UDK 004.056 DOi 10.30837/ bi.2025.1(102).14

**N. Brynza[1,2], V. Borodavka[3], O. Teslenko [1]**

[1]Simon K. KNUE, Kharkiv, Ukraine, natalia.brynza@hneu.net,
ORCID iD: 0000-0002-0229-2874

[2]KhNURE, Kharkiv, Ukraine, nataliia.brynza@nure.ua,
ORCID iD: 0000-0002-0229-2874

[3]V. N. Karazin Kharkiv National University, Kharkiv, Ukraine, vladyslav.borodavka@karazin.ua,
ORCID iD: 0009-0002-3885-1364

[1]Simon K. KNUE, Kharkiv, Ukraine, oleh.teslenko@hneu.net,
ORCID iD: 0000-0003-3105-9323

# IMPLEMENTATION OF MEASURES TO COUNTERACT CYBER THREATS AND ABUSE IN INFORMATION SECURITY BASED ON BLOCKCHAIN TECHNOLOGIES

The selection of security methods for the ISMS of blockchain systems has been researched and justified, considering their specifics and the nature of threats. The key selection factors include ensuring reliability, decentralization, confidentiality, and cyberattack resilience. Various approaches were examined, including consensus algorithms, cryptographic methods, smart contract security, and network protocols. Based on the analysis, the implementation of a monitoring and response method was deemed appropriate, specifically utilizing tools such as iptables, ipset, fail2ban, and dynamic blacklists of IP addresses. This mechanism effectively combines various tools for automation and dynamic updating of security levels. By employing iptables and ipset, the system efficiently filters network traffic, while fail2ban monitors suspicious patterns and blocks malicious IP addresses based on configurable rules. The inclusion of dynamic blacklists adds an extra layer of protection by continuously updating the database of known malicious IP addresses, allowing real-time security adjustments. Monitoring results demonstrate that the system's effectiveness can vary significantly depending on the number of blocked attacks, the cyberattacks that bypassed the system, and the false positive rate.

WEB SERVICE, BLOCKCHAIN, NETWORK, CYBERSECURITY, CYBERATTACK, INFORMATION SECURITY MANAGEMENT SYSTEM, SECURITY MECHANISM, EFFICIENCY

**Н. О. Бринза, В. В. Бородавка, О. В. Тесленко. Імплементація засобів протидії кіберзагрозам і зловживанням у системах керування інформаційною безпекою на базі блокчейн-технологій.** У роботі здійснено аналіз і аргументовано обрано методи захисту, що враховують особливості блокчейн-систем та специфіку потенційних кіберзагроз. Вибір підходів грунтувався на потребі забезпечення високої надійності, децентралізованого управління, конфіденційності даних і стійкості до атак. Розглядалися різноманітні захисні стратегії, включаючи консенсусні алгоритми, криптографічні засоби, захист смарт-контрактів і особливості мережевої взаємодії. За результатами порівняльного аналізу доцільним виявилось застосування моніторингово-реагувальної моделі із використанням таких інструментів, як iptables, ipset, fail2ban і механізми динамічного формування чорних списків IP-адрес. Запропоноване рішення дозволяє поєднати ці засоби в єдину систему автоматизованого управління рівнями безпеки. Зокрема, iptables та ipset відповідають за ефективне фільтрування трафіку, fail2ban виявляє та блокує підозрілу активність, а динамічні чорні списки забезпечують оперативне оновлення бази загроз і адаптацію системи до нових викликів у реальному часі. Проведене тестування свідчить, що ефективність системи значною мірою залежить від кількості заблокованих атак, рівня проникнення загроз і частоти хибних спрацювань.

WEB-СЕРВІС, БЛОКЧЕЙН, МЕРЕЖА, КІБЕРЗАХИСТ, КІБЕРАТАКА, СИСТЕМА УПРАВЛІННЯ ІНФОРМАЦІЙНОЮ БЕЗПЕКОЮ, МЕХАНІЗМ ЗАХИСТУ, ЕФЕКТИВНІСТЬ

## Introduction

The development of information technology (IT) has become an integral part of society. Information is one of the most valuable resources in any business process, which determines the priority of ensuring information security (IS) as a key aspect of effective management. Information security includes a set of measures aimed at preventing and eliminating the risks of unauthorised access, processing, modification, analysis, unauthorised alteration or destruction of data.

Theoretical and practical aspects of IS are actively studied by both domestic and foreign scholars, in particular, Babenko V., Boyko A., Vasylieva T., Gontareva I., Horbenko I., Kachynskyi A., Leonov S., Kuzmenko O., Starkova O., Anderson R., Cardholm L., Kshetri N., Stephanides G., Tsiakis T., and others. Recently, the issue of information policy and IS has become particularly relevant.

In today's world, IT development is the backbone of many industries, including financial, healthcare, logistics and others. One of the key technologies that provides new opportunities for secure storage and transmission of information is blockchain, which ensures transparency of transactions, data immutability and a decentralised management model, making it attractive to many industries.

In a blockchain, data is stored in blocks that form a connected chain. The information in the blocks is chronologically consistent, as any changes or deletions are impossible without reaching a consensus among the network participants. It is a unique data management system that has a number of advanced features. The main difference between blockchains is the decentralised nature of management, which ensures trust in data without the need for centralised control. Traditional databases (DBs) usually lack the ability to share data between different companies. In blockchain networks, each participant has its own copy of the registry, the consistency of which is maintained automatically. In addition, in traditional databases, data can be edited or deleted, while in the blockchain, the data entered remains unchanged, as the system provides a high level of security. Once information is recorded in the blockchain, its modification becomes extremely difficult, which increases the reliability and security of the stored data [1].

