Measuring the Use and Abuse of Brain Wallets

Marie Vasek - University of New Mexico
Ryan Castellucci - White Ops

Introduction - Marie Vasek

- Newly minted assistant professor of computer science at the University of New Mexico
- Discovered Bitcoin in 2011; first Bitcoin-related papers in 2014
- I’m a con n00b

Introduction - Ryan Castellucci

- Cylon^WBot detection researcher for White Ops
- Doing silly things with Bitcoin since 2011
- I went to DEF CON 12 and now I feel old

Typical RSA Key Generation

- Generate a large random number (*p*) - so big special libraries are required
- Make sure the number doesn’t have any small prime factors
- Make sure the number passes probabilistic tests for primality
- Make sure that n-1 is coprime to the public exponent
- Start over if any checks fail, generate a second number (*q*) the same way
- Calculate the private exponent and several other derived values

Typical Elliptic Curve key generation

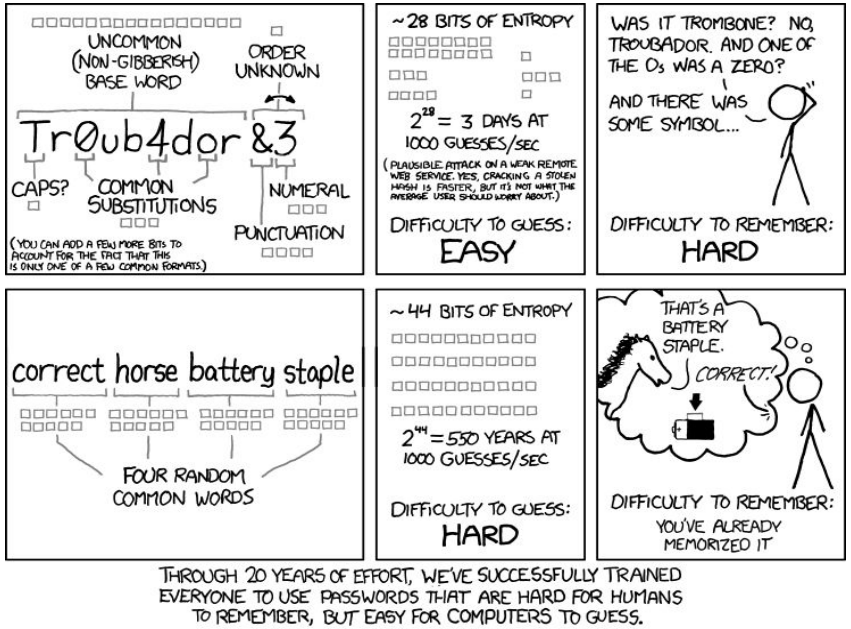
- Generate a large random number in the right range for the curve
- For Bitcoin’s curve, nearly any 256 bit value will work

A perfect storm

- A trivial key generation process to play with
- Paranoid cypherpunks worried about evil random numbers
- Substantial amounts of money on the line
- Results:
  - A simple to use tool where key = SHA256(passphrase)
  - Money piñata for password crackers

Password cracking input - XKCD Phrases

- Took intersection of lists from a few “XKCD password generator” tools
- “correct horse battery staple” ~37.8฿
- “expect pants size clue” ~2.18฿
- “earth air fire water” ~0.0012฿
- “deal iron science food” ~0.00015฿



Password cracking input - Phrases

- XKCD-style (limited wordlist)
- Cryptocurrency IRC chat logs
- Reddit comments
- Wikiquote
- Wikipedia
- BrainyQuote
- Facebook Names
- Urban Dictionary
- Song Lyrics

Password cracking input - Standard lists

- RockYou
- MySpace
- Linkedin
- Openwall
- Keyboard Patterns
- Crackstation
- Naxxatoe (cleaned up)
- Uniqpass
- Everything on SkullSecurity
- Brute force

Password cracking - Process

- Large jobs run on AWS (Thanks UTulsa!)
- NFS server (m4.10xlarge) for data files, software and output
- Spot instances (m4.2xlarge) for compute
- SQS FIFO used for job management
- Simple python script runs `subprocess.call` on whatever it gets from SQS
- One copy of script per virtual core (hyperthreading helps)

