2018

An investigation into a denial of service attack on an ethereum network

Richard Greene
Edith Cowan University, rgreene0@our.ecu.edu.au

Michael N. Johnstone
Edith Cowan University, m.johnstone@ecu.edu.au

DOI: 10.25958/5c526f0b6668b
Greene, R., & Johnstone, M.N. (2018). An investigation into a denial of service attack on an ethereum network. In *proceedings of th*e 16th Australian Information Security Management Conference (pp. 90-96). Perth, Australia: Edith Cowan University.
This Conference Proceeding is posted at Research Online.
https://ro.ecu.edu.au/ism/227

AN INVESTIGATION INTO A DENIAL OF SERVICE ATTACK ON AN ETHEREUM NETWORK

Richard Greene[1], Michael N. Johnstone[1, 2]
[1]School of Science, [2]Security Research Institute, Edith Cowan University, Perth Australia
rgreene0@our.ecu.edu.au, m.johnstone@ecu.edu.au

Abstract

Apart from its much-publicised use in crypto-currency, blockchain technology is used in a wide range of application areas, from diamonds to wine. The most common application of this technology is in smart contracts in supply chain management, where assurance of delivery and provenance are important. One problem for an Ethereum consortium is the potential for disruption caused by a Denial-of-Service attack across the consortium nodes. Such an attack can be launched from a single source or multiple sources to amplify the effect. This paper investigates the impact of various Denial-of-Service attacks on an Ethereum Consortium deployed on the Azure Cloud platform. Our experiments demonstrated that a Denial-of-Service attack on some nodes can be successful. We found that an Ethereum Transaction Server is vulnerable to both Flood and Bandwidth Depletion attacks, but that Ethereum Mining Server nodes appear to be resilient to a Bandwidth Depletion attack.

Keywords
Blockchain, Denial-of-Service, Network Security, Communications, Ethereum

INTRODUCTION

In terms of the principles of information security (confidentiality, integrity and availability), availability ensures that a system can be accessed when needed and at an acceptable response speed. A Denial-of-Service (DoS) attack is the process of overloading a target system with requests to the point it is no longer able to serve legitimate clients. Specht and Lee (2004), describe two main forms of DoS attack, Bandwidth Depletion and Flood, which can be spread over a range of different nodes to form an orchestrated attack or Distributed Denial-of-Service (DDoS). In this paper we investigate the effect of DDoS attacks on a Private Ethereum Consortium network.

Peck (2017) notes the popularity of blockchain technology, cautions that such technology is being touted as the solution for many problems for which it is not suited and offers advice as to whether public or permissioned blockchains are appropriate solutions, as compared to traditional databases. Therefore, blockchains could be classified by whether they are public or private, or alternatively, by how the nodes achieve consensus. Whilst implementations differ, the fundamental ideas of transactions, blocks and consensus are common. Castellanos et al. (2017) state that the peer-to-peer (P2P) structure of blockchains highlights their essential characteristics, viz., they are: distributed, transparent, permanent and secure. Given that a blockchain is effectively a distributed ledger, the blocks could be currency transactions (e.g., Bitcoin) or smart contract elements (e.g., Ethereum), stored via a tamper-proof mechanism. In addition to the data and a time stamp, each transaction encodes a hash of the previous transaction (hence the “chain”), thereby preserving integrity so it is difficult for an intruder to modify a transaction because the hash of all prior transactions in a block would need to be re-computed and changed as well. As noted, the network of devices that hold the ledger exist in a P2P structure, such that consensus must be reached (or some other form of voting) for a transaction to be considered verified and written to a block.

Conventionally, Denial-of-Service is an attack against availability that has proven successful in other (non-blockchain) areas. For example, the Mirai attack in 2016 used Internet-of-Things devices to perform a sophisticated, resilient 1Tbit/sec DDoS attack. DDoS might also be considered an attack against integrity if the purpose is to affect the verification nodes in a blockchain implementation. In figure 1, the transaction webserver node or the Ethereum nodes may be vulnerable to such an attack. This attack might possibly be mitigated in the latter case by having a large number of verification nodes, as each node is equivalent in this P2P model. Therefore, if some nodes were to be lost due to a DDoS attack, as no single node is a key point of failure, and the network is robust, this would not unduly affect processing.

