Enabling a Cooperative, Multi-domain DDoS Defense by a Blockchain Signaling System (BloSS)

Bruno Rodrigues, Thomas Bocek, Burkhard Stiller
Communication Systems Group (CSG), Department of Informatics (IfI)
University of Zürich (UZH)
E-mail: [rodrigues,bocek,stiller]@ifi.uzh.ch

***Abstract*—Distributed Denial-of-Service (DDoS) defense systems are not capable of withstanding by themselves against large-scale attacks. Thus, coordinated protection efforts have become an attractive alternative to extend defense capabilities of a single system. However, existing DDoS signaling protocols are a bottleneck to make a coordinated and distributed defense fully operational. Blockchain technology offers an out-of-the-box solution that not only reduces the complexity of signaling DDoS attack information, but could also provide means of establishing financial incentives, for cooperation at a reduced operational cost. This work presents the Blockchain Signaling System (BloSS), a novel approach deploying hardware to simplify the signaling of DDoS attacks in a cooperative network defense system.**

I. Introduction

As Distributed Denial-of-Service (DDoS) attacks rise in frequency and scale, existing countermeasures are being outperformed effortlessly [4]. Many centralized defense systems lack hardware resources or software capabilities to detect and mitigate large-scale attacks by themselves. Although with a broader defense capacity, cloud-based protection can take away the burden of detection and mitigation, they are equally inefficient against large-scale DDoS attacks as reported in [1].

An alternative is sharing hardware and defense capabilities with other systems, in an approach called cooperative DDoS mitigation. However, existing cooperative approaches require the use of gossip-based protocols, *e.g.*, the Internet Engineering Task Force (IETF) DDoS Open Threat Signaling (DOTS) [4], that usually requires specialized hardware or software in support of its operation.

Blockchain and Smart Contracts (SC) [3] combined provide an out-of-the-box solution that can be used for signaling DDoS attacks information across multiple domains and provide the basis for related financial incentives. Thus, existing approaches using gossip-based protocols can be simplified by using an a Blockchain-based interaction scheme that is simple to deploy and operate without the need to build specialized registries, or related signaling protocols.

The work in [5] outlines the technical approach to the Blockchain Signaling System (BloSS), a novel approach to exchange information in a cooperative network defense using a consortium-based blockchain and SCs for request-confirmation messages.

II. Overview

SCs and decentralized applications (dApp) are the main components of BloSS. While SCs describe how information is exchanged among Autonomous System (AS), the dApp's contains the parameters that define how an AS interacts in a cooperative defense. Also, a consortium-based blockchain provides an intermediary level of trust (in contrast to public and private blockchains) within pre-defined ASes. Solely relying on voluntary contribution creates a favorable environment for *free riding* peers (consuming resources without contributing), and thus incentive or reward mechanisms are required. Hence, the system could use tokens[1] that are given by the operational costs to route a minimal number of IP addresses according to their security policies.

An AS creates a blockchain account (wallet) and notifies a central SC that holds information on IP networks being operated by each AS, the address of their wallet, and the address of their SC. Operating IP networks and the token price could be modified in the central SC.

Individual ASes SCs implement a method to retrieve addresses reported by others ASes. To be aware of addresses of other domains, ASes query the central smart contract to maintain a local lookup table. If this table is outdated, since managed IP networks may change, it may result in defense requests not being accepted.

With the knowledge of IP networks maintained and, possibly, the token prices of other participants, ASes may request protection submitting a transaction to their SC with a list of IP addresses. Subsequently, requested ASes may accept or deny requests based on their security policies or Service Level Agreements (SLAs). A transaction is completed when a log, showing actions, is submitted in response to a defense request.

As of today, BloSS does not implement security mechanisms to prevent intentional or unintentional misuse. However, once a malicious act is proven (*e.g.*, spamming fake addresses), besides of a financial penalty for each transaction, the AS membership may be canceled and legal actions, be defined using SLA.

[1] A functionality providing financial incentives determines future work for BloSS, as it involves so called action-payment verifications.

A. Architecture

The architecture designed (*cf.*, Figure 1) simplifies the integration with existing networking systems. As an example, a Software-Defined Networking (SDN) environment was selected, however, the decentralized application (dApp) is not limited to SDN-based settings.

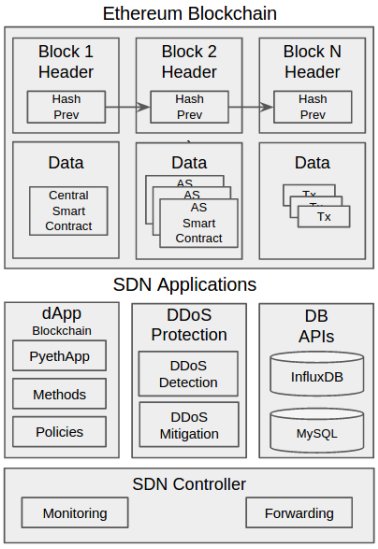

Fig. 1. High-level BloSS architecture deployed with an SDN-based network

The architecture comprises three main layers: (1) the SCs deployed in the Ethereum blockchain; (2) the local dApp interfacing with the blockchain to report/retrieve addresses; and (3) the Ryu SDN controller to monitor/enforce rules in the OpenFlow switches.