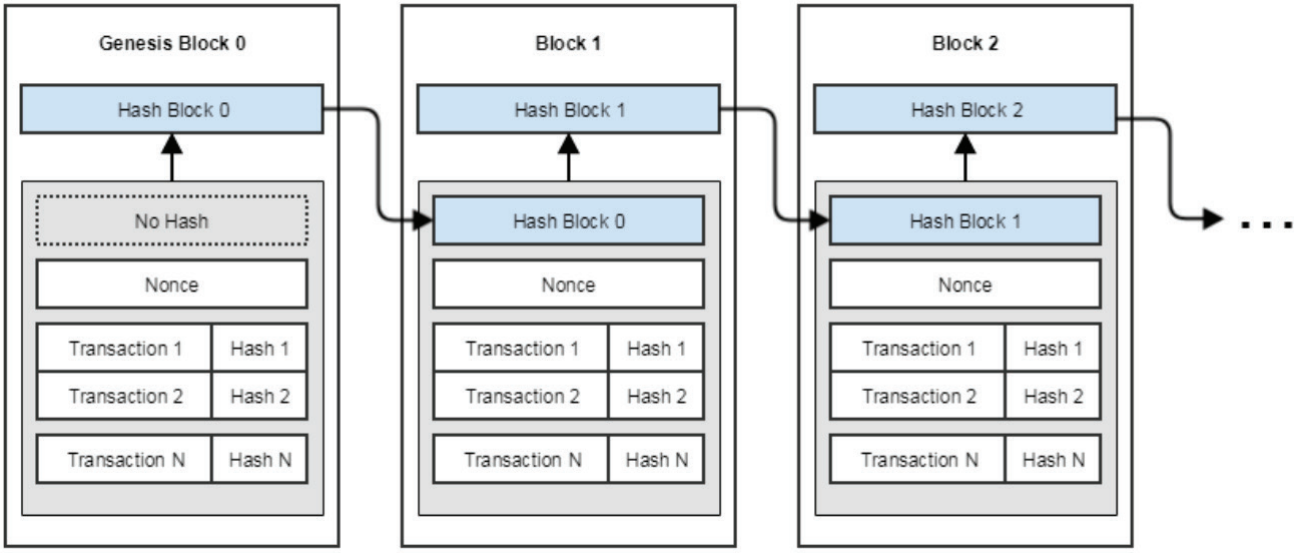


**Fig. 1. The blockchain structure**

Decentralisation in blockchain technology involves the transfer of control and decision-making from centralised entities (individuals, organisations or groups of them) to a distributed network. The transparency of decentralised blockchain systems reduces the dependence of participants on each other, eliminating the need for trust between them. This architecture limits the ability of one party to exert excessive influence or control while maintaining the functionality of the network. Data immutability is a key feature of the blockchain, which makes it impossible to change information once it has been entered into the register. In the event of errors in the records, a new transaction is added to correct them, while all previous transactions remain available for viewing in the history of changes. Thus, the network displays both the original and the corrected record. To confirm new transactions, the system uses consensus mechanisms to ensure that they are registered only if approved by the majority of participants.

Traditional databases pose a number of difficulties in keeping records of financial transactions. For example, when selling real estate, ownership is transferred to the buyer only after payment is made. However, the parties may register transactions separately, which creates risks of distrust: the seller may deny receiving funds, and the buyer may claim that payment has been made, even if this is not true. To avoid such disputes, a trusted third party is usually involved to monitor and confirm the transaction. However, a centralised intermediary complicates the process by creating a single point of vulnerability, which can lead to serious consequences for both parties in the event of a system failure. Blockchain technology offers an effective solution to these problems by creating a decentralised, tamper-proof system for recording transactions. In the case of a real estate transaction, blockchain ensures that there is a single register that is synchronised between the buyer and seller in real time. All transactions have to be mutually approved, and any discrepancies in the records are instantly reflected throughout the system, increasing transparency and trust in the data. Due to its properties, blockchain technology has gained wide popularity in various industries. One of the most famous examples of its use is the creation of the digital currency Bitcoin.

However, with the growing popularity of blockchain comes new security challenges. Blockchain, like any other digital information system (IS), is becoming a target of cyber attacks and misuse. While the decentralised nature of the blockchain provides a certain level of protection, it is not completely immune to various threats. Vulnerabilities can exist at the software level, as well as at the level of access control mechanisms, cryptographic protocols, and network infrastructure.

One of the most serious threats is the '51%' attack, where attackers gain control of the majority of computing power in the blockchain network, allowing them to alter transactions or even create duplicate digital assets. Other threats include phishing attacks, hacking smart contracts, and exploiting vulnerabilities in cryptographic algorithms. Insecure information security management systems (ISMS) can lead to critical data leakage, privacy breaches, and significant financial losses.

development, it is important to ensure reliable security mechanisms in the blockchain MIS. This includes constant monitoring of network activity, implementation of modern cryptographic solutions, creation of multi-level authentication systems, as well as regular software updates and improvements. Ensuring protection against blockchain-related malware and abuse is an important component of ensuring the reliable operation of systems based on this technology. ISMS should constantly adapt to new challenges in cyberspace to maintain a high level of data protection and minimise risks.

### 1. Analysis of mechanisms of protection against cyberattacks and abuse in information security management systems in the blockchain

One of the most well-known attacks is the 51% attack, which occurs when an attacker gains control of most of the computing resources of a blockchain network, allowing him to manipulate transactions and disrupt the chain's integrity. This allows them to control more than 50% of the mining capacity and mine new blocks faster than other participants. This advantage allows attackers to stop or change the order of transaction confirmation, as well as edit parts of the blockchain and cancel transactions that have already been carried out [2]. The 51% attack typically violates blockchain security protocols, and its consequences can range from minor to very serious, depending on the amount of hash power controlled by the attacker. The more computing resources an attacker controls, the greater the likelihood of a successful attack and the more serious the damage [3]. By controlling more than 51% of the capacity, the attacker can secretly create alternative blocks that will be considered valid because of the dominant capacity. This allows him to cancel transactions before they are confirmed, resulting in double spending of coins. In addition, legitimate miners earn less because attackers take their share of the profits from blockchain updates. Some miners, by increasing their computing power, may inadvertently cross the 50% limit of the total network capacity, but this does not pose a threat if they follow the rules and do not interfere with the normal operation of the system [4]. However, if a participant uses their advantage to act dishonestly, it can be considered an attack.

