DefenseChain: Consortium Blockchain for Cyber Threat Intelligence Sharing and Defense

Soumya Purohit, Prasad Calyam, Songjie Wang, RajaniKanth Yempalla, Justin Varghese
University of Missouri, Email: {spvm7, rypbc, jv8dz}@mail.missouri.edu, {wangso, calyamp}@missouri.edu

Abstract—Cloud-hosted applications are prone to targeted attacks such as DDoS, advanced persistent threats, cryptojacking which threaten service availability. Recently, methods for threat information sharing and defense require co-operation and trust between multiple domains/entities. There is a need for mechanisms that establish distributed trust to allow for such a collective defense. In this paper, we present a novel threat intelligence sharing and defense system, namely “DefenseChain”, to allow organizations to have incentive-based and trustworthy co-operation to mitigate the impact of cyber attacks. Our solution approach features a consortium Blockchain platform to obtain threat data and select suitable peers to help with attack detection and mitigation. We propose an economic model for creation and sustenance of the consortium with peers through a reputation estimation scheme that uses ‘Quality of Detection’ and ‘Quality of Mitigation’ metrics. Our evaluation experiments with DefenseChain implementation are performed on an Open Cloud testbed with Hyperledger Composer and in a simulation environment. Our results show that the DefenseChain system overall performs better than state-of-the-art decision making schemes in choosing the most appropriate detector and mitigator peers. In addition, we show that our DefenseChain achieves better performance trade-offs in terms of metrics such as detection time, mitigation time and attack recurrence rate. Lastly, our validation results demonstrate that our DefenseChain can effectively identify rational/irrational service providers.

Index Terms—Blockchain, Threat Intelligence Sharing, Distributed Trust, Cyber Security, Reputation System

I. INTRODUCTION

Cloud-hosted services are targeted by ever-growing Distributed Denial of Service (DDoS) attacks that aim to disrupt the service of major industries, conglomerates and community organizations [1]. Attacks such as Advanced Persistent Threats (APTs) also cause economic damage and leakage of sensitive information through sophisticated malicious attack codes [2]. Another targeted attack type can be seen in the cryptojacking attacks [3], where criminals compromise enterprise resources for illegal bitcoin mining revenue gains.

To defend against such targeted attacks, a co-operative and collaborative attack threat intelligence sharing platform can help raise the situational awareness and foster mechanisms to protect targeted assets through pertinent detection and mitigation of attacks. The platform can produce proactive measurements and actionable information that can be available to multiple domains/entities in a federation [4]. Fig. 1 illustrates various threat sharing scenarios supported by an exemplar platform, where organizations gain actionable threat data by paying an admission fee. Moreover, organizations under a line of attack in close proximity can leverage the platform to form alliances for collaborative defense by sharing the burden [5]. However, such beneficial platform scenarios are vulnerable to ‘false reporting’ and ‘free riding’ issues [6]. False reporting is caused by federation members maliciously reporting cyber attack incidents in order to waste resources of the other federation organizations. Additionally, free riding can occur when an organization takes advantage of platform services without making contributions to the federation in cases that require member co-operation for collective benefit.

Creation of threat intelligence sharing platforms requires overcoming other substantial challenges that include issues such as: why should one domain share its threat intelligence information with another domain? How can we opt-in the domains/entities that are proximal (i.e., in geographic distance or units that are distributed but belong to the same organization) or distant (i.e., relatively far in geographic distance or belonging to different organizations) for collective attack defense? How can the platform be used for co-ordinated threat detection and attack impact mitigation in a timely manner with distributed trust? A subset of these issues have been addressed in prior works using methods such as crowdsourcing with incentives [7] [8]. Reputation systems also have been proposed with algorithms to counter the impact of having false reporting and free riding peers [9] [10] [11]. However, there is a lack of works that use Blockchain solutions that can potentially be used to establish distributed trust, integrate reputation systems and create automated access control for threat intelligence data.
sharing in a scalable and transparent manner.