The system relies on existing DDoS detection and mitigation modules to provide input on addresses to be reported and to enforce requests of the collaborative defense if these match local security policies. The setup of policies, as well as wallet and smart contract addresses, are stored in a MySQL database.

In the Ethereum blockchain, the central SC is used to configure ASes SCs with updated information on their managed IP networks.Addresses of wallets and AS SCs are immutable and can not be changed when included in the central SC. AS SCs store addresses so that the entity that owns the contract performs an action required by other entities. A request is composed of the issuer's wallet address, action (whitelist or blacklist), hash, timestamp, and the IP address list.

The dApp represents the local client connected to the Ethereum blockchain via the *Pyethapp* API. It contains local policies, such as times for retrieving/reporting addresses, the maximum amount of tokens allowed to be sent, and others configurations persisted in a MySQL database. InfluxDB is used to collect network metrics and display graphs in the grafana front-end.

B. Hardware Configuration

A study case including three SDN-based ASes will be presented in the demonstration (*c.f.* Figure 2). ASes are connected to the blockchain via a management network 172.10.15.0/24 to ensure a congestion-free channel. Besides of mining the

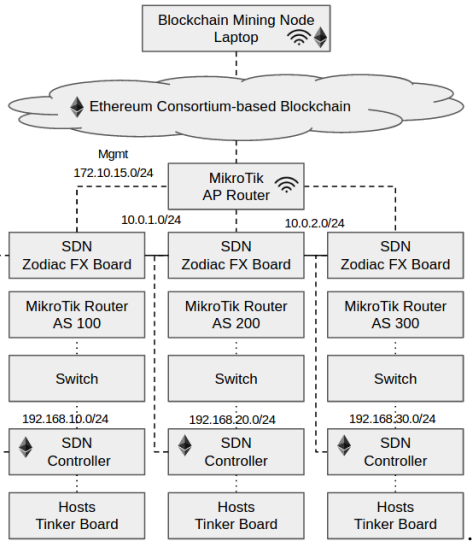

Fig. 2. Hardware configuration presenting three ASes: 100, 200, and 300

blockchain, the laptop act as a CnC (Command and Control) server to generate workload on the hosts. The inter-domain network is configured with the AS 100 connected to the AS 200 using network 10.0.1.0/24, which is connected to AS 300 through the 10.0.2.0/24 network. Hosts are configured with static addresses at each AS.

Fig. 3. Hardware set-up

The OpenFlow-enabled Zodiac FX switches provide an inexpensive alternative to experiment SDN prototypes in hardware. They are connected to a host running a Ryu SDN controller, and the dApp is connected to the blockchain.

Each AS is connected to five hosts deployed in hardware with ASUS Tinker Board devices (Raspberry-Pi like devices). Each board has a Gigabit Ethernet port that is sufficient to overload Fast Ethernet (10/100 Mbits) ports of the Zodiac FX boards, and thus, simulate a DDoS attack.

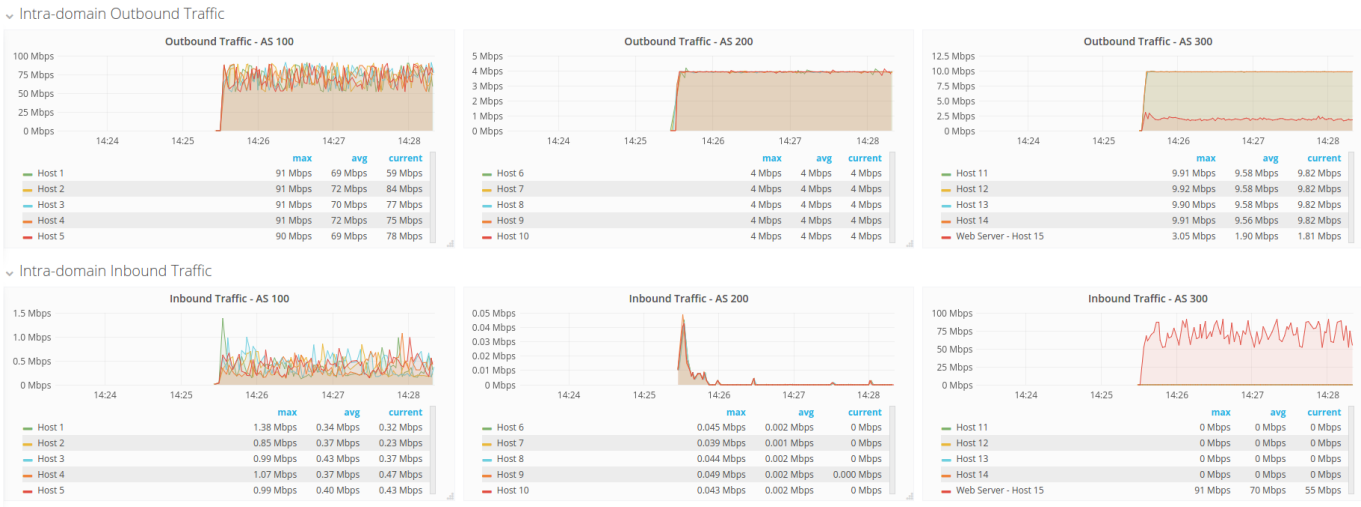

Fig. 4. Grafana Web Interface: ASes inbound and outbound traffic statistics and per-host traffic for hosts of AS 100 flooding Web server hosted in AS 300

