Solar Physics
DOI: 10.1007/•••••-•••-•••-••••-•

Intrusion Detection System
using BlockChain

Aayush Shah · Akshat Shah · Tulsi Patel

Abstract Intrusion Detection System works in the backend which works continuously in the background and detects if there is any suspicious activity going on in the system or not on detecting it, it sends an Alert signal to the System Administrator to carry out an operation to stop the malicious activity from harming the device.
A system called an intrusion detection system (IDS) watches network traffic for suspicious activity and sends out alerts when it is found. It is software that checks a system or network for malicious activities or policy violations. Any illegal activity or violation is often recorded either centrally using a security information and event management (SIEM) system or notified to an administrator. A SIEM system combines outputs from several sources and employs alarm filtering methods to distinguish between legitimate and erroneous alarms. While monitoring networks for potentially harmful behavior, intrusion detection systems are also prone to raising false alarms. Consequently, enterprises must adjust their IDS products after initial installation. It entails correctly configuring intrusion detection systems to distinguish between legitimate network traffic and malicious activities.
Network packets entering the system are also monitored by intrusion prevention systems to look for any malicious activity and immediately send out alerts.

Keywords: Intrusion , Detection; IDS, SNORT; Smart Contract

1. Introduction

1.1. Who is an intruder?

Intruder – A person who is trying to get unauthorized access to the system with criminal intention. Types of intruders:-

- Outside Intruder [Masquerader] – A person not serving an organization trying to penetrate through illegal access.
- Inside Intruder [Misteasor] – A person predetermining serving an organization with limited access and trying to gain the access which the person is not permitted for illegal usage.

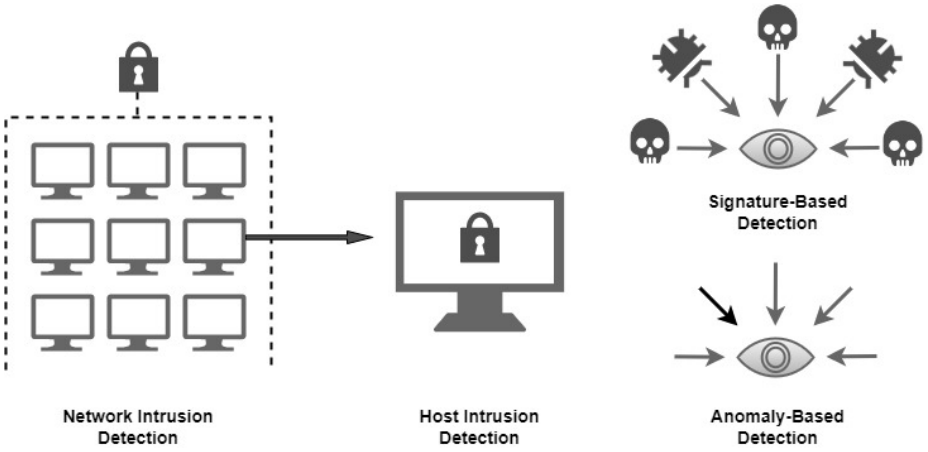

Figure 1. Intrusion Detection System

2. Classification

Classification of Intrusion Detection System: IDS are classified into 5 types:

2.1. Network Intrusion Detection System (NIDS):

Network intrusion detection systems (NIDS) are installed at a predetermined location within the network to monitor all network traffic coming from all connected devices. It carries out an observation of all subnet traffic passing through and compares that traffic to a database of known attacks. The alert can be delivered to the administrator as soon as an attack is detected or unusual behavior is noticed. Installing a NIDS on the subnet where firewalls are to check for attempts to breach the firewall is an example of a NIDS in action.

2.2. Host Intrusion Detection System (HIDS):

Host intrusion detection systems (HIDS) are network applications that run on separate hosts or gadgets. Only the incoming and outgoing packets from the device are monitored by a HIDS, which notifies the administrator of any unusual or malicious behavior. It compares the current snapshot of the system files with the previous snapshot. An alert is given to the administrator to look into if the analytical system files were altered or deleted. Mission-critical equipment, which is not anticipated to modify its layout, is an example of HIDS utilization.

