

# A Blockchain-based Security Assessment Framework

N. Satyanarayana

eSecurity Department, Centre for Development of Advanced, Computing Hyderabad, India  
nanduris@cdac.in

**Abstract**—Using Blockchain Technology for Security Assessment results in effective monitoring capabilities especially when data analytics components are inbuilt in such a system. At present days, we can see the availability of many Security Information and Event Management (SIEM) tools that follow a client-server model for capturing data from different resources and performing data analysis on the server side. However, such tools serve the purpose of a single institute and depend on the trust level in a multi-institute or multi-center-project kind of environment where they can be used. Another limitation could be that if the server is attacked, the whole exercise would be futile. The lack of trust and concerns about data integrity in such an environment makes performing root cause analysis of security risks difficult. Blockchain technology ensures a tamper-proof, time-stamped, and decentralized storage repository that helps in maintaining data integrity even in complex and untrusted multi-institute or multi-center-project environments while assuring data provenance. This article presents a unified and comprehensive security assessment framework that produces a compliance report along with threat perception level by monitoring and assessing resources across multi-institute or multi-center-project in different geographical locations while supporting data privacy by leveraging Blockchain Technology capabilities.

**Keywords** - Blockchain, Security Assurance Policy, Continuous Monitoring

## I. INTRODUCTION

Continuous monitoring of critical digital resources, processes, networks, and resource utilization patterns, and reporting the incident about the observed violations is an essential part of any organization's security framework. The success of an audit process depends on data integrity while running security assessment tools such as SIEM tools. SIEM tools produce reports instantaneously when they run in a machine. For a detailed analysis of security assessment history of events and related data need to be captured in a manner such that the data integrity is maintained forever. Hence, we need a technology that maintains data provenance in a tamper-proof and time-stamped manner so that the security framework is assured of data integrity at any time.

Manuscript received Dec 09, 2022. This work was supported by the Centre for Development of Advanced Computing, Hyderabad through a funded project sponsored by the Ministry of Electronics and Information Technology, Government of India. The manuscript is submitted in the follow-up of the invited journal to the accepted and presented paper entitled "A Blockchain-based Security Information and Event Monitoring Framework" of the 25<sup>th</sup> International Conference on Advanced Communication Technology (2023).

N. Satyanarayana is a Joint Director at the Centre for Development of Advanced Computing, Hyderabad, India. (e-mail: [nanduris@cdac.in](mailto:nanduris@cdac.in)).

Moreover, such a provision will help SIEM tools to produce more effective reports when data analytics components are integrated for fine-grained analysis. Apart from the above, there should be a mechanism using which one can see to what extent the underlying security policy is conformant and its current severity level to indicate a perceived threat. If unique state replication of data provenance is ensured at the premises of the service provider and other stakeholders then compliance to the organization's security policy framework can be provided as a Software Service with continuous monitoring capabilities in a decentralized manner.

This kind of arrangement ensures transparency and trust in the organization's security policy assessment process among the stakeholders. Blockchain Technology ensures unique state replication of data in the underlying network based on consensus.

In a multi-institute or multi-center-project scenario where critical resources are spread across different geographical locations, we can utilize a permissioned Blockchain-based security assessment mechanism for resource monitoring and security analysis. The usage of permissioned Blockchain eliminates the concerns about the integrity of security-relevant data that is available at different geographical locations as it maintains data in a tamper-proof, authentic, and shares information through a secured communication channel. On top of that the permissioned Blockchain also supports data privacy by sharing security-relevant data between intended members/stakeholders only.

Blockchain Technology is a distributed ledger technology that ensures unique state replication across the participating nodes in a tamper-proof and time-stamped manner. The data once stored cannot be modified or deleted. This is because data will be stored in the form of a block whose header contains a hash of the previous block and so on. The same process of appending a new block to the existing chain of blocks in each of the participating nodes will be ensured by the consensus algorithms in the Blockchain network. Hence, to modify data in the Blockchain network, an attacker not only has to recompute the hash of the corresponding block but also the hash of the next blocks up to the end block. And this entire effort has to be replicated in each and every participating node of the Blockchain network. Considering the effort required to modify the data in the Blockchain network we can be assured of the security and integrity of data that is stored in the Blockchain network. There are tools that provide Application Programming Interface to collect various metrics from the cloud providers and leave it to the interested party to deduce the inference, information about violations, and corresponding mitigation plan in a particular machine/VM etc. There is also research work done on using

Blockchain technology to maintain the provenance of data objects corresponding to the cloud platform. To the best of our knowledge, a unified approach where evidence collection in a tamper-proof and time-stamped manner and maintenance of the same based on consensus among its stakeholders in a distributed storage system on one side and presenting the standards conformance along with perceived severity level in the whole system on another side is not present. We addressed this problem in our paper.

Major contributions of this paper include,

- Maintenance of data provenance of security log information in a tamper-proof and time stamped using Hyperledger Fabric Blockchain Platform.
- Design of a unified security assurance platform considering best practices for preventing security threats and a corresponding verification mechanism with information about perceived severity threat level.
- Provision of Security Assessment as a Service with unique state replication across multiple nodes from which stakeholders can ascertain information about current security policy conformation.
- Describes an architecture wherein multi-centre or multi-centre-project stakeholders can perform data analysis in an independent manner while maintaining data in a decentralized environment.
- Briefs about how to circumvent issue of dealing with storage space availability in this distributed and decentralized system.

In section 2, we shall present the background study, our approach detailing the cyber security policy framework model and implementation details in section 3, results in section 4, and conclusion in section 5.

## II. RELATED STUDY

Hyperledger Fabric (HLF) is a Linux Foundation's Blockchain initiative. The transaction flow (data) of HLF, ledger maintenance, and the purpose of the channel can be seen in [3][20][21]. Researchers claimed that up to 2500 tps could be achieved by them when HLF is used as a Blockchain platform [19]. In HLF, a channel is nothing but a logical subnetwork that binds its members together so that any information can be exchanged among the members themselves and others will not be able to see the data exchange. We can logically group auditors, service providers, and customers of a business entity as members of one or more logical organizations and can make them subscribe to a particular channel. In HLF such logical organizations become part of a consortium. We can have multiple consortiums configured in HLF so that no two consortium members can share data between themselves because of a channel as described above.

