

ProtectDDoS: A Platform for Trustworthy Offering and Recommendation of Protections

Muriel Franco, Erion Sula, Bruno Rodrigues, Eder Scheid, Burkhard Stiller

Communication Systems Group CSG, Department of Informatics IfI,
University of Zürich UZH
Binzmühlestrasse 14, CH-8050 Zürich, Switzerland
E-mail: [franco, rodrigues,scheid, stiller]@ifi.uzh.ch, erion.sula@uzh.ch

Abstract. As the dependency of business on digital services to offer their products increases, their vulnerability to cyberattacks increases. In this sense, besides providing innovative and flexible services, business owners must focus on investing in robust cybersecurity mechanisms to prevent cyberattacks. Distributed Denial of Service (DDoS) attacks remains one of the most dangerous cyberattacks, leading to several problems, such as service disruption, financial loss, and reputation harm, for companies and end-users. Even though several providers offer protections for different types of DDoS attacks, there is still a lack of catalogs or solutions that help network operators to access and filter information in order to select the most suitable protection for a specific demand. Thus, in this paper, a platform for offering and recommendation of DDoS protection, named ProtectDDoS, is proposed. ProtectDDoS provides a blockchain-based catalog where DDoS protection providers can announce details regarding their services, while interested users can filter or obtain recommendations of DDoS protection according to their specific demands (e.g., price, attacks supported, and geolocation constraints). ProtectDDoS provides a smart contract that maintains the integrity of the data about the protections available and provides tamper-proof reputations. To evaluate the feasibility and effectiveness of ProtectDDoS, a prototype was implemented and a case study conducted. Further, a discussion concerning additional costs, including the interaction with the smart contract, is provided.

Keywords: Cybersecurity · DDoS Protections · Recommender Systems · Smart Contracts · Marketplaces.

1 Introduction

Denial-of-Service (DoS) attacks represent a significant threat to almost any commercial organization and individuals, which rely on offering Internet-based services. In the last years, such attacks have become more complex and sophisticated, and, in turn, difficult to predict and mitigate [6]. Even more dangerous are the so-called Distributed Denial-of-Service (DDoS) attacks, as the attack itself derives from multiple hosts distributed over the network, such as those using

botnets. As a consequence, the affected targets (*e.g.*, companies and governments) are usually confronted with economic impacts. Not only do these attacks cause financial damages due to the unavailability of the service and loss of online traffic, but in some critical cases, they inflict long-term damage to the corporate reputation, causing drastic drops of the stock prices [1].

Further, the number of DDoS attacks has almost tripled in the last three years, with an averaged financial loss of dozens of thousands of USD (United States Dollar) per hour of the attack. For example, it is estimated that, only in the United Kingdom (UK) in 2019, DDoS will cost more than US\$ 1 billion [4], which includes revenue loss and cyber insurance premiums. These numbers continue to grow due to the increasing amount of exposed Internet-of-Things (IoT) devices and Artificial Intelligence (AI) techniques. Cybersecurity predictions point that the number of DDoS attacks globally will reach 17 million by 2020, causing several economical and societal impacts.

Based on this threat landscape, large companies and governments are spending roughly US\$124 billion on information security products and protection services. However, many of the problems plaguing cybersecurity are still economic in nature [7]. Often systems fail because the organizations do not bear to assess the full costs of a failure neither the risks involved. It is still more prevalent when, for example, considering organizations and users with restrictions of budget or technical expertise to invest in cybersecurity, such as Small- Medium-sized Enterprises (SME). Therefore, it is clear that efficient risks analysis and investments in proper cybersecurity solutions are critical for the next years for both organizations and governments with services or systems exposed on the Internet. These investments must not focus solely on reactive protection against DDoS attacks but also target the planning and decision process of cybersecurity to predict attacks and possible losses arising from a cyberattack. Therefore, multiple layers of precaution to protect the critical services against DDoS attacks are required.

Nowadays, the variety of DDoS protection services has increased as well. While competition in this sector may have benefits for consumers, such as higher quality for the same price or diversified products, organizations often struggle with choosing a protection service that suits their needs. Solutions that help the offering and selection of DDoS protection can support the organization in the decision-making process. More specifically, by providing the user with essential information related to the thousands of DDoS protection services available, taking into account filters and characteristics of the cyberattack (*e.g.*, fingerprints and log files) provided by the user. However, there are no intuitive solutions (*e.g.*, dashboards) that simplify the access to a broad set of DDoS protections while ensuring the integrity of the information from the protections available, *i.e.*, tamper-proof information. Besides that, there is still a lack of integration of catalogs and mechanisms that help to decide which is the most suitable protection taking into account specific DDoS scenarios and user' demands.