A Finney attack is a type of 'double-spending' where a transaction is confirmed by only one transaction confirmation [5]. In this case, the attacker creates a transaction to pay for the goods, while simultaneously preparing a block with a transaction that transfers these funds to another own account, but does not publish this block. As soon as the payment transaction is confirmed by one of the miners and the goods are received, the attacker quickly publishes the prepared block. As a result, two blockchain branches of the same length are formed in the network. If miners start supporting the branch that contains a transaction to the attacker's account, the transaction that was supposed to transfer funds to the seller will be cancelled, and the seller will lose money because the goods have already been shipped [3]. To protect against this attack, the seller can wait for several transaction confirmations, which reduces the risk but does not guarantee complete security. If an attacker controls several nodes in the network and the seller does not wait for enough confirmations, the attacker can create a longer chain with a transaction to his account. After publishing this chain, miners will continue to work with it, maintaining a block with a transaction in favour of the attacker. If both chains have the same length, miners must choose one of them, and in this case, the probability of success of the attack is 50%.

A Race Attack occurs when an attacker makes two transactions at the same time: transaction 'A' to pay for the purchase and transaction 'B' that transfers the same funds to another account. If the seller does not wait for the transaction to be confirmed and ships the goods immediately, he or she runs a risk: with a 50% probability, transaction B can be included in the blockchain without additional actions by the attacker [6]. Even worse, an attacker can increase the probability of success by selecting specific nodes to transmit a particular transaction. The principle of this attack is similar to the Finney attack and is also a form of 'double spending'.

Distributed denial-of-service (DDoS) attacks, although difficult to execute in a blockchain network, are still possible. In a DDoS attack, attackers seek to disable a server by overloading it with a large number of requests, which leads to the depletion of its computing resources. The main purpose of such attacks is to destabilise the functioning of mining pools, e-wallets, cryptocurrency exchanges and other financial services. In addition, the blockchain can be attacked at the application level using DDoS botnets that make massive requests to complicate the network's operation.

Timejacking exploits a potential vulnerability in the way the Bitcoin network handles timestamps. In this attack, an attacker changes the time settings of a node, forcing it to adopt an alternative blockchain. This is possible when an attacker adds several fake peers to the network with incorrect timestamps [7].

An eclipse attack assumes that the attacker controls a large number of IP addresses or uses a distributed botnet network. The attacker modifies entries in the victim's table of 'tested' nodes and waits for the victim's node to

restart [8]. After the restart, all outgoing connections of the victim node are directed to IP addresses controlled by the attacker. This results in the victim being unable to receive the transactions they need. At first glance, the eclipse attack may seem similar to the Sybil attack, as both involve the distribution of fake resources on the network. However, their ultimate goals are different [9]. In an eclipse attack, the attacker tries to completely isolate the victim by redirecting all of its connections to nodes that they control. The attacker creates a ring of controlled IP addresses to which the victim's node is likely to connect after the system is restarted. The restart may be forced (for example, due to a DDoS attack) or occur due to other factors that the attacker may simply wait for.

A Sybil attack is a security threat to online systems where one person attempts to take control of a network by creating multiple fake accounts, nodes, or computers. A simple example is when one person creates multiple accounts on a social network. In the context of cryptocurrencies, this could be a situation where someone runs multiple nodes on a blockchain network at the same time. The name 'Sybil' came from the case of a woman named Sybil Dorsett who suffered from dissociative identity disorder, also known as multiple personality disorder [10].

Cryptojacking is one of the types of cyber threats that has become increasingly widespread in recent years [11]. This type of cybercrime involves the use of computing resources of a user's computer, mobile phone, tablet, laptop, or server for unauthorised cryptocurrency mining without the user's consent or knowledge. The main goal of cryptojacking is to make money on cryptocurrency at the expense of other people's or organisations' resources. This threat is becoming particularly relevant due to the growing popularity of cryptocurrencies such as Monero, Bitcoin, Ethereum, and others, as well as the increasing number of devices connected to the Internet. For attackers, this is an opportunity to make significant profits by using large networks of infected devices for mining [12]. Cryptojacking is often implemented through hidden scripts embedded in websites or mobile applications.

All technologies, including blockchain, have potential attack vectors that cybercriminals can use to their advantage. In the cryptocurrency world, one of the most well-known is the Vector 76 attack. This is a form of double-spend attack that exploits a minor bug in the Bitcoin consensus system. As a result, an attacker can obtain funds and harm their victims [13]. This attack is performed when a fraudulent miner controls two complete networks of nodes. One of them (node A) is connected directly to the exchange service, and the other (node B) is connected to other key nodes in the blockchain network. For a successful attack, an attacker must monitor the transmission and propagation of transactions through different nodes to know which ones are the first to transmit transactions and thus connect to both the exchange service and the key nodes in the network correctly.

The double-spending vulnerability is a common blockchain attack method that exploits the transaction verification process. All transactions in the blockchain must be verified by users to be valid, which takes time [14]. Attackers can take advantage of this delay to trick the system into using the same coins or tokens in multiple transactions. Other types of attacks mentioned earlier have also arisen from this vulnerability [15]. Unlike traditional financial institutions, the blockchain confirms transactions only after a consensus is reached between all network nodes. Until a block with a transaction is verified, the transaction is considered unconfirmed. However, the verification process takes some time, which creates opportunities for CAs. Similar to counterfeiting, double spending leads to inflation by increasing the amount of duplicate currency that did not previously exist. This leads to a depreciation of the currency against other currencies or goods, reduces user confidence, and disrupts the normal circulation and storage of assets.

