BARRIERS TO ADOPTING DISTRIBUTED ONLINE ATTACK DETECTION BASED ON BLOCKCHAIN FOR WEB APPLICATION VULNERABILITIES

M.Mailsamy[1], K.Thangadurai[2], R.Uma Mageswari[3]

[1,2]Associate Professor, Department of Computer Science and Engineering,AEC,Salem

[3]Assistant Professor, Department of Computer Science and Engineering,AEC,Salem

Abstract:

Blockchain is really trendy these days. A distributed ledger on a peer-to-peer network that is completely open to everyone was the block chain. It is composed of blocks that include hash values and data. Before a new transaction can be added to the block chain, researchers must validate it; this process is called mining. Mining is expensive and requires a lot of processing power. Since the block chain is a peer-to-peer network, the data is maintained in every node. The block chain network has increased to 190GB thanks to the increasing number of transactions that are processed through it. It is a problem because a cheap laptop can only hold so much data. This study developed a revolutionary, less expensive system than the block chain method. We choose web applications as our use case since they are increasingly overtaking all other methods of accessing internet services in popularity. The immutability, data security, and data dissemination features of the block chain were all taken into account. The Merle tree concept provides immutability, hashing was used to achieve security, and an open source data distribution tool is used to spread the data. This paper provides innovative methods for preventing malicious data upload using MIME, cross-site programming, and cross-site request manipulation.

Keywords: Blockchain, security, web application, detection, cyber-security

1. Introduction

Web apps are one of the most essential structures that make Internet use more prevalent. This makes online apps easier to use, but it also renders them vulnerable to cyber assaults. To safeguard online applications and avoid cyber threats, Signatures are defined as known attack types in signature-based detection, and only the sorts of assaults designated as signatures.

On their computers, many small businesses employ inventory management systems, financial accounting, and/or acquisition systems. [1] Depending on the size of the business, the database that powers the programme could well be kept on a windows machine on the grounds or on an internet to which the app connects. When the data is separate from the app, database server access (RDA) protocols could be used to link the implementation to the host computer. Although RDBMSs built on the RDA protocols, like remote Wordpress and remote Oracle, are common among software engineers, they are subject to dangers such as illegal users and data tampering at the data layer, bypassing the application software. [4]

The resistance of blockchain to data change should be considered to defend against the possibility of unlawful data tampering. Blockchain is based on a decentralised P2 network, which means there is no central server or database, and the data is secured by decentralised consensus.. Oracle Corporation released Oracle Blockchain in 2017 [7], however its design is excessively complicated [8].

A solution that combined blockchain with RDA protocol-based remote RDBMS would be beneficial in addition to current technologies. [9] Oracle databases, which need the RDA protocol.

As a result, the problem is to come up with a strategy that is easy to use for software developers while yet providing data security at the database level. [11-12] To keep things simple, the approach should use as few third-party libraries as possible. As a result, BIRD (Blockchain- Integrated Distant Database) was created to connect blockchain to a remote RDBMS server.

Many businesses and organisations employ dynamic database web apps to foster collaboration and improve customer service. To make educated judgments, educational institutions, for example, rely significantly on databases holding highly sensitive student records. A single record bridge created by an assault has the potential to lead to a faulty or incorrect choice. Database security is threatened by a variety of attacks, including static leakage, linkage leakage, dynamic leakage, spoofing, and SQL injection assaults, which are referred to as SQLIA in this research. SQLIA puts the security, integrity, functionality, and availability of any online application's database at risk. [14]

Furthermore, they are the most effective way for unlawfully gathering data from databases, allowing hackers to get access to databases and steal sensitive data. Consider a login page where a genuine user enters his username and password to gain access to a secure page where he can examine personal information or post comments on a social networking site. [15] The SQL query is built and submitted to the database for verification when the user submits the data. The user is granted access to the system if it is valid. This implies that the login page and the database communicate in order to validate the username and password combination, with access allowed after verification.

The hacker can use SQL Injection to bypass the login form's validation and see the script by entering specially generated SQL commands. This is only feasible if the inputs are not properly sanitised (i.e. rendered invulnerable) before being transmitted straight to the database through SQL query. SQL Injection vulnerabilities allow an attacker to get access to a database.

2. MATERIAL AND METHODS

The materials and processes are explained in this section. .

A. Blockchain

Blockchain technology has recently grabbed the interest of a number of international organisations, enterprises, and institutions, with some researchers arguing that it is even more powerful than the Internet. The blockchain market was valued $228 million in 2016, according to Allied Industry Research, and could be worth $5.4 billion by 2023. Because to blockchain, people no longer need a third-party agent to provide security and verification in product or service transfer activities.Blockchain's "trust protocol" creates a secure, transparent, and accountable ecosystem. Communication has become incredibly simple all around the world thanks to the Internet.

The authenticity of the activities conducted by the participant nodes ensures the establishment of these blocks. According to Zhao et al., the most essential aspect of the blockchain is that it supports dependable and transparent operations through network-based computations rather than human monitoring or control. The following are some of the benefits of blockchain technology.

1. Data stored on the blockchain cannot be altered or removed.

2. It does not require a central authority to function; its dispersed structure cannot be controlled, cancelled, or closed.
3. Smart contracts may be used to automate certain tasks.