Password cracking - Job runner setup

```
# imports and credential definitions

sqsc = boto3.client('sqs',aws_access_key_id=ACCESS,\
aws_secret_access_key=SECRET,region_name=REGION)
sqsr = boto3.Session(aws_access_key_id=ACCESS,\
aws_secret_access_key=SECRET,region_name=REGION).resource('sqs')

sqsc.list_queues()
queue = sqsr.get_queue_by_name(QueueName=QUEUE)
```

Password cracking - Job runner read queue

```
def get_one_message():
    messages = queue.receive_messages(MaxNumberOfMessages=1)
    if len(messages) == 1:
        return messages[0]
    else:
        return None
```

Password cracking - Job runner loop

```
for message in iter(get_one_message,None):
    print 'RUNNING JOB:\t' + message.body
    job = json.loads(message.body)
    proc = subprocess.call(job['exec'], shell=True)

    proc.wait()

    if proc.returncode == 0:
        print "Job's done!"
        message.delete()
```

Password cracking - Example jobs

```
mp64.bin --hex-charset -1 7e -2 68 ?1?2?a?a?a?a | \
brainflayer.sh -v -o /brainflayer/results/mp_aaaaaa_x_9002_9025.flay

combinator3.bin /brainflayer/wordlists/xkcd/common/925.txt \
/brainflayer/wordlists/xkcd/common/g16.txt \
/brainflayer/wordlists/xkcd/commoncommon.txt | \
brainflayer.sh -v -o \
/brainflayer/results/common_xkcdxkcdxkcdxkcd_925g16.flay
```

Password cracking - Example jobs

```
find /brainflayer/wordlists/skull/ -type f | grep .txt | \
grep -v withcount | grep -v rockyou | xargs cat | tr -d "\r" | \
brainflayer.sh -n 5/16 -v -o /brainflayer/results/skull_5_16.flay
```

```
hashcat-cli64.bin --stdout \
/brainflayer/wordlists/myspace-rockyou-linkedin.txt -r \
/brainflayer/tools/hashcat-3.30/rules/leetspeak.rule | \
brainflayer.sh -v -o /brainflayer/results/mrl___leetspeak.flay
```

Password cracking - Random lessons learned

- Using EBS snapshots to access data files is terrible
- NFS isn’t great either, but gets the job done
- Spot pricing varies (sometimes wildly) between regions

Blockchain analysis

- First pass: use downloaded blockchain to gather transaction data for each brain wallet
- But -- attackers seem to drain too quickly for this!
- Second pass: use blockchain.info API to gather this data down to the second.
  - Sanity check using first method.