## 2. Rationale for technology selection

Justification of the choice of security methods in blockchain ISMS is a key step in increasing the resilience of such systems to acts of corruption and misuse. The choice of specific technologies, mechanisms, and approaches is based on risk assessment, network features, the level of available resources, and the predicted threats that a blockchain system may face. Each security method plays a specific role in the overall security architecture, so the right choice and combination of such methods is critical to maintaining the integrity and reliability of the network.

Consensus algorithms are the basis of the blockchain, which allows decentralised nodes to reach agreement on the state of the ledger without the need for a central controller [16]. The choice of a consensus algorithm determines not only the efficiency and speed of the blockchain, but also its resistance to attacks.

The PoW algorithm provides a high level of protection against 51% attacks, since the attack requires significant computing power, which is economically unprofitable for attackers [17]. However, PoW has disadvantages, such as high power consumption and slow transaction confirmation speed. It is suitable for those blockchain projects that have a large global reach and can afford high security costs (e.g. Bitcoin).

The PoS algorithm relies on the stake that users hold in the network. This approach makes attacks less likely, as an attack on the system requires significant investment, which will be lost in case of failure [17]. PoS is less energy-intensive than PoW and offers faster transaction confirmation, which makes it an attractive choice for new blockchain systems (e.g., Ethereum 2.0). However, it is important to keep in mind that PoS can be vulnerable to a

'trust attack', where large validators can join together and create cartels to control the network.

The DPoS algorithm uses delegation, where participants select validators to confirm transactions. DPoS is very efficient and fast, but is vulnerable to centralisation, as several delegates can gain control of the network [18]. It is suitable for blockchains where speed and scalability are a priority (e.g., EOS).

A reasonable choice between PoW, PoS, and DPoS depends on the needs of a particular system. If the system requires maximum decentralisation and security, PoW may be the best choice. For systems that are focused on speed and environmental efficiency, PoS or DPoS are more suitable.

Cryptography is the main mechanism for ensuring confidentiality, integrity, and authenticity in a blockchain. The choice of cryptographic algorithms should take into account long-term security, including threats from quantum computing.

Asymmetric cryptography - public and private key cryptography is used to protect transactions in the blockchain. The choice of strong algorithms, such as RSA or ECDSA (elliptic curve cipher), provides protection against attackers trying to forge transactions [19-20]. However, with the development of quantum computers, these algorithms may become vulnerable, so future blockchains should start implementing post-quantum cryptographic algorithms (e.g., lattice-based algorithm).

Hashing - cryptographic hash functions such as SHA-256 (in Bitcoin) or Keccak-256 (in Ethereum) are used to ensure the integrity of data in the blockchain [19-20]. Hashing ensures that any change in the block data will be instantly detected. The choice of a strong hash function is important to prevent attacks such as hash collisions or computational hash reversal attacks.

Zero Disclosure Cryptographic Algorithms - Innovative methods such as zk-SNARKs (zero-knowledge succinct non-interactive arguments of knowledge) [21] provide privacy in blockchain transactions by allowing transactions to be confirmed without disclosing details. This is important for private blockchains or blockchains used for financial transactions where confidentiality is critical.

Smart contracts are one of the most powerful tools in blockchain systems, but they can also be a source of vulnerabilities. Therefore, to ensure their security, a number of methods should be applied.

Firstly, it is the audit of smart contracts, i.e. the external audit of the smart contract code by independent experts allows to identify possible vulnerabilities before they are used in a real network. Companies such as CertiK and OpenZeppelin specialise in smart contract security testing.

Also, smart contracts should be thoroughly tested in a sandbox environment to identify possible logical errors or security flaws. Formal verification is important for critical applications, as it allows mathematical proof of the correctness of the contract [22].

Using a modular approach to writing smart contracts makes their code more manageable and secure. Dividing contract logic into separate components allows you to isolate potential problems and reduces the impact of code errors.

Network-level security is important for blockchain because the decentralised nature of the network makes it vulnerable to certain types of attacks. For this purpose, it is necessary to implement encryption at the P2P level - the use of secure communication protocols between nodes blocks the possibility of data interception or manipulation attempts [23]. For example, the TLS protocol ensures reliable data encryption during transmission. It is also necessary to protect against routing attacks - DHT protocols are used to organise data exchange between nodes. To ensure reliability, it is necessary to use advanced mechanisms to protect against routing attacks (e.g., Sybil attacks) to avoid misallocation of data or route falsification.

ISMS should be able to monitor activity in the blockchain network in real time [24]. This allows you to quickly detect and respond to any suspicious activities or threats, including:

– anomaly analysis: the introduction of artificial intelligence and machine learning technologies for analysing blockchain transactions allows detecting anomalies that may indicate potential attacks, for example, a sharp increase in node activity may be a sign of preparation for a DoS attack [25];

– rapid response: the system should have Incident Response Plans that will allow you to quickly isolate suspicious nodes, block malicious transactions or change consensus rules in case of a threat [25].

Social engineering is one of the most common methods of CA, so it is important to consider the human factor in security issues. Educating users on the proper use of private keys, recognising phishing attacks and other fraudulent methods is critical.

A reasonable choice of protection methods in blockchain information security management systems should be based on risk analysis, technological capabilities, and a long-term security strategy. The combination of cryptographic methods, reliable consensus algorithms, secure network protocols and continuous monitoring will minimise the risks of CI and abuse.

According to the rationale for choosing security methods, an effective approach to ensuring security in a blockchain MIS is to use a monitoring and response method. One of the most effective solutions in this direction is the implementation of a protection system based on iptables, ipset, fail2ban and dynamic blacklist IP addresses.

The combination of these tools allows you to create a multi-level protection against man-in-the-middle at the network level. The use of iptables and ipset allows

for flexible configuration of network traffic filtering and blocking of suspicious IP addresses, while fail2ban automatically detects and blocks malicious access attempts, protecting the system from brute force attacks and other hacking attempts. Dynamic blacklisted IP addresses help to keep your system up-to-date and responsive to new threats, ensuring that malicious nodes are blocked in real time.