Xu et al. (2017) state that Ethereum is the most widely used blockchain that supports Turing-complete smart contracts (contrary to the legal sense of contracts, smart contracts are code artefacts that are executed during transactions and express business logic). Dinh et al. (2018) claim that Ethereum is resilient with respect to node failure but there have been some documented attacks on Ethereum systems. For example, on the 18th September 2016 the Ethereum network issued a security bulletin just as its DevCon2 conference in Shanghai began, a DoS attack against the Ethereum network using a security flaw in the Ethereum client known as "Geth" had commenced (Wilcke, 2016a). A payload message within the transaction used seemed to indicate the attack was directed directly at the attendees of the conference. A patch was developed and released within hours to address the issue (Weare, 2016). A few days later, on the 22nd of September 2016, a second DDoS attack began, this time targeting processing nodes. This attack leveraged the `EXTCODESIZE` operation code (used in the Solidity language used to express Ethereum smart contracts) with a low transactional cost but had a high processing demand (Wilcke, 2016b). The result of both attacks was to slow down processing on the Ethereum network.

For public blockchain systems it is the distributed nature of the processing nodes which provides protection from DDoS. Targeted bandwidth depletion attacks, where overloading effects a single machine will not overtly affect the other nodes on the network and ultimately when the attack is mitigated or ceases, the affected node will re-join the network and request any blocks which had not been previously received. The current physical distribution of Bitcoin and Ethereum networks is shown in table 1.

Table 1: Sample Distribution of Bitcoin and Ethereum Networks as at 10th October 2018
(Source: Bitcoin, https://bitnodes.earn.com/nodes/; Ethereum, https://www.ethernodes.org/network/1)

Rank	**BitCoin (bitnodes.earn.com)**		**Ethereum (Ethernodes.org)**	
	Country	**# of Nodes**	**Country**	**# of Nodes**
1st	United States	2333 (23.36%)	United States	5933 (42.04%)
2nd	Germany	1927 (19.29%)	China	2043 (14.48%)
3rd	China	677 (6.78%)	Canada	1068 (7.57%)
4th	France	662 (6.63%)	Germany	539 (3.82%)
5th	Netherlands	499 (5.00%)	Russian Federation	477 (3.38%)

For a Flood attack, the network is overloaded with valid transactions, which can be effective against all processing nodes. However, this type of attack is economically expensive due to the transaction fees charged per node. Any attacker who undertakes this type of attack will be required to pay a fee and as the number of transactions rise, so too will the fee (Nakamoto, 2012). If the attacker does not raise his/her fee, the processing nodes will leave the erroneous transactions on the pending transaction list or "Mempool" in favour of valid transactions offering a greater reward.

In a private consortium there are fewer nodes within a closed network. Processing nodes are pre-selected with additional security measures in place (Buterin, 2015; Buterin, 2016). Although considered partially decentralised, they could be physically co-located and perhaps even on a single network segment which greatly increases the possibility of a successful DDoS attack. The transaction system for the processing nodes is also not based on a monetary reward (as is the case with public blockchains). Processing transaction charges has no value outside of the consortium, which lowers the economic barrier for an attacker. This paper implements a private blockchain in an attempt to verify these assertions. The remainder of the paper describes the experimental environment used, defines the research question and associated hypotheses and discusses the findings of the research.

EVALUATION ENVIRONMENT

We used simulation to quantify the effect of the attacks and review any countermeasures. The chosen network layout consisted of a small centralised consortium of three Ethereum servers, one which acted as a (public-facing) transaction server with the remaining two acting as processing and verification nodes.

The experiments were performed on the Microsoft Azure cloud platform using a network topology shown in Figure 1. Lines noted in the diagram indicate connections between internal and external machines, noting the four types of clients that could be expected to interact with the service, viz. desktop, mobile, an external trusted Ethereum node and a malicious actor performing the DoS attack.

The transaction node operates as the front-end, facilitating user interaction with the consortium record without the need for any infrastructure. Other external and trusted nodes are allowed to join the consortium with the appropriate authentication rights provided. Individuals can interact with the transaction node using a website hosting the Ethereum Web3 client framework.

Particular regard for the security setup on the network was undertaken in following documented good practice as presented by OWASP (2017). This reduced the attack surface to only those exposed ports that would be normally be expected for the correct operation of this type of transaction. Standard HTTP was allowed to the transaction node, Port 4000 for SSH administration of the network and Port 8545 which allows traffic to and from the external Ethereum nodes.

