Peer-to-Peer Networking and Applications (2021) 14:2635–2650
https://doi.org/10.1007/s12083-020-00991-6

Penetration testing framework for smart contract Blockchain

Akashdeep Bhardwaj[1] · Syed Bilal Hussian Shah[2] · Achyut Shankar[3] · Mamoun Alazab[4] · Manoj Kumar[1] · Thippa Reddy Gadekallu[5]

Received: 8 June 2020 / Accepted: 21 August 2020 / Published online: 5 September 2020

Abstract

Smart contracts powered by blockchain ensure transaction processes are effective, secure and efficient as compared to conventional contacts. Smart contracts facilitate trustless process, time efficiency, cost effectiveness and transparency without any intervention by third party intermediaries like lawyers. While blockchain can counter traditional cybersecurity attacks on smart contract applications, cyberattacks keep evolving in the form of new threats and attack vectors that influence blockchain similar to other web and application based systems. Effective blockchain testing help organizations to build and utilize the technology securely withe connected infrastructure. However, during the course of our research, the authors detected that Blockchain technology comes with security considerations like irreversible transactions, insufficient access, and non-competent strategies. Attack vectors, like these are not found on web portals and other applications. This research presents a new Penetration Testing framework for smart contracts and decentralized apps. The authors compared results from the proposed penetration-testing framework with automated penetration test Scanners. The results detected missing vulnerability that were not reported during regular pen test process.

Keywords Attack vectors · Blockchain · Cyber threats · Cybersecurity · OWASP · Smart contracts

Highlights

This research presents a new framework to perform manual penetration testing framework on smart contract application and decentralized apps.

- Results from the new proposed penetration-testing framework and automated penetration test scanners are compared in this research for Blockchain applications. No other framework currently performs such validations.
- The new framework detected missing vulnerabilities that were initially not reported during the regular penetration testing process, which could have made the Blockchain contract app vulnerable to Cyber-attacks and threats.
- While in real-time Cyber space, no one can ensure that the operations would be executed in a predefined order. Any malicious user could cheat the seller if the buyer intentionally changes the order of transactions or execution process. The proposed framework performs validation and compares input as well as any mismatch for actual steps against the predefined properties and process.
- The authors also compared the tool and manual penetration testing results to analyze in the wake of removing the vulnerabilities discovered amid penetration Tests for the smart contract applications.

This article is part of the Topical Collection: *Special Issue on Blockchain for Peer-to-Peer Computing*
Guest Editors: Keping Yu, Chunming Rong, Yang Cao, and Wenjuan Li

✉ Syed Bilal Hussian Shah
bilalshah@dlut.edu.cn

✉ Thippa Reddy Gadekallu
thippareddy.g@vit.ac.in

Extended author information available on the last page of the article

1 Introduction

Blockchain technology has gained enormous growth in terms of research and implementations by various type of industries. Blockchain works on peer-to-peer transactions, being distributed decentralized anonymity with no third party or any centralized control. Smart Contracts [1] are digital programs scripts of codes stored inside a Blockchain. These programs are temper proof, self-verifying, self-executable and self-enforceable [2] digital contracts when certain clauses [3] with specific predefined conditions are met. Smart Contracts are capable of performing transaction in real-time, at low cost and provide a greater degree of security [4]. The network of Blockchain cryptocurrency nodes execute to update the distributed transparent ledger. This update is seen by all nodes and verified [5] before acceptance in the network.

As an example, imagine buying a new car, the traditional process starts by going to a car dealer (intermediary third party), bargaining for your choice of car. Instead of going to a bank for a car loan (another third party) and involving the transport department and insurance, (more third parties for the paperwork. Once all formalities and payments are completed, there is a waiting period before the car's delivery. This process takes time and involves interactions with multiple other third parties.

Assuming the same car details, ownership, papers and offer is available and with no third party involved, with higher-level security and details being available, unchanged and distributed over the Blockchain network. The details validated by each node on the network, but no one person is in absolute control. Execution of the purchase order done using the Smart Contract. This system would be secure and paid by Cryptocurrency in real time [6]. Ownership is transferred immediately as digital identity on the Blockchain Ledger. All nodes update the ledger on the Blockchain network and conclude the transaction [7]. Similar process is followed by Banks or lending organization for processing Loans or receiving automatic payments. Insurance companies can use Blockchain for processing claims. Postal departments can process payment on delivery with Smart Contract systems [8] instead of traditional transaction process.

This concept [6] is implemented for buying or renting apartments that would involve tenant and property owner. Monthly rent or EMIs can be deducted using tokens or cryptocurrencies. So in effect performing any transaction are handled securely and efficiently using Smart Contract systems that are powered by the Blockchain Technology [9]. Global Securities Exchanges in United States [10] and Australia [11] have accepted these. However, much like Cyber threats [10] and attacks on cloud hosted systems and applications, Blockchain networks also suffer attacks like Denial of Service (DoS) [12], Decentralized Autonomous Organization (DAO) [13] and Blockchain specific cyberattacks that are discussed in the subsequent sections in this research. Traditional IT infrastructure and hosted Applications as well as Blockchain environments, both face similar Cybersecurity threats. In most use cases, the attack vectors are same; however, the mitigation strategies can vary. While it may seem that the Blockchain is a perfect solution for transactions, the technology still has points of vulnerabilities. The attack vectors have been categorized based on Network, Applications, Data Integrity and End User levels are mentioned in Table 1.

While designing and implementing the Blockchain based Smart Contract solutions, security threats associated with Smart Contracts relate to various direction, ranging from source code bugs, virtual machine vulnerability, insecure runtime environment to the Blockchain network itself. Some of them are:

- **Complex Technology**: When trying to design and build Smart Contracts from scratch or localized version, the security vulnerabilities lie with the execution and not the system. Average programmers and developers cannot implement Blockchain. This needs specialized skills.
- **Inception Vulnerability:** For a proper Blockchain to perform, thousands of nodes are required to work in unison. If one node or group of nodes, control 51% of the system nodes then they can control the Blockchain outcome. For a small setup of nodes, it is easily possible.
- **Government Control:** Cryptocurrencies can render the government-controlled currencies to become less valuable or go out of use and destabilize the world's economy. Such authorities would always want some regulation and level of control, which goes against the decentralized concept of Smart contracts.
- **Third Party integrations:** Use of non-standard third-party platforms can introduce flaws even as Blockchain network maybe secure, e.g. 400 BTCs were hacked from NiceHash Mining marketplace and $ 60 million stolen as user funds in 2017, Bitcoin Gold was hacked in 2018 losing $18 million awhile Crypto Exchange Zaif confronted $ 60 million bitcoin theft.
- **Security of Keys and Certificate:** Darkweb has over 60 marketplace portals selling SSL and TLS certificates and related services for $ 250 to $ 2000 in March 2019. Blockchain keys and Smart Contracts face yet another set of challenge where Criminals assume identifies of trusted machine nodes.
- **Insecure Source Code**: Source code issues Reentrancy attack can lead to passing on the control to untrusted functions of other Smart Contracts, which can have undefined behavior or use for malicious purposes. Source code bugs in an Ethereum [14] Smart contract cost $80 million in 2016.
- **Virtual Machine Vulnerabilities:** These are low-level attacks using Ethereum Virtual Machine. EVM has been detected to have immutable defects. Blockchain blocks after creation can be changed or cryptocurrency can be lost during transfer or access control of systems by hackers can lead to sensitive functionality access of the Smart Contract.
- **Mining Pools:** Miners unite to combine and create pools of computing power. This helps to mine more blocks and receive more rewards instead of individual miners, which hardly earn any profit or receive any BTCs. Miner Pools [15] increase their reward share by delaying the broadcasts of mined blocks to others. Then suddenly all the blocks are released at once. This makes other miners lose their blocks. The largest pool of Bitcoin Miner is namely AntPool, BTC.com and ViaBTC. Mitigation strategies against such threats are having only trusted miners on the network or modify the Smart Contract protocols to hide the variance between partial or full proof of work inside the Smart Contracts [16].

