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 businesses on digital services increases, their vulnerability to cyberattacks increases, too. Besides providing innovative services, business owners must focus on investing in robust cybersecurity mechanisms to countermeasure cyberattacks. Distributed Denial-of-Service (DDoS) attacks remain one of the most dangerous cyberattacks, *e.g.*, leading to service disruption, financial loss, and reputation harm. Although protection measures exist, a catalog of solutions is missing, which could help network operators to access and filter information in order to select suitable protections for specific demands.
This work presents *ProtectDDoS*, a platform offering recommendations of DDoS protections. *ProtectDDoS* provides a blockchain-based catalog, where DDoS protection providers can announce details regarding their services, while users can obtain recommendations of DDoS protections according to their specific demands (*e.g.*, price, attacks supported, or geolocation constraints). *ProtectDDoS*'s Smart Contract (SC) maintains the integrity of data about protections available and provides tamper-proof reputation. To evaluate the feasibility and effectiveness of *ProtectDDoS*, a prototype was implemented and a case study conducted to discuss costs, including interactions with the SC.

Keywords: Cybersecurity · DDoS Protection · Recommender System · Smart Contract (SC) · Marketplace.

1 Introduction

Denial-of-Service (DoS) attacks represent a significant threat to any commercial organization and individuals, which rely on 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. Consequently, the targets affected (*e.g.*, companies and governments) are usually confronted with economic impacts. Not only do these attacks cause financial damages due to the unavailability of services and loss of online traffic, but in

critical cases, they inflict long-term damage to the corporate reputation, causing drastic drops of stock prices [1].

Furthermore, the number of DDoS attacks has almost tripled in the last three years, with an averaged financial loss of dozens of thousands of USD (US Dollar) per hour of such an attack. *E.g.*, it was estimated that in 2009 only in the United Kingdom (UK) DDoS did cost more than USD 1 billion [4], which includes revenue losses 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 out that the number of DDoS attacks globally will reach 17 million by 2020, causing several economic and societal impacts.

Based on this threat landscape, large companies and governments are spending about USD 124 billion on information security products and protection services. However, many of the problems plaguing cybersecurity are economic in nature [7]. Often systems fail, because organizations do not bear to assess 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 an 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.

As of today, the variety of DDoS protection services has increased as well. While competition in this sector may show 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 with the selection of a DDoS protection can support the organization in the decision-making process. More specifically, by providing the user with essential information related to the many DDoS protection services available, taking into account filters and characteristics of the cyberattack (*e.g.*, fingerprints and log files), the user may simplify decisions. However, there are no intuitive solutions (*e.g.*, dashboards) that ease the access to a broad set of DDoS protections, while ensuring the integrity of the information from 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 presents *ProtectDDoS*, a blockchain-based platform for the offering and support of an recommendation for 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 applying different parameters, such as price, the type of DDoS attack supported, and deployment time. In addition, DDoS attacks fingerprints [10] can be used

as an input to find the most suitable protection for a determined type of attack. This paper's **contributions** are summarized as follows:

- 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 certain scenario determined.
- A dashboard is offered, fully integrated with a recommender system for protection services, called *MENTOR*, allowing the user to use a Web-based interface to obtain a recommendation of the most suitable solution according to his/her demands and predefined filters.

The remainder of this paper is organized as follows. Background and related work are reviewed in Section II. Section III introduces the platform for offering and recommending DDoS protections, including implementation details. Section IV discusses the feasibility of the solution proposed, and a case study is presented. Section V provides a functional evaluation in order to measure the costs of the *ProtectDDoS*. Finally, Section VI summarizes the paper and comments on future work are added.

2 Background and Related Work

As businesses strengthen their digital dependency, they also become more vulnerable to cyberthreats. Therefore, besides the need for speed in 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:

1. **Identify** security risks and associated costs and *(ii)* determine impacts of cybersecurity in the business or service. In turn, it is possible to estimate overall impacts (*e.g.*, financial loss occasioned by a business disruption) in order to decide whether to invest in cybersecurity.
2. **React** against an imminent cyberattack or **assume** risks, paying for the damage or delegating that to third-parties (*e.g.*, cyberinsurers).

