Executive Summary: DDoSCoin

History

- Proof of Concept for useful-proof-of-work cryptocurrency
- Introduced in 2016 via a paper similar to Bitcoin's white paper (https://benvds.com/papers/woot16-paper-wustrow.pdf)
- Designed by Eric Wustrow, an assistant professor at University of Colorado at the time along with Benjamin Vandersloot, a PhD student at University of Michigan at the time
- Never actually deployed, but was implemented here: https://github.com/ewust/DDoSCoin

Key Features

- Miners prove that they contributed to a DDoS attack on a target TLS server
- Anyone can submit "bounties", which include an address for a TLS server to target and a reward sum that can be claimed by a miner that proves they performed the attack

Pros/Cons

- Proof of work is evil- DDoS attacks are malicious
- Never really caught on
- Only works on TLS versions 1.2 and lower

+ Proves that useless hash puzzles aren't the only way do proof of work

Summary

DDoSCoin is a really interesting concept, but never took off. In it, special transactions are allowed that publish bounties for DDoS attacks on TLS servers. Then, miners perform proof of work to fill the bounties and claim the rewards. To prove that they performed the requested number of requests to the target server, a miner submits a signature collected from the target server as part of the TLS protocol. The signature supplied must lead with N bits of zeros, where 2^N is the size of the DDoS attack requested in the bounty.

This concept demonstrates a few things: an evil cryptocurrency and a useful-proof-of-work. It would be interesting to see this take off, but the truth is that there just isn't a very wide demand for public DDoS attacks, and this would probably get taken down by governments due to its evil nature.

Written Report: DDoSCoin

DDoSCoin is a decentralized system for running Distributed Denial of Service (DDoS) attacks on Transport Layer Security (TLS) servers. The system allows anyone to submit a request for a DDoS attack of any size on any provided server, while rewarding miners who participate in the attacks with a quantity of cryptocurrency proportional to the size of attack they conducted. Some of the most interesting features of this system are much like features of other cryptocurrency systems. However, at the same time, many of the features are unique and interesting to investigate. Specifically, the mechanism for proof of work reveals a large area that can be researched further. This report will first touch on the history of DDoSCoin, then the features that are similar to features of other cryptocurrencies, then discuss the features that are unique to this system. Lastly, I will point out the future research that can be conducted on this topic.

In 2016, the idea of DDoSCoin was introduced at USENIX's WOOT '16 conference. The creators were Eric Wurstrow, an assistant professor at University of Colorado Boulder, and Benjamin Vandersloot, a PhD student at University of Michigan. These two researchers worked together to come up with the idea for the system and developed a paper and a presentation for their concept system. Along with their available materials, I found a GitHub repository containing a proof-of-concept model of their system written in C. Apparently, they implemented the system, but never actually deployed and marketed it. So, nobody has ever used DDoSCoin, but a solid base for a system exists in case someone desired to launch it. However, since it was meant to be a proof of concept, this will probably never happen. Additionally, the legal issues involved would most likely get it shut down if it ever launched. DDoS attacks are malicious, and a system supporting the mass distribution of denial-of-service attacks would be considered criminal by most governments.

There are numerous cryptocurrencies that the design of DDoSCoin reflects, but the most similar one is definitely Bitcoin. Some of the key features of Bitcoin are the miners, the miner language called Script, and the general concept of the blockchain and transactions. DDoSCoin actually uses all of these concepts to essentially run a clone of Bitcoin. Transactions are created by users, signed by their private keys, and sent to miners, who then create blocks and append them to the blockchain, which is a read only ledger. The miners use proof-of-work to decide who gets to create the next official block, each having to solve a puzzle that is hard to solve but easy to verify. Both Bitcoin and DDoSCoin use these features, but DDoSCoin expands on them.

Bitcoin has a set of transaction types. These include paying someone, multisignatures, paying to a script, new block payouts, etc. DDoSCoin requires a few others that are added on top of the default Bitcoin transactions. One of the DDoSCoin specific types is submission of bounty, which includes a target server address, a number of TLS connections to be made in the attack, and a payment for whoever fills the bounty. To go along with the bounty submission transaction, the system supports a transaction for adjusting a bounty. The submitter of a bounty makes an update adjusting the number of DoS attempts to be made. When a miner verifies the transaction, it then essentially adds or removes a balance from the bounty submitter, depending on the size of the adjustment.

The rest of the distinction between DDoSCoin and Bitcoin represents the set of features that make it cutting edge. The concept of decentralized bounties for DDoS attacks is unique and implemented in an interesting fashion upon investigation. Bitcoin’s hash puzzle requires a miner to supply a nonce that combines with the block they created such that hashing the block with the nonce returns a digest lower than a target value. The idea is that it will take 2^N attempts to achieve this, where N is the number of leading zeros in the target value. Similarly, a miner in DDoSCoin must prove that they performed 2^N units of work by providing a hash smaller than the target hash with N leading zeros. However, the difference is that the hash must be found through a different means. The miner must start up a TLS connection with the destination server, then use the signature (encrypted version of hash of communication) to obtain a hash less than the target. Then, someone can verify a success by checking the encrypted hash with the public key (publicly available) of the TLS server that was targeted. This methodology is very interesting to me and seems very ingenious.

The concept of DDoSCoin reveals some valuable future research that can be done. It demonstrates that proof-of-work does not need to be a useless hash puzzle. The miners can instead use their resources to perform work that benefits someone, like in the case of performing DDoS attacks. Although the DDoS attacks are not necessarily moral, and could be considered evil, the idea still brings to light other possibilities. I know there are some types of cryptocurrency that use protein folding or gene sequencing simulations as proof of work. To me, it seems that the future lies more in smart contracts, a system where any code can be run on miners. It would be interesting to see a purer distributed decentralized HPC system implemented with blockchain.

Overall, DDoSCoin was a very interesting idea but not all that practical. It demonstrates some important points, but lacks marketability, so it will never truly catch on. I am excited to see what the future holds in terms of useful-proof-of-work and contracting through the blockchain (like the bounty concept demonstrated in DDoSCoin). I believe that it will be less application specific, like DDoSCoin, and more customizable like Ethereum. Decentralizing computation for the good of the public is a genius idea that was nicely demonstrated by these researchers. Hopefully this concept leads to more ethical applications than mass takedowns of target servers paid by bounties in the future.

References

1. https://www.usenix.org/conference/woot16/workshop-program/presentation/wustrow

2. https://benvds.com/papers/woot16-paper-wustrow.pdf

3. https://github.com/ewust/DDoSCoin

Word count: 1008

Time spent: ~5 hours

Completed student ratings: YES

Reviewed by: Lance Parrish

Comments:

- I'm not quite sure how the miner verifies that they actually performed the attack, is there an oracle that checks to see if the server is down?
    - The miner can decrypt the response sent by the TLS server to get the plaintext hash of the nonce they sent back and forth. Then the miner hashes the nonce they were sending back and forth and verifies that it is the same as the decrypted hash.
- And other than the bounty, does it act like bitcoin for the most part?
    - Yes, it is really just Bitcoin with an alternate proof-of-work.

Self-evaluation:

- What did you like about the course?
    - I enjoyed understanding a popular topic and feeling educated about it so I can communicate my opinions with others, even those outside the technical community
- What changes would you recommend?
    - Requiring evidence of additional research outside the class material (i.e. a semester-long project) would make students actually do extra learning.
- What grade do you feel you earned and why?
    - I feel I earned a low A because I learned the material well, but failed to deeply research additional topics.