International Journal of Research in Engineering, Science and Management
Volume-1, Issue-10, October-2018
www.ijresm.com | ISSN (Online): 2581-5792

DDoS Mitigation Using Blockchain

J. Dheeraj[1], S. Gurubharan[2]
[1,2]*Student, Department of Computer Science, SRM Institute of Science and Technology, Chennai, India*

***Abstract*—The rapid growth in the number of insecure portable and stationary devices and a large increase in Internet traffic makes Distributed- Denial- of -Service a top security threat. Existing defensive mechanism lack resources and flexibility to cope with the attacks themselves. Emerging technologies like block chain and smart contracts can be used for the mitigation of DDoS attacks as it allows for the sharing of attack information in a fully distributed and automated fashion. In this paper, an architecture is designed combining these technologies introducing new opportunities for an effective DDoS mitigation. This paper presents the architecture and design of a collaborative mechanism using blockchains and smart contracts. The objective is to create an automated and easy to manage mechanism for DDoS mitigation.**

***Index Terms*— DDoS, blockchains ,ethereum, smart contracts.**

I. Introduction

DDoS attacks have a simple goal of interrupting or suspending internet services for various motivations ranging from personal interests, business tricks etc. A very large DDoS attack was detected in the GITHUB website very recently. Besides frequency the strength and the duration of the attack is also growing rapidly making DDoS attacks more effective. This paper proposes an infrastructure of blockchains and smart contracts which provide the required mechanism without the need to maintain development complexities of such a new protocol. The development of a DDoS attack is mainly dependent on the number of BOTS. By exploting legal srvices on the device the power of a DDoS attack is amplified.A blockchain is a decentralized database consisting of crytographically secured units called 'blocks'. A blockchain keeps growing as data is entered at the end of the chain. The most widely implemented blockchain is Bitcoin. The most popular application for blockchains is crypto currencies. With blockchain technology each page in a ledger of transaction forms a block. That block has an impact on the next block or page trough cryptographic hashing.In other words,when a block is completed, it creates a unique secure code, which ties into the next page or block, creating a chain of blocks, or blockchain. The most remarkable feature about blockchain is that it increase the capacity of the whole network. Even in the highest level financial systems are getting hacked. Bitcoin on the other hand has never been hacked. Blockchain technology has a better security because there is not even a single chance of shutting down of the system.

A. Ethereum

Ethereum is a blockchain protocol inspired from Bitcoin, but not only allows for sending and receiving of tokens, but also offers a scripting language called solidty which allows anyone to write programs which can run on blockchains. Games like Tic-Tac-Toe or Poker are applications that run on Ethereum. Ethereum provides a decentralized Turing complete virtual machine known as the Ethereum Virtual Machine (EVM).It is used for executing scripts using an international network of public nodes. Ether can be transferred between accounts and used to compensate participation mining nodes for computations performed. Ethereum is open sourced.

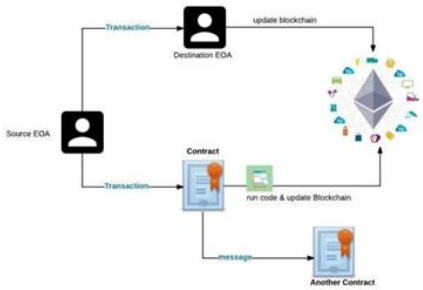

Fig. 1. Working of an ethereum

B. Smart Contracts

Smart contracts are contracts that help any form of transaction without the interference of third party users. Ethereum smart contracts allows for storage of binary information. They help in transactions that mutate the storage. By writing the code, the smart contract creator can control permissions of the users and the conditions and behaviour of the mutations.

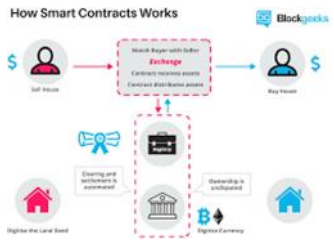

Fig. 2. Working of smart contracts

II. Background

Smart contracts are a piece of software made to facilitate the negotiation or performance of contract, being able to executed,

verified or enforced on its own. A smart contract alone is not "smart" as it needs an infrastructure that can implement, verify, and enforce, the negotiations or performance of a contract by particular computer protocol. It has gained attention in the context of blockchains that provide a fully decentralized infrastructure to run, execute, and verify such smart contracts. Therefore, smart contracts need to run on a blockchain to ensure its permanent storage and obstacles to manipulate the contracts content. A node participating in the blockchain runs a smart contract by executing its script, and storing the contract and its result in a block. Although the Bitcoin blockchain was the first fully decentralized distributed ledger, it is primarily designed for transfer of digital assets, and it is not Turing-complete. Such a Turing-complete contract language allows defining rules to allow or block IP addresses that can be interpreted by an SDN controller. While several projects try to address these issues, the Ethereum blockchain is the most popular that supports a Turing-complete contract language, empowering more sophisticated smart contracts. In Ethereum, smart contracts run in a sand-boxed Ethereum Virtual Machine (EVM) and every operation executed in the EVM has to be paid for to prevent Denial-of- Service (DoS) attacks.