For (1) such overall estimations can be done using different approaches. For instance, the Return On 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 determine a plan to prevent cyberattacks and its impacts proactively. In the context of (2) and once an attack happens, prevention is cheaper than reactions, when an attack already surpassed the infrastructure. If companies do not invest correctly in cybersecurity, the security of their operations depends on luck and impacts of attacks can be devastating, which is not acceptable by companies that have to maintain reputation.

The market for protection services has grown together with investments in cybersecurity. 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 listing 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 investments in cybersecurity. In only one such a repository 1,200 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 is still a lack of platforms that use such information to simplify the decision-process and cybersecurity planning of companies. Table 1 provides a comparison of different cybersecurity-oriented solutions that implement approaches to offer or recommend services.

Table 1: Comparison of Related Work in Terms of the Functionality Designed

-	**Functionalities**				
Solution	**User-friendly Catalog**	**Supports Recommendation**	**Filters**	**Allows Integrity Verification**	**Reputation Mechanisms**
MENTOR	No	Yes	Yes	No	No
Cybertango	Yes	No	Yes	No	No
Tiany et al.	No	Yes	Yes	No	No
Polatidis et al.	No	Yes	Yes	No	No
BUNKER	Yes	No	No	Yes	Yes
ProtectDDoS	Yes	Yes	Yes	Yes	Yes

In previous work, the recommender system for protection services called *MENTOR* was introduced to help during the decision of which is the most suitable protection for demands determined [3] in which a recommendation engine that can suggest recommendations based on a list of parameters and user demands. However, *MENTOR* is still in early stages and does not yet provide user interfaces or a catalog for protection providers to submit their solutions. Furthermore, the reputation of protections based on user feedback is not being considered during the recommendation process. The work of [8] provides a recommender system to predict cyberattacks by identifying attack paths, demonstrating 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 builds protection plans. However, none of them supports neither characteristic of DDoS attacks nor intuitive interfaces for users to add their demands nor log files to receive recommendations.

By using the concepts of Blockchains (BC) and Smart Contracts (SC), different solutions have been proposed to enable the validation of integrity and origin of solutions for different purposes. BCs were initially conceived as a dis-

tributed ledger to be the backbone of the Bitcoin cryptocurrency. However, BCs capacity to provide an immutable, trustworthy, and decentralized collection of records has attracted the attention of both industry and academia [14]. The concept of SCs is implemented by the second generation of BCs, such as Ethereum and NEO. Fees involved in SCs are lower than for traditional systems requiring a trusted intermediary. [11] introduces *BUNKER*, a BC-based marketplace for Virtual Network Functions (VNFs), to provide immutable and trusted information concerning VNF packages acquired by end-users. This solution stores the hash of VNF packages in a BC to guarantee the integrity of the VNF being acquired by end-users. This feature is useful for both providers and users interested in protections, since the integrity of the protection, the provider's identity, and its reputation can be verified for any offered solution, before users decide on one specific cybersecurity solution.

3 The *ProtectDDoS* Platform

The *ProtectDDoS* platform allows users to describe their demands for protections in order to obtain a proper level of protection against different types of DDoS attacks from an extensive list of options available, which facilitates the decision process to select the most suitable protection. These protection services can be acquired proactively before an attack happens or acquired to react during an imminent attack. Thus, *ProtectDDoS* offers mechanisms to support 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, *ProtectDDoS* allows, through a Web-based interface, users to *(i)* upload fingerprints of DDoS attacks to find specific protections, *(ii)* verify, supported by the BC, the integrity and origin of information of different protections, *(iii)* receive the recommendation of the best solution according to its demands, and *(iv)* provide feedback of contracted protections, thus, supporting a reputation system for protections available. The *ProtectDDoS*'s code is publicly available at [13].

3.1 Architecture

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

using the *MENTOR* API (Application Programming Interface) , which is fully integrated with the *ProtectDDoS* architecture, thus, allowing for calls to receive a recommendation of the best protection service according to previously defined filters and configurations of the user.