2 Literature survey

For this research, the authors identified 144 research papers published from 2016 until date on Blockchain and Security Testing, after a four stage selection process shortlisted 38

Table 1 Attack vector classification

Attack Vectors	Process Description
DoS attack	IT infrastructures face denial of service attacks, which typically involve flooding the network pipes and applications with requests. Legitimate users are denied access to the service resources. • Blockchain Smart Contracts face service denial attacks when one or more execute and updates or creation of new blocks requests are submitted to the Blockchain, which is more than what can be handled. Transaction tampering with group routing is another such attacks. Attacker sub-divide the Blockchain network into separate groups. These are not allowed to communicate with each other. Then the transactions are sent to the peer nodes. This makes it impossible for other peers to detect the tampering. • Routing attacks involve partitioning the peer nodes with delays introduced into the network interfering the message broadcasts being sent on the network.
Network Efficiency	Currently in most Blockchain ecosystems, the maximum possible transactions per second is between 3.3 and 7. Credit cards attain around 2000 transactions per second, while Twitter achieves around 5000 transactions per second. • Low efficiency of transactions often holds back Blockchain adoption for potential nodes. This also involves greater processing and throughput efforts inside Blockchain and the miners. • As the Blockchain network grows, complexity increases which in turn interferes with the processing speed and efficiency of the Blockchain network.
Code vulnerabilities	This involves use of multiple iterations of Penetration Testing using secure coding, with manual and automated tools. Smart Contract can be written by any node, which then spreads in the network. Integer Overflow vulnerability was the only major flow detected in Blockchain. • Points of Failure involve use of single primary database server or one master backups can be a glaring vulnerability, IT setups typically use multiple systems and backups and plan for business continuity and disaster recovery. Being Distributed Ledger with multiple nodes involved in the network, there are no such issues visible in Blockchain. • Timejacking exploits the Bitcoin timestamp vulnerability; this is done by altering the node time counter or by adding multiple fake peers having erroneous timestamps. This forces the victim node to agree on using another Blockchain network. Eclipse Attacks has the hacker taking control of large number of distributed nodes as network bots. Once the nodes are restarted, outgoing connections are redirected to the attacker's IP address, which is controlled by the attackers. The victim nodes are then unable to obtain their transactions.
Data Integrity	IT Infrastructure manages data security using the CIA triad. This includes backups and implementation of strong security policies and processes with audits. For Blockchain systems, cybercriminals target user wallet credentials. • Wallet Access involves traditional hacking means like use of phishing emails, dictionary attacks as well as new-sophisticated attacks, which seek vulnerabilities in the cryptographic algorithms. Blockchain utilizes ECDSA Cryptographic algorithm, which automatically generates unique private keys. ECDSA has insufficient entropy vulnerability. This results in the same random value being utilized by more than one signatures. • Fraudulent Modifications are done by Man-in-the-middle and privilege escalation attacks. These are usually mitigated by security policy, data encryption, salting for IT Infrastructure involving databases. Since Blockchain exists in form of sequential chain of blocks, anyone trying to alter records would have to first alter all transactions leading to that specific transaction, which is complicated. However, attackers can alter transaction ID and broadcast that transaction with modified hash value to the nodes. They would try to get it confirmed before the original transaction completes. The initiator would tend to believe the initial transaction might have failed, even as funds in form of BTCs had been withdrawn from their accounts. This is termed as Transaction Malleability. The attacker tricks the victim into paying twice. In 2014, MtGox Bitcoin Exchange was bankrupt due to such a Malleability attack.
End User	• Endpoint threats: Endpoint Security is controlled by enterprise with organization wise policies and console management for monitoring and detection of end user systems and mobile devices [13]. For Blockchain, the nodes are the endpoints, which can be homogeneous, so flaw in one node can be exploited as flaw in Blockchain network systems. • Intentional Misuse: Traditional setup faces insider threats by staff and employees who can steal data and affect the setup. In Blockchain, Miners are incentivized for Proof of Work, who can group together to take control of the network. Majority attack or 51% Attack occur in Blockchain network with one group or hacker harnessing enough computing power to compromise the whole network. Hacker can gain control of network hash rates to create alternate forks and then take precedence over existing forks. • Sybil Attack: is performed by controlling multiple nodes as Bots. These surround the victim node with fake nodes transactions or take time verifying the transactions. Victim node thus becomes is vulnerable to double-spend attacks which are difficult to detect and prevent. The attackers use same coins or tokens for multiple different transactions tricking the Blockchain system to accept the fraud transaction.

relevant publication works as illustrated in Fig. 1 below. Some of the relevant reviews are mentioned in this section. The reason for concentrating on last 3 years was the immense growth and changes on the Blockchain Smart Contract domain has happened primarily in the past few years along with latest cybersecurity attacks, threat vectors and vulnerabilities discovered and exploited by Cybersecurity attackers.

Table 2 represents the overall spread of the research papers, subcategories selected for literature review.

Tonelli et al. (2019) [17] implemented Blockchain based Smart Contract using Micro-Service applications. The authors analyzed and replicated the Smart Contract micro-service architecture in form of a case study using set of Smart Contracts. The results displayed the possibility of implementing simple

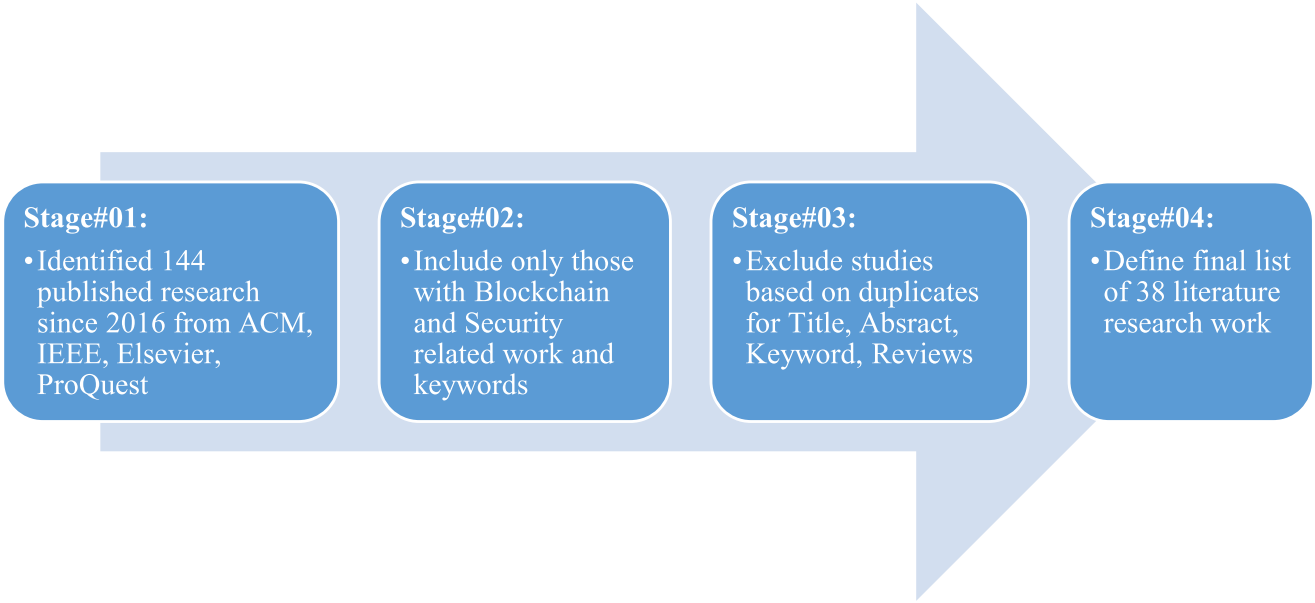

Fig. 1 Staged literature survey selection criteria

micro-services while maintain the similar paradigms and functionality.

Amoordon et al. (2019) [18] proposed a fault tolerant application promoting the awareness and ease of programing in Blockchain. The authors proposed one application per Blockchain displayed the improved performance and reduced weakness against security attacks. This platform could potentially be an ideal Smart Contract application for Blockchain platforms like Ethereum and Bitcoin.

Yamashita et al. (2019) [19] presented a survey on security risks for Blockchain, focusing on the programming languages and development tools. The authors utilized Java and Go language that existed before Blockchain was created, even as these languages are not designed for writing Smart Contracts. The authors focused on 14 primary risks and observed that existing tools would not cover some risks as also developed static analysis detecting tool.

Al-Jaroodi et al. (2019) [20] surveyed the application of Blockchain technologies and Smart Contracts for various industrial domains [21]. The authors observed that deploying Blockchain increased the industrial transparency, security, efficiency and traceability increased even as the cost of deployment and delivery was reduced.

Mohammed et al. (2019) [22] discussed adoption of Blockchain and Smart Contract for industrial sectors primarily the manufacturing industry. The authors observed that for effective integration with multiple systems and components, there were challenges to overcome. The authors proposed adopting middleware approach in order to effectively utilize Blockchain and use the capabilities to full extent leading to smart manufacturing.

Draper et al. (2019) [23] reviewed security applications like PGP, Proxy chain and studied the challenges faced by Blockchain. The authors studied major problems faced and discussed ways of solving the problems like latency, integration, throughput, and regulatory as well as provided direction for future research.