SDN characteristics provide better network visibility by decoupling the control plane from the data plane and by the centralized management to perform tasks such as network diagnosis and troubleshooting. In addition to SDN, the OpenFlow protocol leverages network management by providing a programmable and standardized interface between the data plane and the control plane. It has been recognized that the decoupling of the data plane and the control plane makes SDN a promising solution to enable the enforcement of customizable security services and policies. Various SDN-based solutions have been proposed to deal with DDoS attacks. A survey on these issues is provided in. However, each security/concern category can be sub- divided in fine grained aspects. (eg., authentication, integrity, network communications). In the following are presented mainly research efforts addressing DDoS attacks in SDN networks.

To analyze the impact of DDoS attacks on network performance, the works in and have shown how such attacks may impact on several parameters like the control plane bandwidth (i.e., controller-switch channel), latency, switches flow tables and the controller performance. Other works as and use the SDN capabilities to implement schemes that allow to detect and mitigate DDoS attacks through packet analysis and filtering. These solutions reduce the impact of attacks, but they may cause an overhead in the flow- tables and the SDN performance issues as proposed in (e.g., flow-tables, and controller overloading). Furthermore, they also do not consider DDoS attacks and the collaboration with AS customers as.

SDN-based solutions allow greater agility to openforce decisions that require a global network view. Therefore, infra-domain security policies and mechanisms to prevent and react to DDoS attacks can be made agiler. By combining the intra-domain capabilities provided by SDN and the inter-domain advantages provided by blockchains and smart contracts, the efficiency to mitigate DDoS attacks in both inter-and intra-domains can be improved.

III. Related Works

There are four broad categories of defense against DDoS attacks according to 1) attack prevention, 2) attack detection, 3) attack source identification and 4) attack reaction.

1) Tries to prevent attacks before they become a problem, i.e., as close to the sources as possible. The obvious method to achieve this for amplified or reflected attacks is for the access provider to filter spoofed packets;
2) Can be a difficult task since certain attacks makes themselves as legitimate user traffic or use various traffic types. Due to this complexity, it can be hard to make a confident decision in traffic is part of an attack or special user behavior, e.g., flash crowd.
3) Is applied after an attack was detected. This step is important to efficiently contain or re- route the attack as close to its source as possible.
4) The final step involves taking concrete measures against the attack. The better the result from the more efficiently this can be done

Among the collaborative DDoS mitigation techniques, there are two main approaches using resource management to react against bandwidth attacks. The first takes effect within the victims ISP, i.e., the AS, both techniques apply traffic classification and define specific actions for those classes. Both customer and AS resource management schemes need to classify traffic into several types, and then treat them differently. However, it is rather difficult to give an accurate classification as DDoS attacks can mimic any legitimate traffic. In this regard, some sophisticated techniques can be implemented to classify traffic, but a unified reaction strategies implemented both at the AS and the customer can be more efficient than applying just one.

Other works exist for cooperative defense against DDoS attacks. However, it is still an open issue since DDoS attacks are growing in scale, sophistication, duration and frequency. The IETF is currently proposing a protocol called DOTS (DDoS Open Threat Signaling) covering both intra-organizationand inter-organization communications to advertise attacks. The protocol requires server and clients DOTS agents, which can be organized in both centralized and distributed architectures to advertise black and whitelisted addresses. A DOTS client should register to a DOTS server in advance sending provision and capacity protection information and be advertised of attacks. Then, the DOTS protocol is used among the agents to facilitate and coordinate the DDoS protection service as a whole. Also, a similar approach to the IETF proposal is presented in. The authors use a similar architecture but using an advertising protocol based on FLEX

(Flow-based Event exchange) format, which is used to simplify the integration and deployment of the solution and facilitate the communication process between the involved domains.

The proposed standard advertises the need for defensive measures in anticipation of or response to attack. The main drawback compared to the approach presented herein is the requirement of additional infrastructure requiring trust and collaboration between ISPs. A collaborative defense approach using VNF (Virtual Network Functions) is presented in. The authors propose a cooperation between domains that implements VNFs to alleviate DDoS attacks by redirecting and reshaping excessive traffic to other collaborating domains for filtering. In a gossip-based communication mechanism is proposed to exchange information about the overall observed attacks. The system is built as a peer-to-peer overlay network to disseminate attack information to other listening users or systems rapidly.