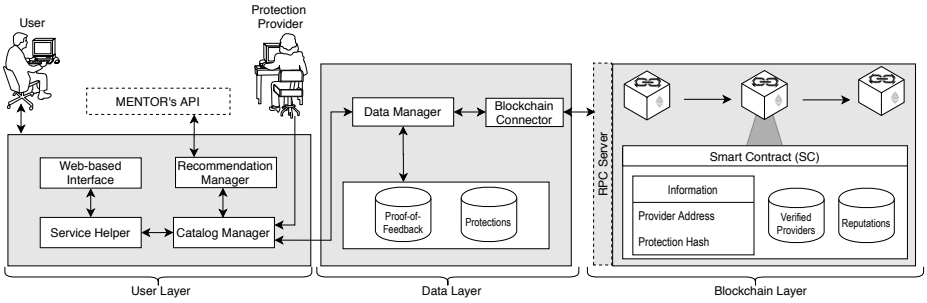

Fig. 1: *ProtectDDoS*'s Conceptual Architecture

The *User Layer* provides a *Web-based interface* (*i.e.*, dashboard) access to the catalog as well as details 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 filters predefined on the entire dataset of protections available, thus, removing protections that are not suitable for user's requirements. The *Catalog Manager* requests information from the Data Layer to build the catalog of protections available, applies these filters, and sends 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 user requirements and information from 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 protections available, such as developer, name, price, and types of attacks supported. Also, a database is provided to store all log files (*e.g.*, pcap) containing information regarding the contracted protection performance, 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 to the *Data Layer* and is in charge of the process to answer requests for information. Furthermore, the *BC Connector* is an adaptor, implemented to enable communications with the SC running inside the BC. The *BC Connector* performs calls to interact with the SC (*e.g.*, verify the protection hash or validate the provider address) by sending BC transactions though a *Remote Procedure Call (RPC) server* provided by the BC.

Finally, the *BC Layer* deploys the SC to store a list of verified providers based on their address on the BC, 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 this information is immutable, which allows any interested party to audit the information following the full history of the information stored.

3.2 Workflow

By accessing the *ProtectDDoS*, users interested in obtaining protection can verify available protection services in a catalog and apply filters to select a set of characteristics that satisfy his/her demands, such as a maximum price or protection against a specific type of DDoS attack (*cf* Figure 2). For that, the user can select a determined attack type from a list of attacks supported or also upload a file containing fingerprints of that DDoS attack for which protection is required. This fingerprint input is used to process the filtering of a list of protections suitable for such a demand. Such a list can be sent to the *MENTOR* recommender system through the API provided in order to receive, as a response, the best protection selected by the recommendation process implemented in *MENTOR*.

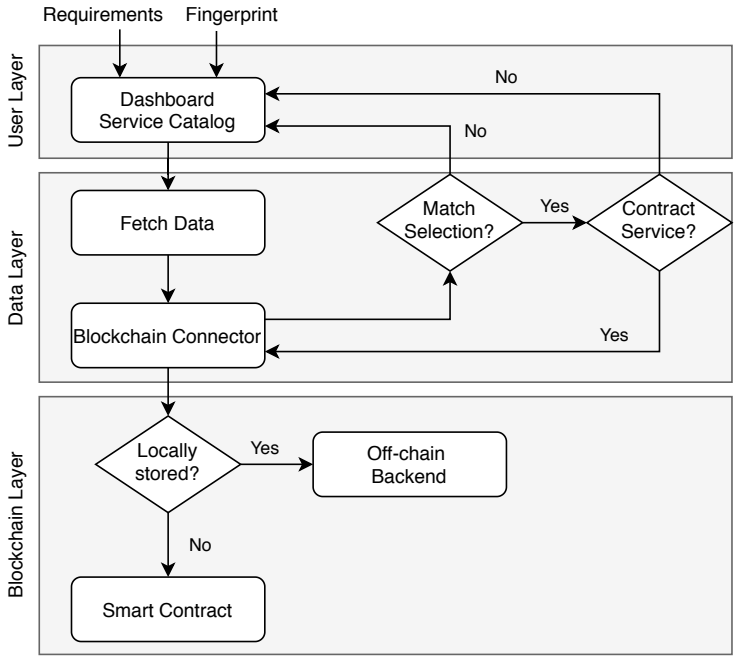

Fig. 2: *ProtectDDoS*'s Workflow

Also providers of protection services can access the Web-based interface and store new protections to become available in the catalog. The process of uploading a new protection service comprises of two essential steps:

1. The service information is hashed, using the SHA-256 algorithm and subsequently submitted to the BC.
2. Upon successful storage, the service provider's address and the transaction hash are retrieved by the RPC server and stored off-chain.