This paper introduces *ProtectDDoS*, a blockchain-based platform for the offering and support of the recommendation of protection services against DDoS attacks. *ProtectDDoS* provides a blockchain-based catalog where protection providers

can announce protections and interested users can filter its protections by using different parameters, such as price, type of DDoS attack supported, and deployment time. In addition, DDoS attacks fingerprints [10] can be used as input to find the most suitable protection for a determined type of attack. Furthermore, a Smart Contract (SC) is implemented to store (i) the hash of protection services and the private address of protection providers to verify the origin and integrity of protections available and (ii) protections' reputations based on users' feedback, which can be used to avoid protections with misbehavior or insufficient performance for a determined scenario. *ProtectDDoS* also offers a dashboard fully integrated with a recommender system for protection services, called *MENTOR* [3]. Thus allowing the user to use a web-based interface to obtain a recommendation of the most suitable solution according to its demands and predefined filters.

The remaining of this paper is organized as follows. Background and related work are reviewed in Section II. In Section III, the platform for offering and recommendation of DDoS protections is introduced, and implementation details are provided. In Section IV, the feasibility of the proposed solution is discussed, and a case study is presented. Section V, a functional is provided in order to measure the additional costs of the *ProtectDDoS*. Finally, in Section VI, a summary of the paper and comments on future work are provided.

2 Background and Related Work

As businesses strengthen their digital dependency, they also become more vulnerable to cyber threats. Therefore, besides the need for speed innovation, decision-makers in cybersecurity (*e.g.*, network operator, company owner, or an expert team) have to be able to implement robust security mechanisms while managing costs and risks associated with the business [9]. Such activities involve (i) identify security risks and associated costs and (ii) determine impacts of cybersecurity in the business or service. In turn, it is possible to estimate the overall impacts (*e.g.*, financial loss occasioned by a business disruption) in order to decide whether to invest in cybersecurity. Such an overall estimation can be done using different approaches. For instance, the Return Over Security Investments (ROSI) [12] offers a benchmark to determine when a specific investment in cybersecurity is recommended based on the potential financial loss given an assessed risk. Based on that, decision-makers have to decide how to handle a possible or imminent threat. Between the different choices, the decision-maker can (i) determine a plan to prevent cyberattacks and its impacts proactively, (ii) react against an imminent cyberattack, or (iii) assume the risks, paying for the damage or delegating that to third-parties (*e.g.*, cyber insurers). In general, if an attack happens, prevention is cheaper than react when an attack already surpassed the infrastructures. If companies do not invest correctly in cybersecurity, the security of their operations depends on luck, and the impacts of attacks can be devastating, which is not acceptable by one that has a reputation to maintain.

The market for protection services has grown together with the investments in cybersecurity. Nowadays, several providers are offering protections for different kinds of attacks (*e.g.*, data leaks, DDoS, and malwares) and demands. For example, [2] provides a repository that lists providers offering many protection services to address different cybersecurity threats, such as advanced threat protection, anti-virus, secure communications, and anti-phishing. The number of protections available is large and is growing in parallel with the investments in cybersecurity. Only on such a repository, 1200 providers are listed, and one can, for example, obtain information to contract more than 80 protection services against DDoS attacks. However, even though there are few catalogs centralizing information from different cybersecurity solutions [2], there still a lack of platforms that use such information to simplify the decision-process and cybersecurity planning of companies. Table 1 provides a comparison among different cybersecurity-oriented solutions that implement approaches to offer or recommend services.