A similar approach was presented in, formalizing a gossip-based protocol to exchange information in overlay network using intermediate network routers. A different approach is presented in, which proposes a collaborative framework that allows the customers to request DDoS mitigation from ASes. However, the solution requires an SDN controller implemented at customer side interfaced with the AS, which can change the label of the anomalous traffic and redirect them to security middle-boxes. In the approach presented in this paper customers and ISPs can take action to mitigate an attack by interfacing directly with a blockchain providing the necessary trust.

Instead of making use of an existing infrastructure such as the blockchain and smart contracts, approaches mentioned above proposes the development of specific gossip-based protocols. In this sense, the deployment and integration of such solutions become complex since existing solutions need to be modified to support these protocols. The IETF proposal focuses on standardizing a protocol to facilitate its deployment. However, its implementation complexity still exists in distributed and centralized architectures to support the different types of communication. Instead, some of the requirements can be inherited from the natural characteristics of blockchains, smart contracts, and SDN, avoiding the complexities of development and adoption of new protocols.

IV. Proposed System Design

A senario is presented in Fig. 3, illustrating the system architecture. A web server based at AS C is under a DDoS attack from devices hosted at various domains (ASes, A, B and C).With a non-collaborative DDoS mitigation approach, the web server relies on defensive mechanisms that are implemented at the AS where it is allocated, which in many cases may be distant from the origin of the attack and therefore overloading the server domain with traffic. The participants of the defense first need to create a smart contract, i.e. register based. Thus when an attacker overloads the web server the customer or the AS under the attack must store the IP addresses of the attackers in the smart contract. The Ethereum creates a new block every 14s, so the subscribed ASes will receive the updated list of IP addresses to be blocked. It will then confirm the originality of the attack of the attack by analyzing the traffic statistics and verifying the authenticity of the target's address.

Once the ASes retrieve the list of attackers and confirm the attack, different mitigation strategies can be triggered according to the security policies and mechanism available in the domain. In scenarios involving multiple domains, once collaborative defense nodes receive information of the attacks, these can apply mitigation operations in agreement with the security policies. In this sense a proper mechanism is required to prevent domains from abusing cooperative defense.

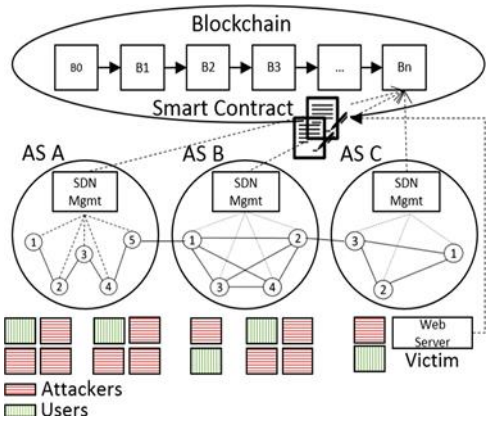

Fig. 3. Application scenario

As DDoS attacks continue to rise and vary in patterns, the need for coordinated responses are also necessary. However, it is important to note that only the collaboration between ASes is an additional approach to existing defense methods. The architecture composed of three parts:

- *Customer:* They report the whitelist and blacklist IP addresses to the Ethereum via smart contracts. Blacklist IP addresses are spammed whilelists refers to addresses that are not spammed.
- *ASes:* They publish the whitelist and blacklist IP addresses. They retrieve the lists containing the published IP addresses and implement DDoS mitigation mechanisms.
- *Blockchain/ smart contracts:*The public Ethereum blockchain which runs the Solidity smart contracts, which comprises the logic to report IP addresses in the blockchain. The architecture is built on the following principles:

1) DDoS detection and mitigation counter measures are provided as on-demand services by either the ASes or third party services.
2) To report and receive the information of the attack it is necessary to dedicate a node connected to the blockchain.

3) To effectively aid attack responses, Blockchain DDoS Mitigation modules are running on the entites such as customer and ASes reporting IP addresses and listening to the blockchain.
4) Only customer or ASes with the proof of ownership of theri IP addresses may report addresses to the smart contracts;
5) Different domains implement different security policies as well as different underlying management systems. Once notified of a DDoS attack in which the customer confirms its authenticity counter measures are defined with respect to the domain security policies.