This approach ensures constant monitoring of activity in the blockchain network and allows for a prompt response to attempts at covert action, which significantly increases the overall security of the blockchain system and reduces the risk of compromise.

This approach ensures constant monitoring of activity in the blockchain network and allows for a prompt response to attempts at CRA, which significantly increases the overall security of the blockchain system and reduces the risk of compromise. The rapid development of web services has increased their vulnerability to various forms of CRA, including DDoS, brute force attacks, SQL injections and unauthorised access attempts, and the growing prevalence of CRA on web services necessitates the development of reliable protection mechanisms to mitigate threats and preserve digital assets. Due to the growing dependence on these services for business operations, financial transactions and personal data storage, the need for reliable protection mechanisms has become more critical than ever. Traditional security measures, such as firewalls and anti-virus software, are no longer sufficient to counter the increasing sophistication of modern spacecraft. In the context of developing a mechanism for protecting web services and blockchain MIS, the key requirement is to ensure high efficiency of detecting and blocking attacks in real time with minimal impact on system performance. For this purpose, it is proposed to use a comprehensive mechanism with dynamic protection, which will consist of the following tools: iptables, ipset, fail2ban, and dynamic blacklist IP addresses. Each of these components has its own unique advantages, allowing for a multi-level security system that adapts to new threats. This approach is aimed at reducing the risk posed by intruders, automating the detection and prevention of acts of cybercrime, and adapting defences in real time to changing attack vectors.

The development will use a complex mechanism with dynamic protection. Let's take a look at its main utilities: iptables, ipset, fail2ban, and dynamic blacklist IP addresses.

Iptables is a utility for configuring and managing packet filtering rules in the network stack of Linux distributions. It allows you to effectively control incoming, outgoing, and redirected traffic by setting rules based on IP addresses, ports, protocols, and other parameters [24]. The main reasons for choosing iptables are the ability to flexibly configure rules to block malicious traffic, high performance even with a large number of rules, embeddedness in the Linux kernel, which ensures reliability and speed, and support for integration with other security tools such as ipset and fail2ban.

Ipset is an extension to iptables that allows you to work with large sets of IP addresses or other parameters within a single rule. Instead of creating a separate rule for each IP address, you can store them in sets, which significantly speeds up the system [25]. The main reasons for choosing ipset are support for large lists of IP addresses without significantly affecting performance, the ability to quickly update and modify lists while the system is running, and increased packet filtering efficiency when using dynamic blacklists.

Fail2ban is a tool for automatically blocking IP addresses that show suspicious activity or commit acts of terrorism based on the analysis of service logs (e.g., sshd, apache, nginx). fail2ban allows you to create rules for blocking attacks at the firewall level after a certain number of failed access attempts [24]. The main reasons for choosing fail2ban are automatic detection of suspicious activity and fast response, the ability to configure it using regular expressions to analyse logs of various services, integration with iptables to create blocking rules in real time, flexibility in setting the blocking period, which allows you to dynamically control access.

The use of dynamic IP address blacklists is necessary to protect against known attacking hosts [25]. These lists can be updated both locally (based on fail2ban rules) and through integration with external sources (reputation lists, known attacks, threats from botnets, etc.). The main reasons for choosing dynamic lists are the ability to automatically update and synchronise with global sources of threat information, dynamically add new IP addresses to the blacklist without the need to restart the firewall, and ensure high request processing speeds through the use of ipset.

### 3. Solution implementation process

At the first stage, basic rules are created using iptables and basic traffic filtering rules are configured, including restricting access to certain ports and services, if necessary, setting rules for blocking known malicious IP addresses, and configuring logging of suspicious traffic for further analysis.

The second step is to configure the ipsets for the dynamic lists. The next step is to create sets of IP addresses using ipset that will be used in iptables rules. This will allow you to quickly update the lists of blocked addresses without rebooting the entire system. Next, sets are created to store the IP addresses of attacking hosts. The sets can be updated with data from external sources or based on local observations.

Integration of fail2ban to automatically block attacks is configured by monitoring web service logs and other critical system components. If suspicious activity is detected (for example, multiple failed authentication attempts),

the system automatically adds the offending IP address to the blacklist via ipset and updates iptables rules.

Dynamic lists of blacklisted IP addresses are integrated with external services or databases to provide up-to-date information about IP addresses that are the source of threats. These lists are regularly updated to ensure protection against new threats.

The advantages of the proposed solution:

– speed and efficiency: using ipset to manage large lists of IP addresses ensures that rules are updated quickly without affecting system performance;

– adaptability: thanks to dynamic blacklists and automatic blocking via fail2ban, the system can instantly respond to new threats and adapt to them;

– flexibility and scalability: the solution can be easily scaled up for use on large web services or blockchain systems without significant changes to the infrastructure.

Thus, the proposed solution based on iptables, ipset, fail2ban and dynamic IP address lists allows you to create an effective multi-level anti-cannonballing system that will respond to threats in real time while maintaining high performance and flexibility.

The effectiveness of the security system can be assessed using a special formula [26]:

$$E = \frac{(M - N)}{(M + V + N)} \cdot 100\% \cdot (1 - B), \qquad (1)$$

where $E$ – system efficiency; $M$ – number of cyberattacks detected and blocked by the system; $N$ – number of cyberattacks prevented by the system; $V$ – number of cyberattacks that have passed the cyber defence system; $B$ – is the percentage of false positives generated by the system. The expression makes sense only if: $M > N \; i \; B \in [0;1)$.

## 4. Software implementation of the security system

The software implementation of the system of protection against man-in-the-middle and abuse in blockchain systems involves the integration of several key tools, such as iptables, ipset, fail2ban, and dynamic blacklist IP addresses. These solutions provide automated protection at the network level and allow blocking suspicious activities and malicious nodes.

The main goal of this implementation is to provide protection against man-in-the-middle at the network level by automatically detecting and blocking malicious actions such as brute force attempts, DoS, DDoS and other abuses.