In this paper, we address the above threat intelligence sharing challenges and propose a novel “DefenseChain” platform for a two-stage (i.e., attack detection followed attack impact mitigation) cyber defense using a Blockchain reference architecture. Specifically, we adopt a permissioned/consortium Blockchain architecture whose benefits include: (a) relatively short deployment duration and less resource-intensive properties in terms of the consensus mechanism, and (b) more effective than a permissionless/public Blockchain architecture for protected data sharing amongst a federation of organizations. Our solution approach also features a novel reputation system and uses a set of protocols to rate the peers objectively in terms of ‘Quality of Detection’ (QoD) and ‘Quality of Mitigation’ (QoM) metrics for cyber defense. The values of the metrics are uniquely calculated using our prior work on the Dolus system that uses a “attack defense by pretense” paradigm to counter targeted attacks such as DDoS, APTs and cryptojacking [12]. These metrics are also utilized in an economic model that we propose for creation and sustenance of the consortium with proximal and distant peers in manner that eliminates false reporting and free-riding issues by e.g., charging a fine in non-ideal cases, and integrating incentives for successfully servicing detection and mitigation requests within deadlines.

Lastly, we implement our DefenseChain platform using the Hyperledger Composer in a NSF Cloud testbed [13]. Our experimental testbed is realistic and comprises of a federation of domains, including a number service provider peers cooperating with different domains in order to perform attack detection and impact mitigation independently, as well as a set of users trying to access a targeted application server, which is being disrupted by a set of attackers. Based on experiment results from this testbed and simulation experiments, we show the benefits of our novel co-operative real-time threat intelligence sharing platform capabilities in comparison with state-of-the-art solutions such as [14] [15] [16] and [6].

The main contributions summary of this paper is as follows:

- We propose a consortium Blockchain based “DefenseChain” platform for real-time threat intelligence sharing as part of a ‘defense by pretense’ strategy.
- We equip our DefenseChain platform to provide distributed trust through analysis of QoD and QoM of peers to help a requester to choose the effective detector(s) and mitigator(s) to defend against targeted cyber attacks.
- We build a reputation system in DefenseChain using an economic model to rate QoD and QoM using both objective and subjective metrics for providing appropriate reward/penalty to the co-operating rational/irrational peers. The reward/penalty uses metrics such as detection time, mitigation time and attack recurence rate.
- We evaluate our DefenseChain implementation through comparisons with state-of-the-art schemes for decision making in choosing the best detector and mitigator peers. Our results show that DefenseChain outperforms the existing schemes by dynamically providing mitigation strategies through enforcement of chaincode-based policies to counter impending/active cyber attacks.

The remainder of the paper is organized as follows: Section II discusses prior related works. Section III presents background on threat intelligence sharing requirements and details various system components within a Blockchain reference architecture. Section IV describes the performance evaluation experiments and results. Section V concludes the paper.

II. RELATED WORKS

A. Threat Intelligence Sharing

Due to the constant increase in the number and complexity of cyber attack incidents, organizations are eager to have proactive and actionable knowledge for efficiently defending their valuable assets i.e., cloud-hosted applications. Towards this end, they need to develop the practice to share threat intelligence information amongst their peers in order to effectively and collectively detect cyber attacks, and stand up robust defenses that mitigate the attack impact on their assets.

Several works have been performed to enable cyber defenders to explore threat intelligence sharing capabilities and construct effective defenses against the ever-changing cyber threat landscape. The authors in [17] and [18] identify gaps in existing technologies and introduce the Cyber Threat Intelligence model (CTI) and a related cyber threat intelligence ontology approach, respectively. The work in [19] details a novel approach based on Structured Threat Information eXpression (STIX) to deal with system diversity during threat information sharing. An encryption strategy for threat intelligence sharing is proposed in [20] in the form of a privacy preserving protocol. The CYBEX work in [21] details an incentivized approach and uses the concept of an admission fee, as well as interaction models organizations for cybersecurity information exchange to defend against attackers in a dynamic game.

The novelty of our work is in the design of a threat intelligence sharing platform using consortium Blockchain in order to implement a ‘defense by pretense’ paradigm for cyber defense as detailed in the work on the Dolus system [12]. We adapt the two-stage ensemble learning scheme to trigger cooperation between multiple domains who collectively provide detection and impact mitigation to defend a domain targeted by attackers through DDoS, APTs and cryptojacking.