Table 1: Comparison of Related Work in terms of designed functionalities

-	Functionalities				
Solution	User-friendly Catalog	Supports Recommendation	Filters	Allows Integrity Verification	Reputation Mechanisms
[3]	No	Yes	Yes	No	No
[2]	Yes	No	Yes	No	No
[8]	No	Yes	Yes	No	No
[5]	No	Yes	Yes	No	No
[11]	Yes	No	No	Yes	Yes
ProtectDDoS	Yes	Yes	Yes	Yes	Yes

In previous work, *MENTOR* [3], a recommender system for protection services was introduced to help during the decision of which is the most suitable protection for determined demands. This system provides a recommendation engine that can recommend the most suitable protection based on a list of parameters and user demands. However, the system is still in its infancy and does not provide user interfaces, or a catalog for protection providers to submit their solutions. Also, the reputation of protections based on user feedback is still not being considered during the recommendation process. In another work, [8] provides a recommender system to predict cyberattacks by identifying attack paths and demonstrates how a recommendation method can be used to classify future cyberattacks. [5] introduced an interactive user interface for security analysts that recommends what data to protect, visualizes simulated protection impact, and helps build protection plans. However, none of them supports the different characteristics of DDoS attacks while offers intuitive interfaces for users to add their demands and log files to receive recommendations.

By using the concepts of Blockchain (BC) and Smart Contracts (SC), different solutions have been proposed to enables the validation of the integrity

and origin of solutions for different purposes. BC was initially conceived as a distributed ledger to be the backbone of the Bitcoin cryptocurrency. However, its capacity to provide an immutable, trustworthy, and decentralized records has attracted the attention of both industry and academia [14]. The concept of SC is implemented by the second generation of BCs, such as Ethereum and NEO. The fees involved in SC are lower than traditional systems that require a trusted intermediary. In [11], for example, *BUNKER*, a BC-based marketplace for Virtual Network Functions (VNFs), was introduced to provide immutable and trusted information concerning VNF packages acquired by end-users. The solution stores the hash of the packages in a BC to guarantee the integrity of the VNF being acquired by end-users. This kind of feature can be useful for both providers and users interested in protections since the integrity of the protection can its origin can be validated as well as its reputations validated before decide for one specific cybersecurity solution.

3 *ProtectDDoS* Platform

ProtectDDoS allows users to describe their demands for protections in order to obtain a proper level of protection against different types of DDoS attacks, which facilitates the decision process, from an extensive list of options available, the most suitable protection. These protection services can be acquired proactively before an attack happens or to react during an imminent attack. Thus, *ProtectDDoS* offers mechanisms to support the decisions required during cybersecurity planning and management. Besides that, protection providers can announce their solutions to build a heterogeneous catalog of protections against DDoS, thus achieving a broad audience of companies and users interested in contract/acquire protections. Also, the *ProtectDDoS* allows, through a web-based interface, the users to (i) upload fingerprints of DDoS attacks to find specific protections, (ii) verify, supported by the BC, the integrity and origin of the information of different protections, (iii) receive recommendation of the best solution according to its demands, and (iv) provide feedback of the contracted protections, thus supporting a reputation system for the protections available. The *ProtectDDoS*'s code is publicly available at [13].

3.1 Architecture

Figure 1 introduces the architecture of the *ProtectDDoS* and its main components. The architecture is divided into three different layers: the (i) User Layer provides the components required to actors interact with the *ProtectDDoS* and the protections available through an intuitive and modern interface, the (ii) Data Layer, which is in charge of the steps involved in process and handle information related to the protections, as well as serving as link for the upper and lower layers, and the (iii) BC Layer, which consists of a SC running inside of the Ethereum BC containing useful information (*e.g.*, hash and reputations of protections) to be used by the other layers, such as to verify the integrity services'

information or its developer. Also, the connection with *MENTOR* recommender system is available by using an API introduced by *MENTOR*, which is fully integrated with the *ProtectDDoS* architecture, thus allowing that calls can be done to receive a recommendation of the best protection services according to the previously filters and configurations defined by the user.