2.3. Protocol-based Intrusion Detection System (PIDS):

A system or agent that consistently resides at the front end of a server, regulating and interpreting the protocol between a user/device and the server, makes up a protocol-based intrusion detection[system (PIDS). By continuously monitoring the HTTPS protocol stream and accepting the associated HTTP protocol, it tries

to secure the web server. Since HTTPS isn't encrypted and doesn't immediately enter the web presentation layer, the system would need to be located within this interface in order to use HTTPS.

2.4. Application Protocol-based Intrusion Detection System (APIDS):

A system or agent called Application Protocol-based Intrusion Detection System (APIDS) typically resides within a server cluster. By observing and analyzing communication on application-specific protocols, it detects intrusions. For instance, this would keep track of the SQL protocol that the middleware explicitly uses when communicating with the web server's database.

2.5. Hybrid Intrusion Detection System :

A hybrid intrusion detection system is created by combining two or more intrusion detection system methodologies. Host agent or system data is merged with network data in the hybrid intrusion detection system to create a comprehensive picture of the network system. In comparison to other intrusion detection systems, hybrid intrusion detection systems are more effective. Hybrid IDS is demonstrated by Prelude.

3. Detection Approaches of IDS:

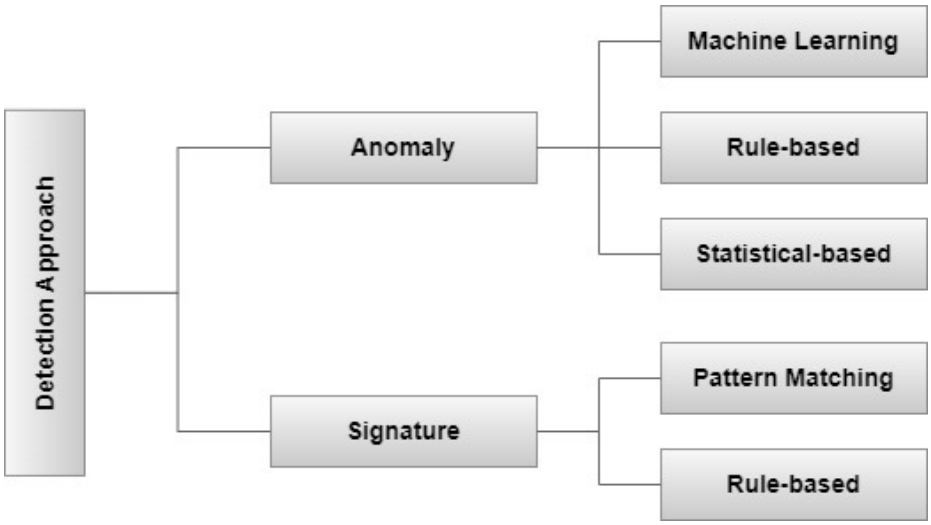

Figure 2. Detection Approaches of IDS

3.1. Signature-based Method

The amount of bytes, number of ones, or number of zeros in the network traffic are only a few examples of the specific patterns that signature-based IDS uses to identify attacks. Additionally, it identifies based on the malware's already-known harmful instruction sequence. Signatures are the patterns that the IDS has identified. A signature-based IDS can quickly identify attacks whose pattern (signature) is already present in the system, but it can be challenging to identify newly discovered malware attacks whose pattern (signature) is unknown.

3.1.1. Pattern Matching Models

There hasn't been much research on the pattern matching model in relation to blockchain technology, despite it being the most frequently utilized model by the signature detection approach. To find malware, pattern or string matching models employ one or more pattern matching algorithms. In order to detect malware, the single pattern approach analyses only one pattern at a time, whereas the multiple-patterns approach examines numerous patterns simultaneously.