Program Description. Iptables is the main tool for configuring a firewall on Linux, allowing you to create rules to control traffic on the network [24]. For a blockchain system, it is important to set up filtering to restrict access to certain ports used in network protocols and prevent attacks.

Basic setup steps:

– identify critical ports used by the blockchain infrastructure (e.g. for transactions and block confirmation);

– create rules to allow traffic to these ports only for certain IP addresses or geographical areas;

– use the DROP policy for all unknown or suspicious traffic: iptables -P INPUT DROP or, for example, allowing the Bitcoin port, iptables -A INPUT -p tcp --dport 8333 -j ACCEPT;

– limiting the number of connections from one IP to prevent DoS or DDoS: iptables -A INPUT -p tcp --dport 8333 -m connlimit --connlimit-above 10 -j DROP.

– ipset is a tool that allows you to efficiently manage large lists of IP addresses, making it easier to implement dynamic blacklists [25]. Using ipset in conjunction with iptables allows you to quickly update rules without the need for a full table reload.

The description of software scripts and configuration files is presented in Tables 1 and 2.

**Table 1**

**Description of software scripts**

| Name | Description |
|---|---|
| blip.sh | BlackIP – is a project that collects and unifies public blacklists of IP addresses and subnets to make them compatible with ipset. Usage.: ./blip.sh або bash blip.sh. |
| check.sh | Checks IP addresses or subnets in the ipset lists (blnet, blip, addonnet, addonnet, addonip, geonet, geoip, f2bip) and the fail2ban ban list (/var/log/fail2ban.log). Usage.: .\check.sh IP_or_SubNET. |
| ipset-check.sh | Checking the ipset list, if there is nothing in the ipset, then ipset lists are created and the script is executed ipset.sh. |
| ipset.sh | Downloading blacklists of IP addresses and subnets from the Github repository, adding them to ipset lists (blnet, blip, addonnet, addonnet, addonip, geonet, geoip), adding IP addresses to the ipset list (f2bip), adding IP addresses and subnets to the ipset whitelist (wlip та wlnet). |
| fail2ban-status.sh | Script to check the fail2ban client status in a file /etc/fail2ban/jail.d/*.conf (nginx-limit-req, nginx-conn-limit, nginx-dos, nginx-badbots, nginx-4xx, sshd). |
| unban.sh | A script to unban (place the banned IPs and subnets in the unban.txt file) the banned IPs and subnets from списків ipset (blnet, blip, addonnet, addonnet, addonip, geonet, geoip, f2bip), з бану fail2ban. |
| wlset.sh | Script to add (place IP addresses and/or subnets in the wlset.txt file) IP addresses and subnets to the whitelist ipset (wlip та wlnet). |

**Table 2**

**Опис конфігураційних файлів**

| Name | Description |
|---|---|
| /etc/ fail2ban/ action.d/ | The directory contains action configuration files that define what fail2ban will do when it detects suspicious activity according to the filter rules. Actions can include: blocking IP addresses with a firewall, sending alerts, executing other commands or scripts. Each file is responsible for a specific action. By default, there are rules for iptables, nftables, tcpwrappers, shorewall. |
| /etc/ fail2ban/ filter.d/ | This directory contains the configuration files for the filters that define which log messages should be considered suspicious or malicious. The filter files contain regular expressions that fail2ban uses to search for specific log entries. Each configuration file can correspond to a specific service (e.g. sshd, apache, etc.). |
| /etc/ fail2ban/ jail.d/ | This directory contains additional or individual configuration files for 'jails'. They supplement or overwrite the basic settings from the jail. conf file. jails is a set of rules that define which filters to use and what actions to take when suspicious activity is detected. This allows you to flexibly configure protection for different services. |
| /etc/ fail2ban/ jail.conf | The main configuration file for setting up jails. It defines the general parameters for each service (or jail), such as the filter to be applied, the time to block IP addresses, the number of failed login attempts before the ban is activated, etc. However, this file is usually not edited directly, as there is a risk that changes may be overwritten when fail2ban is updated. Instead, it is recommended to create or edit separate files in the /etc/fail2ban/jail.d/. |

## 5. Experimental testing of components

The experiment included testing of components such as iptables, ipset, fail2ban and dynamic blacklist IP addresses to assess the system's ability to detect and block malicious traffic and provide resistance to various types of CA in the IS.

For the experiment, a virtual environment with a deployed blockchain network was used to simulate various types of attacks, including brute force, DoS, DDoS, and attempts to gain unauthorised access to blockchain nodes. Test environment:

– operating system: Ubuntu 24.04 LTS;

– security tools: iptables, ipsets, fail2ban;

– type of attacks: brute force on SSH, DoS, DDoS on the blockchain port, simulation of malicious traffic through port scanning, attacks on the web server;

– testing was carried out for 7 days with different loads.

One of the system's key tasks was to detect and block brute force attacks aimed at gaining access to critical blockchain nodes via SSH, web server, and any other open port on the network. In this scenario, fail2ban monitored system logs for failed authorisation attempts and automatically blocked the attackers' IP addresses. Results.:

– the average time to block an IP address after detecting suspicious activity was 1-2 seconds;

– fail2ban the average time to block an IP address after detecting suspicious activity was 1-2 seconds detected and blocked 95% of all attempts brute force;

– the remaining 5% are IP addresses that have carried out low-intensity attacks, but have been added to the blacklist based on the dynamic updating of blacklisted IP addresses.

These results demonstrate the high efficiency of fail-2ban in detecting and blocking brute force attacks, as well as the need to use dynamic blacklisted IP addresses to handle low-intensity attacks (this percentage may increase in a real-world environment).

The experiment also tested the system's ability to protect against DoS and DDoS attacks on blockchain network ports. The attacks used a large volume of malicious requests aimed at overloading the server. Results.:

– using iptables rules and configuring the Nginx web server to limit the number of connections from one IP reduced the intensity of DDoS attacks by 85% by blocking suspicious IPs after exceeding a threshold;

– the system detected and blocked almost all malicious requests through IP address blacklists and ipset;

distributed botnets, where the attack came from many IP addresses, the average blocking time for each malicious IP was 3-5 seconds.