As shown in Fig. 1, the blockchain development may be divided into three phases:.

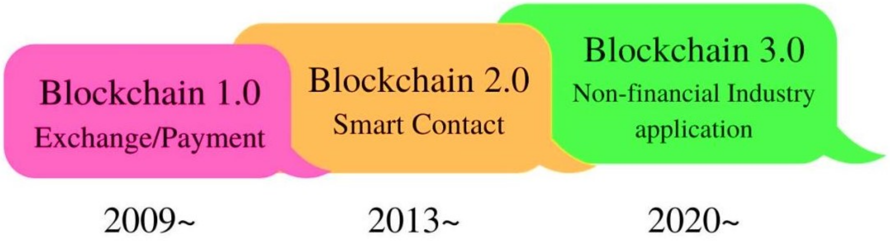


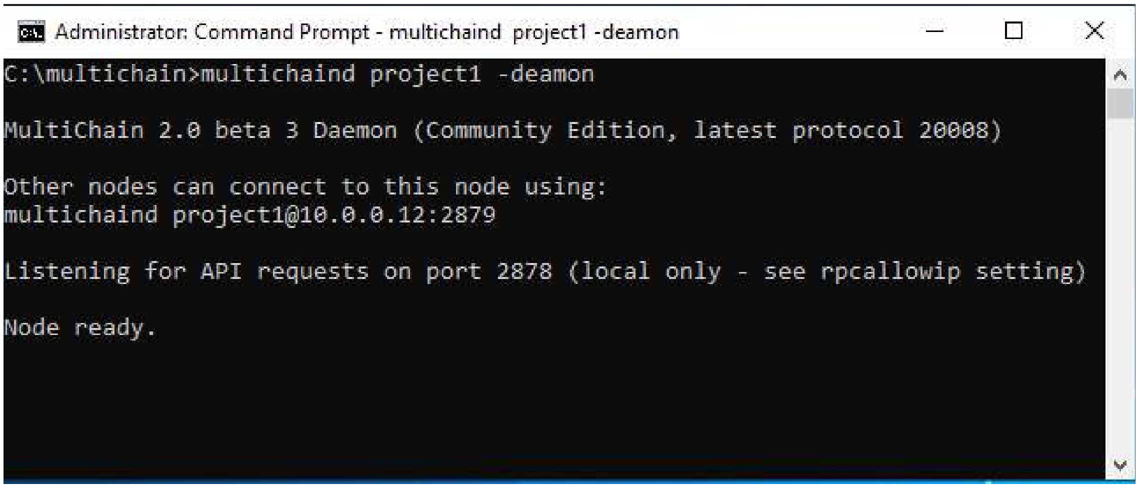
Fig. 1. Evolution of blockchain

B. MultiChain

Our decision to adopt MultiChain as a blockchain application was driven by the advantages of API support for a range of programming languages, ease of use, performance, and documentation. MultiChain is a private blockchain system that regulates data access through a list of registered members. The only people who can read and write blocks in the chain are those who have registered. MultiChain's consensus strategy is the Round Robin (RR) scheduling technique. According to the RR algorithm, each block must have a signature from the participant who desires to produce it. Multichain allows for a wide range of asset issuance and data streams (key-value databases). In the NoSQL paradigm, data streams can be viewed of as a database with varied permissions separated from the rest of the system.

3. A distributed web assault detection application built on the blockchain

The suggested idea is described in detail in this section. Three web servers were used in the test environment: S1, S2, and S3. Each Windows 10 server includes a 2.40 GHz Intel Xeon processor and 32 GB of RAM. The multichain application is installed on each server. As shown in Fig. 2, a chain named project1 was created and uploaded to the S1 server. In the MultiChain structure, other nodes must be authorised to join, read, and write to the chain.

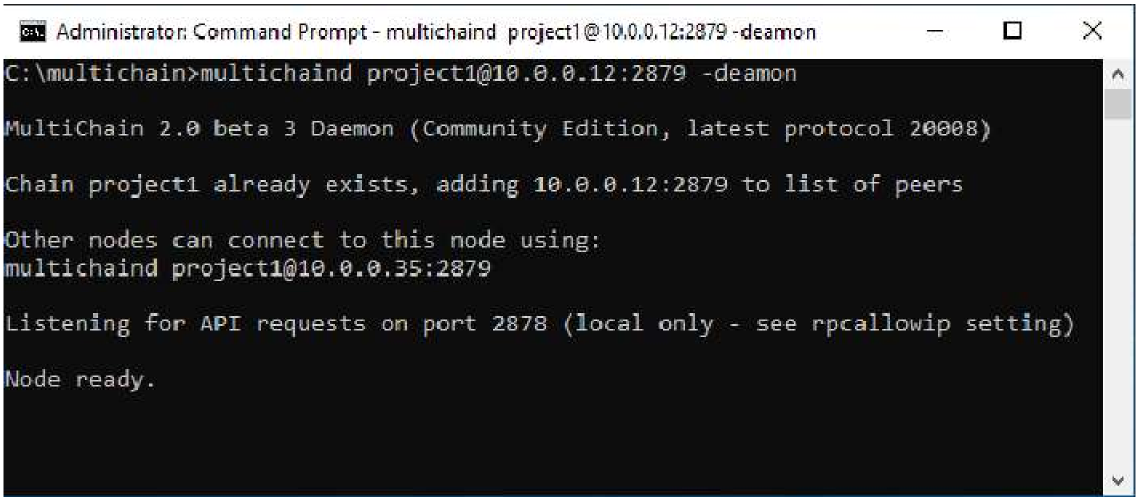

Administrator: Command Prompt - multichaind project1 -deamon

```
C:\multichain>multichaind project1 -deamon

MultiChain 2.0 beta 3 Daemon (Community Edition, latest protocol 20008)

Other nodes can connect to this node using:
multichaind project1@10.0.0.12:2879

Listening for API requests on port 2878 (local only - see rpcallowip setting)

Node ready.
```