B. Reputation Systems

Several different reputation systems have been proposed in prior works that address the issues of false reporting and free riding [8], [9], [10]. The work in [8] proposed the design of a crowdsourcing tournament to maximize a service provider’s utility in crowdsourcing and provide continuous incentives for users by rewarding them based on the rank achieved. The authors in [9] presented schemes to eliminate dishonest behavior with the help from a trusted third party. In a related effort [22], a reputation system is developed that overcomes the limitations in decentralized systems and quantifies the reputation by removing human opinion from the transactions. E-commerce applications [23] have also adopted reputation
systems that use Blockchain solutions for implementation of privacy-preserving mechanisms involving Proof of Stake for determining any new block to be accepted instead of accepting the highest difficulty block. The authors in [24] designed a trust model that evaluates trust based on the reputation built up on historical interactions and indirect opinions about the sender. The work in [25] introduces a proxy to transfer reputation values between anonymous contributions, and a reputation anonymization scheme is shown to prevent the inadvertent leakage of privacy.

The closest related work to our work is in [6]. Therein, a reputation and reward scheme is proposed that considers potential information frauds and allows automatic smart contract execution based on malicious peers. We adapt their Beta reputation that is used for probabilistic rating and to identify and reward honest participants. Our work also borrows the idea of using a InterPlanetary File System (IPFS) [26] for creation of the reputation system and to store device attributes as well as threat data in an off-chain manner in our Blockchain architecture. We include the concept of a deposit, and a request/response deadline to eliminate free-riding cases and false reporting similar to the work in [6]. Furthermore, we propose a novel objective evaluation of attack detection and impact mitigation through real-time threat intelligence sharing using novel QoD and QoM metrics. Our reputation system also features a trust-based model implemented using threat detection and attack impact mitigation protocols that are motivated by prior work in [10] for incentivizing domains in a federation to co-operate and trust each other.

C. Blockchain for Building Trust

There have been several studies that utilize Blockchain as a solution in order to solve the problems inherent in traditional transactional models. CrowdBC [27] is an exemplar work that implements a reward/penalty scheme using smart contracts, and explores the ability to abstract a user’s real world identity for providing a unique method to ensure data privacy. In the area of IoT and sensor networks, works such as [28] proposed security models based on Blockchain to ensure the validity and integrity of cryptographic authentication data. A Blockchain-based security model is proposed for forensic evidence preservation [29] in order to allow storage of metadata e.g., pieces of evidence using smart contracts amongst the different entities involved in an investigation process. Similarly, iShare [30] features a security model that leverages Blockchain to collect cyber attack information and shares it across organizations in an anonymous fashion. The anonymity afforded by this approach serves as inspiration to our approach to threat intelligence sharing across a federation of proximal/distant domains. Anonymity issues have also been tackled in [31], where Blockchain is used to enable anonymous reputation estimation as part of establishing privacy-preserving trust for vehicular ad hoc networks.

Our work on DefenseChain is motivated by the above works in the context of designing our reputation system using Blockchain technologies, and for incentivizing federation peers via an economic model based on a deposit fee received from potential detector and mitigator peers.

III. DEFENSECHAIN CONCEPTS AND SYSTEM DESIGN

A. Threat Intelligence Sharing

With the growing intensity and scale of cyber attacks on cloud-hosted applications, it is critical to share the threat intelligence data within multiple domains in order to rapidly detect cyber attack threats and effectively mitigate the attack impacts in a collaborative manner. The shared threat information can be classified as the requester’s IP, attack start and attack stop times, bytes and packets captured, etc. Such information attributes can be communicated across federation peers so that detection and mitigation can be carried out using co-operation strategies that create a win-win for the peers who are impacted and who are willing to offer cyber defense services.

The attack/defense model we consider is similar to the exemplar model considered in the Dolus system [12], where a federation of autonomous systems co-operate using a ‘defense by pretense’ paradigm to effectively block attacker traffic closer to the source side. The defense by pretense strategy also buys time for the cyber defenders by creation of illusion of attack success, while a robust defense strategy is being put in place through the co-operation of the federation peers. In our federation of users, we assume threat intelligence sharing is carried out differently for proximal and distant peers through separate controller nodes, as depicted in Fig 2. The Proximal Controller with a Hyperledger (i.e., an exemplar permissioned Blockchain technology) setup performs the attack traffic redirection from the slave-switch and the root-switch to a Quarantine Virtual Machine (QVM). This action safely allows the requester machine to access the new machine without any disruption. In the case of a Distant Controller offering a mitigation service, the transfer of attack traffic from the slave-switch to the root-switch takes a relatively longer time due to the geographic distance or due to the time needed to establish trust. Similarly, attack traffic redirection to the QVM in the case of the Distant Controller happens same as in the case of the Proximal Controller, however with a delay.