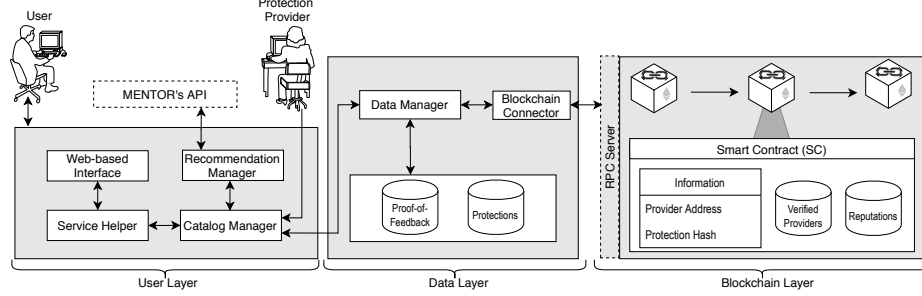


Fig. 1: *ProtectDDoS*'s Conceptual Architecture

In the *User Layer*, a *web-based interface* (*i.e.*, dashboard) provides access the catalog, details of protections, and the recommendation process. The *Service Helper* plays a crucial role in the integration of the catalog and the recommendation process by applying the predefined filter on the whole dataset of protections available, thus removing protections that are not suitable to the user's requirements. The *Catalog Manager* requests information from the *Data Layer* to build the catalog of available protections, apply the filters, and send the list of protections to start the recommendation process. Finally, the *Recommendation Manager* is in charge of constructing the calls for the recommendation API (*i.e.*, *MENTOR*'s API). For that, this component transforms the user requirements and information from the selected protections into a defined JSON data structure [13] containing all relevant information for the recommendation.

The *Data Layer* contains the *Protections database* to store all information of the protections available, such as developer, name, price, and types of attacks supported. Also, a database is provided to store all of the log files (*e.g.*, pcap) containing the information regarding the performance of the contracted protection, which helps during the audition and validation of bad or good feedback provided by users. This database is managed by the *Data Manager*, which is the interface with the *Data Layer* and is in charge of process and answer to requests for information. Furthermore, the *BC Connector* is an adaptor implemented to enable the communication with the SC running inside of the BC. The *BC Connector* performs the calls to interact with the SC (*e.g.*, verify the protection hash or validate the provider address) by sending BC transactions thought a *Remote Procedure Call (RPC)* server provided by the BC.

Finally, in the *BC Layer*, the SC is deployed to store a list of verified providers based on their address on the BC, the reputations of each protection according to the users' feedback, the hash of the proof-of-feedback files, and the hash of the protection associated with the address of the provider that submitted the service. It is worth reinforcing that all of this information is immutable and tamper-proof, which allows any interested party to audit and trust in the information following the whole history of the stored information.

3.2 Workflow

By accessing the *ProtectDDoS*, users interested in obtaining protection can verify the available protection services in a catalog and apply filters to select a set of characteristics that satisfies his/her demands, such as a maximum price or protection against a specific type of DDoS attack. For that, the user can select a determined attack-type from a list of attacks supported or also upload a file containing the fingerprints of a DDoS attack of which protection is required. Based on that fingerprint, the *ProtectDDoS* can process such information to filter a list of protections suitable for such a user's demands. Furthermore, such a list can be sent to the *MENTOR* [3] recommender system, through a provided API, in order to receive, as a response, the best protection selected by the recommendation process implemented in *MENTOR*.

Also, providers of protection services can access the web-based interface and store new protections to be available in the catalog. Such protections can be described according to its price, deployment technology, region, and technical aspects (*e.g.*, types of attacks supported and performance). Besides, the dependencies of protection can be described in order to help the user to decide which protection is suitable to run together with protections already setup on-premise, *i.e.*, an infrastructure which already has an open-source running might have different requirements or conflicts than an infrastructure running a vendor-specific intrusion detection system. During the submission of new protection for the catalog, the *ProtectDDoS* can store immutable information regarding the protections and providers on the BC. This information is used to validate the origin and integrity of the protections, thus resulting in a trustworthy catalog since any user can verify and trust that the information stored by the protection developer was not modified. The reputations of each protection are also stored according to the feedback from the user and an optional log file uploaded as a proof-of-feedback. This log file is stored on the platform database, and the hash of the file is stored in the BC in order to make it available for a further audition by any interested.

The process of uploading a new protection service comprises of two essential steps. In the first step, the service's information is hashed, using the SHA256 algorithm, and then subsequently submitted onto the BC. This operation is handled using an SC and will cost an amount of Ether as a fee to be completed (*cf.* Section V). When interacting with an SC, an Ethereum account is required. Ethereum was defined to be used because of its popularity and simplified way to build SCs. However, the *ProtectDDoS* is designed to operate with other BCs, as well. Upon successful storage, the protection information, along with the

transaction hash, i.e., an identifier used to identify a transaction on the BC, is then stored in the platform database. Based on that, it is possible to retrieve the hash of any protection from the BC, thus offering the possibility to any user checks whether the service’s information has been compromised or if the provider is not verified. In order to do that, the service hash is required, and a specific function in the SC needs to be invoked. Therefore, such integration with BC allows that not only the recommendation of protections can be made, but also evidence of the integrity and origin (*i.e.*, who is the provider) of the protection can be provided.