Fig. 2.Using MultiChain to make a chain

Figure 3 shows the connection between the IP address of the S2 server, 10.0.0.35, and the MultiChain formed in the IP address and port of the S1 server, 10.0.0.12:2879. Once the S2 machine's node is attached to the chain, and the S2 node is now one of project1's owners, data synchronisation occurs. The remaining participating nodes, for example, can still use chain named project if the S1 node departs the system.

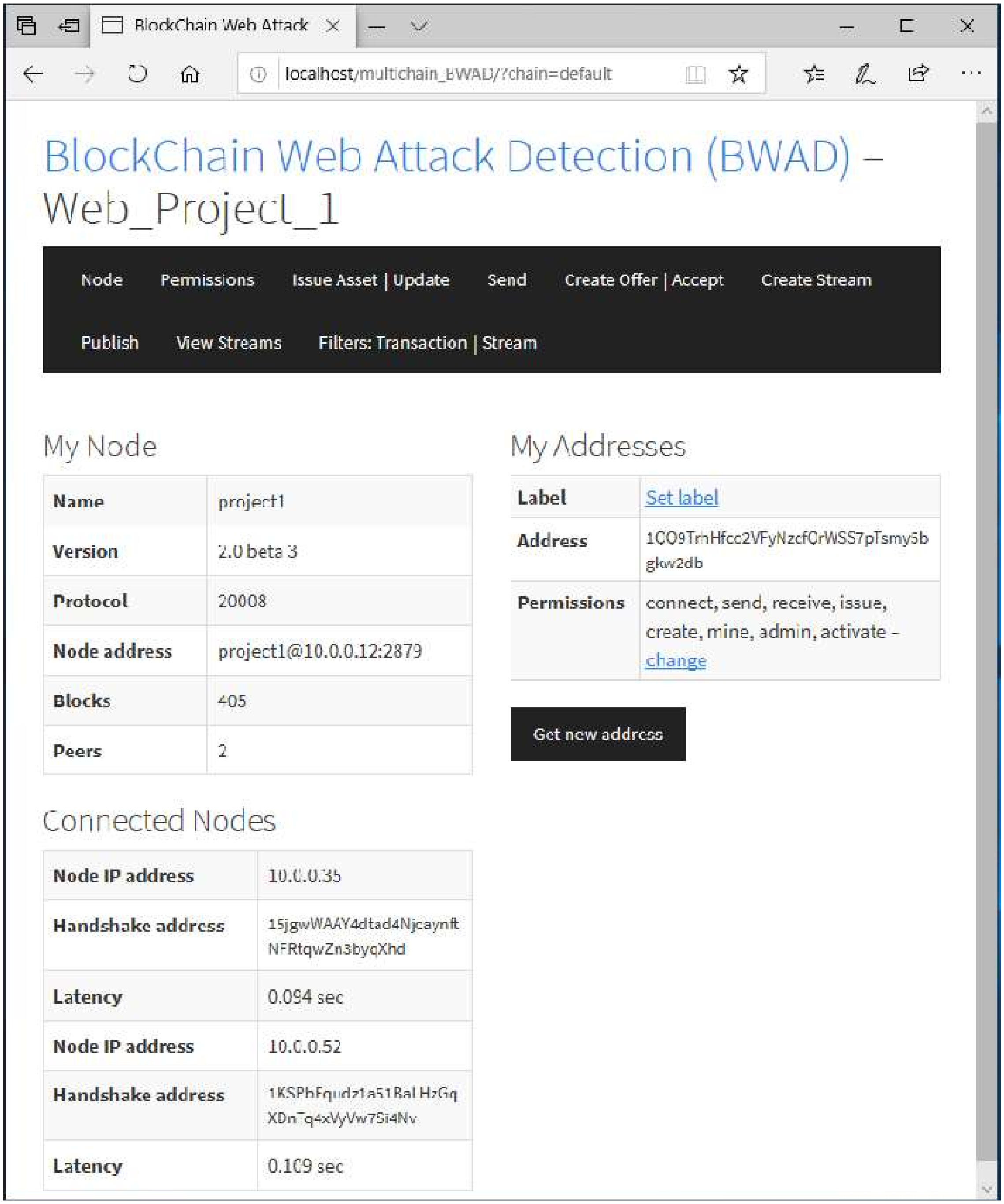
.