3.1.2. Rule-based Models

A set of rules in rule-based models is compared to network traffic or audit data. If the rules are correct, they can identify any attack. However, a rule-based model must be combined with another method because utilizing it alone to detect malware is insufficient.

3.2. Anomaly-based Method

As new malware is generated quickly, anomaly-based IDS was launched to identify unknown malware threats. In anomaly-based IDS, machine learning is used to build a reliable activity model that is compared to anything arriving and is labelled suspicious if it is not found in the model. In comparison to signature-based IDS, machine learning-based methods have a better generalised property because these models can be trained using different applications and hardware configurations.

3.2.1. Machine Learning Models

By enhancing the system's feature selection, machine learning can help IDSs detect new and ongoing attacks automatically without any human interaction. In recent years, a variety of machine learning methods, including support vector machines, artificial neural networks, and genetic evolutionary algorithms, have been included into intrusion detection systems (IDSs) to improve system security.

3.2.2. Rule-based Models

In order to establish whether an event is normal or abnormal, rule-based models keep track of the actions taken by a system whose rules are kept in a database. The model has one flaw: if the database has few rules, it fails to identify anomalous events.

3.2.3. Statistical Models

An IDS with a statistical model relies on data analysis and correlation, followed by the application of statistical theories to such data to identify attacks. Each statistical variable's threshold is also specified by the user. However, because confidentiality, integrity, and availability have not been taken into account within the present statistical models' guiding principles, current statistical models are insufficient in terms of genetic architecture. There isn't any literature on using a statistical model based on blockchain technology as of the time of this review.

Enabling Traditional Security Mechanisms in Cloud:

1. Encryption
2. Hashing
3. Digital Signature
4. Public Key Infrastructure (PKI)
5. Identity and Access Management (IAM)
6. Single Sign-On (SSO)

What is Intrusion Prevention System? An intrusion prevention system (IPS) is a network security tool that continuously scans a network for harmful activity and responds to it when it does occur by reporting, blocking, or dropping it. It can be either hardware or software.

It is more sophisticated than an intrusion detection system (IDS), which can only alert an administrator and simply detect harmful activities. A next-generation firewall (NGFW) or unified threat management (UTM) solution may include intrusion prevention systems. They must be strong enough to scan a large volume of traffic without impairing network performance, like many network security technologies.

Any business security system that continuously scans network traffic for suspicious activity and takes action to stop it is known as an intrusion prevention system (IPS), sometimes known as an intrusion detection and prevention system (IDPS). IPS solutions, which are largely automated, assist in filtering out this dangerous activity before it gets to other security devices or controls, effectively decreasing the manual work required of security teams and enhancing the performance of other security products.

Additionally, IPS solutions are particularly good at spotting and preventing vulnerability exploitation. Before a security patch can be installed, threat actors frequently have a window of opportunity to exploit a vulnerability after it has been found. Here, a method for preventing intrusions is employed to swiftly stop these kinds of attacks.

In the middle of the 2000s, IPS appliances were initially created and made available as standalone products. However, unified threat management (UTM) systems for small and medium-sized businesses as well as next-generation firewalls at the corporate level now provide this capabilities. Now that next-generation IPS solutions are integrated with cloud computing and network services, they may offer a comprehensive defense against the rising number of cybersecurity threats that affect both local and international enterprises globally.

Why IDS/IPS?

- Maintaining data transmission confidentiality and user authentication with passwords or digital certificates are insufficient for securing distributed systems.
- Each node needs to have a monitor that will inform other nodes in the area if an assault happens.
- Because cloud-specific assaults don't always leave "traces" in a node's operating system, an attack on a cloud computing system may go undetected.
- This makes it difficult for conventional security systems to detect suspicious activity in a cloud environment.
- The 48
- Data and network encryption are found to be the top two cloud security technologies, followed by the use of IDS and IPS.
- The levels of security in the cloud will be increased by combining conventional security measures with IDS and IPS.

