Security Management and Visualization in a Blockchain-based Collaborative Defense

Christian Killer, Bruno Rodrigues, Burkhard Stiller
Communication Systems Group (CSG), Department of Informatics IfI, University of Zurich UZH
Binzmühlestrasse 14, CH-8050 Zürich
E-mail: [killer,rodrigues,stiller]@ifi.uzh.ch

***Abstract*—A cooperative network defense is one approach to fend off large-scale Distributed Denial-of-Service (DDoS) attacks. In this regard, the Blockchain Signaling System (BloSS) is a multi-domain, blockchain-based, cooperative DDoS defense system, where each Autonomous System (AS) is taking part in the defense alliance. Each AS can exchange attack information about ongoing attacks via the Ethereum blockchain. However, the currently operational implementation of BloSS is not interactive or visualized, but the DDoS mitigation is automated. In real-world defense systems, a human cybersecurity analyst decides whether a DDoS threat should be mitigated or not. Thus, this work presents the design of a security management dashboard for BloSS, designed for interactive use by cybersecurity analysts.**

I. Introduction

Cybersecurity concerns have globally risen to one of the top priorities in both research and development [1]. Distributed Denial-of-service (DDoS) attacks are one major concern in cybersecurity, because their perpetration requires little effort on the attacker's side, while massive damage is inflicted to the victim. Recent statistics on security reports show not only a steady increase in the number of DDoS attacks, but also the number of long-duration attacks (*e.g.*, the most extended attack was longer than 12 days [2]).

One approach to defend against DDoS attacks is sharing hardware and defense capabilities with other systems, an approach called cooperative DDoS mitigation [3]. Thus, if an attack is highly sophisticated and there are no countermeasures available, it is possible to request for cooperative mitigation to any domain participating in the alliance. However, challenges involve, for example, trust between its members, the need for financial incentives fostering the cooperative behavior and aspects that make such defense operational.

A cooperative defense is ideally an additional protection mechanism combined with traditional in-house (*e.g.*, firewall, intrusion detection systems) or exterior (cloud-based defense) protection mechanisms. By itself, the detection and mitigation of DDoS attacks is not a straightforward task as there are different types and patterns of DDoS attacks [2]. A cooperative defense introduces another layer of support, but also complexity requiring that network operators decide not only when and whether it is necessary to request or accept a mitigation service, but also to or from which Autonomous Systems (AS) to interact with, and whether the incentives involved cover operational expenses.

978-1-7281-1328-9/19/$31.00 ©2019 IEEE

The Blockchain Signaling System (BloSS) [4] is prior work that uses a blockchain as a platform to exchange information about attacks and to distribute necessary financial incentives (*cf.*, Figure 1). Among the technical challenges, the visualization and management of security are crucial. This paper tackles the challenge of building an interactive interface that facilitating the decision making of ASes concerning their participation in a collaborative defense based on a blockchain.

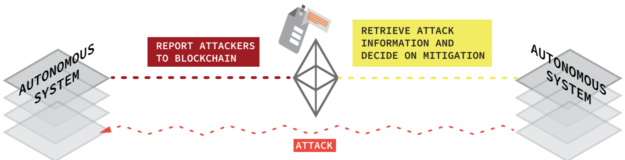

AUTONOMOUS SYSTEM
REPORT ATTACKERS TO BLOCKCHAIN
RETRIEVE ATTACK INFORMATION AND DECIDE ON MITIGATION
AUTONOMOUS SYSTEM
ATTACK

Fig. 1: Cooperative mitigation request and acceptance in a blockchain-based defense.

In a context where ASes rely on cybersecurity specialists to make critical decisions regarding threats, it is necessary to structure and categorize data such that visualization ”makes sense” to the analyst [5]. As a cooperative defense involves multi-disciplinary concepts and the decision-making process usually requires a low response time from the user, selecting an appropriate type of graphical representation and flow of interaction is not a straightforward task [6].