The data structure introduced by *MENTOR* recommender system is used as a basis to describe the different parameters that represent the users’ requirements to contract a protection service. Besides, to specify details of DDoS attacks, additional information is considered to represent the attack type since information directly obtained from DDoS fingerprints can be used (*e.g.*, attack sources, the protocol used, and destination ports). Also, the structure of a DDoS fingerprint supported by the *ProtectDDoS* is those proposed by the DDoSDB platform, which is a public DDoS attack database for improving defense strategies [10]. The parameters that can be configured for the catalog filter or recommendation of protections are: (i) Service Type, which can be reactive or proactive, (ii) Attack type (*e.g.*, SYN Flood or DNS Amplification) defined directly from a list or identified by a using a fingerprint filed optionally uploaded by the user, (iii) Coverage Region to indicate the location (*e.g.*, continents, countries, or even cities) where a cloud-based protection has to be deployed, (iv) Deployment time, which represents how long (*e.g.*, in seconds, minutes, or hours) it may take until the protection is deployed and active, and (v) the budget available by the user to pay for a protection. In order to achieve a better-personalized recommendation or filter, the user can also, by using the web-based interface, specify the priority of the following parameters: deployment time, leasing period, or budget. For instance, a budget priority set to *high* would lead to the best money-saving protection service with the other characteristics close to the chosen ones.

4 Proof-of-Concept and Case Study

A Proof-of-Concept (PoC) was implemented to showcase the *ProtectDDoS*. The source code is available online [13]. The *User Layer* was implemented using *ReactJS 16.8*, a popular JavaScript library for building user interfaces, in its latest version. This library facilitates the overall process of develop the user interface components, using the *JSX* syntax extension. It also boosts productivity and facilitates further maintenance. The *Service Helper*, *Catalog*, and *Recommendation Manager* were implemented using *Python 3.6.5*, while the *MENTOR*’s API is implemented using *Flask 1.0.2*. *SQLite 3.30.1* was defined as the database to store information at the *Data Layer*, and its connection is implemented by using the *SQLAlchemy 1.3*, which is an open-source SQL toolkit and object-relation mapper. For the *BC layer*, Ethereum was defined as the BC technology to be

used. For the SC development, *Solidity*, a well-known contract-based programming language for Ethereum, was used.

A case study is conducted to provide evidence of the feasibility and the usability of *ProtectDDoS*. For this case is considered a scenario where (i) a protection provider wants to submit a new protection to be listed in the platform and (ii) a user wants to contract reactive DDoS protection against an application layer attack that is affecting his/her infrastructure. However, the user has a budget of US\$ 5000, and the protection has to be deployed in a server running in Europe, in order to ensure the legal aspects of the General Data Protection Regulation (GDPR). The interface to configure such requirements is publicly available at [13].

Firstly, protections providers have to submit new services to be listed on the platform, populating the catalog with different protections against DDoS attacks. This is done through a *Service Upload Tab*. Each protection service comprises two parts: *General Information* and *Technical Details*. The generated hash and the provider account address is then stored in the BC for further validations. The Metamask extension is used to enable users/providers to send transactions (e.g., hash of protections and feedbacks) to be stored on the BC. The costs involved in this interaction are discussed in Section V.

After populating the database, protection services are made available in the platform's catalog for the user. After configuring their demands, the user can upload a fingerprint of the DDoS attack to filter services that are more suitable to protect against this attack. This is done automatically by the *ProtectDDoS* by processing the fingerprint and extract useful information that gives evidence of the attack type. After submitting their demands, the filter is applied, and a list of protections that are suitable for this case is available. This list is then sent to the *MENTOR* recommender system in order to receive an ordered list with the most recommended protection on the top.

The user can verify at any time whether the offered protection service has been manipulated or not, i.e., validate the integrity and origin of the protection information being provided. In order to do so, the *Service Hash* and *Transaction hash* are required. This information can be easily obtained by clicking on the *See More* button of a specific service. At this point, the user could either copy the service hash or have a closer look at the transaction itself by clicking on the transaction hash. If the user decides to go for the second option, an *Etherscan* page will be opened, which will, in turn, provide further details regarding the transaction itself. Otherwise, if the user decides to go for the first option, an Ethereum account and the browser extension Metamask will be required to execute the validation. Thus, through the *Verify Page*, the user can fastly validate a particular service by its hash. In the verification interface, the protection is verified, meaning that this particular service is stored onto the BC and the hash is corrected (i.e., the information regarding the protection was not modified after its submission). However, in this case, the provider linked to this service hash is highlighted as untrusted, meaning that the real identity of this provider can not be ensured.