Mahmood et al. (2019) [24] focused on providing enhanced safety and productivity of logistics operations using Smart Contracts, Big Data and ICT. Implementation of Supply Chain for tracking containers in real time was presented with Email and SMS alerting system for customers. Customers to track international and national delivery of their consignments utilized the systems.

Tateshi et al. (2019) [25] presented a unique model to auto generate executable Smart Contracts in Blockchain based Hyper

Table 2 Blockchain related literaturec review categorization

Paper Classifications	Stage 1	Stage 2	Stage 3	Stage 4	Final Review	Breakup %
Smart Contract	38	29	17	12	10	26.8%
Blockchain Threat	33	26	18	14	9	23.7%
Attack Vectors	38	30	21	16	10	26.3%
Blockchain Cybersecurity	35	28	20	15	9	23.2%
	144	**140**	**98**	**66**	**43**	

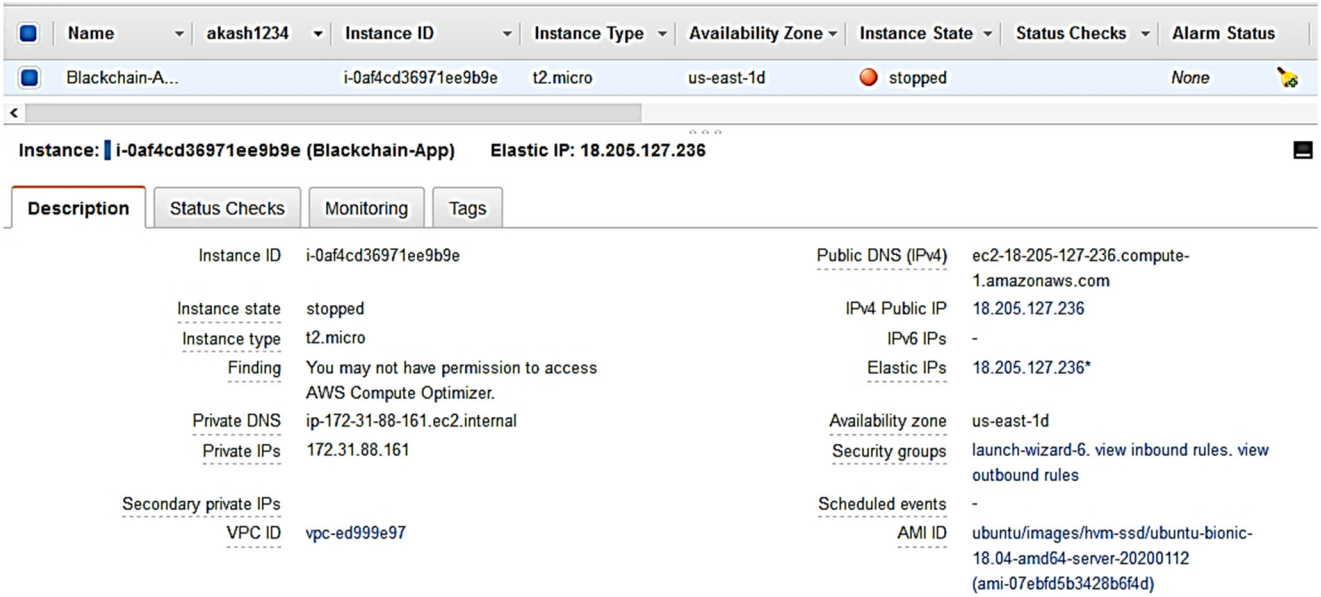

Fig. 2 AWS Node Instance setup

ledger using human written and understandable Contract document. The authors created this using a template with a controlled natural language and evaluated the results using case studies from real world Smart Contacts in various domains.

Wang et al. (2019) [26] proposed comprehensive overview of Smart Contacts based on Blockchain. The authors introduced the platforms and operating mechanisms of Smart Contracts and six-layer architecture framework. The authors also reviewed legal and technical challenges [27] and discussed the application security issues as well as provided references for future research.

Ozyilmaz et al. (2019) [28] designed Blockchain-based Internet of Things using emerging technologies like Swarm, Ethereum and LoRa. The authors addressed the issues of data storage, high availability, mining and denial of service attacks for Smart Contracts systems that typically employ trustless nodes in decentralized manner for distributed storage in Blockchain networks.

Wan et al. (2019) [14] focused on industrial IoT nodes [15] to restructure the original architecture and designed a new decentralized model [16] based on Blockchain network. This improved the security and privacy [29] as compared to traditional architecture as well as optimized application delivery. As the size of network and number of nodes increase, the traditional architecture was unable to provide efficient support while the proposed architecture emerged as a viable solution.

Suliman et al. (2019) [30] utilized the features of Blockchain Smart Contract as a concept for carrying out

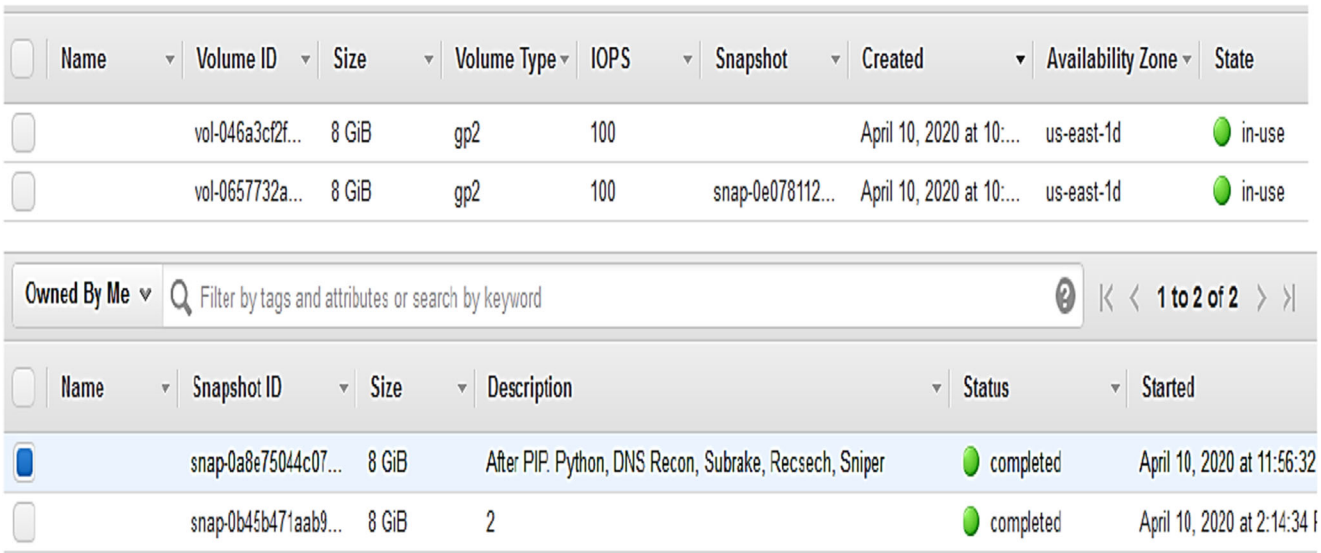

Fig. 3 AWS Node Volume and Snapshots for changes

Table 3 Blockchain environment setup prerequisite

Tool Name	Installation Steps	Tool Description

MIST Browser	$ sudo git clone https://github.com/ethereum/mist.git
	$ cd mist
	$ yarn
	$ curl –o –L https://yarnpackg.com/install.sh bas -s
		Browser for decentralized applications using Yarn package manager

Install Google Chrome	$ sudo wget https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb $ sudo apt install. /google-chrome-stable_current_amd64.deb	Download the Google Chrome package and then install

Nodejs & NPM	$ sudo apt install nodejs
	$ node –version
	$ sudo apt install npm
		Install JavaScript runtime for Chrome engine and node package manager

Metamask	Open https://metamask.io/ on Google Chrome Use "Get Chrome Extension" to install Metamask Select add to Chrome ➔ Add Extension ➔ click on Metamask Logo and Agree terms to use	Allows user accounts and key management, including hardware wallets instead of having keys on central server.
Solidity Complier	$ sudo npm install solc	Setup Solidity compiler

transactions. The authors discussed the architecture, application logic, entity and the interaction workflow using decentralized and highly trusted network having no intermediary. This model is based on using for live data exchange using Smart Contracts for Ethereum, Wood et al. (2016) [31].

Alladi et al. (2019) [32] presented existing trends in research related to blockchain implementations for industrial sectors. The authors discussed implementation challenges and also presented issues hampering the adoption of the blockchain technology for industry 4.0 and discussed future application areas.