Administrator: Command Prompt - multichaind project1@10.0.0.12:2879 -deamon

```
C:\multichain>multichaind project1@10.0.0.12:2879 -deamon

MultiChain 2.0 beta 3 Daemon (Community Edition, latest protocol 20008)

Chain project1 already exists, adding 10.0.0.12:2879 to list of peers

Other nodes can connect to this node using:
multichaind project1@10.0.0.35:2879

Listening for API requests on port 2878 (local only - see rpcallowip setting)

Node ready.
```

Fig. 3.Joining the node to the chain

The MultiChain Explorer tool, which allows web-based scanning of blockchain operations, was provided to the developers by MultiChain. MultiChain Explorer has been installed on all servers, and the necessary procedures have been performed. As a result, nodes' blockchain activity may be monitored. Figure 4 shows a screenshot of the MultiChain Explorer web interface, revealing that the three nodes belong to the project1 chain. As indicated in the diagram, the server with the IP address 10.0.0.12 and port 2879 started the chain named project1. Nodes with the IP addresses 10.0.0.35 and 10.0.0.52 are then added to the chain.



BlockChain Web Attack Detection (BWAD) – Web_Project_1

My Node

Name	project1
Version	2.0 beta 3
Protocol	20008
Node address	project1@10.0.0.12:2879
Blocks	405
Peers	2

My Addresses

Label	Set label
Address	1CQ9TmHfcc2VFyNzcfQrWSS7pTsmy5bgkw2db
Permissions	connect, send, receive, issue, create, mine, admin, activate - change

Connected Nodes

Node IP address	10.0.0.35
Handshake address	15jgwWAAY4dtad4NjcaynftNFRtqwZn3byqXhd
Latency	0.094 sec
Node IP address	10.0.0.52
Handshake address	1KSPhFqudz1a51BaLHzGqXDnTq4xVyVw7Si4Nv
Latency	0.109 sec

Fig. 4.MultiChain Explorer programme screenshot

PHP-based web applications and the Apache web server make up the server. MultiChain's PHP API allows web applications on servers to access data from MultiChain nodes that have already been installed. The created application is depicted in Figure 5 as a block diagram.

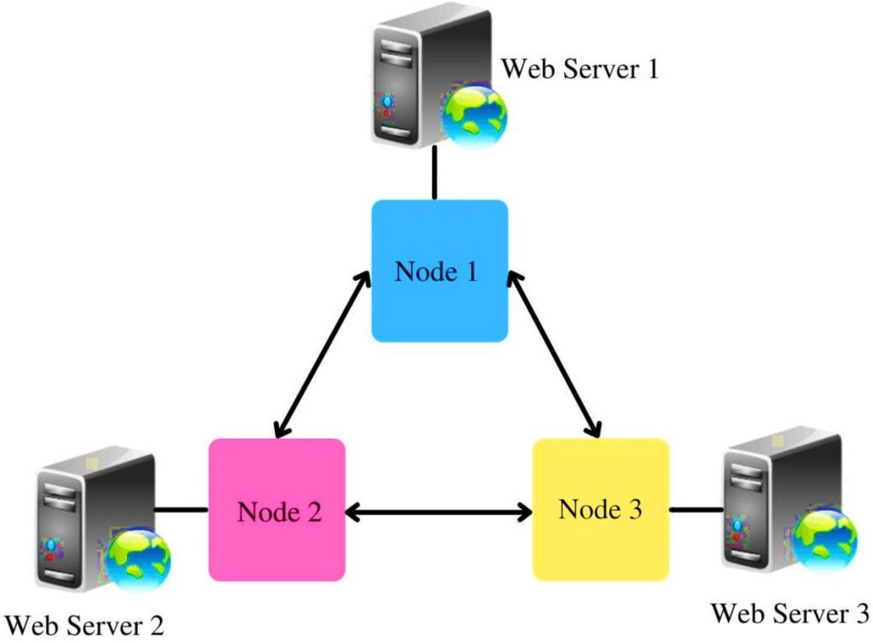


Fig. 5.The generated application's block diagram

In this inquiry, the blockchain is used as a public and transparent signature pool. The MultiChain's data stream functionality is excellent for this in terms of efficiency and data storage structure. All nodes have joined the BAWD Signature List data stream, which has been built as a public signature pool on the blockchain. The blockchain black list was treated as if it were empty in our investigation, and we made sure that this pool was replenished with attack signatures recognised by the system's web apps.

Blacklists are created for each online programme in order to detect web-based attacks. Signatures were used to detect web-based attacks using the CSIC 2010 dataset. SQL Injection, Cross-Site Scripting, and other well-known online security vulnerabilities