Major security vulnerabilities that were exploited in Cloud services were reported in Top Security Threats in Cloud Computing [4] by Cloud Security Alliance (CSA). From the information provided in [4] it can be inferred that 24x7 monitoring of security baseline controls such as password quality verification, continuous monitoring of behavioral anomalies w.r.t users, processes etc., and network policies etc., would provide a protective or vigilant cloud environment. The CSA Cloud Controls Matrix (CCM) is a cyber-security control framework for cloud computing, composed of 133 control objectives that are structured in 16

domains. It can be used as a guide to determine which security controls should be implemented by which actor for the systematic assessment of a cloud implementation. The controls in the CCM are mapped against industry-accepted security standards, regulations, and control frameworks including but not limited to: ISO 27001/27002/27017/27018, NIST SP 800-53, AICPA TSC, ENISA Information Assurance Framework, German BSI C5, PCI DSS, ISACA COBIT, NERC CIP, and many others [5].

Cloudwatch[6] from Amazon collects monitoring and operational data in the form of logs, metrics, and events providing the end user with a unified view of AWS resources, applications, and services that run on AWS and on-premise servers. This is a service provided by Amazon itself for monitoring the health information of its instances and other relevant data to the end-users. Lynis [7] is a security tool for systems running Linux, macOS, or Unix-based operating systems. It performs an extensive health scan of systems to support system hardening and compliance testing. CIS-CAT (CIS-Configuration Assessment Tool) [8] compares the configuration of target systems to the security configuration settings recommended in machine-readable content, provided the content conforms to Security Content Automation Protocol (SCAP). Both the tools (Lynis and CIS-CAT) generate security-relevant information of critical resources in the same physical asset where the tool is deployed and depends on the running status of certain services such as "audit" in Linux-based systems for the collection of information. These tools either leave the job of security assessment inference to the end-users through an API framework or they can be used in single nodes.

Several researchers have identified the need for addressing cloud security and explored using Blockchain in Cloud Platforms for security-related aspects [1][2][15][16][17]. Blockchain-based data provenance architecture for cloud environments has been proposed by [9][14][18]. In their approach, the focus was mainly on providing the ability to audit or privacy protection of cloud data objects or related operations of a cloud platform. The emphasis was more on using the data provenance capability of the underlying blockchain network for security analysis.

## III. OUR APPROACH – MODEL, POLICY FRAMEWORK AND IMPLEMENTATION

### A. Security Assurance Model

From the background study, we can observe that several efforts have been made in Cloud security by different organizations or standard bodies, primarily CSA, ISO, and individual cloud platform vendors. However, it is confusing which standard to follow from a plethora of standards by different organizations or else which platform to choose from the available platforms in the market as each one has its own metrics resulting in vendor lock-in. Each effort is in its own direction with a common goal of ensuring cloud security assurance. We need a holistic view of the security assurance framework. To fill this gap, we developed a security assurance policy framework based on the model defined in Fig. 1. and implemented monitoring of 14 controls as given in Table I below.

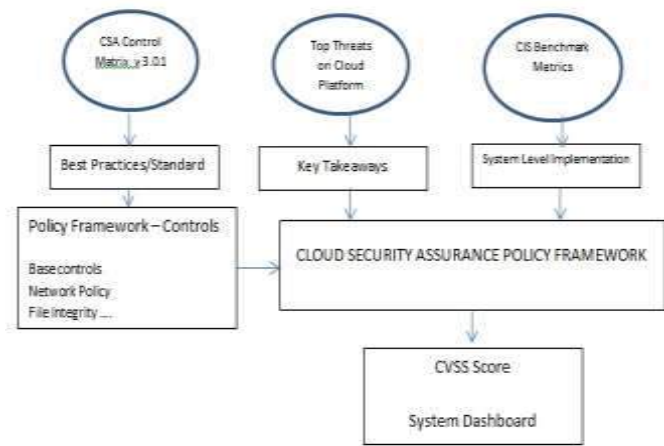


Fig. 1. Security Assurance Policy Framework Methodology

The list is not limited to these controls itself and can be extended in the future. The controls are all relevant to the key takeaways that were studied based on CSA Top Cloud Threats and corresponding preventive mechanisms. These controls have been implemented for continuous monitoring using the Centre for Internet Security benchmark metrics [10]. These controls have been mapped to different control domains of CSA Control Matrix v 3.0.1 in order to present the compliance w.r.t Cloud Control Matrix best practices. In Table I below, column 1 represents the description of the control being monitored, and column 2 represents the Cloud Control Matrix control domain to which the control name is mapped. When monitoring of a control name is implemented based on CIS benchmark metrics, its security threat perception can be graded in accordance with the Common Vulnerability Scoring System (CVSS) score. The combination of the control name, its mapping to the control domain, and its CVSS score thus form the security assessment policy framework from which the overall compliance of all critical resources of the business entity can be ascertained. The CVSS score could be one of None, Low, Medium, High, or Critical. The scores would be calculated in pursuance to [12]. The rule engine to calculate the CVSS score for each policy rule can be obtained based on the below-shown pseudo code.

TABLE I  
UNIFIED CLOUD SECURITY ASSURANCE POLICY FRAMEWORK

| Control Name                 | CCM Control Domain                                                       |
|------------------------------|--------------------------------------------------------------------------|
| File Integrity               | AIS-04 Data Security/Integrity                                           |
| Apache Loaded Modules        | IVS-07 (OS Hardening and Base Controls)                                  |
| User Login Attempts          | IVS-07 (OS Hardening and Base Controls)                                  |
| Password Policy              | IVS-07 (OS Hardening and Base Controls)                                  |
| Secure Boot Setup            | IVS-07 (OS Hardening and Base Controls)                                  |
| Process Monitoring           | IVS-07 (OS Hardening and Base Controls)                                  |
| Network Policy (IP Tables)   | IVS-06 (Network Security)<br>IAM-01 (Audit Tool Access)                  |
| Cron Service                 | IVS-07 (OS Hardening and Base Controls)                                  |
| Syslog Configuration         | IVS-07 (OS Hardening and Base Controls),<br>IAM-01 (Audit Tool Access)   |
| Secure Service User Accounts | IVS-07 (OS Hardening and Base Controls),<br>IAM – 01 (Audit Tool Access) |

|                      |         |                                                                            |
|----------------------|---------|----------------------------------------------------------------------------|
| Apache Configuration | Logs    | IVS-07 (OS Hardening and Base Controls),<br>IAM – 01 (Audit Tool Access)   |
| Apache Configuration | User    | IVS-07<br>(OS Hardening and Base Controls), , IAM – 01 (Audit Tool Access) |
| TCP Wrapper          |         | IAM – 01 (Audit Tool Access), IVS-06 (Network Security)                    |
| SSH Configuration    | Service | IVS-07 (OS Hardening and Base Controls),<br>IAM – 01 (Audit Tool Access)   |

Implementation of each control name may have a dependence on the occurrence of more than one event associated with it. For example, Process monitoring control compliance would be verified based on various events such as (a) Whether a process is system-oriented or network-oriented, (b) if it is network oriented whether access restrictions are present in the network policy or not (c) whether the log information corresponding to the process has appropriate access permissions or not (d) if it is network oriented whether the process owner has “nologin” shell or not. For each event, the CVSS score would be computed and the highest severity level among all the events that correspond to a control name would become that control’s severity level.