As cybercrimes are increasing day by day, the evaluation of such attacks to provide protection measures were suggested by Ch et al. (2020) [33]. Use of manual methods with technical approaches often fail to control cyberattacks [34, 35]. The authors proposed a machine learning computational application that can analyze and classify the rate of cybercrimes as per country or state locations. The authors implemented security and data analytics to analyze and classify structured and unstructured data. The testing analysis reportedly produced an accuracy of 99%.

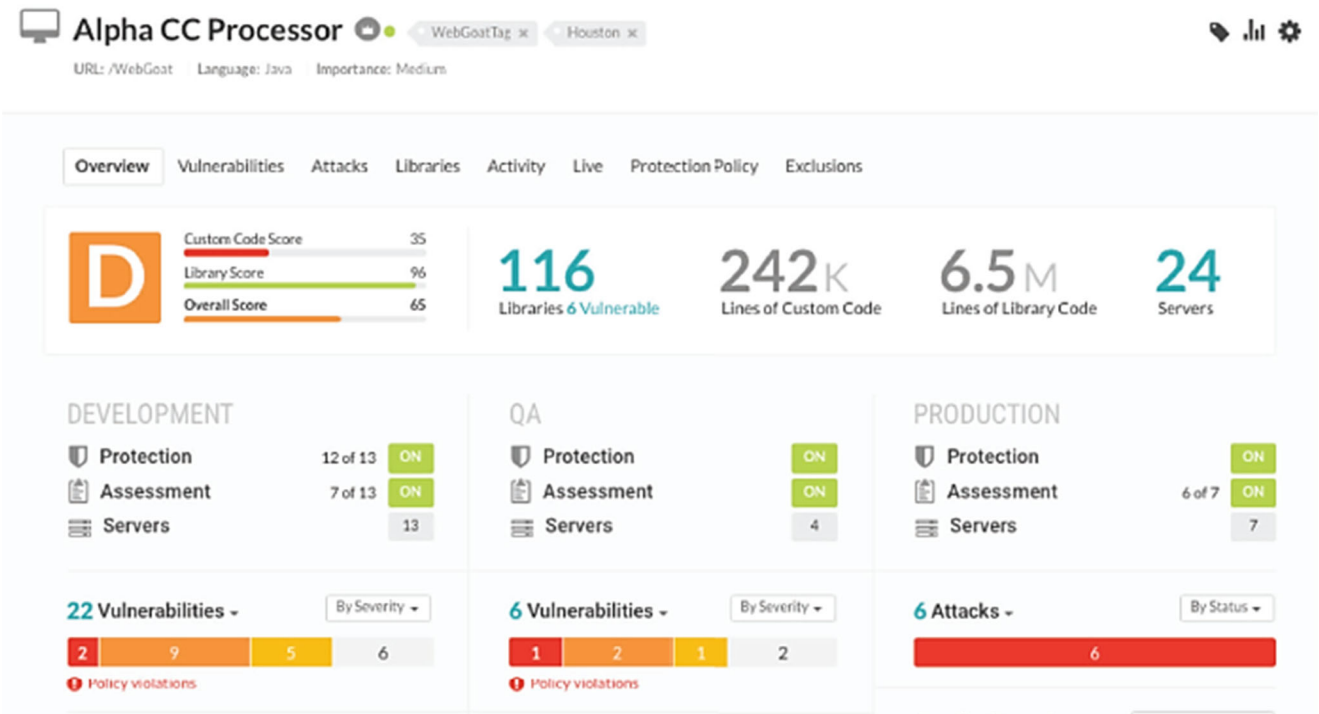

Fig. 4 AWS Setup Console for the Smart Contract Blockchain

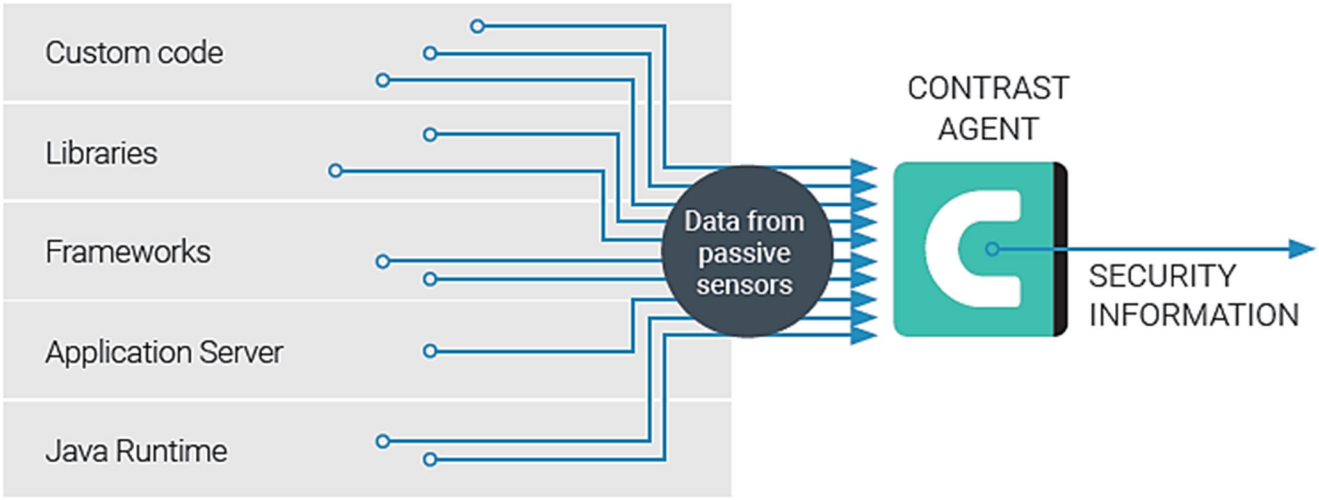

Fig. 5 Deep level application security test

2.1 Gaps identified

The authors reviewed research papers on Blockchain and Security Tests and identified that there are gaps that need to be addressed.

- Classification of the research papers themselves is a huge problem as new classifications needs to be defined related to blockchain and penetration testing as compared to web and application security testing or OWASP.
- Other challenges like latency and robustness of the application and systems are also researched by lots of organizations and researchers.
- Survey and research on the legal and regulatory compliance issues due to different countries rules and regulations.
- Cyber-risks and privacy are accorded the highest priority as some of the most difficult features to implement and deploy. Since blockchain is permission less, public systems in the form of nodes can be controlled and utilized for unlawful purposes. This further complicates the process as the global transactions are completely anonymous) transactions without any check or involvement of any centralize authority
- Scalability of the nodes and storage related to cryptocurrencies is being able to handle the dynamic transaction rate in a centralized system even as its technology core remains unchanged.

3 System model

Blockchain environment setup includes installation of few prerequisites as part of essential tools required for Blockchain nodes. The authors setup Amazon Web Service general-purpose instances running multiple nodes with Ubuntu OS 18.04. Each node created on the AWS use T3 instance model with dedicated single tenant hardware. Each node has been designed to run the Smart Contract application on m 5.4x large with 8 vCPU (Alpha CC), 32 GB RAM and 300 GB SSD drive each. In order to connect the nodes, the authors utilized Amazon Web Services Instances accessible via RDP, Putty and SSH using IP v4 Public address as shown in Fig. 2.

AWS Instance Volume and Snapshots were taken after each major application and configuration change at regular durations as illustrated in Fig. 3 below. The systems have

Layers	Blockchain			Environment
Application Layers	Node ID	Smart Contract	Virtual Machine	Graphical User Interface
Data Level Layer	State Transaction	Record	Transaction Event	Database Store
Consensus Layer	Proof-of-Work	Proof-of-Stake	Incentive Values	Data Integrity Validation
Network Layer	Auto Node Discovery	Propagation Delay	Transaction Hashing	Shared Infrastructure

Fig. 6 Blockchain environment setup

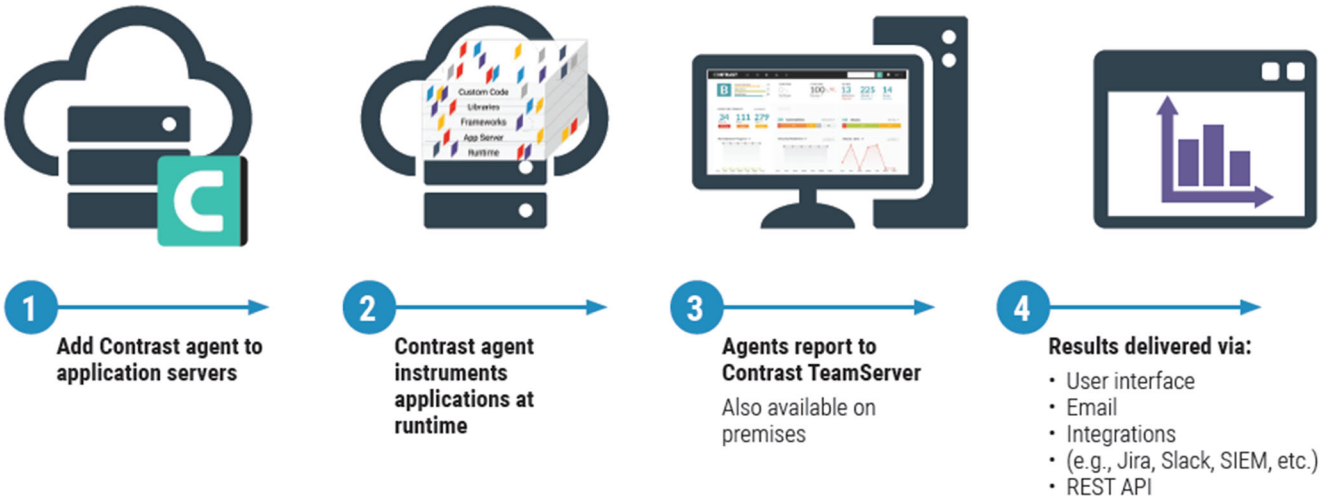

Fig. 7 Proposed architecture

3500 Mbps of committed EBS transmission capacity up to 10 Gbps. This performs weakness evaluation utilizing latent sensors [36, 37]. (Table 3).