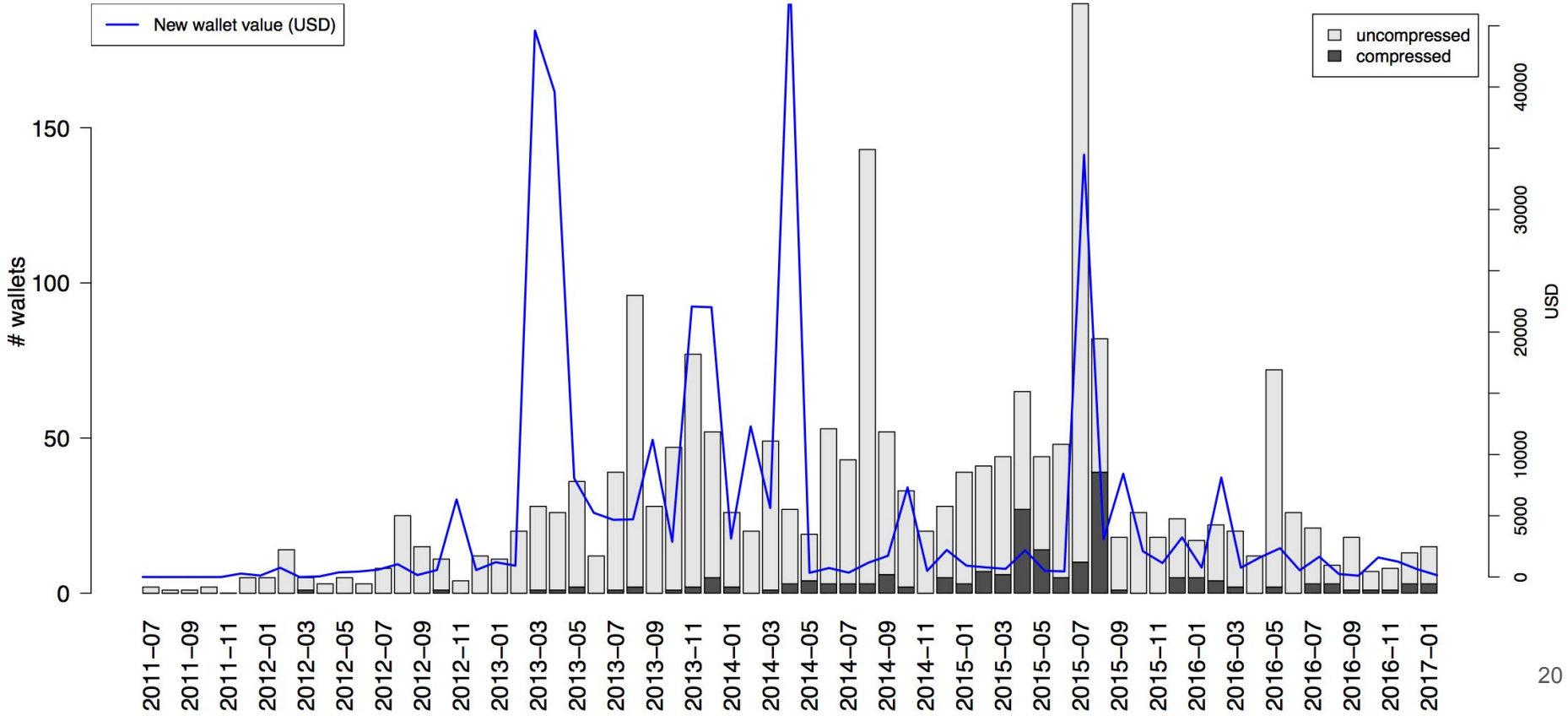
Password Corpora: 3.9 trillion Candidate Passwords

Source	# Wallets	(non-empty)	Unique	90% # drains	Total BTC	Total USD
			Word lists			
xkcd	155	3	8	8	126.94	8 857.49
Urban Dictionary	244	0	1	3	51.01	54 41.56
Password dumps	815	0	44	3	199.20	39 155.22
Industry lists	876	0	32	3	364.91	37 096.71
Facebook names	364	0	23	4	107.78	14 425.13
BitSig	235	0	71	8	1 586.78	63 818.81
Bitcoin IRC	454	1	17	6	777.52	25 355.79
Reddit	843	8	120	3	2 175.42	99 089.43
WikiQuote	281	0	3	7	113.60	17 700.88
Lyrics	438	0	17	4	270.28	19 257.41
Wikipedia	176	0	5	6	565.77	15 645.48
Llamasoft	275	0	275	3	372.42	51 799.40
			Non-word lists			
Brute Force	586	2	84	2	96.44	23 796.09
Misc	268	7	268	1	73.67	26 941.39
Overall	2 005	21	763	3	3218.65	312 591.70