We model the detection and mitigation as two separate steps. The detection involves the identification of vulnerability of attack, suspiciousness score of a domain resource node,
Fig. 3: Proposed DefenseChain reference architecture that features on-chain/off-chain components within a federation of peers involving a cloud-hosted application, dedicated controllers with Hyperledger configurations, IPFS and QVMs integration.

and the attack time duration. Also, the mitigation critically focuses on the type of mitigation performed, redirection of attack traffic and spoofing of the server’s IP. With our proposed economic model detailed later in Section III-E, we address these problems by harnessing an incentive-based approach that develops a foundation of distributed trust.

B. DefenseChain Platform Overview

Fig. 3 illustrates our proposed reference architecture in a federation where a cloud service provider is hosting several servers belonging to different organization peers that may be vulnerable to cyber attack threats. Roles of the peers involved in the federation area defined in Section III-C. The central part of our DefenseChain architecture is the consortium Blockchain-based trust setup created on top of the Dulos defense by pretense implementation as outlined in [12]. Within this federation, we assume that there are organization peers requesting for a detection and mitigation service from cooperating domains. Furthermore, each domain can perform their service using a suitable mitigation strategy such as e.g., moving target defense, defense by pretense, network firewall defense using blacklisting, etc. Our DefenseChain rates the detection and mitigation service quality of the peer(s), and provides the requesting peer(s) with the flexibility to choose the domain that can provide the higher levels of service quality measured through the QoD and QoM metrics that are detailed in Section III-D. Furthermore, through our economic model described in Section III-E, we implement an incentivized approach that allows the mitigator domains to collaborate and also eliminates the issues of free riding and false reporting.

Additionally, our platform design includes on-chain and off-chain components for storage, processing and sharing of threat intelligence information. We elaborate on these components in the following:

On-Chain: this component fetches and displays the details such as e.g., attacker IP, source IP, number of packets, spoofed IP, blacklisted IP from the IPFS. These details are fed into the detection and mitigation chaincodes that initialize and manage ledger state through transactions submitted by applications. In our DefenseChain, they help in calculation of QoD and QoM in a federation of peers, respectively.

Off-Chain: this component stores information such as the packet capture, bandwidth capture and device attributes data. Depending on the number of transactions and the attacks encountered, the storage of the related data will require large amounts of storage (in the order of tera bytes or even peta bytes in core network domain scenarios). For this purpose, we utilize the IPFS concept from [26] as an off-chain storage that interacts periodically through the Oraclize [32] service. The hashes of the IPFS data are referenced in our chaincodes. This approach allows us to deliver a dynamic and efficient data retrieval in a peer-to-peer manner.

C. User Interface

We utilized the Hyperledger Composer playground for configuration, deployment, and testing of our business network i.e., federation. A new federation can be created by a requesting peer organization. The business networks are a combination of identity and profile, and hence they are viable for permission and access control. We created userIds and secret passwords for the peers to connect into a business network. In our security policy, we use a Federated Identity and Access Management scheme, where mapping of peers in the Hyperledger Composer matches with the real-world identity of the peers. Initially, the requester, detector and mitigator roles pay a deposit fee that initiates the transaction process, as shown in Fig. 4. Upon successful validation by a detection or mitigation chaincode, the requester can search and view the detectors/mitigators.

We share the threat information such as assets affected, attack tools, QVM IP addresses, blacklisted IP addresses,
attack data is stored on the controller of the peer node, which is then fetched through the IPFS as shown in Fig. 4. This dynamic fetching off-chain occurs rapidly. At this step, some delays can be experienced depending upon the network performance. Furthermore, proximal peers which are likely to be attacked can benefit from this information by our mitigation protocol. We allow the requester to choose the proximal peers, which are acceptable in the chance of getting attacked based on their geographical location or domain affinity.

D. System Roles

There are four different roles i.e., Requester, Detector, Mitigator and Watchdog that serve as central actors in our DefenseChain implementation. In the following, we provide their definitions:

1) Requester: Requester is an actor who is affected by cyber attacks and submits the detection/mitigation requests to the federation. Once a threat is identified, the requester has the option to search for detector(s) and mitigator(s), and can decide which of the peer service providers are ideal to trust. All the actors have to pay a deposit fee, which includes the transaction fee and a collateral. The transaction fee is refunded in the case a mitigation strategy could not be effectively devised; otherwise, the fee is credited to the detector(s)/mitigator(s) providing the services.