The approach in this work defines workflows with significant state management events and the architecture that supports a blockchain-based collaborative defense to provide an interactive dashboard to the security analyst. The approach is evaluated through a use case based on [4], [7], presenting the front-end component displaying the on-going processes, and enabling a cybersecurity analyst to react on an individual threat level. Section II presents the background on security visualization, followed by Section III highlighting the architecture. The use case evaluation and a short discussion is presented in Section IV. Finally, conclusions are drawn in Section VI.

II. Security Visualization

In order to apply visualization techniques to the computer security field, specifically to a collaborative defense, knowledge of different disciplines needs to be combined. The dichotomy of security visualization is that systems try to visualize security data, they know the data and their semantics, but do not necessarily understand the main design principles of visualization [8]. Therefore, it is essential to apply best-practices from visualization theory. For example, the idea of Situation Awareness (SA) is not restricted to any particular domain, but can be applied to the cyber domain [9]. Different challenges reach SA for a decision maker, being a key to identify and model relevant activities of interest [10]: (a) current situation: include the identification of the type of attack, while a recognition is acknowledging that an attack occurs; (b) impact of attack: involves the assessment of the current and future impact; (c) evolution: awareness of how the situation evolves which includes tracking the situation; and (d) causality: understanding of how and why the current situation is caused, including causality analysis, back-tracking, and the use of forensics.

[11] defines the following method to visualize data: (a) overview first, (b) zoom and filter (c) details-on-demand. Thus, it is necessary to grasp the situation and its possible outcomes to collect an overview, *e.g.*, the analysis of a DDoS attack determines whether it is necessary to request or accept/deny a cooperative defense. Afterwards, the zoom and filter step relates to the detailed analysis of the specific event and the filtering of unrelated events that can introduce unwanted noise into the analysis. Finally, details are provided on demand to the analyst.

Tools specifically designed to visualize DDoS attacks vary from representations for non-technical audiences [12] to visualizations specifically designed for cyber-security analysts. While the former is related to the design of infographics and charts for a business audience, the latter digests detailed attack data and transforms these data into a visual format optimized for cybersecurity analysts [13], [14]. Although there does not exist a one-size-fits-all visualization tool due to the subjective nature of the human perception process, the research in visualization either leads to the adaption of existing visualization techniques to the cybersecurity domain or it leads to novel ways to specifically visualize cybersecurity data.

Furthermore, dashboards are commonly used to analyze data in (near) real-time. The focus is to understand the current state of a system, ongoing tasks. or events of interest. [15] describes various factors to consider during the design of dashboards, such as not overload the dashboard with visual features, to make prominent visual features showing essential messages, and to design a dashboard within constraints, even though they still should be pleasant to view. Unnecessary decorations, or overuse of colors and other visual properties are to be exploited carefully.

III. Architecture

BloSS uses a private permissionless instance of the Ethereum blockchain to exchange attack information, *i.e.*, read and write privileges are limited to the members of the alliance. The system architecture is composed of three components in which the original `bloss-core` component is extended by the `bloss-node` and a `bloss-dashboard` component as shown within Figure 3.

The `bloss-core` component is responsible for the communication between the blockchain, network management system, and IPFS [16]. In the event of an attack, the attack information is first stored on IPFS. Afterwards only the IPFS hash is stored on the Ethereum blockchain. Prior to saving the attack reports on IPFS, they are encrypted with the public key of the recipient AS (domain where attack originates from), thus, preventing information leakage and decreasing the amount of data saved on-chain.

A RESTful interface on the `bloss-core` is used to exchange information (*e.g.*, attack reports, blocking status, traffic breaches) with a `bloss-node` instance. The `bloss-node` acts as a relay server for communications between the `bloss-core` and `bloss-dashboard`. A WebSocket interface is used to exchange data with the front-end `bloss-dashboard`. Additionally, a `bloss-node` handles the state management of attack reports. The `bloss-dashboard` is a front-end dashboard displaying relevant information for the human analyst. The `bloss-dashboard` is implemented as a Single-Page Application (SPA) and attached to the `bloss-node` via WebSockets.