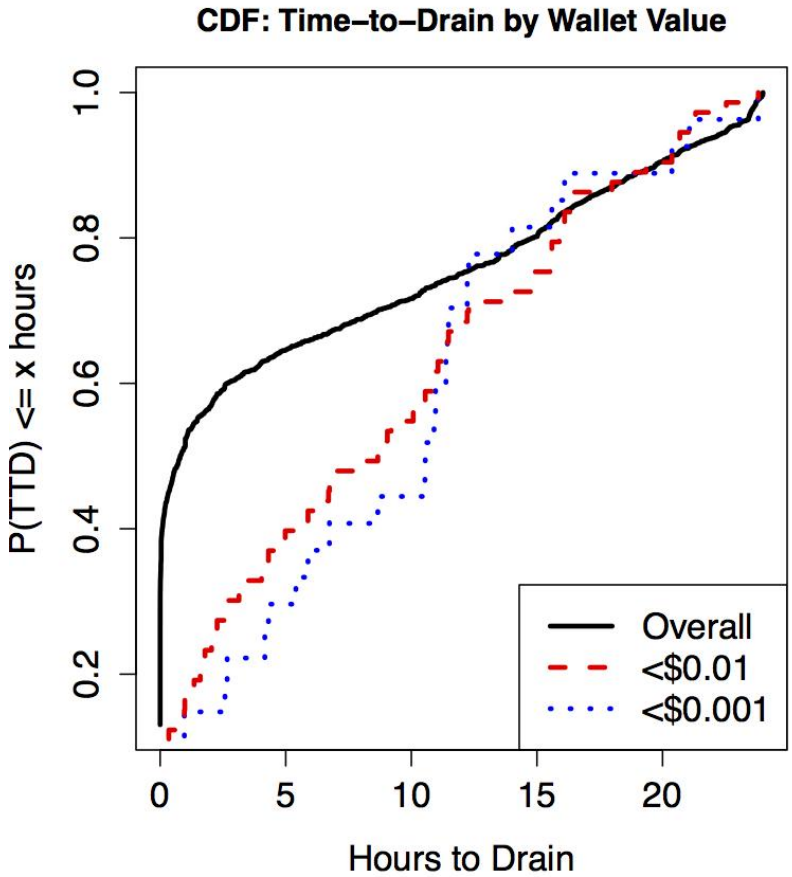
Brain Wallet Usage

- 2,005 distinct brain wallets
- 1,959 passwords and passphrases
- 3,219 BTC (approximately 312 K USD)
- Notable Passwords/phrases:
  - This string contains 0.25 BTC hiding in plain sight.
  - “”
  - bitcoin is awesome

