIN
- ~10,000 guesses/second
- I’ll feed it a bunch of wordlists I have - It’ll be fun!
- I was not prepared for the result
WTF!?

- “how much wood could a woodchuck chuck if a woodchuck could chuck wood”
- 250 BTC - $20k at the time
- Mine for the taking, but ethics
- With great power comes great responsibility
- Don’t want someone else to steal it either
- What to do?
WTF!?  

- “how much wood could a woodchuck chuck if a woodchuck could chuck wood”  
- 250 BTC - $20k at the time  
- Mine for the taking, but ethics  
- With great power comes great responsibility  
- Don’t want someone else to steal it either  
- What to do?  
- Call Dan Kaminsky (a friend and admitted whitehat)
How to prevent a pwning

- All I have is a Bitcoin address - how to warn?
- Send “Chuck” a few cents, then take it back
  - “Oh no! Somebody must have my private key! I better move my money somewhere else while I still can!”
- Vanitygen -> address with chosen prefix
- What prefix?
- Must be short, exponentially harder w/ length
How to prevent a pwning

- All I have is a Bitcoin address - how to warn?
- Send “Chuck” a few cents, then take it back
  - “Oh no! Somebody must have my private key! I better move my money somewhere else while I still can!”
- Vanitygen -> address with chosen prefix
- What prefix?
- Must be short, exponentially harder w/ length
- My wife suggested “yoink”
Yoink!
The Plan

● Send a little money to Chuck
● Take that money back
  ○ There’s no “Balance” on an address
  ○ Just a collection of previous TX outputs
  ○ The 250BTC output is separate from the little output
● Hope Chuck worries why a “yoink” address can take money back
What actually happened

- Send a little money to Chuck
- Try to take specifically that money back
  - Not a thief!
- **Actually** draw from the 250BTC at risk
  - Took 0.00031337 BTC, sent it to “Yoink”
  - Took 249.99968663 BTC, sent it Somewhere Else
- WHERE DID THE MONEY GO!? 
- WHAT IS THIS ADDRESS!?
Change Addresses

- I thought exact change would be used
- Big, old outputs are prioritized
- Bitcoin software made up a new address
- Sent the “change” from the 250BTC to it
- Found it, sent the money back
How to prevent a pwning

● New strategy: follow the BTC
● DeepBit mining pool owner passed on my email
  ○ After convincing him of no malicious intent
● Hilarious conversation ensues
Smart people pick can bad passphrases too

- Chuck wasn’t stupid
- How many people understand how effective cracking tools are?
Meet Brainflayer

- Speed improvements - now uses libsecp256k1
- Does 130k guesses/second on my machine
- Running on EC2, $1 checks 560 million passphrases.
- With 1,000 instances, $175 will check one trillion passphrases in 9 hours
Remember this?

~28 bits of entropy

\( 2^{38} = 3 \text{ days at 1000 guesses/sec} \)

Difficulty to guess: EASY

Was it trombone? No, troubadour. And one of the Os was a zero?

And there was some symbol...

Difficulty to remember: HARD

~44 bits of entropy

\( 2^{100} = 550 \text{ years at 1000 guesses/sec} \)

Difficulty to guess: HARD

Correct horse battery staple

Through 20 years of effort, we've successfully trained everyone to use passwords that are hard for humans to remember, but easy for computers to guess.
XKCD: Not Always Right

- Brainflayer could cover that search space in less than a week on EC2
- ...and, bad guys use botnets, probably capable of checking $2^{48}$ passphrases per day
Want it faster?

- Low level optimization and fancy math are possible
- GPU acceleration is possible
- FPGA acceleration is possible
- ASIC acceleration is possible, but unlikely
  - Mining BTC ≠ Calculating Brainwallets
How does Brainflayer work?

1. Download entire blockchain
2. Extract all the unique addresses
3. Pre-process for high-speed lookups
4. Generate candidate passphrases and calculate corresponding addresses
5. Check for matching addresses from the blockchain
6. Win
Analyzing the blockchain

- 80,000,000 BTC addresses
- Bloom filter can check all addresses simultaneously, providing either “no match” or “probable match”
- Second, slower check could identify any false positives
Analyzing the blockchain

- 80,000,000 BTC addresses
- Bloom filter can check all addresses simultaneously, providing either “no match” or “probable match”
- Second, slower check could identify any false positives
- Crack multiple blockchains at the expense of just a few more false positives
Bloom filter?