4. What is SNORT?

The C programming language is used to create the network-based intrusion detection system known as SNORT. In 1998, Martin Roesch created it. Cisco is now the company creating it. The software is open-source and free. To keep track of the system in real time, it can also be utilised as a packet sniffer. It can be used by the network administrator to monitor all incoming packets and identify any that pose a threat to the system. It is based on a tool called library packet capture. The rules are fairly simple to develop and put into practice, and they can be used in any operating system and network environment. The primary reason this IDS is more well-liked than others is that it is free to use and open source, allowing any user to utilise it whatever he pleases.

A potent open-source intrusion detection and prevention system (IDS and IPS), SNORT offers real-time network traffic analysis and data packet tracking. To find potentially malicious activities, SNORT employs a rule-based language that integrates anomaly, protocol, and signature inspection techniques.

Network administrators can detect Common Gateway Interface (CGI) attacks, buffer overflows, stealth port scans, and denial-of-service (DoS) and distributed DoS (DDoS) attacks using SNORT. A set of rules developed by SNORT define malicious network activity, spot malicious packets, and notify users.

SNORT is a piece of open-source software that is available for both personal and business use. Which network traffic should be gathered and what should happen when malicious packets are detected are determined by the SNORT rule language. This snorting function can be used to find malicious packets in the same way that sniffers and network intrusion detection systems do, or as a full network IPS solution that keeps an eye on network traffic and finds and prevents potential attack pathways.

5. Challenges in Distributed intrusion detection

Collaborating between several IDS and sharing logs with a centralised system is fraught with difficulties. Genuineness and trust are important problems that must be tackled. The following criteria should be met by a distributed intrusion detection system:

- **Integrity** - It's crucial that the alerts that an IDS generates be accurate. Logs and alerts must be impenetrable to tampering, and under no circumstances should they be viewable or modifiable by an attacker or any other person.
- **Scalability** - The computational demand for the centralised server should be able to adjust as the network size may grow and shrink.
- **Consensus** - All participating IDS should share a common understanding of the type and quality of alert generation.
- **Privacy** - Participating individual IDS should have control over alert data accessibility and selective disclosure.

6. How blockchain can be used for IDS

In recent days, it has been seen that attackers are using more sophisticated and intricate tactics to assault the system undetected. It's also feasible that the network administrator's focus might be diverted if any part of the network is improperly configured or hacked by the attackers. The legitimacy of the logs and alarms received from the various IDS raises serious concerns because they are all connected to the same central server. Blockchain appears to be the most dependable option to make the system as a whole reliable and trustworthy. We will concentrate on the difficult difficulties with DIDS in this part and how blockchain can assist to overcome them.

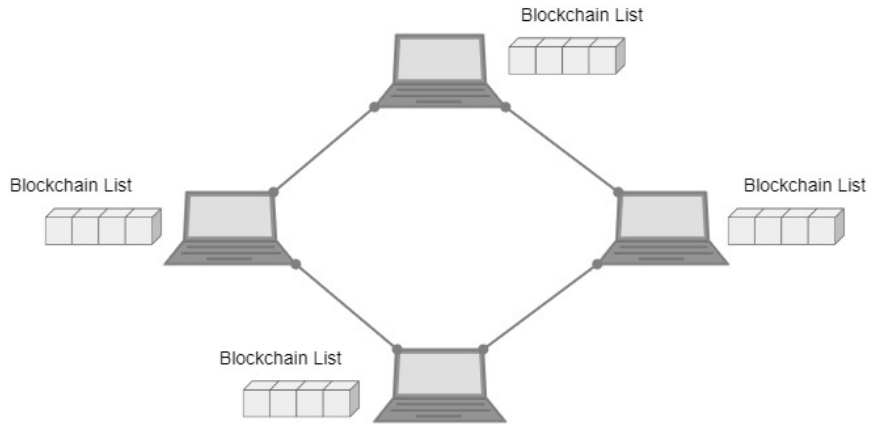

Figure 3. Proposed Blockchain model

The most promising answer to meet the aforementioned needs and obstacles is provided by the blockchain. It is suggested that a secure distributed ledger be implemented to share the logs and alarms that each IDS generates.