Protection against DDoS attacks proved to be effective, but response times were slightly longer for significant distributed attacks due to the dynamic nature of updating IP blacklists.

Dynamic lists of blacklisted IP addresses, which were updated through external sources, played a key role in protecting the system from new and little-known attacks. These lists were constantly updated with the latest data on malicious IP addresses, which helped maintain a high level of security. Results.:

– automatic updating of the lists took place twice a day, which ensured timely blocking of new threats;

– automatic updates of the lists occurred twice a day, which ensured timely blocking of new threats more than 375 IP addresses from dynamic lists that were not detected during traffic monitoring by iptables or fail2ban.

The load on the system from the use of security tools was minimal. The use of iptables, ipset, and fail2ban did not cause significant delays in the operation of the blockchain network. Results.:

– the CPU load during the blocking of DDoS attacks increased by about 5%, but this did not affect the overall system performance;

– the response time of blockchain nodes to requests remained stable even during attacks, which indicates the high performance of solutions.

The operands for calculating the effectiveness of a security system are the number of attacks that were blocked after the system warned of them, the total number of attacks that were blocked, missed, or warned, and a multiplier that reduces the overall effectiveness based on the percentage of false positives. If the system frequently blocks legitimate traffic, its effectiveness decreases. The effectiveness of the system depends on the number of attacks that are successfully blocked or prevented by the system [26].

Table 3 shows the results of a week's worth of monitoring of the targeted web server. In the study, the system blocked or prevented most attacks, but false positives reduced the overall effectiveness of the system. To improve efficiency, the number of false positives can be reduced and/or the accuracy of malicious activity detection can be increased. According to the results of the weekly monitoring, 663 attacks were detected, and the final efficiency of the protection system was approximately 88%. This means that the system blocked and prevented most attacks, but some attacks (25 cases) passed through the system, there was a small number of false positives (20 cases), and 18 cases of malicious acts were detected that were prevented by the system, i.e. detected by cyber defence tools but not blocked, and the system sent notifications about such threats so that measures could be taken to further neutralise them.

**Table 3**

**Description of the configuration files**

| M | N | V | B | Efficiency (%) |
|---|---|---|---|---|
| 50 | 1 | 3 | 0.12 | 79.85 |
| 100 | 3 | 6 | 0.09 | 80.98 |
| 150 | 4 | 10 | 0.0667 | 83.09 |
| 200 | 7 | 13 | 0.05 | 83.34 |
| 250 | 10 | 15 | 0.04 | 83.78 |
| 300 | 11 | 16 | 0.04 | 84.84 |
| 350 | 12 | 18 | 0.04 | 85.39 |
| 400 | 14 | 20 | 0.0375 | 85.6 |
| 450 | 16 | 21 | 0.0356 | 85.94 |
| 500 | 17 | 22 | 0.032 | 86.74 |
| 550 | 17 | 22 | 0.031 | 87.69 |
| 600 | 18 | 25 | 0.031 | 87.71 |

The results demonstrate that the effectiveness of the system can vary significantly depending on the number of blocked attacks, the number of spacecraft that passed through the system, and the percentage of false positives. To achieve maximum efficiency, it is important to minimise the number of missed attacks and false positives.

## Conclusions

The software implementation of the blockchain security system using iptables, ipset, fail2ban, and dynamic blacklist IP addresses provides reliable multi-level protection at the network level. It allows you to quickly detect and block suspicious activity, protecting the system from brute force attacks, DoS, DDoS, and other threats, providing continuous monitoring and automatic response to threats in real time. This implementation of a dynamic multi-level security mechanism offers an effective and scalable solution for protecting web services from man-in-the-middle attacks. The system's adaptability and the ability to integrate real-time threat intelligence increase its effectiveness in countering evolving cyber threats. The system's ability to automatically detect and block malicious IP addresses combined with low resource consumption makes it a scalable and practical solution for both small and large web applications. Future improvements could focus on further automation and incorporation of machine learning and artificial intelligence techniques for adaptive threat detection.

The study demonstrates the high efficiency and feasibility of implementing the proposed protection mechanisms in the ISMS in the blockchain and web services. The developed approaches not only provide a high level of protection against cyberattacks and abuse, but also have the potential for further development and integration with other protection systems in the information infrastructure and IS of modern enterprises.

**References**

[1] Arun J. S., Gaur N., Cuomo J. Blockchain for Business. Addison-Wesley Professional: 1st edition. 2019. 224 p.

[2] The 51% attack on blockchains: A mining behavior study / Aponte-Novoa F. A., Orozco A. L. S., Villanueva-Polanco R., Wightman P. IEEE. 2021. Vol. 9. pp. 549 – 564. DOI:10.1109/ACCESS.2021.3119291.

[3] Anita N., Vijayalakshmi M. Blockchain security attack: A briefsurvey in Proc. 10th Int. Conf. Comput., Commun. Netw. Technol. (ICC-CNT). 2019. pp. 6 – 11. DOI:10.1109/ICCCNT45670.2019.8944615.

[4] A survey on the security of blockchain systems / Li X., Jiang P., Chen T., Luo X., Wen Q. Future Generation Computer Systems. 2020. Vol. 107. pp. 841 – 853. DOI:10.1016/j.future.2017.08.020.

[5] What is a finney attack? URL: https://bitcoin.stackexchange.com/questions/4942/what-is-a-finney-attack (дата звернення: 04.03.2025).

[6] Aggarwal S., Kumar N. Chapter Twenty – Attacks on blockchain. Advances in Computers. 2021. Vol. 121. pp. 399 – 410. DOI: https://doi.org/10.1016/bs.adcom.2020.08.020.

[7] Vyas C. A., Lunagaria M. Security concerns and issues for bitcoin. National Conference cum Workshop on Bioinformatics and Computational Biology, NCWBCB. 2014. pp. 10 – 12. URL: https://research.ijcaonline.org/ncwbcb/number2/ncwbcb1414.pdf (дата звернення: 04.03.2025).