- A big bitmask - 512MiB
- Brainflayer maps each address to 20 locations in the bitmask
- Set all of them to insert
- Read to check - return false on a miss
- Normally bloom filters uses hashes, but those are slow
- hash160 -> already hashed - just bitslice
Generating candidate passphrases

- Wordlists are easy to find, but phraselists require some creativity
- Scrape song lyric sites, wikiquote, project gutenberg, forums, mailing lists, reddit, etc.
- Normalize the raw data
- Apply rules (with/without punctuation, vary capitalization, with/without spaces, etc.)
Some results

“Down the Rabbit-Hole”- held about 85 BTC in July 2012

“The Quick Brown Fox Jumped Over The Lazy Dot” - held about 85 BTC in December 2011

“” - had 50BTC last week, stolen in seconds
Some more results

- “gate gate paragate parasamgate bodhi svaha”
- “The Persistence Of Memory”
- “QTC”
- “644122178”
- “8964009”
- “que me lleve la muerte”
- “one two three four five six seven”
- “it’s a secret to everybody”
- “Ph’nglui mglw’nafh Cthulhu R’lyeh wgah’nagl fhtagn”
A few more results (for the lulz)

- “my hovercraft is full of eels”
- “Interior Crocodile Alligator”
- “No need to worry, my accountant handles that”
- “tomb-of-the-unknown-soldier-identification-badge”
- “permit me to issue and control the money of a nation and i care not who makes its laws”
- “who is john galt”
- “Live as if you were to die tomorrow. Learn as if you were to live forever.”
All together...

- I looked up the peak balances on everything I cracked - it adds up to 733 BTC
- Hard to tell what was moved away safely and what was stolen
- I didn’t take any of it
Don’t be that guy. Don’t be Chuck.

- Just about anything you can come up with can be replicated by a sufficiently clever password guessing algorithm.
- If someone else came up with it, it will eventually make it into a phraselist.
- There are better ways.
Alternatives

- Electrum, which generates a 12-word phrase for you
- WarpWallet supports a salt and uses scrypt for key stretching
- Encrypted paper wallets
Determining passphrase strength

- Strength of a computer generated random key measured in bits
  - Adding a bit doubles the strength
  - Adding 10 increases about 1000x

- What if key is not computer generated, but chosen by a person trying to be random?
  - Need to determine how many “bits” of entropy
  - Several tools attempt this, notably Dropbox’s zxcvbn
  - Failure cases, including limited dictionary size
Strength of chosen passphrases

- “kwyjibo#” rated as having 42.2 bits of entropy - but it was on the Simpsons without the hash - and in big wordlists
- Bad estimates are rampant
  - “one two three four five six seven” rated as 92.9 bits “centuries to crack”
  - Microsoft study estimated average user’s password is ~40 bits (wat?)
Key stretching

- Makes passwords harder to crack by making passwords more expensive to test
  - Brainflayer tests 130K/sec
  - Stretching could make this 1/sec
- scrypt, bcrypt, sha512crypt, pbkdf2, etc
- Password Hashing Competition just announced their winner, “Argon2”
Extreme key stretching

• Hard to go past 1/sec and still be usable
• Could split the problem
  ○ Store part of the system on a hard drive
  ○ Without the part, both attacker and passphrase holder might be able to try 1/day
  ○ Default attacker does not have the “shortcut” and runs slow, default defender does and runs fast
• More research required
How to get a secure password

- Generate it randomly!
- You may not remember it, but your password manager can
- Your password manager needs a master password - and backups
- The backups could be cracked
- Doesn’t seem to be any getting around the need to remember at least one good one
A field report

- About half a dozen active thieves
- They’re sophisticated, and in competition
- They must be fast - they do that with bots
- Cracking with Brainflayer is not real-time
- Rainbow tables aren’t instant
- Lookup tables are instant
- They have big lookup tables
How I’d do it

- Disk-backed key-value store
- Truncated hash160 key, passphrase and/or private key value
- Extract hash160s from transaction, check if private key is available
- Use private key to take the funds
- Be faster than the other guys
Estimating table size

- 64 billion on a $120 4TB disk
- My probes imply at least 10 billion
- Every 5 character password I try gets swiped in seconds
- So does anything on common wordlists
- And lyrics, and stuff on wikiquote
- 6 random characters is a bit much for a table
- Brainflayer can do that for $1,300
Cryptomnemonics

- Diceware
- Electrum’s scheme
- Pronounceable passwords
- Structured generators
- Proposal for .onion URLs
- Active area of research, many others
Remember:

- Meat is predictable. Don’t get robbed.
Let’s have some fun

- So, DEFCOIN exists
- The guys from “Crack Me If You Can” generated a bunch of passwords and passphrases for me
- Brainflayer should be online at https://rya.nc/brainflayer shortly
- Get yourselves some DEFCOIN :D
- Follow me on twitter @ryancdotorg
Questions?