2) Detector: Detector is an actor who provides cyber attack detection services to the attack defense requests from a Requester. Upon providing the service, the Detector receives the transaction fee as payment for a successful detection. Detectors can be identified as \( D = \{ D_1, ..., D_i, ..., D_n \} \) to detect an ongoing attack. To incentivize Detectors to provide high-quality services, a reward \( v \) and a monetary penalty \( \pi_D \) are required as deposit (\( \text{deposit}_D = v + \pi_D \)) in the DefenseChain. This deposit cannot be redeemed before the detection deadline.

3) Mitigator: Mitigator is an actor who provides mitigation services to all the attack defense requests from Requesters. Upon providing the service, the Mitigator receives the transaction fee as payment for successful mitigation. Mitigators are identified as \( M = \{ M_1, ..., M_i, ..., M_m \} \), who accept the mitigation request and provide attack mitigation solutions corresponding to a given attack scenario. Each Mitigator must make a deposit of \( \pi_M \) in the DefenseChain, which serves to significantly reduce the possibility of Sybil and Collusion attacks (see Section III-G for definitions). Efficient strategies by \( M_j \) will result in a corresponding reward \( v_R \).

4) Watchdog: The Watchdog is a system Daemon which is essentially an admin role that is used to: (a) analyse the detection and mitigation data, and (b) rate whether the detection or mitigation has been successful. The Watchdog will also flag false reports by analyzing the data that it monitors. False reporters will be penalized in their transactions and could loose out on their collateral and in their ability to perform future transactions. The output given by the Watchdog determines the rating and reward that will be received by a Detector/Mitigator who claims successful service completion.

If a peer is flagged as a False Reporter in more than 75 percent of its transactions, Mitigators can have the ability to refuse assignment of mitigation services to the False Reporter. Popular Mitigators can have the ability to raise their False Reporter tolerance bar by refusing to accept False Reporters who were flagged in less than 75 percent of their transactions. Watchdog, identified by \( W = \{ W_1, ..., W_m \} \) is the role that analyzes the mitigation solutions provided by Mitigator \( M_j \) for request by Detector \( D_i \). The analysis of the Watchdog determines the efficacy of the strategy submitted by the Mitigator \( M_j \). The output given by \( W_i \) determines what rating and reward will be received by \( M_j \).

E. QoD and QoM Protocols for Decision Making

Our reputation system allows objective rating of the detectors/mitigators after the threat data has been shared by the requesters. We devise two protocols to govern the detection and mitigation performed within DefenseChain. These protocols allow threat data transmission sequentially through a software-defined network (SDN) infrastructure.

1) Quality of Detection (QoD): Fig. 5 describes the process of the attack detection protocol. When an attack is active on a federation peer, the traffic within a cloud provider’s network through the SDN switches can be monitored using a Frenetic run-time enabled monitoring sub-component [12]. Next, in order to learn and classify the attacks, the DefenseChain employs a two-stage ensemble learning scheme used in the Dolus system on the incoming traffic, both from the attackers’ side and from the benign users’ side. In order to differentiate attackers from benign users, the first stage handles outlier
detection to identify salient events of interest (e.g., connection exhaustion), whereas the second stage handles outlier classification to distinguish different attack event types.

The QoD is thus given by:

\[ QoD = \begin{cases} A \cdot y & \text{if } t_r \leq t_{\text{deadline}} \\ A \cdot y & \text{if } t_r > t_{\text{deadline}} \end{cases} \]  

2) Quality of Mitigation (QoM): Fig. 6 describes the process of the attack mitigation protocol. Once attack detection is done, the Requester then submits the request for attack mitigation to the mitigator(s) based on trust considerations. The mitigation chaincode has the script that triggers the appropriate mitigation policy (i.e., to automate the mitigation mechanism). Once the appropriate mitigation policy is set, the controller redirects the attack traffic to the QVMs. Meanwhile the mitigation mechanism for a particular attack checks the resource availability and submits a response time to the mitigation chaincode. The mitigation chaincode takes into account the availability of resources, the service response time and detection effectiveness in order to calculate the QoM.

\[ QoM = \frac{\sum_{i=1}^{n} S_r}{\sum_{i=1}^{n} t_{m}} \]  

F. Incentives for Sharing

We also consider domain reputation as a factor in performing decision making. It allows the DefenseChain to choose a detector or a mitigator based on their respective historic reputations. We follow a semi-legal approach, where we focus on determining the reputation of a detector/mitigator based on their service performance and deposit fee factors. With the historic reputation information, and owing to the design of the detection/mitigation protocols in our scheme, we enable a trust building platform in DefenseChain for threat detection and mitigation. A higher reputation score leads to a higher probability of a peer being selected for providing detection/mitigation services in the future.
Our DefenseChain implementation is designed to reduce the occurrence of Sybil, Collusion and Ballot Stuffing attacks on the threat intelligence sharing platform [6].