Individual IDS alerts are recorded on the blockchain as a transaction. Before a transaction is added to the blockchain, all of the individual IDS connected to the centralised server run a consensus protocol to verify it. Such a setup ensures that the centralised server only receives updated alerts that have been verified and authenticated. All participating IDS can access these tamper-proof alerts.

7. Smart Contract

7.1. What is Smart contract?

Digital contracts known as "smart contract" are executed automatically when certain circumstances are satisfied and are maintained on a blockchain.

Simply put, smart contracts are blockchain-based algorithms that execute when certain criteria are satisfied. They are often used to automate the implementation of an agreement so that all parties may be confident of the conclusion right away, without the need for an intermediary or additional delay. Additionally, they can automate a workflow, causing the subsequent action to be taken when the necessary criteria are satisfied.

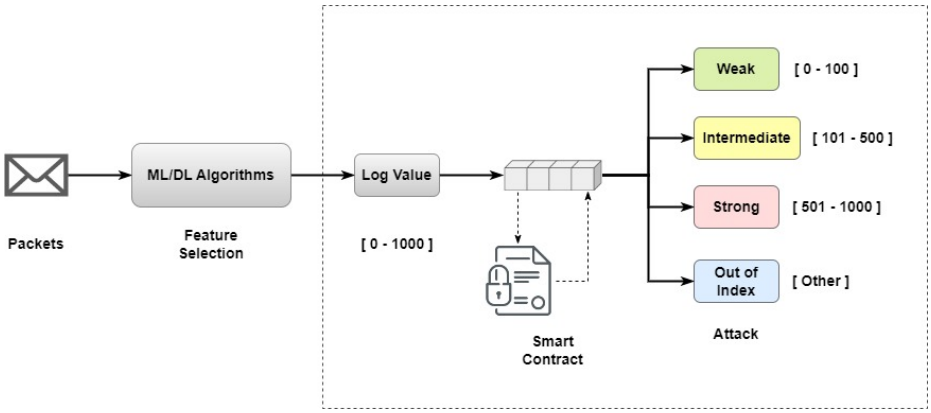

Figure 4. Working of smart contract

7.2. How smart contracts work?

Simple "if/when...then" phrases that are typed into code and placed on a blockchain are how smart contracts operate. When predefined circumstances have been verified to have been satisfied, a network of computers will carry out the actions. These can entail paying out money to the right people, registering a car, sending out notices, or writing a ticket. When the transaction is finished, the blockchain is then updated. As a result, the transaction cannot be modified, and only parties to whom permission has been granted can view the outcome.

As many conditions as are required to reassure the participants that the activity will be successfully accomplished can be included in a smart contract. Participants must agree on the "if/when...then" rules that govern those transactions, consider any potential exceptions, and design a framework for resolving disputes in order to set the terms. Participants must also decide how transactions and their data are recorded on the blockchain.

A developer can then construct the smart contract, however more and more businesses using blockchain for business are using templates, web interfaces, and other online tools to make creating smart contracts easier.

7.3. Benefits of smart contracts

- **Speed, efficiency, and accuracy** - The contract is promptly carried out if a condition is satisfied. Smart contracts are digital and automated, so there is no paperwork to complete or time wasted fixing mistakes that frequently occur when documents are filled out manually.
- **Trust and transparency** - There is no need to wonder whether information has been changed for one participant's personal gain because there is no third party engaged and participants exchange encrypted records of transactions.
- **Security** - Because the blockchain transaction records are encrypted, they are incredibly difficult to hack. Additionally, hackers would need to alter the entire chain in order to change a single record on a distributed ledger since each record is linked to the records that came before and after it.
- **Savings** - Smart contracts do away with the need for middlemen to handle transactions, along with the fees and wait times that go along with them.

7.4. Smart Contract implementation

We have discussed the proposed model of the implementation of the Intrusion Detection System using the Blockchain in the above section. Also, we discussed the usefulness of Smart Contracts in Blockchain implementation.