#### Pseudo code for determining CVSS score

```

foreach (observed_error) {
  if (user_account has loginshell) {
    if (network_outter_allowed) {
      if (public_ip) {
        av = network;
      } else {av = adjacent;}
    } else {av = local;}
  } else {av = physical;}

  if (user_account == nologin_shell) {
    ac = high; scope_change = false; pr = high;
    confidentiality = none;
    availability = none;
    integrity = none; }
  elseif (passwd_policy == weak &&
          user_account == login_shell) {
    ac = low; and scope_change = true; pr = low;
    confidentiality = high;
    availability = high;
    integrity = high;
  }elseif (passwd_policy == strong && user_account ==
login_shell) {
    ac = high; and scope_change = false; pr = high;
    confidentiality = low;
    availability = low;
    integrity = low; }
}

```

This model gives three important dimensions from which the security assessment of the business entity can be ascertained. Security Threat Perception of a critical resource being monitored. (2) Strength/Weakness of the critical resource in a particular domain in a particular control domain in accordance with the CCM. (3) Overall compliance to the security assessment policy framework of the business entity considering all the critical resources.

### B. Implementation of Security Assurance Framework – System Architecture

In order to realize the effectiveness of the policy framework the policy has to be implemented as a software module. Fig. 2. depicts the system architecture which implements the proposed security policy framework using Blockchain Technology.

We used Hyperledger Fabric (HLF) [11] as a blockchain platform. In each resource that is being monitored, there will be an agent program running hereinafter called 'Agent'. The agent program reads the log contents from the respective resource corresponding to each observable control as defined in Table 1, encapsulates the same in a JSON object format, and transfers the same to the Blockchain platform. During this process each resource that is being monitored encrypts the JSON object using its private key and the same will be sent to the HLF Blockchain node. The array containing error ids represents events where policy violations have been detected. A hash value of all the fields in the JSON object is computed and is also made part of the JSON object. This is useful for data validation upon receipt by the Blockchain node. The JSON object format to be stored in the Blockchain is as follows.

```
{
  "hashvalue": "ab232lalkn23,nlk....",
  "data": {
    "host": "xx.xx.xx.xx",
    "timestamp": "02-02-2021",
    "valid": "false" //in case of detection of policy violation
    "result": [errid1, errid2, errid3....] //list of observed violations
    "metricid": "3232" //policy control id
    "condition": {
      Loginshell: [portno, exists or not exists, uname]
      public_ip: yes or no //error can occur from outer network
    }
  }
}
```

Blockchain node upon receiving the JSON object would decrypt the received JSON object using the resource public key and recalculates the hash of all the fields in the JSON object and verify its integrity. Modified JSON objects would be rejected and will not be stored in the Blockchain node. This verification mechanism is implemented as a smart contract so that the same validation rules will be applied to each and every node of the Blockchain network.

### C. Role of Blockchain Nodes

The advantages of using Blockchain in this architecture are (a) The security policy-relevant information collected from each resource gets replicated in all the Blockchain nodes in a tamper-proof and time-stamped manner based on the underlying consensus mechanism (b) Reduces log processing load on the resource being monitored by delegating the data processing to the Blockchain platform. (c) Due to consensus-based unique state representation of data across all Blockchain nodes, all stakeholders viz. service providers, consumers, and auditors of a business entity would see the same data provenance or its analysis report at any given time

without loss of generality. This unique state representation of data in the HLF blockchain network is supported by the Raft consensus algorithm [13] which is considered a robust crash-tolerant consensus algorithm. These Blockchain nodes each can be kept at different geographical locations or in a single data center as per the business entity's need. Since this is a permissioned Blockchain only authorized users can access/retrieve data from the Blockchain nodes. A more detailed use case scenario is explained in the next section.

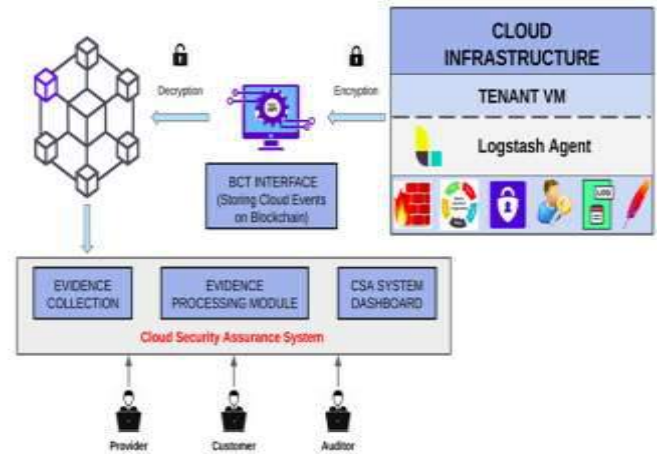


Fig. 2. The architecture of the Security Assurance Framework powered by Blockchain

In this system architecture, we defined three logical organizations namely Customer, Service Provider, and Information Security Auditor. Each logical organization can be assumed as a representation of real-time entities of respective stakeholders. The advantage of using three different logical organizations is that the nodes which are part of these organizations will maintain a unique state of log data as well as users of respective logical organizations only can initiate transactions (insert/query) on Blockchain network. Since logical organizations have been defined based on the functionalities of respective stakeholders who are concerned about the security aspects of the business entity, and all stakeholders are ensured of unique state replication with tamper-proof capability the security assessment of the business entity can be regarded as a fool-proof, robust and trustworthy system. Users from different geo-graphical locations can access such a system as Software-as-a-Service. Having been informed about the role of the Blockchain network and its necessity, we now focus on how the system reports its findings. For this purpose, we have incorporated two different metrics (a) Compliance rate – Which informs to what extent the service provider is compliant with the pre-defined security assurance policy as per table 1, (b) Severity level – Which informs the perceived threat severity level as per the criteria defined by CVSS scores to measure the overall health report of the resources controlled by the business entity.