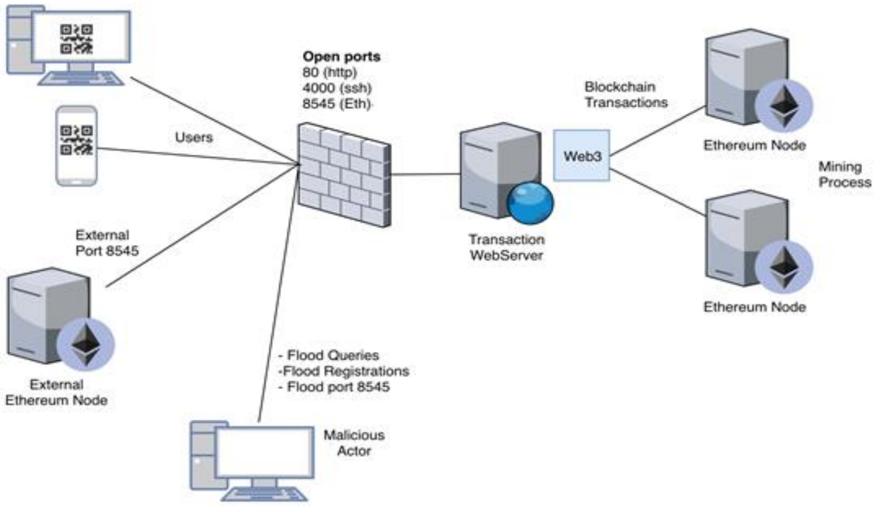

Figure 1: Consortium Network Topology

The firewall also acted as a simple load balancer, directing the HTTP traffic to the transaction server and transaction data between external and internal processing to the verification nodes.

RESEARCH METHOD

The research was designed as a set of laboratory experiments, enacted via simulation. The specific research question was:

1. Are blockchain consortium nodes vulnerable to Denial of Service attacks?

    a. Is an Ethereum Transaction Server vulnerable to a Bandwidth Depletion attack?
    b. Is an Ethereum Transaction Server vulnerable to a Flood attack?
    c. Is an Ethereum Mining Server vulnerable to a Bandwidth Depletion attack?

The hypotheses supporting the research questions are listed in **Error! Reference source not found.**. Suitable experiments were designed to test the hypotheses, the results of which are discussed in the next section.

Table 2: Hypotheses derived from research questions

Hypotheses
H_1: An Ethereum Transaction Server node is vulnerable to a Bandwidth Depletion attack.
H_2: An Ethereum Transaction Server node is vulnerable to a Flood attack.
H_3: An Ethereum Mining Server node is vulnerable to a Bandwidth Depletion attack.

ANALYSIS AND DISCUSSION

Transaction Server Bandwidth Depletion Attack

For the Bandwidth Depletion attack the Python script in figure 2 was executed from an external server and directed at the transaction node. This script simply requested the home page many times.

```
[ ... snip ...]
request = urllib2.Request(url + param_joiner + buildblock(random.randint(3,10)) +
'=' + buildblock(random.randint(3,10)))
       request.add_header('User-Agent', random.choice(headers_useragents))
       request.add_header('Cache-Control', 'no-cache')
       request.add_header('Accept-Charset', 'ISO-8859-1,utf-8;q=0.7,*;q=0.7')
       request.add_header('Referer', random.choice(headers_referers) +
buildblock(random.randint(5,10)))
       request.add_header('Keep-Alive', random.randint(110,120))
       request.add_header('Connection', 'keep-alive')
       request.add_header('Host',host)
       try:
                    urllib2.urlopen(request)
       except urllib2.HTTPError, e:
                    #print e.code
                    set_flag(1)
                    print 'Response Code 500'
                    code=500
       except urllib2.URLError, e:
                    #print e.reason
                    sys.exit()
       else:
                    inc_counter()
                    urllib2.urlopen(request)
       return(code)
[ ... snip ...]
```

Figure 2: Python Code used in Bandwidth Depletion Attack

When executed, the public-facing transaction server became overloaded with requests and ceased serving pages to the user or returned partial results. This can be seen from the traffic volume shown in figure 3, captured on the server for the duration of the attack phase. Although this attack did not adversely affect the Ethereum processing nodes, it did prevent normal operation of the application and would have prevented users from interacting with the website.