We have defined a scenario where every time a new packet enters the system, the contract checks the packet attributes and applies various ML/DL Algorithms on the dataset attributes to generate log value. In order to reduce the complexity of the structure, we take a user input log value and look for the attack class type. The below table shows the conditions for the different types of attack.

We have classified attack ranges as weak attack for 0 to 100 log values, intermediate attack for 101 to 500 log values, strong attack for 501 to 1000 log values and out of index attack for other log values.

```
1 pragma solidity ^0.4.24;
2
3 contract IDS {
4     uint256 private log; // To get the user input about the level of
       the attack(intrusion) [0, 1000]
5
6     function setLog(uint256 _log) public {
```

```
7         /*This function takes an uint256 input _log which is
      assigned to the log value. */
8         log = _log;
9     }
10
11    function getResult() public view returns(string) {
12        /*This function outputs the result */
13        if (log>=0 && log<=100) { // Range [0, 100] => WEAK Attack
14            return "Weak Attack";
15        } else if (log>=101 && log<=500) { // Range [101, 500] =>
      INTERMEDIATE Attack
16            return "Intermediate Attack";
17        } else if (log>=501 && log<=1000){ // Range [501, 1000] =>
      STRONG Attack
18            return "Strong Attack";
19        } else { // Random value is assigned
20            return "Out of Index Attack";
21        }
22    }
23 }
```

Listing 1: Solidity Code

Table 1. Attack Classification

Input log value	**Result**
0 - 100	Weak Attack
101 - 500	Intermediate Attack
501 - 1000	Strong Attack
other	Out of Index Attack