1) Sybil and Collusion attacks: A peer to peer system is prone to the creation of pseudonymous identities. Reputation systems also are susceptible to Sybil and Collusion attacks. In our system, Hyperledger Composer provides verifiable identity through the Membership Service Provider. Through the generated certificates with the user’s identity, pseudonyms creation is not possible, hence such attacks can be controlled in the DefenseChain.

2) Ballot Stuffing: Another type of attack that can happen in a reputation system is Ballot Stuffing. It occurs when users try to create a fake chaincode in order to get positive ratings. As our detection and mitigation chaincode focuses on objective evaluation, DefenseChain is immune to Ballot Stuffing.

IV. PERFORMANCE EVALUATION

In this section, we evaluate our DefenseChain platform by performing real-time threat sharing of DDoS, APTs and cryptojacking in realistic experiments. We show how DefenseChain effectively allows a federation of peers to leverage the attack threat information that is shared across multiple domains. Further, we compare the performance of our DefenseChain solution with state-of-the-art schemes proposed in related works such as [14] [15] [16] and [6].

A. Experiment Testbed Setup

We implemented our DefenseChain using the NSF Cloud [13] infrastructure as shown in the Fig. 7. In this testbed, we created a peer federation network, where each peer organization is connected through a central root switch. Each peer organization has a dedicated QVM and a Controller to perform detection and mitigation protocols. We installed the Hyperledger Composer platform on the controllers of each organization. We introduced different channels on the Blockchain by having the concept of inter and intra-organization and deploying them on virtual machines. All of these components were connected via a network switch that facilitated interactions between the federation of users.

The Missouri InstaGENI and UMKC InstaGENI nodes Fig. 7 act as proximal peers, whereas the Michigan InstaGENI acts as a distant peer. We leverage the IPFS deployed on a GPO ExoGENI node that is connected through the Oraclize service via a REST API. The interaction between on-chain and off-chain components is done through REST API calls and the Oraclize service. Reputation scores and threat metadata can also be queried using REST API calls. Moreover, peers can query the transaction history, which includes fields such as: reputation score, number of interactions, and source IP. Through this information set, peers have the option of accepting or rejecting a service offered by a peer organization. The communication between the networked components occurs through a trusted gateway component.

B. Network Feature Analysis

Our methodology can be best understood through the Fig. 8 illustration that summarizes the Dolus system steps for cyber defense by pretense [12]. The network traffic is collected through the slave switches when legitimate users and attackers try to access the cloud-hosted services. Users participate in the consortium Blockchain by requesting for the detection and mitigation services. We monitor the attack traffic targeted to a target node and capture e.g., bytes transmitted, number of

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>t_d</td>
<td>time taken to detect an attack</td>
</tr>
<tr>
<td>t_m</td>
<td>time taken to mitigate an attack</td>
</tr>
<tr>
<td>t_max</td>
<td>assumption made about maximum time taken to detect/mitigate an attack</td>
</tr>
<tr>
<td>t_f</td>
<td>service response time</td>
</tr>
<tr>
<td>S_r</td>
<td>Successful Rate for each type of attack</td>
</tr>
<tr>
<td>y</td>
<td>penalty factor</td>
</tr>
<tr>
<td>e^r</td>
<td>attack recurrence rate</td>
</tr>
<tr>
<td>n</td>
<td>number of interactions</td>
</tr>
<tr>
<td>β</td>
<td>initial reputation</td>
</tr>
<tr>
<td>β_k</td>
<td>average reputation</td>
</tr>
<tr>
<td>β_w</td>
<td>overall reputation</td>
</tr>
<tr>
<td>f</td>
<td>fee</td>
</tr>
</tbody>
</table>