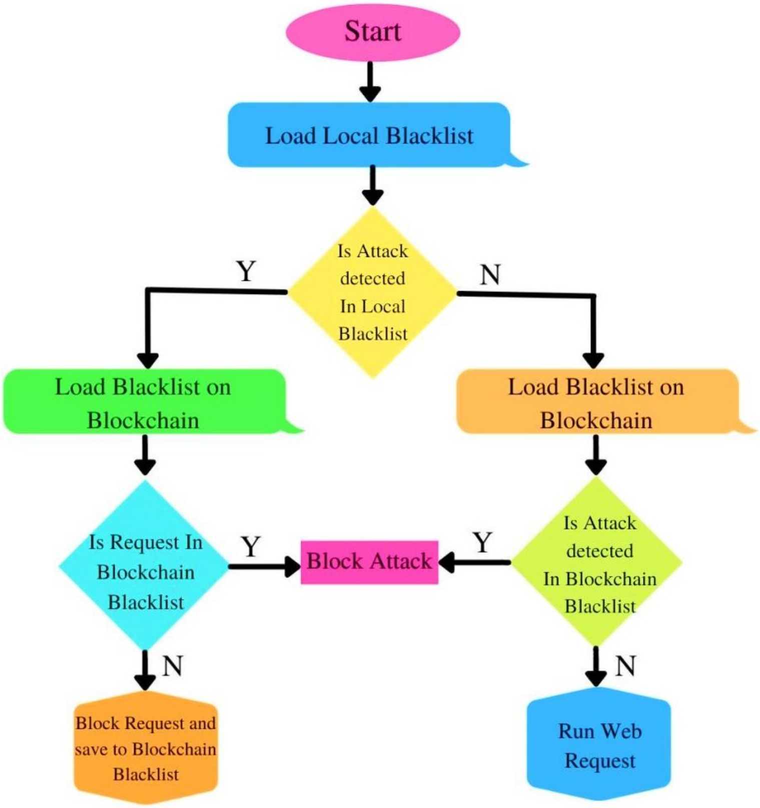
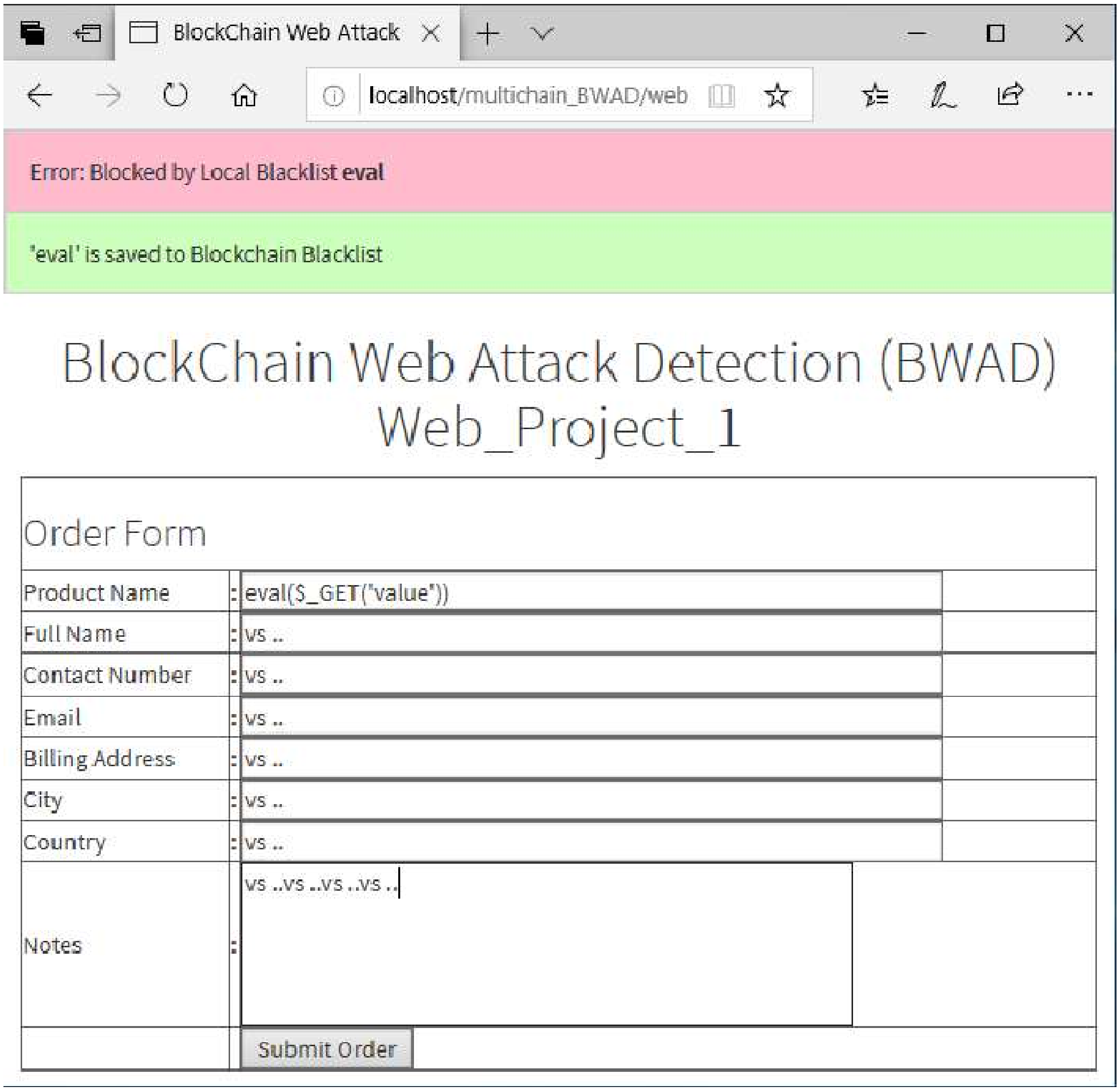

Start
Load Local Blacklist
Is Attack detected In Local Blacklist
Y
N
Load Blacklist on Blockchain
Load Blacklist on Blockchain
Is Request In Blockchain Blacklist
Y
Block Attack
Y
Is Attack detected In Blockchain Blacklist
N
N
Block Request and save to Blockchain Blacklist
Run Web Request