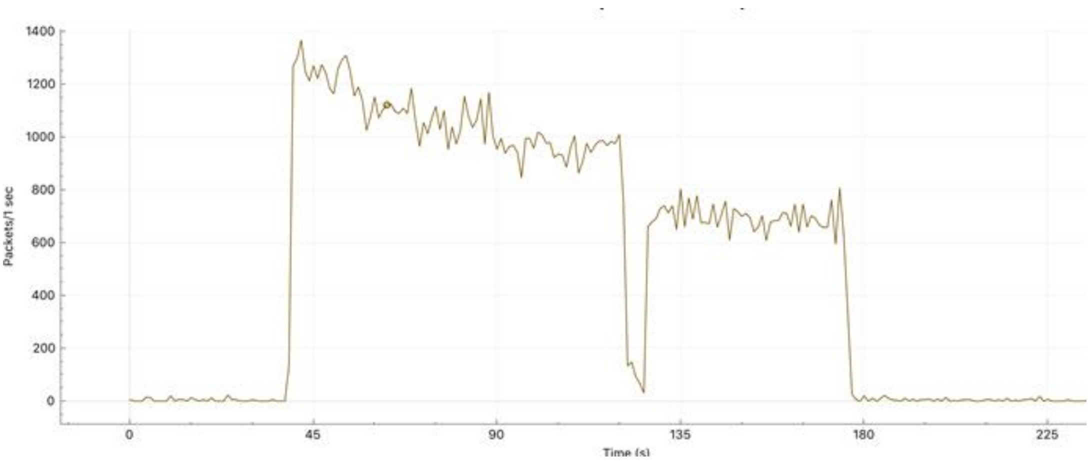

Figure 3: Packet Trace during Bandwidth Depletion Attack

The result of this experiment supports H_1 (that an Ethereum Transaction Server is vulnerable to a Bandwidth Depletion attack).

Transaction Server Flood Attack

For the transaction server flood attack the script listed in figure 3 was applied to the transaction server however this time attempting to retrieve several smart contracts and wallets on the network. This is functionality provided by the transaction server via the Web3 Framework.

The result was a sharp increase in the traffic being sent to the internal network, which overloaded the website and the transaction server failed to process additional requests. The processing nodes continued without issue as the requests failed to have an impact on the servers operating the blockchain.

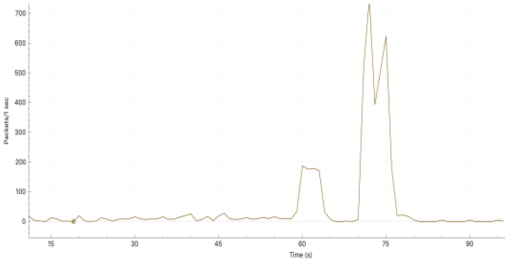

Figure 4: Transaction Server Packet Trace.

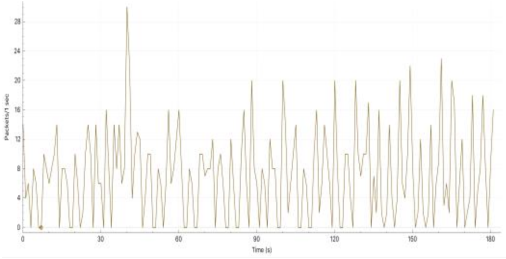

Figure 5: Processing Node Packet Trace.

The result of this experiment supports H_2 (that an Ethereum Transaction Server is vulnerable to a Flood attack).

Mining Server Bandwidth Depletion Attack

The only open port to the processing servers was via the standard Ethereum port 8545 (SANS, 2018), therefore the experiment involved sending a large number of frames to that port and verifying the effect on the processing servers. Due to permission restrictions, frames sent to the processing servers failed to have any impact on the effective operation of the network. Additionally, external nodes were able to connect to the network without issue and the external clients could create accounts and interact with the contracts. The result of this experiment did not support H_3 (that an Ethereum Mining Server node is vulnerable to a Bandwidth Depletion attack).

The research question asked, “Are blockchain consortium nodes vulnerable to Denial of Service attacks?” In relation to blockchain-based systems (such as Ethereum), vulnerabilities introduced by lack of availability could cause significant risk in a trusted network, where transaction integrity is paramount. Results from the first two experiments show that a DoS attack can effectively deny service and result in loss of message transmission between devices on the consortium network. With the acceptance of both H_1 and H_2, this paper argues that the use of vulnerable protocols increases the ability of cyber criminals to hamper the availability of critical systems operating in a network.

CONCLUSION

This research set out to examine a specific security flaw in blockchain nodes (that they could be vulnerable to Denial-of-Service attacks). There was some evidence that this type of attack could be successful. The experiments undertaken suggest that Ethereum transaction server nodes are vulnerable to a specific type of Denial-of-Service attack, answering Research Questions 1a and 1b. Ethereum mining server nodes, however, appear to be resilient to a Bandwidth Depletion attack.