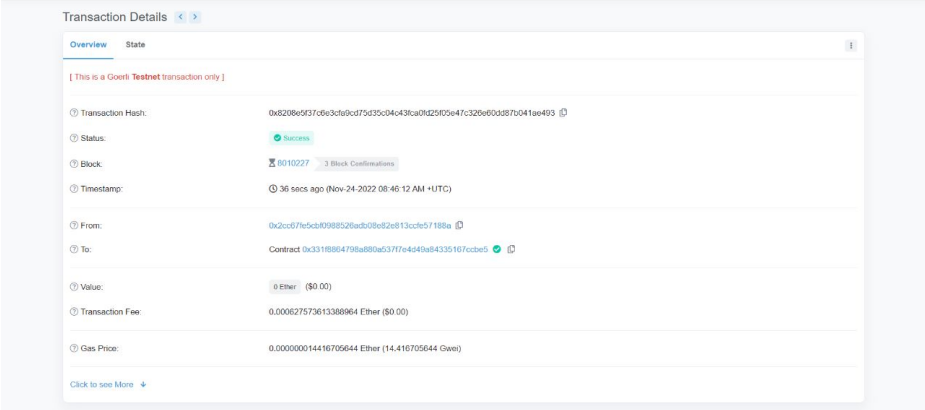

Figure 5. Etherscan.io output

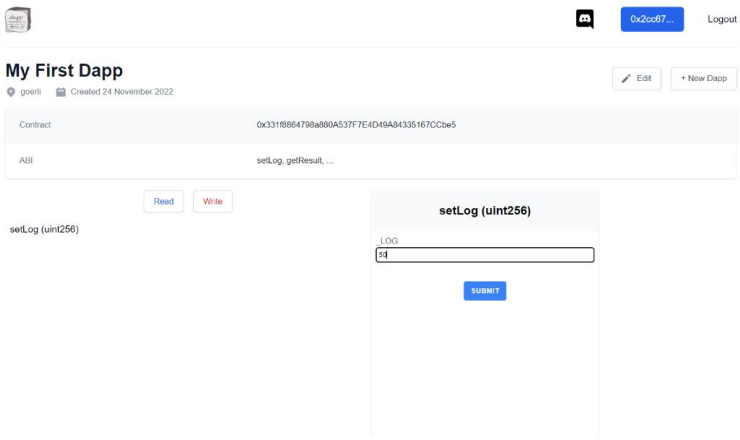

Figure 6. Weak attack log input

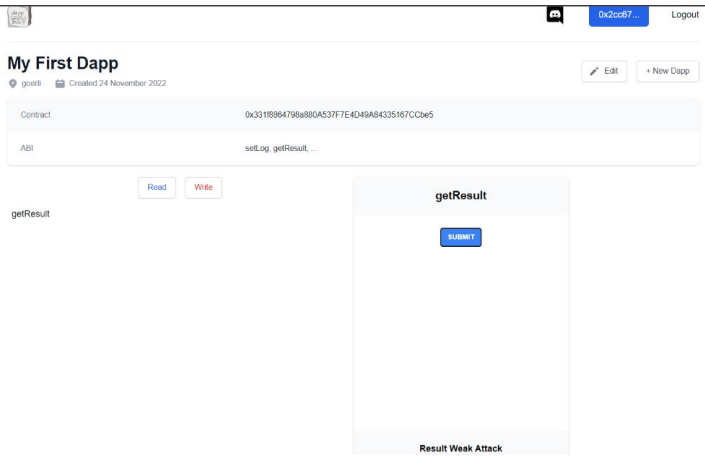

Figure 7. Weak attack output

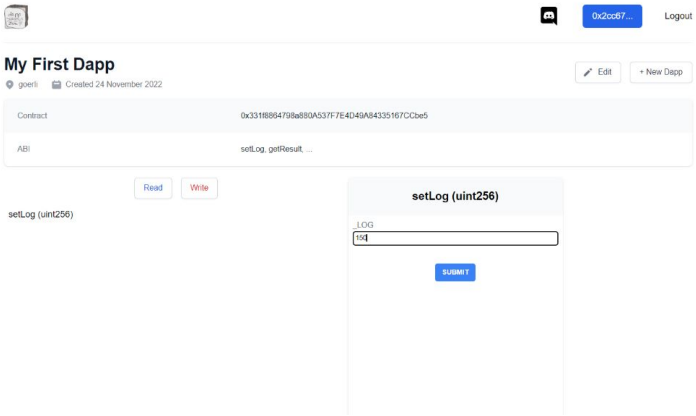

Figure 8. Intermediate attack log input

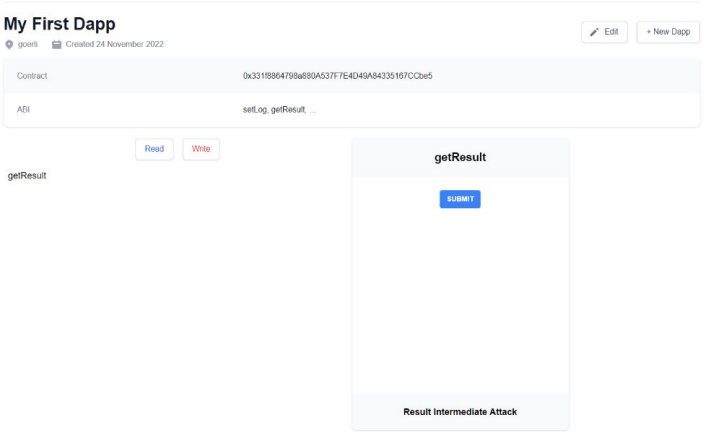

Figure 9. Intermediate attack output

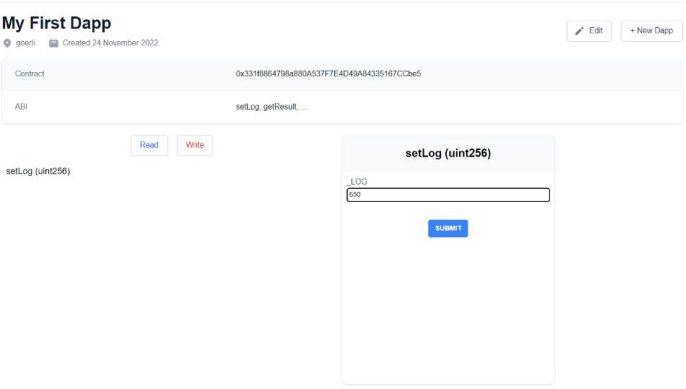

Figure 10. Strong attack log input

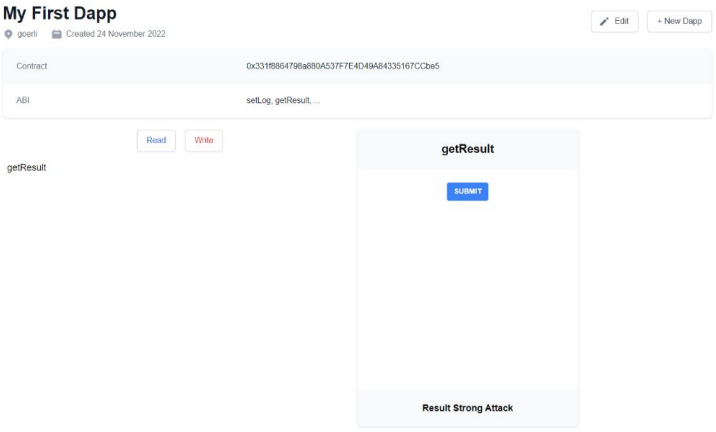

Figure 11. Strong attack output

8. Case Study - Kuwait Government IDS Survey

Due to the significance of Kuwait's position as an oil producer, several nations approach Kuwaiti official websites in pursuit of crucial information about the nation. A survey was conducted in order to evaluate the effectiveness of IDS in Kuwait in protecting governmental computer systems. The study included 90 employees from 16 different governmental organizations. From the Public Authority for Youth and Sports to the Ministries of Justice, Electricity, and Justice to the Public Authority of Industry, the entities studied span a wide spectrum.

System software specialists with five to nine years of expertise made up the majority of the employees. Information technology bachelor's degrees were held by about half of those surveyed. In contrast to 27% of respondents who claimed there had been attacks, 73% of respondents said there had not. 86% of respondents said they used firewalls to track down and stop attacks, while 14% said they didn't. 31% said they did not use IDS for detection, compared to 69% who said they did. 45% of respondents said they did not use NIDS, while 55% said they did. While 27% said they had no ties to other systems, 73% said they were connected to other organizations. 19% of respondents said they did not use the remote access option, while 81% said they use a service with 5 or more employees.