The second is the Centralized administration server that gathers and reports on vulnerabilities recognized by the operators, and controls the organization local mix with different instruments like IDEs and CI/CDs supporting highlights for announcing, warnings and API get to procedure with RESTful API for custom integrations as illustrated in Fig. 4 below.

4 Proposed framework

The Penetration Testing framework comprises of core testing strategies and services, such as cloud testing services, functional testing, API testing, integration testing, security testing, and performance testing. It also includes Blockchain specific testing strategies such as block testing, smart contract testing and peer/node testing. The authors propose utilization of Static Application Security Analysis at introductory stage, before execution of the Blockchain code. This includes custom application code alongside Runtime stage and incorporates the Blockchain Application Server, Framework and Code Libraries. Regularly, Dynamic Application Security Testing just includes utilization of devices that adventures the running Blockchain applications. This is performed utilizing reproduced focused on assaults or exceptionally made HTTP inputs [38]. By dissecting the HTTP reaction, the vulnerabilities are distinguished. Nevertheless, DAST is oblivious in regard to what happens inside the application, and gives just restricted inclusion. Like SAST, DAST [39] instruments are moderate, with an average examining movement taking hours, if not days, to finish. This performs a full runtime information and control stream examination, joined with static investigation of all the code, as depicted above, while likewise dissecting all the inbound and outbound HTTP traffic produced amid typical testing of the application. This permit performing dynamic investigation like, however more powerful than DAST, without requiring any devoted security tests, misuse of the objective application, or security specialists to be associated with the testing procedure is illustrated in Fig. 5. Since, evaluate works from inside the application, this gives more precise examination than customary Penetration (Pen) Testing apparatuses. What's more, not at all like either SAST or DAST items. The authors performed Software Composition Analysis (SCA) to assemble a stock of all outsider segments (for example libraries, structures and so on.), including open source programming (OSS), that are utilized by the application. Use of proper Penetration Testing tools is equally important. This helps differentiate between the known and hidden ambiguous vulnerabilities in the application and modules. The authors performed Blockchain Pen Tests using two specific tools and recommend them for all potential Blockchain Pen Testers. The first is Truffle Framework provides simple and easy Pen Testing and Management environment for Smart contracts related

Table 4 Threat Severity Levels

Rating	Severity	Description
1	Insignificant	Result of low or irrelevant log entry, can be ignored,
2	Minor	Alert due to more than one node or transaction, can be false positive
3	Moderate	Verified security event leading to a true positive event
4	Major	Ongoing security breach, requires significant management intervention

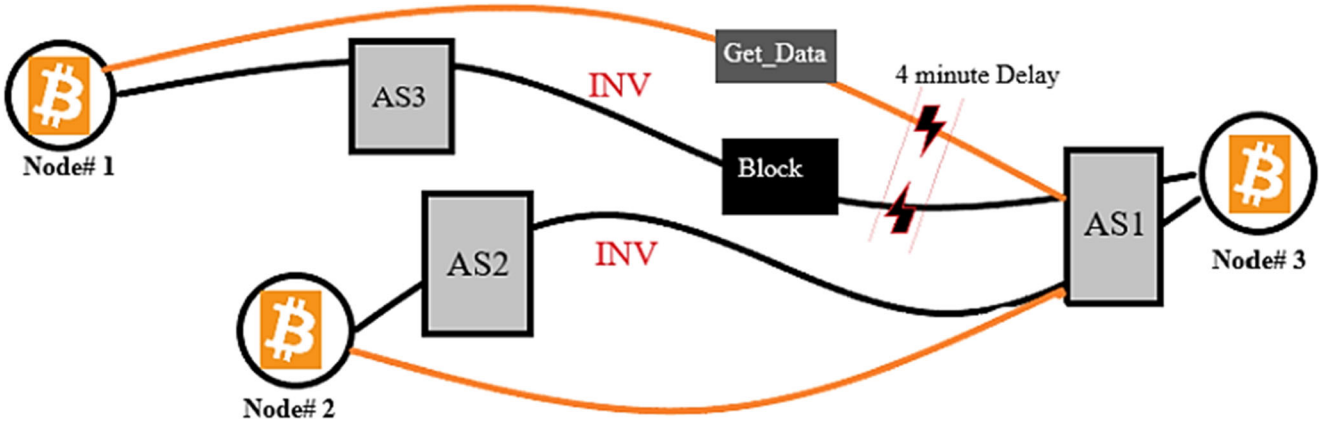

Fig. 8 Blockchain node transaction Delays

applications. This framework includes support of complex to standard Blockchain based implementations, customized deployment as well as linking libraries.

The framework even offers JS and Solidity development environment to run automated use case and codes. Pen testers can also run automated scripts for migration and deployments as well as build pipeline for end-to-end support for custom Blockchain processes and perform asset rebuilding during development phase. The second is Ethereum Tester tool to perform full test suite with customized API support to improve the Pen Tester and Developer efficiency, time and efforts. These tools in particular helped detect and block vulnerabilities that were never found and reported any time earlier during the pre-penetration testing reconnaissance phase. Blockchain architecture and execution environment is illustrated in Fig. 6 below. Cybercriminals have been abusing Blockchain requesting ransoms in type of digital currencies, ransomware assaults. In any case, presently the attacks focus on Blockchain Smart Contract vulnerabilities as the primary wellspring of income, assaults. Proposed Penetration Testing architecture is presented in Fig. 7.

The authors estimate the risk level by determining the total relationships for each threat as per the incident.

In order to calculate the threat level, first treat level estimation is done by applying thresholds and then use weighted methodology. Threat point levels are collaborated with the Threat rating. This represents the threat severity range from one to four as illustrated in Table 4 below to determine the Total Risk Points. This is calculated as the sum of risk points with threat severity weight, as per the risk point and ratings.

Risk Points = [Risk Point (Maximum) * Rating (Major)] + [Risk Point (High) * Rating (Moderate)] + [Risk Point (Low) * Rating (Minor] + [Risk Point (Minimum) * Rating (Insignificant)].

$$\text{Sum of Risk Point Sum (RP)} = \left\{ \begin{array}{c} [RP(max)*SR(major)] \\ [RP(high)*SR(moderate)] \\ [RP(low)*SR(minor)] \\ [RP(min)*SR(insignificant)] \end{array} \right\}.$$

$$\text{Severity Rating SR} = \left\{ \begin{array}{c} (4\ (Major) if\ RP > HTi) \\ 3\ (Moderate)\ if\ RP \geq HTi \\ 2\ (Minor)\ if\ RP = HTi \\ 1\ (Insignificant)\ if\ RP \leq HTi \end{array} \right\}.$$

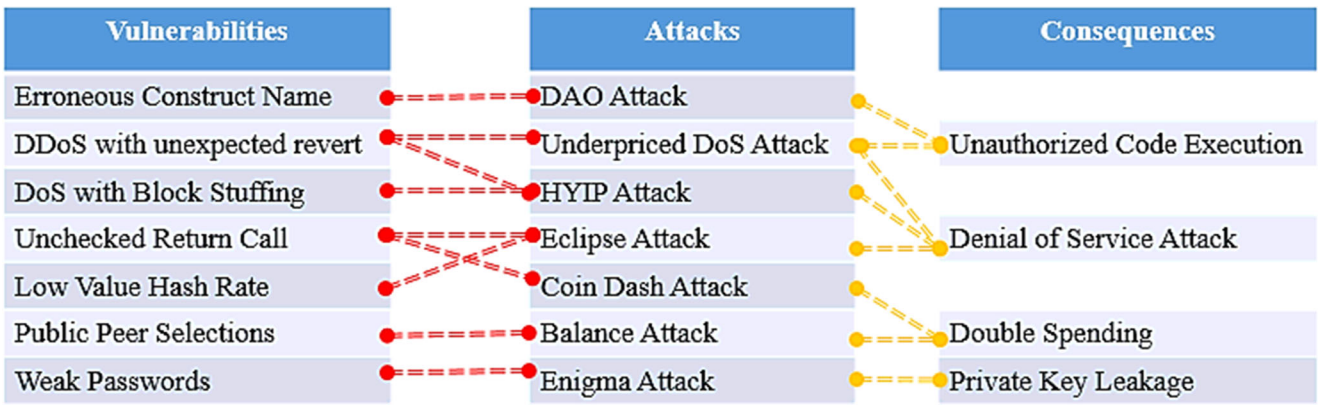

Fig. 9 Vulnerability, Attack and Consequence Relations

The authors implemented the below mentioned algorithm for Client-side Authentication and Validation.