Compliance Rate (CR) =  $(X_{t-1} - X_t) / X_{t-1}$  where  $X_t$  is no of errors observed w.r.t security assurance policy at time  $t$  and  $X_{t-1}$  is at time  $t-1$ . CR represents the observed rate of change w.r.t identified non-compliance factors of security policy. The CR represents the compliance rate against the security policy pertaining to the most recent time window. Time

window represents the gap between two successive vulnerability analysis attempts. Time window is a configurable parameter that defines how frequently the security log and other critical information have to be collected from the critical resources to be monitored.

Average Compliance Rate =  $1/n \sum CR_i$  where 'CR<sub>i</sub>' is the value of the compliance rate at a given instant of a time window and 'n' is the no of such time windows chosen in a given period (e.g., in the last 24 hours, in the last 30 days etc). Severity Level is one of 'Critical', 'High', 'Medium', 'Low', 'None' labels which is decided based on Base Score. The base score is computed as shown below. The base score and other formulae computation are done in accordance with section 7.1 of CVSS specification.

|                       |                                                                                 |
|-----------------------|---------------------------------------------------------------------------------|
| Impact =              |                                                                                 |
| If Scope is Unchanged | $6.42 * ISS$                                                                    |
| If Scope is Changed   | $7.52 * (ISS - 0.029) - 3.25 * (ISS - 0.02)^{15}$                               |
| Exploitability =      | $8.22 * AttackVector * AttackComplexity * PrivilegesRequired * UserInteraction$ |
| Base Score =          |                                                                                 |
| If Impact ≤ 0         | 0, else                                                                         |
| If Scope is Unchanged | $Roundup(\text{Minimum}[(\text{Impact} + \text{Exploitability}), 10])$          |
| If Scope is Changed   | $Roundup(\text{Minimum}[1.08 * (\text{Impact} + \text{Exploitability}), 10])$   |

(Source: <https://www.first.org/cvss/v3.1/specification-document>)

The Impact Sub-score (ISS) is calculated as  $[1 - [(1 - confidentiality) * (1 - integrity) * (1 - Availability)]]$

The severity level is decided based on base score value ranges as follows.

| Rating   | CVSS Score |
|----------|------------|
| None     | 0.0        |
| Low      | 0.1 – 3.9  |
| Medium   | 4.0 – 6.9  |
| High     | 7.0 – 8.9  |
| Critical | 9.0 – 10.0 |

#### IV. IMPLEMENTATION & RESULTS

We have implemented an 'Agent' code using "logstash" for each control that is described in the security policy framework depicted in Table 1 above. The log information from each resource is routed through an intermediate gateway developed using Node.JS server-side program which acts as a client to the HLF Blockchain network. The Node server upon receiving the encrypted JSON object sends the same unaltered to the Blockchain network. The smart contract later decrypts the JSON Object and stores the same in the Blockchain network. The flow of user interaction with the system is depicted in Fig. 3. below. A Dashboard component also has been developed using the Angular programming framework which can be accessed from the web by end-users who are members of service providers, customer, and auditor organizations. The dashboard component displays the no of violations against the security policy framework as described in Table I in a graphical format. It also displays various metrics such as no of resources audited in the platform, the average compliance rate over a period of time (e.g., last 24hrs, 7 days, 30 days). We also wanted to understand the effectiveness of the security policy compliance rate being computed by the system. To observe this we relied upon measuring severity levels in accordance with CVSS scores. If the security policy has any impact, then it should get reflected in observed severity levels.

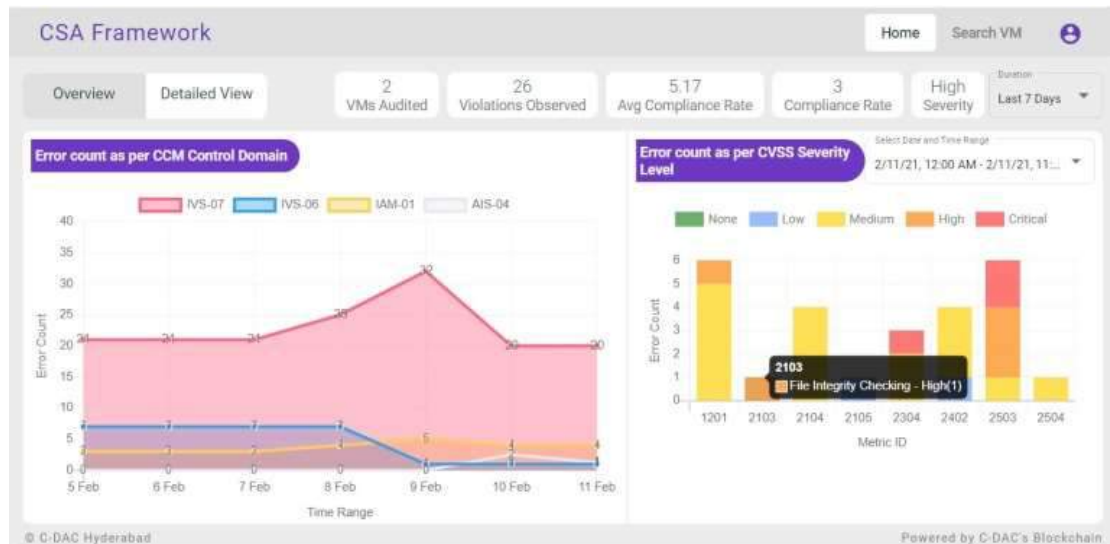


Fig. 3. Display of resources being monitored and their stats

Fig. 3. above depicts the snapshot of the dashboard where no of resources audited, compliance rate, severity level etc. The dashboard also provides a detailed error report comprising an error description, its impact, and a mitigation plan for each and every violation that is observed by the system.

In Fig. 3., it can be observed that the left side graph shows the observed events information which are considered as violations as per the underlying security policy. The graph shows the details in accordance with the CCMv3 guidelines of Cloud Security Alliance. The right side of the graph

displays the no of violations observed corresponding to each category of events in terms of their CVSS score.

Fig. 4., depicts the user interface screen wherein the end-user can observe the different types of errors or violations that have been monitored by the underlying system, its severity level, and also the mitigation plan. From this screen one can

observe how many controls with which the resources being monitored are compliant can be observed by clicking on the chart icon on the top right corner.

The screen can be used to observe the error description of each violation that is observed in a resource, its impact as well as mitigation plan also.