As the collaborative defense is to serve as a complementary defense to in-house and off-house mechanisms for monitoring and detecting attacks, the focus of the dashboard is on events and components related to the cooperative defense, such as information about the status of relevant services and the status of individual attack reports submitted to the blockchain.

Figure 2 illustrates the three main sections of the visualization system, the status of the system components, the visualization of requests for collaborative defense on the Mitigator's (*M*) side, and the alarms tab allowing the request for collaborative help on the Target's (*T*) side. An AS may act either as *M* or *T*, so these tabs are available to the network operator. These tabs display three columns with progressing states from left to the right, in which the left row which contains new events [17].

The middle row comprises all events in progress and updates them accordingly to changes in their status. The right row represents a log of all elements finished or declined. The *REQUESTS_TAB* contains all incoming mitigation requests. The *ALARMS_TAB* contains all alarms that were triggered as soon as pre-defined inbound traffic threshold breach occurs. The *REQUESTS_TAB* contains all incoming mitigation requests.

IV. Use Case Evaluation

Figure 4 groups all states involved in this Use Case (UC). The UC assumes three ASes being involved: AS400, AS500, and AS600. The precondition for this UC is that all these services are active and operating correctly. The left side of the dashboard shows the status of these services, as well as the option to activate or deactivate respective modules with a click. In the first step, as soon as the inbound traffic breaches

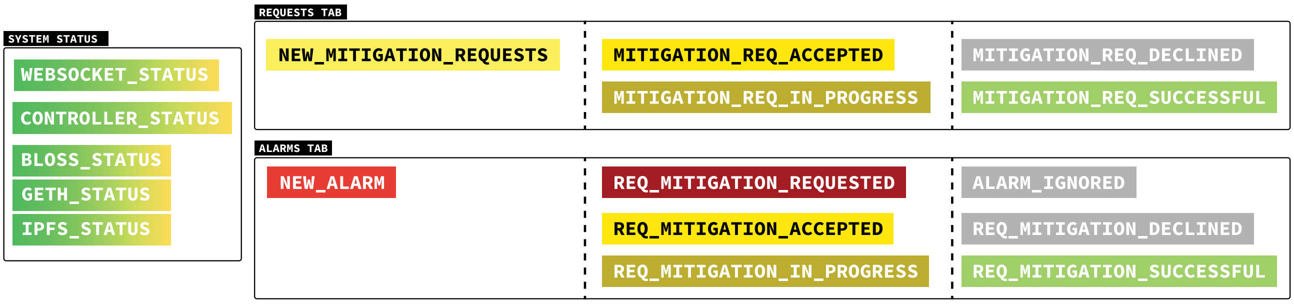


Fig. 2: Schematic view of the network operator's dashboard.

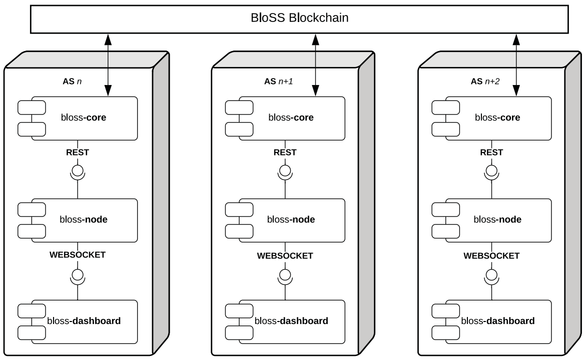


Fig. 3: Component Diagram of the BloSS architecture.

the pre-defined threshold, alarms are sent to the dashboard and the operator has to decide whether to react or ignore these alarms: the dashboard display a message ● NEW_ALARM.