The first operation is handled using an SC and will cost a fee to be completed (*cf.* Section V). As the costs to store all information required would be high, only the hash of this service information is stored. When interacting with an SC, an Ethereum account is required. Ethereum was used, since it offers a clear path program SCs in Solidity. The second operation retrieves the hash of any protection from the BC, thus, offering to any user checks, whether this service information has been compromised or the provider cannot be verified. Hence, the service hash is required and a specific function in the SC is invoked.

Also, a DDoS fingerprint is supported by *ProtectDDoS* as defined in the DDoSDB platform [10]. Parameters that be configured for the catalog filter or recommendation of protections and include: *(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 and 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 determined 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 fund this protection.

4 Proof-of-Concept and Case Study

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

A case study was conducted to provide evidence of the feasibility and usability of *ProtectDDoS*. This case considers 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 a reactive DDoS protection against an application layer attack that is affecting his/her infrastructure. The user holds a budget of USD 5,000.

The protection has to be deployed in a server running in Europe to ensure legal compliance to the General Data Protection Regulation (GDPR). The interface to configure such requirements is available publicly, too [13].

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

After populating the database, protection services are made available within the platform's catalog for the user. After configuring his/her demands, the user can upload a fingerprint of the DDoS attack to filter services that are suitable to protect against this attack. This is done automatically by *ProtectDDoS*, which processes the fingerprint and extracts useful information, providing evidence of the attack type. After submitting user demands, the filter is applied and a list of protections suitable for this case is available. This list is sent to the *MENTOR* recommender system in order to receive an ordered list with the most recommended protection on the top. Based on this list containing suitable services, for example, the recommendation engine can decide that the best option is a service with a deployment time in seconds, which includes features to mitigate this type of attack with the cost of USD 2,400. Although other solutions may be cheaper, they are providing features that are not considered ideal, taking into account all user demands and the fingerprint of the attack being addressed.

The user can verify, whether the protection service offered has been manipulated or not, *i.e.*, by validating the integrity and origin of the protection information being provided. Thus, the *Service Hash* and *Transaction Hash* are required. This information can be obtained by clicking on the *See More* button of a specific service. At this point, the user can either copy the service hash or have a closer look at the transaction itself by selecting the transaction hash. If the user decides to go for the transaction hash, an *Etherscan* page will be opened, which will provide further details regarding the transaction itself. Otherwise, if the user decides select the service hash, an Ethereum account and the browser extension Metamask will be required to execute the validation. Thus, through the *Verify Page*, the user can quickly validate a particular service by its hash. Within this verification interface the protection is verified, meaning that this particular service is stored inside the BC and the integrity of this service is ensured (*i.e.*, the information regarding the protection was not modified after its submission). However, in this case here, the provider linked to this service hash is highlighted as untrusted, meaning that the real identity of this provider cannot 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's database and its hash is stored in the BC in order to ensure that changes in the log can be identified during further audits or analysis. By using such a reputation approach, the platform can be configured to remove from the recommendation process or even from the catalog protections that are representing misbehavior, such as not delivering the functionality promised or with a worse performance while mitigating the DDoS attack specified.

5 Functional Evaluation

Despite these benefits introduced by the *ProtectDDoS* platform, costs and security have to be considered upon using a public BC.

5.1 Costs

Costs are concerned with additional fees and the time to store information. These fees are not high, but should be considered to store, for instance, a large number of protections and their reputations. Thus, an analysis of the current state of the Ethereum BC was conducted to investigate costs. Fees exist for every transaction that requires to store data in an SC. This fee is described in "Gas", which is the price being paid for BC miners to successfully conduct the execution of a transaction or a contract. This fee is paid using Ether (ETH), which is Ethereum's cryptocurrency. Besides ETH, fees can also be represented in sub-units "Gwei": 1 ETH is ≈1 billion Gwei. For the costs analysis, the price of 1 ETH was equaling USD 144 as of the quotation in December 2019.