G. Exception Handling

\[
\beta_w = \begin{cases} 
-1, & \text{if } QoD/QoM \in [0,3], \beta \leq \beta_k \text{ and } f = 0 \\
0, & \text{if } QoD/QoM \in [3,7], \beta \geq \beta_k \text{ and } f = 0 \\
1, & \text{if } QoD/QoM \in [7,10] 
\end{cases}
\]
Fig. 8: DefenseChain execution pipeline that begins with collection of network traffic from a requesting peer organization, and involves subsequent request routing into a controller with detection/mitigation chaincodes deployment in order to: (a) implement attack impact mitigation policies, and (b) share attack threat intelligence amongst the federation peers.

packets, source, and destination IP address. We also collect data from the IPFS and send the data to the detection chain-code. Therein, the QoD is calculated by taking suspiciousness score and attack-related data factors into consideration. QoD/QoM scores are calculated through a chaincode based on the complexity/effectiveness of the detection/mitigation mechanisms evidenced by e.g., available resources, and service response times.

We use Frenetic (an open-source SDN controller platform [12]) to execute Python scripts to identify suspicious packets, gather attack patterns in order to direct switches via SDN to redirect packets to pertinent QVMs. We then broadcast this information to the neighboring switches where the IP addresses of the attackers are blacklisted by updating a corresponding network policy. We randomize the attack data for DDoS, APTs and cryptojacking by changing e.g., the total bytes transferred, rate of transfer, connections made, and attack duration. This allows us to get dynamic suspiciousness scores of domain nodes for different targeted attacks. For example, in case of a DDoS attack, we exhaust the targeted application duration. This allows us to get dynamic suspiciousness scores of domain nodes for different targeted attacks. For example, in case of a DDoS attack, we exhaust the targeted application.

Formulating 

\[ \text{Device suspiciousness for trace } t: \]

\[ s_{ss, t} = \sqrt{d_{st, t}^2 + f_{flows, t}^2 + b_{bytes, t}^2} / 3 \]  

\[ d_{st, t} = w_{dst} \times \frac{\text{numDst}_{t} - \text{numDst}_{Min, t}}{\text{numDst}_{Max, t} - \text{numDst}_{Min, t}} \]  

\[ f_{flows, t} = w_{flows} \times \frac{\text{numFlows}_{t} - \text{numFlows}_{Min, t}}{\text{numFlows}_{Max, t} - \text{numFlows}_{Min, t}} \]  

\[ b_{bytes, t} = w_{bytes} \times \frac{\text{numBytes}_{t} - \text{numBytes}_{Min, t}}{\text{numBytes}_{Max, t} - \text{numBytes}_{Min, t}} \]

We evaluate our DefenseChain through experiments using metrics such as detection time, mitigation time, attack recurrence and peers’ reputation. We compare the performance of DefenseChain with state-of-the-art schemes i.e., First Come First Serve (FCFS) [14], Random [15], Distance-based [16] and the Social Reputation models [6]. We performed simulation experiments by choosing a different set of Detectors and Mitigators over 25 iterations. This is to simulate real-world situations that allow us to create a fair chance of interactions. Each Mitigator and Detector have different values of data corresponding to the detection and mitigation strategies that they employ. Our DefenseChain picks a Detector or Mitigator from a non empty set and a two-stage simulation experiment is conducted. In these experiments, we consider the DDoS attack as our exemplar cyber attack scenario.

Our first experiment was to evaluate the decision making process in choosing a Detector and Mitigator, when a request arrives from a Requestor who is under a targeted attack. We simulated a total set of 20 Detectors and Mitigators. From this total set, we randomly generated a subset of 5, 10, 15, 20 Detectors and Mitigators for evaluating the various decision making schemes. Our DefenseChain, the Requestor ended up choosing the Detector(s)/Mitigator(s) based on their calculated QoD and QoM scores. The other schemes uses different algorithms for choosing the Detector and Mitigator. For instance, the Random scheme will randomly pick one Detector/Mitigator, the FCFS scheme will choose the first peer who has responded to the request submitted by the Requestor, and the Distance-based scheme will choose the nearest Detector/Mitigator for detection/mitigation of the cyber attack. Our results show that Mitigators and Detectors chosen by our DefenseChain have better overall QoD/QoM scores when compared to the state-of-the-art schemes. This can be seen in Fig. 9 (a, b), where DefenseChain improved performance ranges from 1.3x - 4x times higher in terms of QoD/QoM values. This improvement obtained by our DefenseChain is due to the fact that we consider a comprehensive set of parameters to determine the Detection and Mitigation
Fig. 9: Performance comparison of DefenseChain with First Come First Serve, Random, and Distance-based schemes for: (a) detectors chosen to determine QoD, and (b) mitigators chosen to determine QoM.