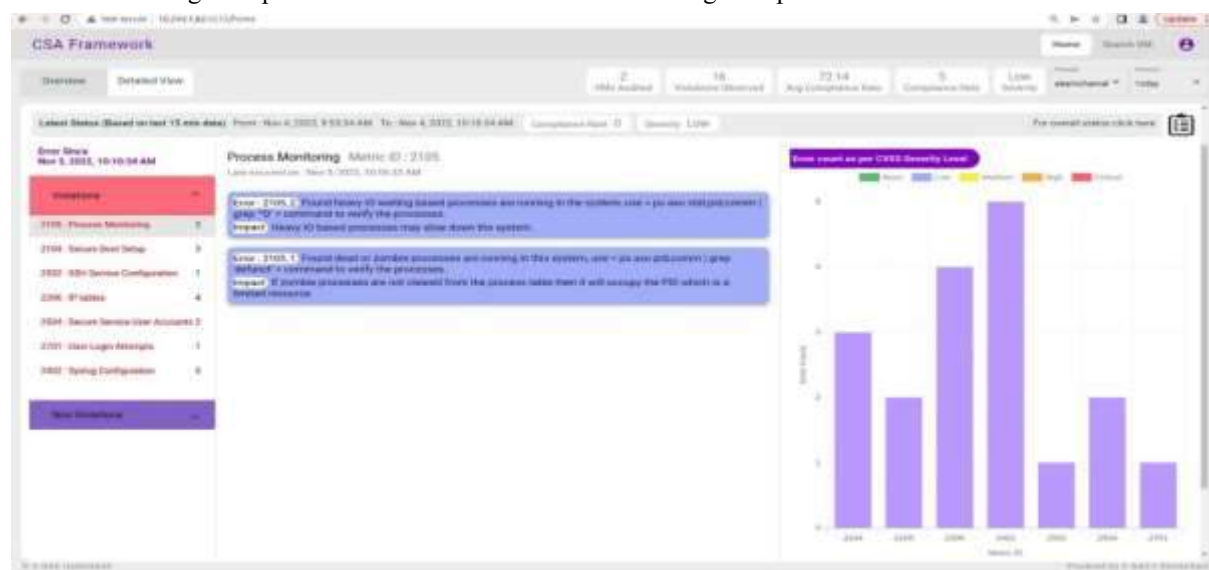


Fig. 4. Observed e/vents and corresponding event details

#### A. Use Case Scenario

The use case scenario will give a clear picture of how the organizations can get benefitted using this software. The software can be used in a multi-institute or multi-center-project environment wherein different stakeholders from different geographical locations work in a collaborative manner. Multi-institutional or multi-center-project-based resources can be monitored for security vulnerability based on a pre-defined security policy framework and maintain relevant data in a Blockchain network, hosted by relevant stakeholders, spread across different geographical locations for data analytics. Since the data is maintained in Blockchain all relevant stakeholders are ensured of the uniqueness and correctness of data at their premises and use such data for security analysis purposes in an independent manner. Hence, Institutions working in a multi-center, multi-project, or collaborative manner with other institutes and having the requirement of 24x7 monitoring of critical resources for security breaching based on a pre-determined security policy framework that applies universally among the participating entities can use this model or system. One such example scenario could be the Cyber Insurance domain where the cyber insurer requires a mechanism wherein monitoring of resources can be done in a seamless manner even in an untrusted network environment. Another such example could be where a consortium project is being executed by multiple stakeholders in a consortium manner while utilizing resources located at different places under different network administration teams.

The functional requirements of deployment in this case is as follows.

- ✓ **Blockchain Network:** As far as this use case is considered Blockchain network represents Hyperledger Fabric-based Blockchain network architecture. In this

architecture, the network administrator can define a consortium of logical organizations that would like to share data among members of respective organizations based on pre-defined signing policy and membership.

- ✓ **Logical Organization:** An organization from the perspective of a Blockchain network is one that binds peer nodes and users through a common membership. We can use this concept to create several logical organizations that group peer nodes and users, belonging to different physical institutes/centres/project groups. Once the logical organizations are defined then data received from Blockchain client applications can be shared among the participants of those organizations only.
- ✓ **Channel:** In order to share the data between the logical organization's peers of respective organizations have to join the specific channel which is nothing but a logical subnetwork that supports data exchange among its members only. Hence, a channel has to be created so that peers of different organizations can join the channel.
- ✓ **Critical Resource:** Each critical resource that has a public and private key certificate pair and is capable of sending its security log information in a specific data format through a secured transmission medium has to be identified. The critical resource then sends the data in the desired format to the blockchain client application in the respective centre/project group/institute through an agent program. The Blockchain client application then submits the data to the Blockchain network.

- ✓ **Blockchain Client Node:** A Blockchain client application accepts the data in the specified format from a critical resource and validates its authenticity and integrity based on the resource's certificates. The client application then submits the same as a blockchain transaction to the Blockchain network along with information like channel configuration, user details, and request parameters. The data to be inserted will be forwarded to all the peer nodes that are part of the logical organizations that are participating in the Blockchain network as a consortium. The Blockchain client also receives requests from another entity other than critical resources i.e., the User application to fetch data from the underlying Blockchain network corresponding to a limited time period like the last 24 Hours, last 7 days, etc., in a specific format.
- ✓ **Blockchain Peer Node:** Each transaction that is submitted by different client applications will be received by one of the peer nodes in each logical organization that is being administered by the Blockchain network. Upon successful processing of transaction data through a Smart Contract program deployed in each peer node, the data will eventually be added to the existing peer ledger in all the Blockchain nodes. Each peer will run the same version of the smart contract. At the same time, each peer can also run multiple smart contracts each bearing a specific version number. Each center/institute/project group can designate one peer node as a member of the blockchain-based logical organization.
- ✓ **User App:** The user application can be accessed by authorized users with valid certificates representing respective stakeholders such as Cyber Insurer, and System/Network Administrators representing logical organizations of the Blockchain network. Upon receiving the request for the resource's current security status from the end users through the user app, the Blockchain client application issues a query request to the underlying Blockchain platform to fetch the relevant information and then runs a machine learning model or data analytics program to ascertain the overall threat perception level and security compliance rate. Once the threat perception has been computed, the information will be sent as a response to the user's request. The authorized users then can view information like the number of critical resources that are being monitored across different institutes/project groups/centers, the average and overall compliance rate of specific resources being monitored, detailed information of captured events, their impact, and mitigation plan, threat perception level in accordance with CVSS score. Only registered Users Applications can communicate with the Blockchain client node.
- ✓ **Ordering Service Nodes:** The user application initially sends data to endorsing nodes (Blockchain Peer Nodes) in each organization and collects the read/write set upon executing a Smart Contract on endorsing node. The data will not be committed at this point in time. The User application after collecting read/write set information from each peer will send the same to the ordering service nodes for ordering transaction data and bundle all transactions in a Block in accordance with the pre-defined block size limit. Ordering service nodes after the creation of blocks will broadcast them to all the Blockchain peer nodes in each organization which commits the same upon successful revalidation of transaction data to observe whether any discrepancies are there in the transaction data from the time the read/write sets are collected earlier and before committing them in the same peer node. Blocks will then commit the data either as valid or invalid depending on the validation results. The Ordering Service nodes can be deployed at each participating Institute/Project Group/Centre or one of them can host the ordering service nodes. The same ordering service nodes can be shared and used for multiple project groups/institutes/centers.