Furthermore, the user can access the web-based interface and provide feedback regarding a previously contracted protection, which includes a rating from *zero* to *three*, comments, and a log file (*e.g.*, pcap format) containing the proof-of-feedback. This proof-of-feedback is stored in the platform database and its hash stored in the BC in order to ensure that changes in the log can be easily identified during further auditions or analysis. By using such a reputation, the platform can be configured to remove from the recommendation process or even from the catalog protections that are presenting misbehavior, such as not delivering the promised functionalities or with a bad performance to mitigate the specified DDoS attack.

5 Functional Evaluation

Despite the benefits introduced by the platform, drawbacks have to be considered when using a public BC. The drawbacks are mainly regarding additional fees and time to store information. The fees are not high but should be considered in order to store, for instance, a large number of protections and its reputations. An analysis of the current state of Ethereum BC was conducted to investigate costs. In the Ethereum BC, there are fees for every transaction that requires to store data in an SC. This fee is described as Gas, which is the pricing value paid for the miners to successfully conduct to execute a transaction or execute a contract on the BC. This fee is paid using Ether (ETH), which is the cryptocurrency used in the Ethereum BC. In the Ethereum, besides the ETH, their fees can also be represented as a sub-unit called Gwei. This sub-unit is used to describe small amounts of ETH since 1 ETH is ≈ 1 billion of Gwei. For the costs analysis, the price of 1 ETH is equal to US\$ 144, as of the quotation in December 2019.

To execute the functionalities provided by the SC, the contract needs to be first compiled and successively deployed to the desired network. When doing so, the owner of the SC will be confronted with costs that occur only once, *i.e.*, during deployment. The deployment of the latest, fully working SC at the time of writing generated a total cost of 0.01041256 ETH, which amounts to US\$ 1.50. This cost can be further broken down into two main components: 520,628 units of Gas used to deploy the actual contract and 20 Gwei gas price paid per unit. Important to notice is that whenever the SC gets updated, the owner will have to deploy it again, and if a new feature is added to the SC, the cost will increase. In addition, then there is a cost of 0.0076 ETH ($\approx$ US\$ 1.10) to add a new provider as *Verified*. Such costs can be paid by the owner of the catalog or even by providers that want to announce on the platform.

When designing the functionality of storing a protection service to the BC, two possible approaches were investigated: *(i)* store the full protection service information or *(ii)* store only a hash of the protection. Although the approach *(i)* enables user to, eventually, verify every characteristic of the protection, the costs of writing large amounts of data on the BC increase exponentially. Therefore, the approach *(ii)* is a more suitable alternative in terms of costs since the amount to be paid to store a new protection service is lower than writing the full protection's

information. Upon submission, the provider would pay 0.002154 ETH ($\approx \text{US\$ } 0.31$) to store the generated hash and its address. In case a new account address for the generated hash has been stored, the system would allow to store and submit the service again with a cost of 0.001082 ETH ($\approx \text{US\$ } 0.16$). Also, there are costs concerning the storage of ratings provided by the user and the reputation of each protection service. This cost has to be paid by the SC owner (*i.e.*, the platform) not to encumber the user with this fee. It is important to mention that there are no fees to retrieve information from the SC. Hence, the functions *verifyService()* and *getReputation()* does not have any cost involved.

6 Summary and Future work

This paper presented *ProtectDDoS* a web-based platform that introduces a trustworthy catalog and recommendation of protections against DDoS attacks. *ProtectDDoS* builds on BC-based SCs to allows the validation of the integrity and the origin (*i.e.*, provider) of protections available. Also, by using SCs, the reputation of protections can be stored in a tamper-proof way. Moreover, *ProtectDDoS* explores the recommendation of protections by integrating with a cybersecurity recommender system, thus allowing users to receive recommendations of the best protections according to specific demands. *ProtectDDoS* also allows, through a user-friendly web interface, the upload of DDoS attacks fingerprints and the configuration of different parameters to specify user’s specific demands and characteristics of attacks in order to find the most suitable protection against a DDoS attack. The feasibility of the solution was evaluated in a prototype implementation and the dedicated case study discussion. Also, an evaluation was provided in order to measure the benefits and drawbacks of blockchains (*e.g.*, additional costs).