Fig. 10: Performance comparison of DefenseChain with First Come First Served, Random, and Distance-based schemes for: (a) evaluating detectors on the basis of the time taken for detection, (b) evaluating mitigators on the basis of the time taken for mitigation, (c) studying their performance trade-offs in the context of the attack reoccurrence rate.

Fig. 11: Performance comparison of DefenseChain with a SocialReputation model proposed in [6] in order to evaluate reputation values for rational and irrational mitigators.

capabilities, rather than randomly choosing from a set of Detectors/Mitigators or using simplistic decisions considering only the order in response to the request or the distance from the Requestor, as in the other FCFS, Random, and Distance-based schemes being compared, respectively.

Upon choosing the Detector/Mitigator using our DefenseChain, we analyze the performance trade-offs in the detection time, mitigation time with the attack reoccurrence metric. As shown in Figs. 10 (a, b, c), DefenseChain takes up to 2 times more time in detection and mitigation of cyber attacks as compared to the other schemes, due to its multiple stages, i.e., policy update, attack traffic redirection and spoofing of the IP address during the detection/mitigation processes. However, these processes only consume a few minutes and these overhead times can be compensated by using the defense by pretense strategies that buy time for federation peers to create a robust cyber defense solution.

It is however important to note that our DefenseChain produces the least attack reoccurrence rate, which is 10-100 times lower than other schemes, as shown in Fig. 10 (c). This is because of our policy enforcement approach to mitigate attacks with more secure mechanisms using the Dolus system. The reoccurrence of the cyber attack is an important measurement of the quality of detection and the effectiveness of the mitigation services, and all Requestors will inherently give a much higher weight in real-world scenarios. Thus, we show that our DefenseChain has much higher performance considering trade-offs in comparison to other state-of-the-art decision-making schemes in choosing the appropriate Detectors and Mitigators in a federation.

Lastly, we compared our reputation system implementation with the SocialReputation based model [6] to evaluate its efficacy in determining and dealing with rational and irrational mitigators. In this evaluation, we initialized reputation scores of rational and irrational mitigators as 11 and 6. Choosing a baseline helped us to get results comparable to a real-world setting. For each iteration performed, we show the
cumulative reputation scores of both DefenseChain and Social-Reputation. Using our feedback from our Watchdog service described in Section III-E, our DefenseChain identifies the rational/irrational Detectors/Mitigators and provides them with pertinent feedback based on their historic data and social data in the Blockchain. As shown in Fig. 11, with the increase in number of iterations, our DefenseChain shows a faster increase in reputation for rational mitigators due to its ability to choose the most capable and reliable Detectors/Mitigators using our comprehensive QoD/QoM scoring. Additionally, the reputation of irrational mitigators decreases at a much faster rate as our DefenseChain is more capable of identifying false reporters and free riders, who are assigned negative scores.

V. CONCLUSION

In this paper, we developed a novel DefenseChain platform that leverages advancements in Blockchain technology for providing threat intelligence sharing platform capabilities to defend against cyber attacks such as DDoS, APTs and cryptojacking. DefenseChain can be used to perform attack detection/mitigation via threat intelligent sharing among a federation of domains using distributed trust principles. We devised novel QoD and QoM metrics to determine which Detector and Mitigator can be selected by a Requester in a trustworthy manner, based on factors such as e.g., accuracy, suspiciousness score, service time, attack type and attack recurrence. Our consortium Blockchain reference architecture implementation allows threat data sharing before and after attacks, so that the Requester is able to request for Detector(s) and Mitigator(s) services to effectively defend targeted attacks in a timely and robust manner. Our evaluation results from a realistic experimental testbed and from simulation results show that our DefenseChain is effective in choosing Detector(s) and Mitigator(s) based on QoD and QoM values, and outperforms state-of-the-art schemes such as SocialReputation model [6] in identifying and handling rational/irrational Detectors and Mitigators within a federation of co-operating peers/domains.

As a part of our future work, we plan to collaborate with regional network service providers to integrate our Blockchain-based solutions in their infrastructures. We will identify additional real-world scenarios where distributed trust principles can be applied to authorize access of protected threat data access to defend against targeted cyber attacks.

REFERENCES

[31] Hyperledger Provably - https://docs.provable.xyz/