In Fig. 5., it is envisaged that two Institutes A and B have joined hands to execute a collaborative project X whose resources are located in both A & B. The architecture is not limited to two only but can be used for many collaborating agencies. Both A & B would like to monitor the security compliance report or threat perception level of all the resources. Based on the above-depicted deployment architecture now it is possible for A and B to run their respective Blockchain client applications to receive data from their respective institute's critical resources and forward them to the Blockchain network.

The Blockchain network in this architecture has two logical organizations namely Org A and Org B which binds one peer node each (more peers can be run to ensure high availability if needed) along with users. A blockchain channel also has been created and made all peers of respective logical organizations members of the same for sharing data among the peers of those organizations only.

Once the Blockchain network receives data from different client applications they forward them to all the peers of both Org A and Org B through Ordering Service nodes and the data will be stored in their respective peer ledgers permanently.

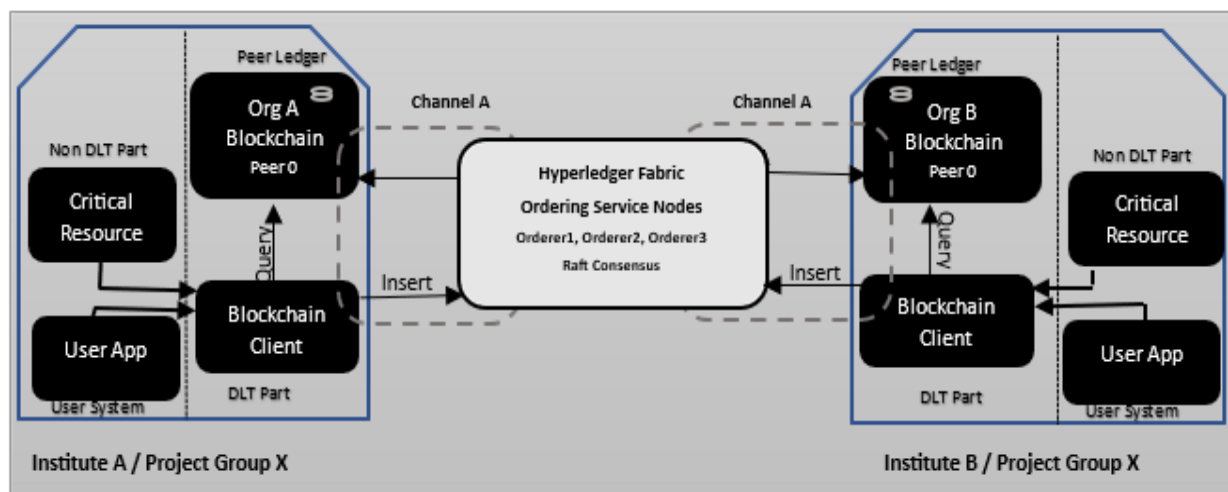


Fig. 5. Security Assessment Blockchain Architecture

The User Apps running at institutes A & B can have a unique representation of the data pertaining to Project X at any given point of time and can run their own analytics programs to infer security compliance information and threat perception level independently. Thus, this way the architecture supports a decentralized network with data replication capabilities while ensuring each participant relies upon a unique representation of data and facilitates each user app to run data analytics independently.

The architecture can support multiple consortiums with multiple channels to delineate the data and data analytics into separate groups and monitor independently.

### B. Security and Identity Management

**Blockchain Peer and Orderer Node:** A peer and orderer node can join the Blockchain network by establishing its membership within an organizational context and generating a public and private key for the peer and orderer nodes.

**User Registration:** Users with 'READER' and 'WRITER' roles can be registered with the system. READERS are allowed for querying the data and WRITERS are allowed for inserting data into the Blockchain network. User accounts with the WRITER role are used during critical resource registration time and the READER role is used for those user accounts which are meant for accessing the User App components.

**Non-DLT System & User System:** These are the components which are not related to Blockchain network but are nothing but data producers or consumers. The data source's (or critical resource being monitored) integrity protection is ensured by way of computing the hash of all the fields in the data source object and encrypting it using the data source's private key. The name and password used during key pair generation must be a registered user with the role of 'WRITER'. Since the data source's public key is shared with the Blockchain client (BCTClient), the client can verify the authenticity and integrity of the data once it receives the same and it can insert data into the Blockchain network.

**User App:** This component's IP address MUST be registered with the BCTClient in order to accept requests from only registered applications. Apart from that the request object coming from User App MUST also provide a username

and password for authentication purposes. When the request is from a registered User App and the user is a registered user of the Blockchain network then only the request will be processed further.

**Privacy:** Blockchain peers, clients, and orderers of different organizations can exchange data among themselves provided they all join the same Blockchain 'channel'. Members of different channels cannot exchange information between themselves. This way by binding peers, clients, and orderers of different organizations that needs to exchange data among themselves only data privacy is achieved.

**Event API:** The user app that is installed at the respective stakeholder's premises can invoke an API call for retrieving data from the blockchain system periodically to compute overall and individual threat perception levels.

### C. Performance of the tool

The software module i.e., the intermediate gateway in our overall architecture, used for retrieval of data from the Blockchain network and to verify security policy conformance, should be designed in such a manner that it can withstand the vast amount of data that is required to be processed and improve the system response time.