Future work includes *(i)* the support to lease of protections directly from the platform by using SCs, thus storing and enforcing automatically the agreements between providers and users, *(ii)* the development of mechanisms to process and extract meaningful information from different configurations and log files provided by users, thus extending the information supported by *ProtectDDoS*, and *(iii)* the proposal of DDoS visualizations to help users to understand attack behaviors as well as performance of contracted protections. Also, an in-depth analysis of the recommendation process and the performance of protections recommended for each DDoS attack has to be conducted. Furthermore, the integration with cybersecurity economics-aware solutions [9] might be done in order to provide a more accurate and cost-effective offering and recommendation of protections.

Acknowledgements

This paper was supported partially by *(a)* the University of Zürich UZH, Switzerland and *(b)* the European Union’s Horizon 2020 Research and Innovation Program under Grant Agreement No. 830927, the CONCORDIA project.

References

1. Abhishta, R. Joosten, L. J. Nieuwenhuis: Comparing Alternatives to Measure the Impact of DDoS Attack Announcements on Target Stock Prices. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)* **8**(4), 1–18, December 2017
2. Cybertango: The Cybersecurity Directory - DDoS Protection Companies, 2019, <https://www.cybertango.io/cybersecurity-vendors/DDoS>, Last visit May 1, 2020
3. M. F. Franco, B. Rodrigues, B. Stiller: MENTOR: The Design and Evaluation of a Protection Services Recommender System. In: 15th International Conference on Network and Service Management (CNSM 2019). Halifax, Canada, October 2019, pp. 1–7
4. B. Hellard: DDoS attacks could cost the UK £1bn, 2019, <https://www.itpro.co.uk/security/33279/ddos-attacks-could-cost-the-uk-1bn>, Last visit May 1, 2020
5. T. Li, G. Convertino, R. K. Tayi, S. Kazerooni: What Data Should I Protect?: Recommender and Planning Support for Data Security Analysts. In: 24th International Conference on Intelligent User Interfaces (IUI 2019). ACM, Los Angeles, USA, March 2019, pp. 286–297
6. S. Mansfield-Devine: The Growth and Evolution of DDoS. *Network Security* pp. 13–20, October 2015
7. T. Moore: Introducing the Economics of Cybersecurity: Principles and Policy Options. In: Workshop on Deterring CyberAttacks. Washington, DC, USA, April 2010, pp. 1–21
8. N. Polatidis, E. Pimenidis, M. Pavlidis, H. Mouratidis: Recommender Systems Meeting Security: From Product Recommendation to Cyber-Attack Prediction. In: G. Boracchi, L. Iliadis, C. Jayne, A. Likas (eds.) *Engineering Applications of Neural Networks*. Springer, Athens, Greece, August 2017, pp. 508–519
9. B. Rodrigues, M. F. Franco, G. Paranghi, B. Stiller: SEconomy: A Framework for the Economic Assessment of Cybersecurity . In: 16th International Conference on the Economics of Grids, Clouds, Systems, and Services (GECON 2019). Springer, Leeds, UK, September 2019, pp. 1–9
10. J. Santanna, K. van Hove: DDoSDB: Collecting and Sharing information of DDoS attacks, 2019, <https://ddosdb.org/>, Last visit May 1, 2020
11. E. Scheid, M. Keller, M. F. Franco, B. Stiller: BUNKER: a Blockchain-based trUsted VNF pacKagE Repository. In: 16th International Conference on the Economics of Grids, Clouds, Systems, and Services (GECON 2019). Springer, Leeds, UK, September 2019, pp. 1–8
12. W. Sonnenreich, J. Albanese, B. Stout, et al.: Return On Security Investment (ROSI)- A Practical Quantitative Model. *Journal of Research and practice in Information Technology*, **vol. 38**, 45–52, 2006
13. E. Sula, M. Franco: Web-based Interface for the Recommendation of DDoS Attack Protections, 2019, <https://gitlab.ifi.uzh.ch/franco/ddosrecommendation>, Last visit May 1, 2020
14. T. Bocek and B. Stiller: Smart Contracts - Blockchains in the Wings. *Digital Marketplaces Unleashed*, Heidelberg, Germany, January 2017