Next, the analyst decides whether to request the cooperative mitigation or to ignore the alarm. If a mitigation is required, a request is sent to the bloss-core component, which submits a transaction to the blockchain and the request is moved to the column "In Progress" with the status ● REQUEST_MITIGATION_REQUESTED. Then, target operators on AS 500 and AS 600 can either decline the request for mitigation and the status of the attack_report changes to ● REQ_MITIGATION_DECLINED or accept the request for mitigation and the status of the attack_report changes to ● REQ_MITIGATION_ACCEPTED. These screens follow Shneiderman's process [11] to overview first (*e.g.*, events are displayed and grouped in the "Requests" column), to zoom and filter involving the analysis of the operator to decide whether to request or ignore the alarm, and to provide details on demand by clicking on events to show detailed information.

As soon as the mitigator starts blocking, *i.e.*, applying a mitigation action, such as blackholing traffic or blocking hosts listed as attackers, the attack_report status changes to ● REQ_MITIGATION_IN_PROGRESS. After the expiration of the maximum block duration (of the attack), the attack_report is completed and ends in the status ● REQ_MITIGATION_SUCCESSFUL. Furthermore, the history of requests, besides being registered on the blockchain (not disclosing any details, *e.g.*, blacklisted addresses) and available to all members of the alliance, events involving each domain are saved and grouped in the "Log" column. This offers the possibility to investigate details on demand.

V. Discussion

This interactive dashboard facilitates the visualization and management of a collaborative defense by a network operator, and enables the management of these events. The bloss-dashboard includes features to visualize the status of mitigation requests from the perspective of the mitigator as well as the requester.

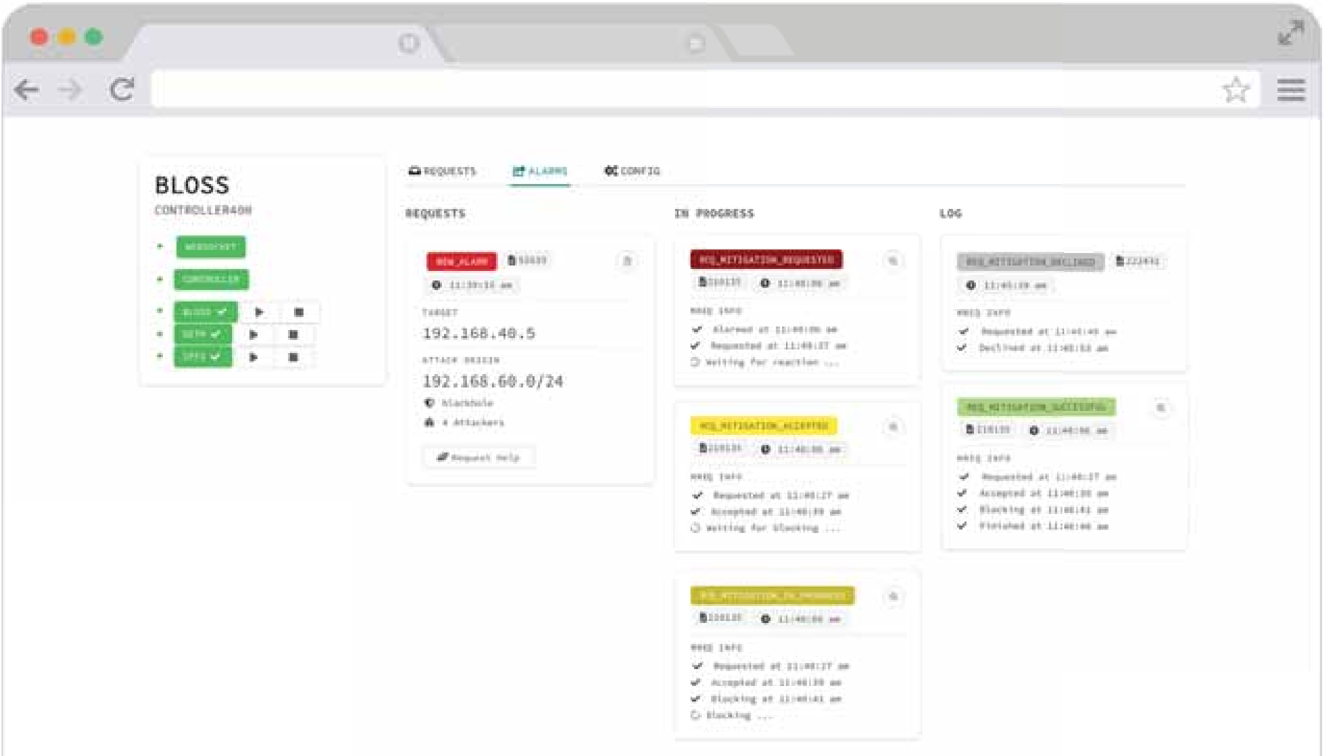
Tasks involving the detection and mitigation of a DDoS attack are not trivial, hence, the analysis in a collaborative environment grows in complexity. The challenge of this dashboard here, however, is to reduce the complexity of information provided to a network operator to support decision-making without adding overhead of unnecessary information. In the presented use case, a new alarm is displayed to and evaluated by the human operator. An alarm can either be classified as a valid threat and whether mitigation should be requested, or the alarm can be ignored.