To execute the functionality provided by the SC, the contract needs to be compiled and successively deployed to the desired BC network. At this moment the owner of the SC will be confronted with costs that occur only once, *i.e.*, , during the deployment. The deployment of the latest, fully working SC here at the time of writing generated a total cost of 0.01041256 ETH, which amounts up to USD 1.50. This cost can be broken into two main components: 520,628 units of Gas used to deploy the actual contract and a 20 Gwei gas price paid per unit. Important to note is that whenever the SC is 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, the cost of 0.0076 ETH (≈ USD 1.10) resembles to add a new provider as *Verified*. Such costs can be paid by the owner of the catalog or by providers that want to announce themselves on the platform.

Upon the design of the functionality to store 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 users 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. Upon submission, the provider paid *0.002154 ETH* (≈ *USD 0.31*) to store the hash generated and its

address. In case a new account address for the hash generated has been stored, the system allows for a storage and submission of this service again with a cost of *0.001082 ETH* (≈ *USD 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) ensuring that the user is not burdened with this fee. It is important to mention that there are no fees to retrieve information from any SC. Hence, the functions *verifyService()* and *getReputation()* do not show any cost involved.

5.2 Security

One of the main characteristics of a BC is its ability to unearth, causing applications to remove trusted third parties, while trust levels can be relatively increased by the transparency and immutability of the process. In the context of security applications, such as the *ProtectDDoS*, two additional concerns exists with the exposure of confidential data and the handling of protection service requirements. Therefore, it is important to consider the solution's deployment approaches in order to ensure that the information stored is not exposed or tampered with. In this sense, a possible deployment absorbing requests from multiple clients (*i.e.*, on external premises) implies a centralization process, which is just the opposite of the decentralization proposed by BCs (*cf.* Figure 3).

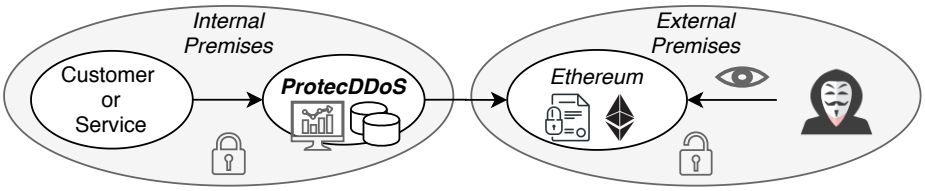

Fig. 3: *ProtectDDoS*'s Deployment

In this sense, Figure 3 presents an ideal implementation approach of the service as a decentralized application and is maintained within internal premises. Thus, the *ProtectDDoS* platform operates as a decentralized application, where public data on protection services are announced and the instance in internal premises can act as a reverse proxy selecting, among services advertised, which ones have all characteristics desired. Similarly, protection service advertisers also operate instances of the *ProtectDDoS* platform on internal premises. Henceforth, aspects of confidentiality and integrity related to the security needs of customers are maintained on internal premises and characteristics of the service advertised cannot be tampered with either.

6 Summary and Future Work

This paper developed and evaluated *ProtectDDoS* a Web-based platform that introduces a trustworthy catalog and recommendation of protections against DDoS attacks. *ProtectDDoS* builds on BC-based SCs to allow for the validation of integrity and the origin (*i.e.*, provider) of protections available. Also, by using SCs the reputation of protections can be stored in an immutable manner. Moreover, the *ProtectDDoS* platform explores the recommendation of protections by integrating them with the cybersecurity recommender system *MENTOR*, thus, allowing users to receive recommendations of the best protection according to specific demands. *ProtectDDoS* also allows through the user-friendly Web interface the upload of DDoS attack fingerprints and the configuration of different parameters to specify specific user 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 prototypical implementation based on the dedicated case study. The evaluation provided measures the benefits and additional costs in the context of BCs in use.

Future work includes *(i)* the support of leasing protections directly from the platform by using SCs, thus, storing and enforcing automatically respective 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 and the performance of protections contracted. Furthermore, an in-depth analysis of the recommendation process and the performance of protections recommended for each DDoS attack will be conducted. Finally, an integration of *ProtectDDoS*with cybersecurity economics-aware solutions [9] might be performed in order to provide for an 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