It is observed that if logs are collected at 10min intervals in a day, a total of 144 iterations will be required to store the log information in the Blockchain platform. We observed that with 14 control data that are collected 144 times a day and each JSON object size of 594bytes on average, approximately a total of 1.14MB of data would be stored from each resource that is being monitored. For 30 days 34.27MB of data will have to be processed corresponding to a particular resource. After observing this, we designed the below data structure to cope up with data generated by multiple resources. In our experiment, we used nearly 8VMs for monitoring purposes and improved the system response time. The end-user, upon selecting the duration for which the analysis has to be performed in the dashboard, data pertaining to that period will be retrieved from the Blockchain network and maintained in accordance with the below data structure.

```
{
  metric_id: val  # id of the control name
  result: [error1, error2, error3, error4 ....] #stores id
  of every error that is observed
  timestamp: [ts1, ts2, ts3 ....] #stores timestamps at
  which the errors have been observed
  machine_id: xxxx  #id of the resource
  hostname: val  #name of the host
}
```

By default, the data collection interval is set to 10min in our approach. In this format instead of processing individual JSON objects representing each log entry, we rearranged similar entries based on the time of their occurrence while responding to the request from the dashboard component. This has enhanced the system response time considerably.

Since the data is collected from different resources for a specific period it makes the data analysis more effective as events can be related and inferences can be drawn accordingly. We used the pseudo-code explained above for computing threat perception scores by related various events that occurred during the specified period.

#### D. Storage Management

It is important to understand how the storage space is utilized in each peer node in the Blockchain network. Whenever a peer node confirms a transaction block that it has received from the ordering service node in the Hyperledger network, the peer node stores the same in its ledger. The ordering service ensures an atomic broadcast of the blocks once they are ordered as explained above. Usually, when a peer node in the Blockchain network goes down for any reason and when it rejoins the same network, it will try to contact the other peer nodes to obtain the missed entries in its peer ledger to be in sync with the blockchain network.

Here, the question arises whether we should ensure that all peer nodes must be homogeneous in terms of their storage capabilities. What happens if we use nodes in different logical organizations with different storage spaces?

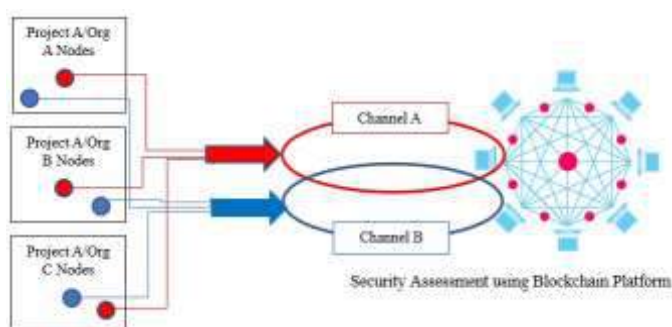


Fig. 6. Schematic diagram of transition from one channel to another channel to cope with available storage space in blockchain nodes

It is explained that from a single resource, approximately 1.14MB of log data has to be maintained in peer ledgers of Blockchain nodes in the network. Apart from that the same data will get replicated in other nodes of the Blockchain network. When multiple resources are monitored those many nodes data has to be stored in the Blockchain node.

When the storage space gets filled in the respective blockchain nodes it is not possible to replace the one which got filled first compared to other nodes. This is because if we

replace a node that gets filled first with another node the newly added node will start trying to be sync with other blockchain nodes which are members of the same channel. This situation necessitates that all nodes of Blockchain nodes must be homogenous. This is a serious limitation as it will restrict blockchain usage in a multi-institute or multi-center-project environment where ensuring homogenous nodes is a difficult phenomenon.

We studied the impact of such a requirement and its practical applicability. In general, no production system or network can be brought down temporarily and hence we need to devise an alternative mechanism.

In this regard, we conducted a trial experiment in our lab following the below steps which resulted in a smooth transition of the existing production-grade blockchain network that could handle the above storage management-related issue and also eliminated the requirement of strictly using homogenous nodes across all the organizations.

As shown in Fig. 6., we maintained a single peer node in each logical organization of the Blockchain network and made the signing policy an IMPLICIT MAJORITY so that members of any of the two organizations, when they sign the transaction data, would be sufficient. We also made another node in each logical organization ready with new storage space available and made them members of a new 'channel' that binds only these newly added peers of respective organizations as members. Now we have two channels (please refer to what purpose a channel serves above), one which was there from the beginning and is being used currently. The newly created channel contains new peers of respective organizations. Once this setup is made (for which we need not bring down the production network), we deployed the same smart contract that is used for validating transaction data (the log data) in the existing channel on to the new channel also. Once the arrangements are completed, we made all subsequent requests from client applications to divert their future request submission to the new channel. Since new channel and smart contract combination is made available on newly added peers, they maintain their own peer ledgers on respective nodes. This way we handled the smooth transition of the production network from one channel to another channel with the same smart contract. This solution not only helped in dealing with storage management effectively but also eliminated the need for worrying about having homogeneous nodes only across the Blockchain network.

#### E. Comparative Analysis

Table II below depicts the comparative analysis of developed solutions with different security analysis tools. Other tools given in Table II have certain shortcomings when compared to our solution as both of them generate a report in the system in which those tools are deployed. We used freeware versions. Whereas our solution provides a web interface to view the health report of all resources being monitored in one place. Moreover, we also have the CVSS score that displays the severity level to alert the concerned stakeholders. Another important feature of the developed solution is that it automatically collects data at regular intervals and sends them to the Blockchain platform for permanent storage in a tamper-evident manner. This way, we

have designed and implemented a unified security policy framework with a 24x7 monitoring and alert system. The system facilitates a unique state of representation of security

log information due to the usage of Blockchain because of which inferences can be drawn beyond any doubts on data integrity.

**TABLE II**  
COMPARATIVE ANALYSIS OF DEVELOPED SOLUTION WITH OTHER SIMILAR INITIATIVES

| Controls                                         | Lynis                                  | CIS-CAT                         | Our Solution                    |
|--------------------------------------------------|----------------------------------------|---------------------------------|---------------------------------|
| File Integrity                                   | No (Depends on auditd)                 | No (depends on auditd)          | Yes (Own Implementation)        |
| Continuous monitoring                            | No                                     | No                              | Yes                             |
| Reports                                          | Yes (On VM where it runs)              | Yes (On VM where it runs)       | Sends information to Blockchain |
| Impact Details                                   | Partial                                | Yes                             | Yes                             |
| Firewall rules for all open ports                | No                                     | Yes                             | Yes                             |
| File System Base control                         | Yes (Doesn't say whether it is right?) | Yes (Say's whether it is right) | Yes (Say's whether it is right) |
| No login shell for system users/network services | No                                     | Yes                             | Yes                             |
| Presentation                                     | Text                                   | Yes (html)                      | Yes (html)                      |

## V. CONCLUSIONS

We presented a unified and comprehensive security assessment framework that is supported by Blockchain Technology. The security policy framework was developed based on inputs drawn from key takeaways from top threats in the cloud platform, existing cloud best practices/standards such as Cloud Security Alliance, and implementation techniques to provide an early detection mechanism using CIS Benchmarks etc. In this attempt, we demonstrated how Blockchain technology supported the data provenance related to security aspects in the decentralized network and log management in a time-stamped and tamper-evident manner based on consensus among the relevant stakeholders. We also demonstrated an inbuilt mechanism for monitoring the effectiveness of the security policy framework by way of computing severity levels in various resources. During the course of using the Blockchain network the role of blockchain nodes, and challenges w.r.t storage management have been studied carefully and the system has been designed accordingly to provide better performance.