The complexity of the this bloss-dashboard is minimal and based on the known layout from software development (*e.g.,* the Kanban board [17]), leading to a low entry barrier of understanding the user interface. Also, the bloss-dashboard remains predictable as the main layout does not change and empty states (*e.g.*, no new MREQ) indicate where new elements have to appear in the layout.

VI. Conclusion and Future Work

This work presented an applicable and operational management dashboard reducing the operational complexity of a blockchain-based cooperative defense. Thus, a human decision-maker (*e.g.*, a security analyst) can decide whether a threat is severe and defines a course of action based on an overview of all attack-relevant data in the security management tool.

A future version of the dashboard will provide further insights into the reputation history of members involved in



TARGET
192.168.40.5

ATTACK ORIGIN
192.168.60.0/24

Blackhole

4 Attackers

MREQ INFO
✓ Alarmed at 11:40:06 am
✓ Requested at 11:40:27 am
Waiting for reaction ...

MREQ INFO
✓ Requested at 11:40:27 am
✓ Accepted at 11:40:39 am
Waiting for blocking ...

MREQ INFO
✓ Requested at 11:40:27 am
✓ Accepted at 11:40:39 am
✓ Blocking at 11:40:41 am
Blocking ...

MREQ INFO
✓ Requested at 11:41:49 am
✓ Declined at 11:41:53 am

MREQ INFO
✓ Requested at 11:40:27 am
✓ Accepted at 11:40:39 am
✓ Blocking at 11:40:41 am
✓ Finished at 11:40:46 am

Fig. 4: `bloss-dashboard` Mock-up UI displaying possible states of an example attack_report in the `ALARMS_TAB`

a mitigation service by providing more data for an analyst to decide on a mitigation request's acceptance.

Acknowledgements

This work had been partially funded by the EU H2020 Program under Grant Agreement No. 830927 and partially by the University of Zurich UZH, Switzerland.

References

[1] K.-C. Lee, C.-H. Hsieh, L.-J. Wei, C.-H. Mao, J.-H. Dai, and Y.-T. Kuang, "Sec-Buzzer: Cyber Security Emerging Topic Mining with Open Threat Intelligence Retrieval and Timeline Event Annotation," *Soft Computing*, Vol. 21, No. 11, pp. 2883–2896, June 2017. [Online]: https://doi.org/10.1007/s00500-016-2265-0

[2] A. Khalimonenko, O. Kupreev, and E. Badovskaya, "*DDoS attacks in Q1 2018*," [Online] https://securelist.com/ddos-report-in-q1-2018/85373/, April 2018, last visit March 6, 2019.

[3] S. T. Zargar, J. Joshi, and D. Tipper, "A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks," *IEEE Communications Surveys & Tutorials*, Vol. 15, No. 4, pp. 2046–2069, March 2013.