With a consortium-based blockchain, to increase the probability of a successful external attack, the target should be the transaction server rather than processing nodes. Therefore, the code used to interact with the Web3 framework would need to be designed with DoS/DDoS mitigations in mind. In future work, we intend to evaluate the security properties of the Solidity codebase used in the system, using both manual methods and automated tool support. Although a large-scale co-ordinated DDoS attack against the transaction server would be difficult to defend, some approaches might be to rate limit inward edges of traffic on the network (Kumar, Joshi, and Singh, 2006) or use virtualisation and dynamic scaling of infrastructure (Riteau, 2011) to reach a point where malicious traffic can be absorbed or deflected. Testing implementations derived from these approaches is also an area worthy of future research.

Additional blockchain attacks, such as the 51% attack, where a majority of processing nodes are hacked preventing new transactions from gaining confirmations (Park and Park, 2017) would still be hypothetically possible, but as membership of the Consortium is controlled and members undergo greater scrutiny this is unlikely to be a factor.

REFERENCES

Buterin V. (2015). On Public and Private Blockchains, Ethereum.Org, retrieved on 11th October 2016 from https://blog.ethereum.org/2015/08/07/on-public-and-private-blockchains/

Buterin V. (2016). Transaction spam attack: Next Steps, Ethereum Blog, retrieved from https://blog.ethereum.org/2016/09/22/transaction-spam-attack-next-steps/ on 6th September 2018

Castellanos, J.A.F, Coll-Mayor, D. and Notholt , J.A. (2017). Cryptocurrency as Guarantees of Origin: Simulating a Green Certificate Market with the Ethereum Blockchain. Proc. 5th IEEE International Conference on Smart Energy Grid Engineering. pp. 367-72.

Dinh, T.T.A., Liu, R., Zhang, M., Chen, G., Ooi, B.C. and Wang, J. (2018). Untangling Blockchain: A Data Processing View of Blockchain Systems. IEEE Transactions on Knowledge and Data Engineering. 30(7). pp. 1366-85.

Kumar K., Joshi R.C. and Singh K, (2006). An Integrated Approach for Defending Against Distributed Denial-of-Service (DDoS) Attacks, Indian Institute of Technology Roorkee, retrieved on 10th October 2018 from https://pdfs.semanticscholar.org/cd24/ea2c151c5d04dd12b35f2902acebf96bf66a.pdf

Nakamoto S. (2012). Bitcoin: A peer-to-peer electronic cash system, Bitcoin.org,
retrieved on 10th October from https://bitcoin.org/bitcoin.pdf

OWASP. (2017). What is Attack Surface Analysis and Why is it Important?, Attack Surface Analysis Cheat Sheet, retrieved on 2nd September 2018 from https://www.owasp.org/index.php/Attack_Surface_Analysis_Cheat_Sheet

Park, J.H. and Park J.H. (2017). Blockchain Security in Cloud Computing: Use Cases, Challenges, and Solutions, *Symmetry*, 9(8), 164.

Peck, M.E. (2017). Do You Need a Blockchain? IEEE Spectrum. 54(10), pp.38-39,60.

Riteau P. (2011). Building Dynamic Computing Infrastructures over Distributed Clouds. Proc. First International Symposium on Network Cloud Computing and Applications, IPDPS 2011, Toulouse, France.

SANS, (2018), Port 8545 (tcp/udp) Attack Activity, Internet Storm Center, retrieved on 5th October 2018 from https://isc.sans.edu/port.html?port=8545

Specht S. and Lee R. (2004). Taxonomies of Distributed Denial of Service Networks, Attacks, Tools, and Countermeasures. Proceedings of the ISCA 17th International Conference on Parallel and Distributed Computing Systems, September 15-17, 2004, The Canterbury Hotel, San Francisco, California, USA.

Wilcke J. (2016a). Security alert – All geth nodes crash due to an out of memory bug, Ethereum.org retrieved on 10th October 2018 from https://blog.ethereum.org/2016/09/18/security-alert-geth-nodes-crash-due-memory-bug/

Wilcke J. (2016b). The Ethereum network is currently undergoing a DoS attack, Ethereum.org, retrieved on 10th October 2018 from https://blog.ethereum.org/2016/09/22/ethereum-network-currently-undergoing-dos-attack/

Weare K., (2016). Ethereum Security Alert Issued, Ethereum Foundation Responds with “From Shanghai, With Love”, InfoQ retrieved on 10th October 2018 from https://www.infoq.com/news/2016/09/Ethereum-DOS-Attack

Xu, X., Weber, I., Staples, M., Zhu, L., Bosch, J., Bass, L., Pautasso, C. and Rimba, P. (2017). A Taxonomy of Blockchain-Based Systems for Architecture Design. Proc. 2017 IEEE International Conference on Software Architecture. pp.243-52.