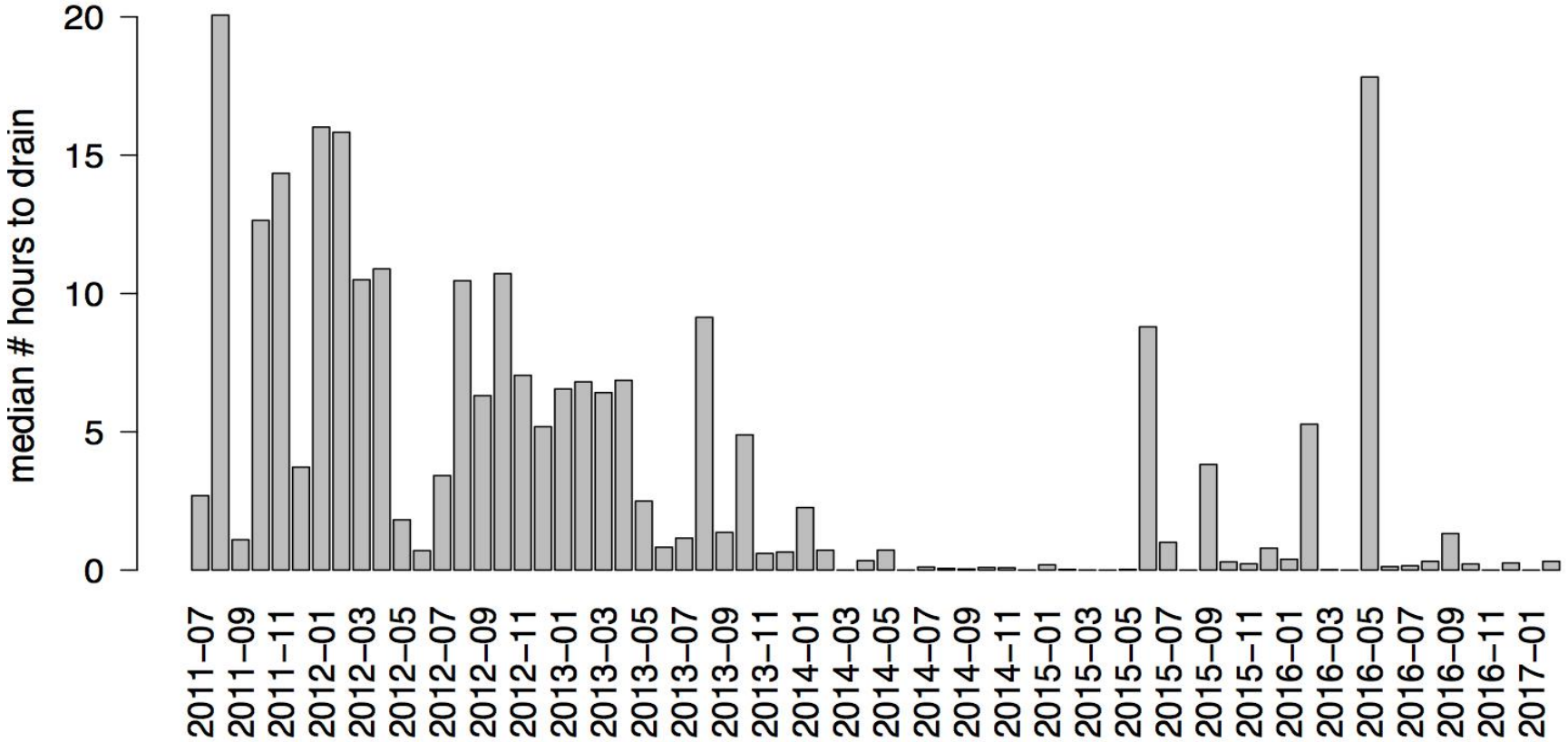
New Brain Wallet Usage by Month



Brain Drain Time



Brain Wallet Drains over Time



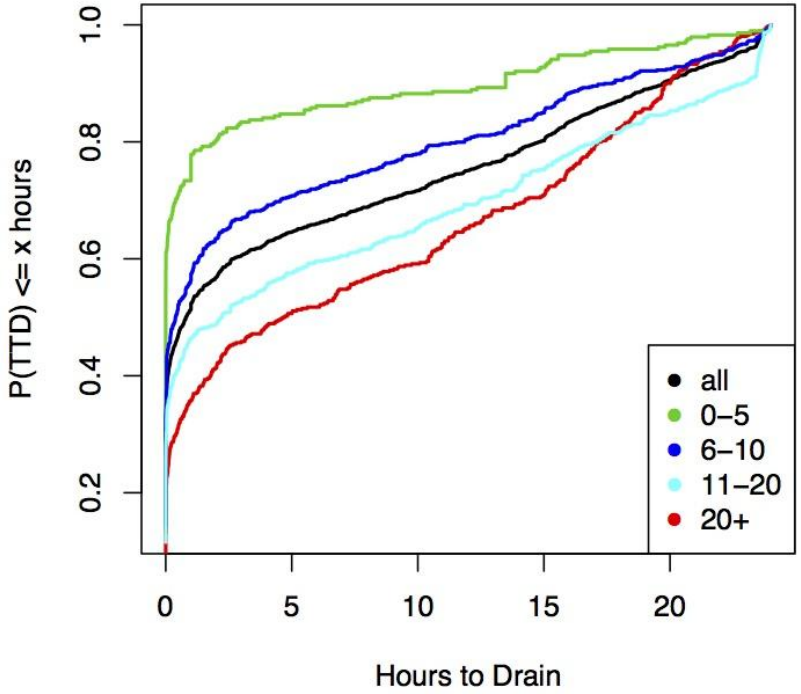
Draining: complexity

- zxcvbn: password complexity metric developed by Wheeler at dropbox.
- Tried to see if more complex passwords affected user or attacker behavior
  - Used Spearman's rank-correlation coefficient
- Found no significant effect ($p > 0.1$ in all cases)

What does this mean?

- Users don't pick stronger passwords when securing more money.
- Attackers don't prefer less complex passwords.