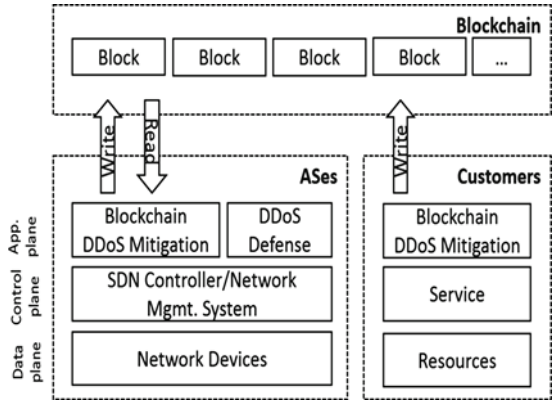

Fig. 4. Proposed system architecture

To mitigate DDoS attacks different techniques can be used which involves analyzing internet traffic using diiferent algorithms for attack detection and filtering them. The collaborative approach decreases the necessity of such algorithms in the detection phase using information from other domains. Any domain participating must create a smart contracts identified with the IP addresses and the range of IP addresses certified by an authority. Then the smart contract is registered so that participation can easily be tracked and thus relevant smart contracts can be identified.

V. Discussion

The use of Ethereum Virtual machine allows for multiple domains involved in an attack scenario to invoke functions in smart contracts which reports attacks or maintains a list of trusted IP addresses to be operating in case of an attack. The support of blacklist or whitelist IP addresses is the decision that depends on the security policies of the particular domain. Therefore smart contracts have been developed in such a way to support both types of lists using a flag indicating which type of address it is. The existing and distributed storage infrastructure reduces the complexity in the development of the approach as it supersedes the design and standardization process of a gossip- based protocol. Also, the EVM smart contracts support in a decentralized and native way the logic to control who is reporting the attack and who are the attacker.

Through a high level comparison with the ongoing IETF proposal, instead of making use of an existing infrastructure such as blockchain and smart contracts, IETF proposes the development of such protocol with several requirements to be deployed in a distributed architecture. In this sense the protocol development becomes complex since it must be deployed in a centralized architecture to support different types of communication. Instead it can be argued that some of the requirements can be inherited from the characteristics of the blockchains and smart contracts. This avoids complexity in the development and adoption of new protocols.

However, this smart contracts works well only for a small number of attacks, while for a large number of attacks the approach is rather costly. Therefore to keep the complexity of the architecture low only the data of the IP address must be stored in the smart contracts. The cost of adding 50 source IP address in a freshly deployed smart contract is 9.3 USD, while 100 IP addresses cannot be stored in one contract.

VI. Conclusion

This paper proposes a collaborative architecture using smart contracts and blockchains to enable DDoS mitigation across multiple domains. As a distributed and primarily public storage, the blockchain determines a straightforward and efficient structure to develop a collaborative approach towards DDoS attack mitigation. The proposed architecture can be considered as an additional security to already existing techniques.

It can be combined with existing solutions to reduce the DDoS attacks. Coupled with current solutions, the DDoS detection and mitigation overhead process comprising multiple domains can be reduced. The architecture enables ASes to deploy their DPS and generate added value for their customers without transferring control to their network to a third party. Future work will help in investigating ways to compress the list eg. With bloom filters, and its advantages and disadvantages.

Acknowledgement

This research was supported by our teachers, friends, parents supervised by the SRM Institute of Science and Technology.

References

[1] Akamai: How to Protect Against DDoS Attacks - Stop Denial of Service (2016). https://www.akamai.com/us/en/resources/protect-against-ddos-attacks.jsp. Accessed 10 Jan 2017
[2] Bocek, T., Stiller, B.: Smart Contracts - Blockchains in the Wings, pp. 1–16. Springer, Heidelberg. Tiergartenstr. 17, 69121, January 2017
[3] CloudFare: Cloudflare advanced DDoS protection (2016). https://www.cloudflare.com/static/media/pdf/cloudflare-whitepaper-ddos.pdf
[4] Dao, N.N., Park, J., Park, M., Cho, S.: A feasible method to combat against DDOS attack in SDN network. In: 2015 International Conference on Information Networking (ICOIN), pp. 309–311, January 2015
[5] Dridi,L., Zhani, M.F.: SDN-guard: Dos attack mitigation in SDN networks. In: 2016 5th IEEE International Conference on Cloud Networking (Cloudnet), pp. 212– 217, October 2016
[6] Fund, E: Ether unit converter. http://ether.fund/tool/converter

[7] Jonker, M., Sperotto, A., van Rijswijk-Deij, R., Sadre, R., Pras, A.: Measuring the adoption of DDoS protection services. In: Proceedings of the 2016 ACM on Internet Measurement Conference, IMC 2016, Santa Monica, California, USA(2016)
[8] Kandoi, R., Antikainen, M.: Denial-of-service attacks in openflow SDN networks. In: 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), pp. 1322–1326. IEEE (2015).
[9] Kreutz, D., Ramos, F.M., Verissimo, P.E., Rothenberg, C.E., Azodolmolky, S., Uhlig, S.: Software-defined networking: a comprehensive survey. Proc. IEEE 103(1), 14–76 (2015).
[10] Mansfield-Devine, S.: The growth and evolution of DDoS. Netw. Secur. 10, 13–20 (2015).