## ACKNOWLEDGMENT

The author acknowledges all their colleagues who supported them in coming out with the results and implementation of the software and MeitY, Government of India.

## REFERENCES

- [1] G Ramachandra et., "A Comprehensive Survey on Security in Cloud Computing", *Procedia Comput. Sci.*, vol. 110, no. 2012, pp. 465–472, 2017.
- [2] Monjur Ahmed, Mohammad Ashraf Hussain, "Cloud computing and security issues in the cloud", *International Journal of Network Security and Applications.*, Vol.6(1), 2014, pp. 25-36
- [3] Z. Zheng, S. Xie, H. Dai, X. Chen and H. Wang. (2017) "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends," *IEEE International Congress on Big Data (BigData Congress)*, Honolulu, HI, pp. 557-564.
- [4] Technical report from Cloud Security Alliance, "Top threats to cloud computing: Deep Dive – A case study analysis for 'The Treacherous 12: Top Threats to cloud computing' and a relative security industry breach analysis.
- [5] *Cloud Controls Matrix* from Cloud Security Alliance. Available: <https://cloudsecurityalliance.org/research/cloud-controls-matrix/> on July 16, 2020
- [6] *CloudWatch User Guide for monitoring Amazon instances health information.* Available: <https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/WhatIsCloudWatch.html>
- [7] *Lynis Documentation* Available: <https://cisofy.com/documentation/lynis/>
- [8] *CIS-CAT Lite tool* Available: <https://www.cisecurity.org/blog/introducing-cis-cat-lite/>
- [9] X. Liang, S. Shetty, D. Tosh, C. Kamhoua, K. Kwiat and L. Njilla, "ProvChain: A Blockchain-Based Data Provenance Architecture in Cloud Environment with Enhanced Privacy and Availability," *2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, Madrid, 2017, pp. 468-477, doi: 10.1109/CCGRID.2017.8.
- [10] *CIS Benchmarks for Ubuntu Linux* Available: <https://www.cisecurity.org/cis-benchmarks/>
- [11] *Hyperledger Architecture, Volume 1 (2017) Introduction to Hyperledger Business Blockchain Design Philosophy and Consensus.* Available: [https://www.hyperledger.org/wp-content/uploads/2017/08/Hyperledger\\_Arch\\_WG\\_Paper\\_1\\_Consensus.pdf](https://www.hyperledger.org/wp-content/uploads/2017/08/Hyperledger_Arch_WG_Paper_1_Consensus.pdf)
- [12] *Common Vulnerability Scoring System v3.1: Specification Document* Available: <https://www.first.org/cvss/v3.1/specification-document>
- [13] D. Ongaro, J. Ousterhout. (2014) "In search of an understandable consensus algorithm", *Proc. USENIX Conf. USENIX Annu. Tech. Conf. (USENIX ATC)*, pp. 305-320.
- [14] C. Yang, L. Tan, N. Shi, B. Xu, Y. Cao and K. Yu, "AuthPrivacyChain: A Blockchain-Based Access Control Framework With Privacy Protection in Cloud," in *IEEE Access*, vol. 8, pp. 70604-70615, 2020, doi: 10.1109/ACCESS.2020.2985762.
- [15] Park JH, Park JH. Blockchain Security in Cloud Computing: Use Cases, Challenges, and Solutions. *Symmetry*. 2017; 9(8):164. <https://doi.org/10.3390/sym9080164>
- [16] S. Singh, Y.-S. Jeong, and J. H. Park, "A survey on cloud computing security: Issues, threats, and solutions," *J. Network and Computer. Applications.*, vol. 75, pp. 200–222, Nov. 2016.
- [17] B. Duncan, D. J. Pym and M. Whittington, "Developing a Conceptual Framework for Cloud Security Assurance," *2013 IEEE 5th International Conference on Cloud Computing Technology and*

*Science*, Bristol, 2013, pp. 120-125, doi: 10.1109/CloudCom.2013.144.

- [18] Sachin Shetty, Val Red, Charles Kamhoua, Kevin Kwiat and Laurent Njilla “Data provenance assurance in the cloud using Blockchain”, *Proc SPIE 10206, Disruptive Technologies in Sensors and Sensor Systems*, 1020601 (2 May 2017); Available: <https://doi.org/10.1117/12.2266994>
- [19] *Channels* — *hyperledger-fabricdocs main documentation* Available: “<https://hyperledger-fabric.readthedocs.io/en/release-2.2/channels.html>”
- [20] *Transaction Flow* — *hyperledger-fabricdocs main documentation*. Available: “<https://hyperledger-fabric.readthedocs.io/en/release-2.2/channels.html>”
- [21] *Ledger* — *hyperledger-fabricdocs main documentation* Available: “<https://hyperledger-fabric.readthedocs.io/en/release-2.2/ledger/ledger.html>”



**N Satyanarayana** lives in Hyderabad, India, and is born in 1976. He did his Master of Technology a post-graduation degree in computer science from Jawaharlal Nehru Technological University, Hyderabad, India. Prior to this, he completed his Master of Computer Application post-graduate degree in computer applications from Sri Venkateswara University, Tirupati, India in the year 1999.

He is currently working as Joint Director in the Centre for Development of Advanced Computing (CDAC), Hyderabad, India. He has been working for CDAC for the past twenty years and played an instrumental role in developing various applications in the areas like eLearning, Peer to Peer, Network Management, and Blockchain Technology.

Mr. N Satyanarayana published several papers at National and International conferences in various areas such as Peer to Peer Computing, Network Management, e-Learning, and Blockchain. His current research interest includes Blockchain consensus algorithms and reference architectures. He is also contributing towards best practices/standards in the respective fields of his work being a member of the technical committees of the Bureau of Indian Standards.