1. **Start Server & Client**

2. **Blockchain use key pair**

Public Key → GC5X3F3CKLDWV2Z6YPHTYOJDDWEQX
Private Key → SML4S25HDAMJYZJYACXMHIEUMAPQ

User & Application → identified through respective key pairs only

Both entities key each other's public key (ONLY)

Owners → Signing & Encrypt messages

Crypto Signatures → message sent by real owner in Blockchain, open with Private key

End-to-end Encryption → not relevant in this research use case scenario

Application with known Public Key → owns its Secret Key

User with known Public Key → owns their Secret Key

3. **User Access Blockchain App to request a challenge**
   Involves simple HTTP Request to Blockchain Endpoint
4. **Server Challenge**
   Server performs single data entry operation (Add, Set, Delete, Modify)
   Data_EntryResult(name, value)
   If success
   User Challenge Validated
   Else
   Return_error(value) as
   0: Data_not_supported
   1: Data_not_found
   2:Data_has_low_XML_reserves
   3: Invalid_data_name

Check Input (Challenge) = Client (Public Key) && Time(Current) = within transaction time

5. **User Validation Challenge Signature Check**
   If Signature = Valid
   Sign and Return → back to application
   Else
   Exit

6. **Application Validation**
   Perform Challenge Transaction check
   User Signature check
   If both are TRUE
   Issue Access Token

User can now access the service

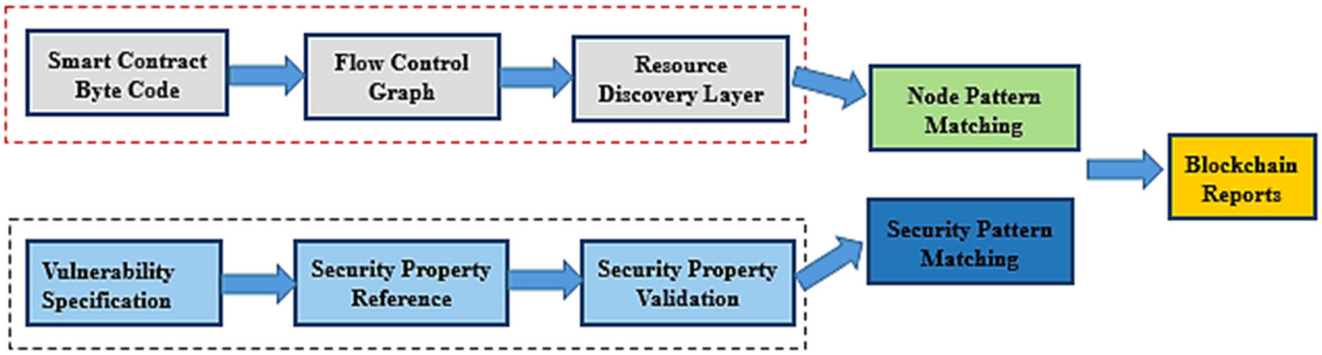

Fig. 10 Workflow for vulnerability detection

4.1 Research performed

The authors performed Penetration Testing on a production ready, commercial Blockchain application; the testing was performed in the pre-production environment, with the critical vulnerabilities as mentioned below. These vulnerabilities map the critical vulnerabilities found and mapped to OWASP Top10 for the Blockchain Smart Contracts.

- **Vulnerability Type: Injection**

Threat Level: High

Process: Validated strings with white listing before the Database SQL query.

Issue: Buffer-out-of-Bound issue detected on system in the Smart Contract Parsing module. This poor sensitization of input allowed authentication to be bypassed and unauthorized commands to be executed. This vulnerability on the Sandbox launched a reverse shell on the infected nodes on the network. The authors found three functions in Data sub-directory code that were using string concatenation query for performing Database operations on packages supplied parameters.

- **Vulnerability Type: Broken Authentication**

Threat Level: High

Process: Design issues in LISK Cryptocurrency do not bind short addresses immediately to Public Keys. Attackers can overtake any unclaimed account.

Issue: Incorrect implementation of Near-Swap feature makes it prone to different attacks. Best option is not opening Web server access for everyone. There should be some level of authentication in place. The application's feature for Near-Swap, allows a third person to snoop into the communication and download files from either of the two user's device, without their permission.

- **Vulnerability Type: Transaction Routing Attack**

Threat Level: High

Process: Hack peer nodes to change the state of transactions before they are committed on the network.

Issue: Divide the Smart network into groups, in order to delay the transactions, tamper the propagating messages sent on the network and even divert the Blockchain traffic as illustrated in Fig. 8.

The below code illustrates the nodejs connectivity to the node.

```
// web3 is an Ethereum client library
const Web3 = require('web3');
const web3 = new Web3();
web3.setProvider(new
web3.providers.HttpProvider('http://localhost:8545'));

// This file is generated by the Solidity compiler to
easily interact with
// the contract using the web3 library.
const loginAbi =
require('./solidity/build/contracts/Login2.json').abi;
const LoginContract = web3.eth.contract(loginAbi);
module.exports = LoginContract;
```

To demonstrate the advantages of using manual penetration testing approach against the automated scanner, the authors compared the manual results against two state-of-

Table 5 Comparing Manual and Automated for benchmarks reported for project effectiveness

Vulnerability types	Manual V Automated	Manual – Automated	Automated – Manual
Timestamp Value	522	671	103
Reentrancy Routine	15	129	17

Table 6 Analysis of resulting rates after complete Penetration Testing for Random Samples

Benchmark	Manual FP Rate	Manual FN Rate	Automated FP Rate	Automated FN Rate
Timestamp	6%	11%	39%	31%
Reentrancy	15%	8%	44%	39%

the-art Penetration Testing analyzer. For sake of confidentiality, the names cannot be revealed. One of the tools is based on Symbolic Execution while the other tool is based on dynamic random testing. This ensured testing of the smart contract was performed any vulnerable related to double-dealing. The authors performed functional and non-functional testing in order to validate and resolve any smart contract anomalies. During Non-Functional Testing the Smart Contract performance and security is taken into account at highest level. Security Pen test ensured Common Vulnerabilities and Exploits reentrancy, buffer under and overflow, call for delegate or visibility while the Performance guaranteed peak transaction amount for contract behaviors. While in the Functional Testing, business requirements and rules were validated using various use cases that included boundary test rules, valid/invalid argument and argument combinations as illustrated in the Figs. 9 and 10.

5 Results

The shows an untested contract that is vulnerable to cheating. In the parallel/decentralized world, no one can ensure that the operations are executed in the predefined order. A malevolent buyer could cheat the seller of Product X if the buyer intentionally changes the order of transaction execution. Comparison with the first tool, Smart Contract is taken as input and checked for any match for concrete traces in the tools predefined security properties [40–43]. This is compared to the manual Penetration Testing results obtained. The authors performed two comparisons that analyze in the wake of relieving the vulnerabilities discovered amid Penetration Tests for the Smart Contract. Right off the bat, the viability of this present reality vulnerabilities was resolved and furthermore, computerized Penetration Testing apparatuses are looked at which are used in the business for Smart Contracts Penetration Testing. The creators included more than 30,000 Smart Contracts with the most extreme assault program size set to three, having a delay timeout of 15 min for each Smart Contract. To comprehend the adequacy of the Manual Static Penetration Testing performed, correlation performed utilizing computerized dynamic Penetration Testing apparatuses. The outcomes got have been displayed in the Tables 5 and 6.

To confirm the final release of the pen tested Blockchain, the authors compared the results with previous version releases. Table 7 displays this based four major security features as Tamper proof, Authentication, Decentralization and Authorization. Thus, is validated that the production release after undergoing multiple pen test iterations show no major issues related to the four security feature, as compared to the pre-pen test or the multiple pen test iterations.