[4] B. Rodrigues, T. Bocek, and B. Stiller, "Enabling a Cooperative, Multi-domain DDoS Defense by a Blockchain Signaling System (BloSS)," *43rd IEEE Conference on Local Computer Networks (LCN 2018), Demonstration Track*, Singapore, Singapore, October 2017, pp. 1–3. [Online]: https://goo.gl/5TMFUt

[5] D. Staheli, T. Yu, R. J. Crouser, S. Damodaran, K. Nam, D. O'Gwynn, S. McKenna, and L. Harrison, "Visualization Evaluation for Cyber Security: Trends and Future Directions," *11th Workshop on Visualization for Cyber Security (VizSec 2014)*, New York, NY, U.S.A., November 2014, pp. 49–56. [Online]: http://doi.acm.org/10.1145/2671491.2671492

[6] X. Li, Q. Wang, L. Yang, and X. Luo, "Network Security Situation Awareness Method Based on Visualization," *Third International Conference on Multimedia Information Networking and Security (MNES 2011)*, Shanghai, China, November 2011, pp. 411–415.

[7] S. Mannhart, B. Rodrigues, E. Scheid, S. S. Kanhere, and B. Stiller, "Toward Mitigation-as-a-Service in Cooperative Network Defenses," *3rd IEEE Cyber Science and Technology Congress (CyberSciTech 2018)*, Athens, Greece, August 2018, pp. 362–367.

[8] R. Marty, *Applied Security Visualization*, Addison-Wesley, Upper Saddle River, NJ, U.S.A., 2008.

[9] G. P. Tadda and J. S. Salerno, "Overview of Cyber Situation Awareness," *Cyber Situational Awareness: Issues and Research*, S. Jajodia, P. Liu, V. Swarup, and C. Wang, Eds. Boston, MA, U.S.A., Springer, 2010, pp. 15–35.

[10] P. Barford, M. Dacier, T. G. Dietterich, M. Fredrikson, J. Giffin, S. Jajodia, S. Jha, J. Li, P. Liu, P. Ning, X. Ou, D. Song, L. Strater, V. Swarup, G. Tadda, C. Wang, and J. Yen, *Cyber SA: Situational Awareness for Cyber Defense*. Boston, MA, U.S.A., Springer US, 2010, pp. 3–13. [Online]: https://doi.org/10.1007/978-1-4419-0140-8_1

[11] B. Shneiderman, "A Grander Goal: A Thousand-fold Increase in Human Capabilities," *Educom review*, Vol. 32, No. 6, pp. 4 – 10, November 1997.

[12] AO Kaspersky Lab., "Cyberthreat Real-Time Map," [Online] https://cybermap.kaspersky.com, last visit March 6, 2019.

[13] S. McKenna, D. Staheli, C. Fulcher, and M. Meyer, "BubbleNet: A Cyber Security Dashboard for Visualizing Patterns," *Computer Graphics Forum (EuroVis 2016)*, Vol. 35, No. 3, Groningen, Netherlands, 2016, pp. 281–290. [Online]: http://dx.doi.org/10.1111/cgf.12904

[14] A. Yelizarov and D. Gamayunov, "Visualization of Complex Attacks and State of Attacked Network," *6th International Workshop on Visualization for Cyber Security (VizSec 2009)*, Atlantic City, NJ, USA, November 2009, pp. 1–9.

[15] J. Jacobs and B. Rudis, *Data-Driven Security: Analysis, Visualization and Dashboards*, Wiley, Hoboken, NJ, U.S.A., 2014.

[16] J. Benet, "IPFS - Content Addressed, Versioned, P2P File System," *CoRR*, Vol. abs/1407.3561, 2014. [Online]: http://arxiv.org/abs/1407.3561

[17] N. Oza, F. Fagerholm, and J. Münch, "How does Kanban Impact Communication and Collaboration in Software Engineering Teams?" *6th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE 2013)*, San Francisco, CA, U.S.A., May 2013, pp. 125–128.