Draining: Passphrases vs. Passwords



Time to drain by password or passphrase length

Beyond Brain Wallets

- Bad Nonces
- Small/large value keys
- Math.random()
- User selected seeds
- Published “example” addresses with keys
- Arbitrary constants
- Blockchain data
- File hashes
- Ethercamp
- Parity Wallet
- P2SH (Multisig, weird stuff)

Beyond Brain Wallets

```
954f7d96502b5c5fe2e98a5045bca7f5e9ba11e3dbf92a5c0214a6aa4c7f2208
9c41a5f0461dbd3145aedfb838c98c9ccd41ea5cf757ae44c4cbe2734fd89c9b
452821e638d01377be5466cf34e90c6cc0ac29b7c97c50dd3f84d5b5b5470917
0000000000000000000000028bc56c889111335c23e6715a0aeb92e0adeb2e6
0000000000000000000000000000000000000000000000000031323334
ffffffffffffffffffffffffffffffebaaedce6af48a03bbfd25e8cd03640d8
000000001976a914742a9aacda1b402be98dbcb20c3f9248cda5e83a88ac7276
000000004257f209a21040d1baeb99c1ea852271238ef67a124a2c5349e1d870
020203040506070809 0a0b0c0d0e0f101112131415161718191a1b1c1d1e1f20
b1674191a88ec5cdd733e4240a81803105dc412d6c6708d53ab94fc248f4f553
0000000000000000000000000000000000000000000000000000000000000001
```

Beyond Brain Wallets

SHA256(Chrysanthemum.jpg)
Bitcoin Transaction ID (late 2014 spam tx)
65th through 128th hexadecimal digit of the fractional part of pi
0 padded hash160 of address ‘14iPehNvQRjQgDHvFdMhZbBqhytS2teZVu’
NULL padded ASCII ‘1234’
n (order of curve) - 105
chunk of raw data from some random Bitcoin transaction
block hash of Bitcoin block 1234
No idea why this string of bytes was on my hard drive
SHA256(bitcoin.pdf)
¯_(ツ)_/¯

Pay-to-Script Hash Attacker

- 139 Bitcoin wallets found
- First seen in March 2012
- 43 still undrained -- total of 0.165 BTC or 421 USD left
- Others all drained in under 24 hours
- Median drain time 9 hours
- 13 of these attackers also drained brain wallets
  - At least 8 were spam attackers

Large Bitcoin Collider

Large Bitcoin Collider

- Distributed incremental key search
- Only searching for keys with a balance
- Admin *claims* sweeping funds as “lost and found” legal in his country.
- GPU accelerated
- Scary perl client that executes arbitrary arbitrary code from server :-)
- Actually hit some keys with small balances, likely original keys, not collisions
  - `0000000000000000000000000000000000000000000000006cd610b53cba`
  - `000000000000000000000000000000000000000000000000e09c93a2ec81`
  - `0000000000000000000000000000000000000000000000005e1667c899783`
- Also has been finding keys for a “puzzle transaction”
- Won’t steal anyone’s coins unless they used a deliberately bad RNG

Drains by Bitcoin Mining Pools

- 8 mining pools
- 157,710 drains
- 88,708 transactions
- September 2013 - May 2017
- 15 brainwallets
- 1.58 BTC (437 USD)

- Drain to transaction fees, rather than to an address

But why?

- Clean up unspent transaction outputs (UTXOs)
- Unneccesary UTXOs eat valuable diskspace on nodes
- Bitcoin network “stress test”
  - Advocating for bigger block sizes by causing mass disruption
  - Create a 30 day backlog of transactions
  - June 13- August 28, 2015
  - 15 brainwallets
  - 20,172 transactions
  - 6.6 BTC

Thanks!

- Joseph Bonneau, Cameron Keith, and Tyler Moore
- Filippo Valsorda
- “Llamasoft” https://github.com/llamasoft

Why Brain Wallet Is The Best ?

Source: http://blog.ether.camp/post/138376049438/why-brain-wallet-is-the-best

Questions?

https://secon.utulsa.edu/vasek/ @mjvasek

https://rya.nc/ @ryancdotorg