Fig. 6. Flow diagram of the developed application

Figure 6 depicts the flow diagram of the created application. Requests from web servers are verified against a local blacklist in the first stage. The request is stopped if an attack is discovered, and it is assessed whether the attack signature is defined in the blockchain's blacklist. If the necessary attack signature isn't on the blockchain blacklist, this signature is added. As a result, the blockchain blacklist has been updated, and other participant web servers will be able to recognise this attack signature in the future. If an attack is not identified on the web server's local blacklist, the request is verified against the blockchain blacklist in the second stage. The relevant request is performed if an attack is not discovered as a result of this detection. Otherwise, the request will be turned down. A scenario with screenshots is shown to better understand the functionality of the produced programme. In this situation, the "eval" signature was used to attack the web application on the S1 server that contained the order form. The request is denied because the "eval" signature is on this web application's local blacklist. A snapshot of this circumstance is shown in Figure 8.



Error: Blocked by Local Blacklist **eval**

'eval' is saved to Blockchain Blacklist

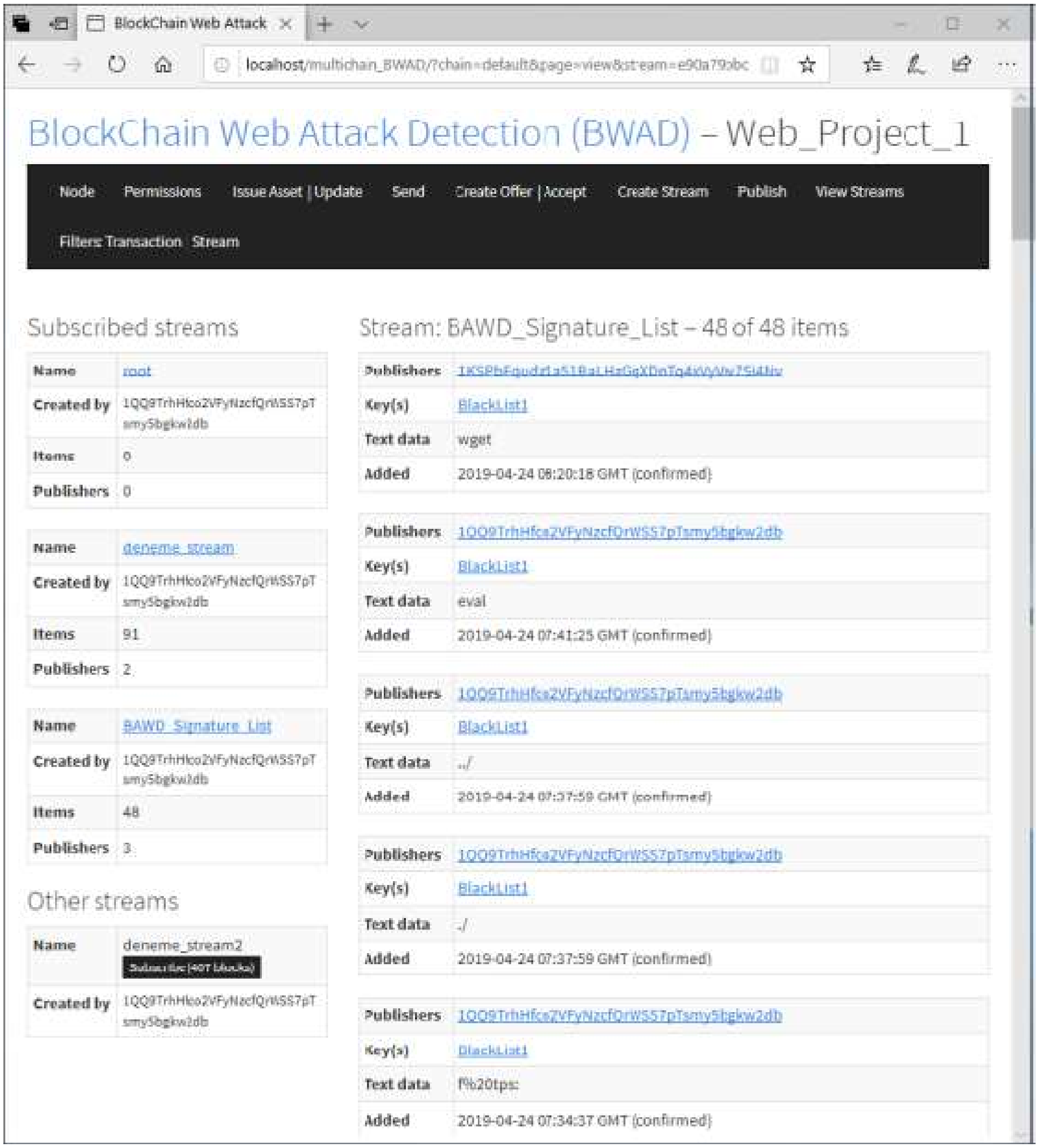
BlockChain Web Attack Detection (BWAD)
Web_Project_1

Order Form

Product Name	:	eval($_GET("value"))
Full Name	:	vs ..
Contact Number	:	vs ..
Email	:	vs ..
Billing Address	:	vs ..
City	:	vs ..
Country	:	vs ..
Notes	:	vs ..vs ..vs ..vs ..