III. Demonstration

The demonstrator presents live networking statistics (*c.f.* Figure 4). A use case with hosts connected to AS 100 simulating a flood attack scenario to a host connected on AS 300 will be demonstrated. Upon the attack detection, the AS 300 will request a cooperative defense submitting a transaction to the SC of the AS that is managing the IP network address of the attacker. Then, a message (*c.f.*, Figure 5) shows a request to block IP addresses of attacker's. The exchange of blockchain messages can be seen through the presenter's, which accesses via the SSH terminal the controller of each AS. Then, the requested ASes connected to the blockchain will receive the blocks mined and decide on their actions based on their security policies and DDoS thresholds.

```
**************************************************
REPORTING ADDRESS(ES):
Timestamp: 2017-07-03-09:33:57
Issuer: 192.168.30.15
Attacker(s): ['192.168.10.2', '192.168.10.3', '192.168.10.4', '192.168.10.5', '192.168.10.1']
Action: blacklist
Hash: bd10b992b461e8d337661840c54f2cebd461dce04e02a60dbee645a4ec9e6102
**************************************************

**************************************************
RETRIEVING ADDRESS(ES):
Timestamp (TS): 2017-07-03-09:34:30
Delta TS (s): 33
Issuer: 192.168.30.15
Attacker(s): ['192.168.10.2', '192.168.10.3', '192.168.10.4', '192.168.10.5', '192.168.10.1']
Action: blacklist
**************************************************
**************************************************
CONFIRMATION ADDRESS(ES) BLOCKED:
ASN 100
['192.168.10.2', '192.168.10.3', '192.168.10.4', '192.168.10.5', '192.168.10.1']
Timestamp: 2017-07-03-09:34:30
Hash: 50a7ca9f3a2608ee32a898cfa82846866bae3fc789f52d5039946e86ecda8b46
**************************************************
```

Fig. 5. Request-confirmation messages for the cooperative defense.

In the presented use case, as the AS 200 is configured as a transit AS, the message will be ignored. However, AS 100 will retrieve the message and block addresses accepting the request, if the tokens were sufficient. Otherwise, the message is ignored. The visitor at the demonstration, will see live networking statistics from boards and request/confirmation messages sent to the SC.

IV. Preliminary Considerations

Blockchain has the potential to simplify the operability and interoperability between ASes to advertise DDoS attacks in a cooperative defense. The absence of a centrally managed system and the support of vendor-specific communication protocols makes blockchain a suitable platform for a collaborative defense application. However, an off-chain storage infrastructure such as IPFS (Inter Planetary File System) [2] is required to scale the number of addresses. Within an operational system in the future, the blockchain will have an IPFS hash pointing to the object mounted on the distributed file system.

Overall, for systems like BloSS, security and performance are key requirements to be taken into account. On one hand, security-levels can be increased through a consortium-based blockchain, which delimits the access to the information being exchanged in the blockchain. Also, it provides a higher trust (in contrast to a public blockchain) over the entities able to report/receive the information exchanged. On the other hand, performance is measured according to the overall time elapsed between all the steps after reporting an attack, and the mitigation confirmation by other ASes, which includes latencies for block's processing and propagation and retrieval of addresses stored off-chain in IPFS.

Acknowledgments

The authors like to acknowledge discussions with David Hausheer for insightful comments during the design phase.

References

[1] Akamai, "How to Protect Against DDoS Attacks - Stop Denial of Service," 2016. [Online]. Available: https://goo.gl/pfcWph

[2] J. Benet, "IPFS-Content Addressed, Versioned, P2P File System," *arXiv preprint arXiv:1407.3561*, 2014.

[3] T. Bocek and B. Stiller, "Smart Contracts - Blockchains in the Wings," in *Digital Marketplaces Unleashed*, C. Linnhoff-Popien, R. Schneider, and M. Zaddach, Eds. Berlin, Heidelberg, Germany: Springer, 2017, pp. 1–16. [Online]. Available: https://goo.gl/M6LWrH

[4] K. Nishizuka, L. Xia, J. Xia, D. Zhang, L. Fang, and C. Gray, "Inter-organization Cooperative DDoS Protection Mechanism," Internet-Draft, Draft, December 2016. [Online]. Available: https://goo.gl/Z2yMD8

[5] B. Rodrigues, T. Bocek, A. Lareida, D. Hausheer, S. Rafati, and B. Stiller, "Blockchain-based Architecture for Collaborative DDoS Mitigation with Smart Contracts and SDN," in *11th International Conference on Autonomous Infrastructure, Management and Security (AIMS 2017)*. Zürich, Switzerland: Springer, LCNS Vol. 10356, July 2017, pp. 16–29.