[8] DeCusatis C., Zimmermann M., Sager A. Identity-based network security for commercial Blockchain services. IEEE 8th Annual Workshop and Conference on Computing and Communication. 2018. pp. 474 – 477. DOI:10.1109/CCWC.2018.8301713.

[9] Sharma P. K., Moon S. Y., Park J. H. Block-VN: a distributed Blockchain based vehicular network architecture in smart city. Journal of Information Processing Systems. 2017. Vol. 13. No. 1. pp. 184 – 195. DOI: 10.3745/JIPS.03.0065.

[10] Swathi P., Modi C., Patel D. Preventing Sybil Attack in Blockchain using Distributed Behavior Monitoring of Miners. 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT). 2019. pp. 1 – 6. DOI:10.1109/ICCCNT45670.2019.8944507.

[11] Mining on someone else's dime: Mitigating covert mining operations in clouds and enterprises / Tahir R., Huzaifa M., Das A., Ahmad M., Gunter C. A., Zaffar F., Caesar M., Borisov N. 20th International Symposium on Research in Attacks, Intrusions and Defenses (RAID), Atlanta. 2017. pp. 287 – 310. DOI: https://doi.org/10.1007/978-3-319-66332-6_13.

[12] Saad M., Khormali A., Mohaisen A. End-to-end analysis of in-browser cryptojacking. CoRR. 2018. Vol. abs/1809.02152. 15 p. DOI: https://doi.org/10.48550/arXiv.1809.02152.

[13] Blockchain Attack Vectors: Main Vulnerabilities of Blockchain Technology. URL: https://www.apriorit.com/dev-blog/578-blockchain-attack-vectors (дата звернення: 04.03.2025).

[14] Misbehavior in bitcoin: a study of double-spending and accountability / Karame G. O., Androulaki E., Roeschlin M., Gervais A., apkun S. ACM Transactions on Information and System Security. 2015. Vol. 18. No. 1. pp. 1 – 32. DOI:10.1145/2732196.

[15] Nicolas K., Yi W. A novel double spending attack countermeasure in blockchain. IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON). 2019. pp. 383 – 388. DOI:10.1109/UEMCON47517.2019.8992991.

[16] Блокчейн інфраструктура для захисту кіберсистем / Адамов О.С., Хаханов В.І., Чумаченко С.В., Абдуллаєв В.Г. Радіоелектроніка та інформатика. 2018. №4 (83). С. 64 – 85. URL: https://journals.indexcopernicus.com/search/article?articleId=2146726 (дата звернення: 04.03.2025).

[17] Proof of Stake versus Proof of Work White Paper. URL: https://bitfury.com/content/downloads/pos-vs-pow-1.0.2.pdf (дата звернення: 04.03.2025).

[18] Everything you Wanted to Know about the Blockchain / Puthal D., Malik N., Mohanty S. P., Kougianos E., Das G. IEEE Consumer Electronics Magazine. 2018. Vol. 7. No. 4. pp. 06 – 14. DOI:10.1109/MCE.2018.2816299.

[19] Analysis of cryptographic authentication and manipulation detection methods for big data / Havrylova A. A., Korol O. G., Voropay N. I., Sevriukova Y. O., Bondarenko K. O. Сучасний захист інформації. 2024. Vol. 1(57). pp. 97 – 102. DOI: https://doi.org/10.31673/2409-7292.2024.010011.

[20] Горбенко І. Д. Гриненко Т. О. Захист інформації в інформаційно-телекомунікаційних системах: Навч. посібник. Ч.1. Криптографічний захист інформації. Харків : ХНУРЕ, 2004. 368 с.

[21] Internet of Things, Blockchain and Shared Economy Applications / Huckle S., Bhattacharya R., White M., Beloff N. Procedia Computer Science. 2016. Vol. 98. pp. 461–466. DOI:10.1016/j.procs.2016.09.074.

[22] Blockchain Technology in Financial and Banking Sector / Ragha L., Dixit A., Rodrigues B., Yadav K. International Journal of Trend in Research and Development. 2018. Vol. 1. pp. 41 – 44. URL: http://www.ijtrd.com/papers/IJTRD15855.pdf (дата звернення: 04.03.2025).

[23] Курочкіна М. Г. Блокчейни – новітня технологія криптографії в цифровому світі. Світ телекомунікації та інформатизації: матеріали Міжнародної науково-технічної конференції студентства Державного університетутелекомунікацій. Київ:ДУТ, 2017. С. 209–212. URL: http://www.dut.edu.ua/uploads/n_5218_58757739.pdf (дата звернення: 04.03.2025).

[24] Bhuyan M. H., Bhattacharyya D. K., Kalita J. K. Network anomaly detection: Methods, systems and tools. Proceedings of IEEE Communications Surveys Tutorials. 2014. Vol. 16(1). pp. 303 – 336. DOI: https://doi.org/10.1109/SURV.2013.052213.00046.

[25] Unsupervised anomaly detection via variational autoencoder for seasonal kpis in web applications / Xu H., Chen W., Zhao N., Li Z., Bu J., Li Z., Liu Y., Zhao Y., Pei D., Feng Y., Chen J., Wang Z., Qiao H. Proceedings of Proceedings of the 2018 World Wide Web Conference. 2018 pp. 187 – 196. DOI: https://doi.org/10.1145/3178876.3185996.

[26] Савіцький Л. М., Безносенко С. Ю., Горбач Р. Я. Концептуальні погляди на побудову системи захисту від кібератак із застосуванням методів штучного інтелекту в інформаційно-комунікаційних системах. Сучасні інформаційні технології у сфері безпеки та оборони. 2024. № 1(49). С. 77 – 85. DOI: 10.33099/2311-7249/2024-49-1-77-85.

***Date of submission of the article to the editorial board: 10.05.2025***