6 Conclusion and future work

The authors compared manual Penetration Testing with two Application Security Testing tools for automated synthesis of Smart Contracts that can exploit the vulnerabilities of victim nodes. To ensure the synthesis is tractable, summary-based symbolic evaluation was introduced. This reduced the number of data paths that tools needs to traverse and explore while maintaining the precision of the vulnerability queries. By building on the summary-based symbolic evaluation, manual Penetration Testing further introduced optimizations that enabled parallel exploration and other form of cyberattacks. The authors encoded known Smart Contact vulnerabilities in the search query and evaluated the entire data set with over 25,000 Smart Contracts. The experimental results show manual Pen Testing significantly outperformed the automated Smart Contract tools in terms of execution time, precision and soundness of issues detected. In addition, manual Penetration Testing uncovered over 12 previously unknown instances with the Batch Overflow vulnerability.

Even as Blockchain technology for Smart Contract applications is relatively new, this holds huge promise for future of contracts. The Blockchain attack vectors which can exploit the vulnerabilities and perform cybersecurity attacks on the

Table 7 Comparison of Pen Testing Solution with Previous/ untested versions

Security Feature	Pre-Pen Test-1	Pen Test	Post-Pen Test-1	Post-Pen Test-2	Production
Tamper Proof	X	X	X	√	√
Authentication	X	√	√	√	√
Decentralized	X	X	√	X	√
Authorization	√	X	√	√	√

Blockchain networks. This can in turn slow down the adoption process. Most of the attack vectors at end user or data integrity level can easily be avoided by creating awareness and effective Blockchain implementation users, others like Network and Application levels can only be mitigated with professional expertise. OWASP Top10 vulnerabilities are mapped to threats and attacks on Blockchain, which also illustrates that most Cybersecurity attacks, can be performed on both cloud-hosted applications and Blockchain-based Smart Contract applications.

References

1. Greenspan G (2018) Why Many Smart Contract Use Cases Are Simply Impossible. Retrieved March 10, 2020, from https://www.coindesk.com/three-smart-contract-misconceptions
2. Tsankov P (2018) Security practical security analysis of smart contracts. ArXiv preprint, arXiv: 1806.01143v2
3. Wang F, Yuan Y, Rong C, Zhang J (2018) Parallel Blockchain: an architecture for CPSS-based smart societies. IEEE transactions of. Comput Soc 5(2):303–310
4. Zhang Y (2018) Smart contract-based access control for internet of things (IoT). ArXiv Preprint arXiv 1802(04410):2018
5. Xu L, Mcardle G (2018) Internet of too many things in smart transport: the problem, the side effects and the solution. IEEE Access 6: 62840–62848. https://doi.org/10.1109/ACCESS.2018.2877175
6. Li Y, Cheng X, Cao Y, Wang D, Yang Y (2018) Smart choice for the smart grid: narrowband internet of things (NB-IoT). IEEE Internet Things J 5(3):1505–1515. https://doi.org/10.1109/JIOT.2017.2781251
7. Amani S, Bégel M, Bortin M, Staples M (2018) Towards verifying Ethereum smart contract Bytecode in Isabelle/HOL. Proceedings of 7th ACM SIGPLAN international conference for certified program proofs (CPP), Los Angeles, 66–77
8. Wang S (2018) A preliminary research of prediction markets based on Blockchain powered smart contracts. Proceedings of IEEE international conference of Blockchain, 1287–1293
9. Chang T, Svetinovic D (2019) Improving Bitcoin ownership identification using transaction patterns analysis. IEEE Trans Syst Man Cyber Syst Pub 50:9–20. https://doi.org/10.1109/TSMC.2018.2867497
10. Australian Securities Exchange (2018) CHESS Replacement. Retrieved February 15, 2020 from https://www.asx.com.au/services/chess-replacement.htm
11. US Securities and Exchange Commission (2018). Investor Bulletin: Initial Coin Offerings. Retrieved February 5, 2020, from https://www.sec.gov/oiea/investor-alerts-and-bulletins/ib_coinofferings
12. Zhang J (2018) Cyber-physical social systems: the state of the art and perspectives. IEEE Trans Comput Soc 5(3):829–840
13. What is a DAO? (2018) Retrieved February 17, 2020, from https://blockchainhub.net/dao-decentralized-autonomous-organization
14. Wan J, Li J, Imran M, Li M, Fazal A (2019) Blockchain-based solution for enhancing security and privacy in smart factory. IEEE transactions on industrial informatics (early access), IEEE systems, man, and cybernetics society. https://doi.org/10.1109/TII.2019.2894573
15. Pouttu A, Liinamaa O, Destino G (2018) 5G test network (5GTN) — environment for demonstrating 5G and IoT convergence during 2018 Korean Olympics between Finland and Korea," IEEE INFOCOM 2018 - IEEE conference on computer communications workshops (INFOCOM WKSHPS), Honolulu, HI, 2018, pp. 1–2, https://doi.org/10.1109/INFCOMW.2018.8406996
16. Choo K, Gritzalis S, Park J (2018) Cryptographic solutions for industrial internet-of-things: research challenges and opportunities. IEEE Trans Industrial Info 14(8):3567–3569. https://doi.org/10.1109/TII.2018.2841049
17. Tonelli R, Lunesu M, Pinna A, Taibi D, Marchesi M (2019) Implementing a microservices system with Blockchain smart contracts. IEEE international workshop on Blockchain oriented software engineering (IWBOSE), Hangzhou. https://doi.org/10.1109/IWBOSE.2019.8666520
18. Amoordon A, Rocha H (2019) Presenting Tendermint: Idiosyncrasies, Weaknesses, and Good Practices. IEEE international workshop on Blockchain oriented software engineering (IWBOSE), Hangzhou. https://doi.org/10.1109/IWBOSE.2019.8666541
19. Yamashita K, Nomura Y, Zhou F, Pi B, Jun S (2019) Potential risks of hyper ledger fabric smart contracts. IEEE international workshop on Blockchain oriented software engineering (IWBOSE), Hangzhou. https://doi.org/10.1109/IWBOSE.2019.8666486
20. Al-Jaroodi J, Mohamed N (2019) Industrial applications of Blockchain. IEEE 9th annual computing and communication workshop and conference (CCWC), Las Vegas. https://doi.org/10.1109/CCWC.2019.8666530
21. The Energy Web Foundation (2018) Promising Blockchain Applications for Energy: Separating the Signal from the Noise. Retrieved April 2, 2020, from http://www.coinsay.com/wp-content/uploads/2018/07/Energy-Futures-Initiative-Promising-Blockchain-Applications-for-Energy.pdf
22. Mohamed N, Al-Jaroodi J (2019) Applying Blockchain in industry 4.0 applications. IEEE 9th annual computing and communication workshop and conference (CCWC), Las Vegas. https://doi.org/10.1109/CCWC.2019.8666558
23. Draper A, Familrouhani A, Cao D, Heng T, Han W (2019) Security applications and challenges in Blockchain. IEEE international conference on consumer electronics (ICCE), Las Vegas, NV https://doi.org/10.1109/ICCE.2019.8661914
24. Mahmood S, Hasan R, Ullah A, Sarker U (2019) SMART security alert system for monitoring and controlling container transportation. 4th MEC international conference on big data and Smart City (ICBDSC), Muscat. https://doi.org/10.1109/ICBDSC.2019.8645574
25. Tateishi T, Yoshihama S, Sato N, Saito S (2019) Automatic smart contract generation using controlled natural language and template. IBM J Res Dev (Early Access), IBM. https://doi.org/10.1147/JRD.2019.2900643
26. Wang S, Ouyang L, Yuan Y, Ni X, Han X, Wang F (2019) Blockchain-enabled smart contracts: architecture, applications, and future trends. IEEE transactions on systems, man, and cybernetics: systems (early access), IEEE systems, man, and cybernetics society. https://doi.org/10.1109/TSMC.2019.2895123
27. Hildenbrandt E (2018) KEVM: A complete formal semantics of the Ethereum virtual machine. IEEE 31st computer Security Foundation symposium (CSF), 204–217
28. Ozyilmaz R, Yurdakul A (2019) Designing a Blockchain-based IoT with Ethereum, swarm, and LoRa: the software solution to create high availability with minimal security risks. IEEE consumer electronics magazine, volume: 8, issue 2, 28–34. IEEE Consum Electron Soc 8:28–34. https://doi.org/10.1109/MCE.2018.2880806
29. Knirsch F, Unterweger A, Engel D (2018) Privacy-preserving Blockchain-based electric vehicle charging with dynamic tariff decisions. Compute. Sci. Res. Develop. 33(1–2):71–79
30. Suliman A, Husain Z, Abououf M, Alblooshi M, Salah K (2019) Monetization of IoT data using smart contracts. IET Networks 8(1): 32–37. https://doi.org/10.1049/iet-net.2018.5026