Fig. 7.Detecting attacks and keeping the blockchain blacklist up to date

The "eval" signature was then discovered to be missing from the blockchain blacklist, and this attack signature was added to the blockchain's BAWD Signature List data stream. This data stream is shown in Fig. 8 on the MultiChain Explorer programme.



BlockChain Web Attack Detection (BWAD) – Web_Project_1

Subscribed streams

Name	root
Created by	1QQ9TrhHfco2VFyNzcfQrWSS7pT smy5bgkw2db
Items	0
Publishers	0

Name	deneme_stream
Created by	1QQ9TrhHfco2VFyNzcfQrWSS7pT smy5bgkw2db
Items	91
Publishers	2

Name	BAWD_Signature_List
Created by	1QQ9TrhHfco2VFyNzcfQrWSS7pT smy5bgkw2db
Items	48
Publishers	3

Other streams

Name	deneme_stream2
Created by	1QQ9TrhHfco2VFyNzcfQrWSS7pT smy5bgkw2db

Stream: BAWD_Signature_List – 48 of 48 items

Publishers	1KSPhFqudrJa51BaLHzGgXDnTq4xVyVw7Si4Nv
Key(s)	BlackList1
Text data	wget
Added	2019-04-24 08:20:18 GMT (confirmed)

Publishers	1QQ9TrhHfca2VFyNzcfQrWSS7pTsmy5bgkw2db
Key(s)	BlackList1
Text data	eval
Added	2019-04-24 07:41:25 GMT (confirmed)

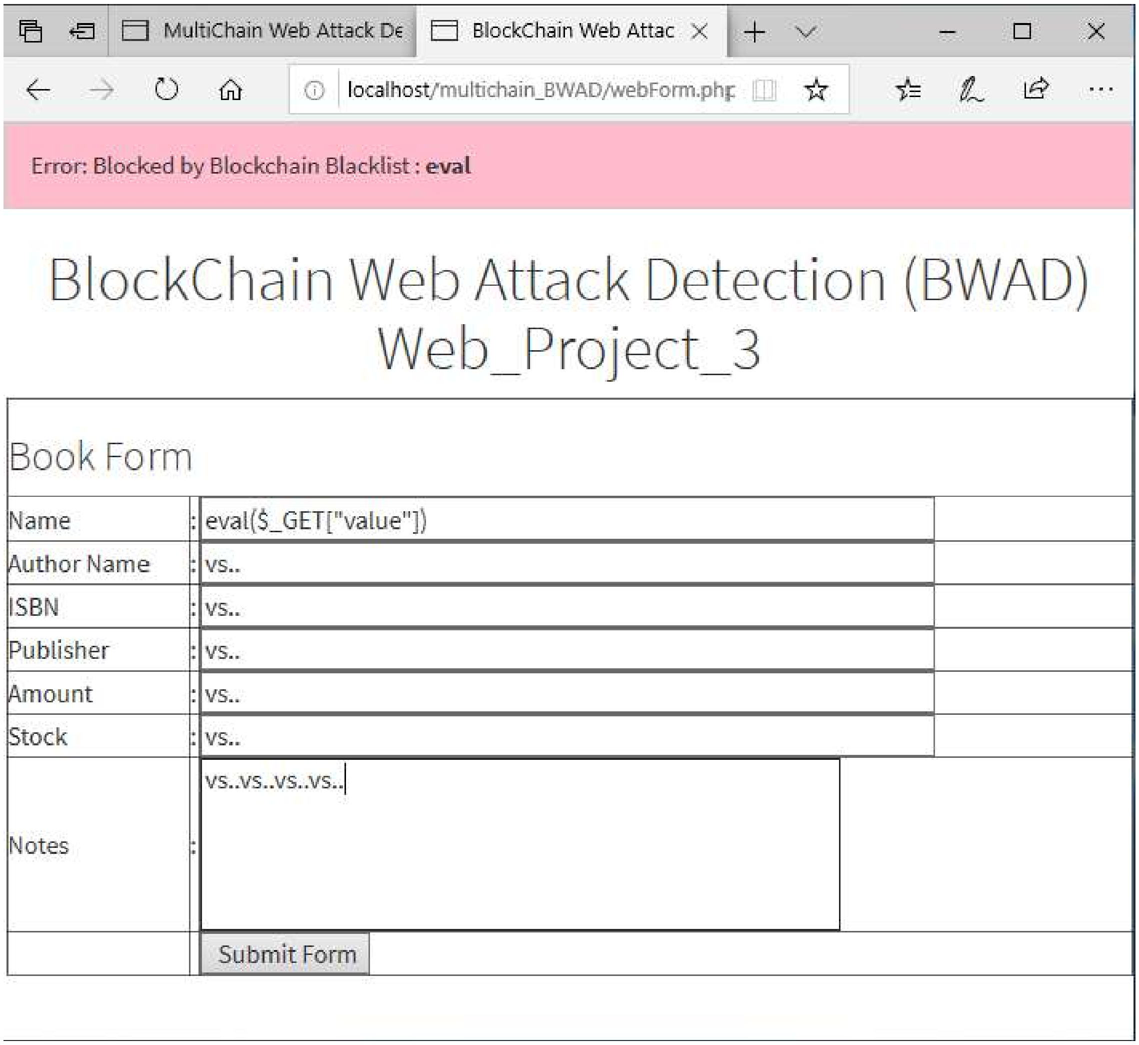
Publishers	1QQ9TrhHfca2VFyNzcfQrWSS7pTsmy5bgkw2db
Key(s)	BlackList1
Text data	../
Added	2019-04-24 07:37:59 GMT (confirmed)

Publishers	1QQ9TrhHfca2VFyNzcfQrWSS7pTsmy5bgkw2db
Key(s)	BlackList1
Text data	./
Added	2019-04-24 07:37:59 GMT (confirmed)

Publishers	1QQ9TrhHfca2VFyNzcfQrWSS7pTsmy5bgkw2db
Key(s)	BlackList1
Text data	f%20tps:
Added	2019-04-24 07:34:37 GMT (confirmed)

Fig. 8.An example of a blockchain blacklist can be seen in the image below.

In a continuation of this scenario, the "eval" signature was used to attack the web application on the S3 server that contained the book form. The "eval" signature was not identified in this web application's local blacklist in the first step, hence the attack went undetected. The blockchain blacklist was checked for the "eval" signature in the second phase, and the attack was discovered this time. Figure 9 depicts this attack scenario in book form.



Error: Blocked by Blockchain Blacklist : **eval**

BlockChain Web Attack Detection (BWAD)
Web_Project_3

Book Form

Name	:	eval($_GET["value"])
Author Name	:	vs..
ISBN	:	vs..
Publisher	:	vs..
Amount	:	vs..
Stock	:	vs..
Notes	:	vs..vs..vs..vs..

Fig. 9. Attack detection through blockchain blacklist

Conclusions:

The block chain has been extremely popular in recent years. This system is costly to implement, and because it is a peer-to-peer network, it is difficult to keep large amounts of data in every node. In this research, we used a web application as a use case to demonstrate immutability and distribution of data aspects of block chain. We use XSS, CSRF, and malicious data upload protection in our web application.

References

1. Ghiasi, Mohammad, et al. "Cyber-attack detection and cyber-security enhancement in smart DC-microgrid based on blockchain technology and Hilbert Huang transform." *Ieee Access* 9 (2021): 29429-29440.

2. Guha Roy, Deepsubhra, and SatishNarayanaSrirama. "A blockchain-based cyber attack detection scheme for decentralized Internet of Things using software-defined network." *Software: practice and experience* 51.7 (2021): 1540-1556.
3. Xu, Weicheng, et al. "Blockchain-based secure energy policy and management of renewable-based smart microgrids." *Sustainable Cities and Society* 72 (2021): 103010.
4. Dehghani, Moslem, et al. "Cyber-attack detection in dc microgrids based on deep machine learning and wavelet singular values approach." *Electronics* 10.16 (2021): 1914.
5. Zhang, Jun, et al. "Deep learning based attack detection for cyber-physical system cybersecurity: A survey." *IEEE/CAA Journal of AutomaticaSinica* 9.3 (2021): 377-391.
6. Abdel-Basset, Mohamed, et al. "Federated intrusion detection in blockchain-based smart transportation systems." *IEEE Transactions on Intelligent Transportation Systems* 23.3 (2021): 2523-2537.
7. Arifeen, Md, et al. "A blockchain-based scheme for sybil attack detection in underwater wireless sensor networks." *Proceedings of International Conference on Trends in Computational and Cognitive Engineering*. Springer, Singapore, 2021.
8. Madichetty, Sreedhar, and Sukumar Mishra. "Cyber Attack Detection and Correction Mechanisms in a Distributed DC Microgrid." *IEEE Transactions on Power Electronics* 37.2 (2021): 1476-1485.
9. Chen, Jian, et al. "A multi-layer security scheme for mitigating smart grid vulnerability against faults and cyber-attacks." *Applied Sciences* 11.21 (2021): 9972.
10. Ferrag, Mohamed Amine, et al. "Deep learning-based intrusion detection for distributed denial of service attack in Agriculture 4.0." *Electronics* 10.11 (2021): 1257.
11. Almaiah, Mohammed Amin. "A new scheme for detecting malicious attacks in wireless sensor networks based on blockchain technology." *Artificial Intelligence and Blockchain for Future Cybersecurity Applications*. Springer, Cham, 2021. 217-234.
12. Kumar, Prabhat, et al. "A Distributed framework for detecting DDoS attacks in smart contract-based Blockchain-IoT Systems by leveraging Fog computing." *Transactions on Emerging Telecommunications Technologies* 32.6 (2021): e4112.
13. Kumar, Prabhat, Govind P. Gupta, and RakeshTripathi. "An ensemble learning and fog-cloud architecture-driven cyber-attack detection framework for IoMT networks." *Computer Communications* 166 (2021): 110-124.
14. Zhu, Li, et al. "Joint Security and Train Control Design in Blockchain-Empowered CBTC System." *IEEE Internet of Things Journal* 9.11 (2021): 8119-8129.
15. Inedjaren, Youssef, et al. "Blockchain-based distributed management system for trust in VANET." *Vehicular Communications* 30 (2021): 100350.