31. Wood G (2016). Ethereum: A secure decentralized generalized transaction ledger. Retrieved March 15, 2020, from https://ethereum.github.io/yellowpaper/paper.pdf
32. Alladi T, Chamola V, Parizi R Choo R (2019) Blockchain applications for industry 4.0 and industrial IoT: a review. IEEE access, special section on distributed computing infrastructure for cyber-physical systems, volume 2019 (7). https://doi.org/10.1109/ACCESS.2019.2956748
33. Ch R, Gadekallu T, Abidi M, Al-Ahmari A (2020) Computational system to classify cyber crime offenses using machine learning. MDPI J Sustainability 12. https://doi.org/10.3390/su12104087
34. Azab A, Alazab M, Aiash M (2016) Machine learning based botnet identification traffic. In 2016 IEEE Trustcom/BigDataSE/ISPA (pp 1788-1794). IEEE
35. Reddy GT, Sudheer K, Rajesh K, Lakshmanna K (2014) Employing data mining on highly secured private clouds for implementing a security-asa-service framework. J Theor Appl Inf Technol 59(2):317–326
36. Qin R, Yuan Y, Wang Y (2018) Research on the selection strategies of Blockchain mining pools. IEEE Trans Comput Soc 5(3):748–757
37. Gatteschi V, Lamberti F, Demartini C, Pranteda C, Santamaria V (2018) Blockchain and smart contracts for insurance: is the technology mature enough? IEEE Future Internet 10(2):20–26
38. Lin C, Wang Z, Deng J, Wang L, Ren J, Wu G (2018) mTS: temporal-and spatial-collaborative charging for wireless rechargeable sensor networks with multiple vehicles. IEEE INFOCOM 2018 - IEEE conference on computer communications. Honolulu, HI 2018:99–107. https://doi.org/10.1109/INFOCOM.2018.8486402
39. Struye J, Braem B, Latré S, Marquez-Barja J (2018) The CityLab testbed — large-scale multi-technology wireless experimentation in a city environment: neural network-based interference prediction in a smart city, vol 2018. IEEE INFOCOM 2018 - IEEE conference on computer communications workshops (INFOCOM WKSHPS), Honolulu, pp 529–534. https://doi.org/10.1109/INFCOMW.2018.8407018
40. Shah B, Chen Z, Yin F, Khan I, Ahmad N (2018) Energy and interoperable aware routing for throughput optimization in clustered IoT-wireless sensor networks. Futur Gener Comput Syst 81:372–381
41. Shah B, Zhe C, Yin F, Khan I, Begum S, Faheem M, Khan F (2018) 3D weighted centroid algorithm & RSSI ranging model strategy for node localization in WSN based on smart devices. Sustain Cities Soc 39:298–308
42. Numan M, Subhan F, Khan WZ, Hakak S, Haider S, Reddy G, Alazab M (2020) A systematic review on clone node detection in static wireless sensor networks. IEEE Access 8:65450–65461
43. Bhattacharya S, Kaluri R, Singh S, Alazab M, Tariq U (2020) A novel PCA-firefly based XGBoost classification model for intrusion detection in networks using GPU. Electronics 9(2):219

Akashdeep Bhardwaj is currently working in School of Computer Science, University of Petroleum and Energy Studies, Dehradun, India. He completed his Bachelors of Engineering in Computer Science at Pune University, Pune, India, Post Graduate Diploma in Management, AIMA-CME, New Delhi, India and Ph.D (Computer Science), University of Petroleum and Energy Studies Dehradun. His areas of research are Cyber Security, Digital Forensics, Cloud Security, Information Security, IT Management, IT Infrastructure. Mailing Address: School of Computer Science, University of Petroleum and Energy Studies, Dehradun, India. Pin Code: 248001. E-Mail id: bhrdwh@yahoo.com

Syed Bilal Hussain Shah is currently a Postdoctoral Researcher with the School of Software, Dalian University of Technology, China. He authored/coauthored more than 25 research articles in reputable journals and conferences, such as Peer-to-Peer Networking and Applications, Future Generation Computer Systems IF, and Sustainable Cities and Society. Furthermore, he published articles in ACM, the IEEE, and Springer conferences. His main research interests include wireless sensor networks, the IoT, throughput optimization in WSN, node localization, energy efficient routing in smart wireless sensor networks, distributed and centralized clustering in WSN, IoT-based cognitive radio, opportunistic networks, and Industry 4.0 technology. He presented his article in a conference at Cambridge, U.K., in July 2017. Mailing Address: School of Software, Dalian University of Technology China- 116,000. E-Mail id: bilalshah@dlut.edu.cn

Achyut Shankar Amity School of Engineering and Technology is currently working as an Assistant Professor in Amity University, India. He completed his Ph.D in Vellore Institute of Technology, India. His areas of research are Computer Networks, Security, Blockchain. Mailing Address: Department of Computer Science, Amity University, Noida, Uttar Pradesh, India. Pin Code: 201313. E-mail: ashankar2711@gmail.com

Mamoun Alazab (Senior Member, IEEE) received the Ph.D. degree in computer science from the School of Science, Information Technology and Engineering, Federation University of Australia. He is currently an Associate Professor with the College of Engineering, IT, and Environment, Charles Darwin University, Australia. He is also a Cyber Security Researcher and a Practitioner with industry and academic experience. His research is multidisciplinary that focuses on cyber security and digital forensics of computer systems with a focus on cybercrime detection and prevention. He has more than 150 research articles in many international journals and conferences, he delivered many invited and keynote speeches, 24 events in 2019 alone. He convened and chaired more than 50 conferences and workshops. He also works closely with government and industry on many projects, including Northern Territory (NT) Department of Information and Corporate Services, IBM, Trend Micro, the Australian Federal Police (AFP), the Australian Communications and Media Authority (ACMA),Westpac, and United Nations Office on Drugs and Crime (UNODC). He is also the Founding Chair of the IEEE NT Subsection. Mailing Address: College of Engineering, IT and Environment, Charles Darwin University, NT 0909 Australia. E-Mail: alazab.m@ieee.org.

Manoj Kumar is currently working in School of Computer Science, University of Petroleum and Energy Studies, Dehradun, India. He completed his Bachelors in Technology in Computer Science Engineering at Kurukshetra University, Masters in Technology in Computer Science Engineering at ITM University, India, M.Sc. (Information Security & Digital Forensics), ITB, Ireland, and Ph.D. (Computer Science)(DIF), The Northcap University, India. His areas of specialization are Digital Image Forensics, Image Processing, Information Security, Machine Learning, Artificial Intelligence. Mailing Address: School of Computer Science, University of Petroleum and Energy Studies, Dehradun, India. Pin Code: 248001. E-Mail: wss.manojkumar@gmail.com.

G Thippa Reddy is currently working as Assistant Professor (Senior) in School of Information Technology and Engineering, VIT, Vellore, Tamil Nadu, India. He obtained his Bachelor of Technology degree in Computer Science and Engineering from Nagarjuna University, Andhra Pradesh, India, Master of Engineering in Computer Science and Engineering from Anna University, Chennai, Tamil Nadu, India and completed his Ph.D. in Vellore Institute of Technology, Vellore, Tamil Nadu, India. He has 14 years of experience in teaching. He produced more than 25 international/national publications. Currently, his research interests include Machine Learning, Deep Learning, Computer Vision, Big Data Analytics, Blockchain. Mailing Address: SITE, VIT University, Vellore, Tamil Nadu, India- 632,014. E-Mail id: thippareddy.g@vit.ac.in.

Affiliations

Akashdeep Bhardwaj[1] · Syed Bilal Hussian Shah[2] · Achyut Shankar[3] · Mamoun Alazab[4] · Manoj Kumar[1] · Thippa Reddy Gadekallu[5]

Akashdeep Bhardwaj
bhrdwh@yahoo.com

Achyut Shankar
ashankar2711@gmail.com

Mamoun Alazab
alazab.m@ieee.org

Manoj Kumar
wss.manojkumar@gmail.com

[1] School of Computer Science, University of Petroleum and Energy Studies, Dehradun, India

[2] School of Software, Dalian University of Technology China, Dalian, China

[3] Department of Computer Science, Amity University, Noida, Uttar Pradesh, India

[4] College of Engineering, IT and Environment, Charles Darwin University, Brinkin, NT 0909, Australia

[5] School of Information technology and Engineering, Vellore Institute of Technology, Vellore, India