One-half of the respondents stated that some attacks were missed by their systems. In 16% of the organizations, updates are not made. 13% of organizations don't secure the network's IP range. 31% of organizations leave network users' default passwords unchanged.

The case study came to the conclusion that the surveyed organizations has numerous flaws in their systems' security . The authors of the case study suggested raising employee knowledge, enhancing the efficiency with which the firm uses IDS, and carrying out normal security maintenance, such as deploying updates, enforcing strong passwords, and creating a tighter security strategy.

9. Analysis of Blockchain-IDS Models

As was already established, anomaly and signature approaches form the foundation of blockchain-based IDS models, both of which have unique problems that blockchain technology may be able to address. The difficulties of IDS in both approaches are discussed in this subsection. Additionally, it analyses and contrasts the current blockchain-based IDS models now in use.

The anomaly detection approach suffers from a high number of false alarms, and it is unable to detect encrypted packet that occurs by cyberattacks. Moreover, it has difficulty constructing a normal profile for dynamic systems, its alarms are not classified, and initial training is required. In contrast, the main limitation in the signature detection approach is that it is unable to detect a new cyberattack in the system. Therefore, this approach needs to be updated frequently, and it is an inappropriate choice for detecting a multi-step attack.

The existing blockchain-based IDS models also suffer from different issues. Below table provides a description of each model along with their strengths and

weaknesses. As aforementioned, the common challenge between all models is that they have no standard design.

Table 2. Summary of the blockchain-based IDS models

Ref.	Description	Strengths	Weaknesses
[10]	It proposes blockchain protocol (CIoTA) based on a distributed and collaborative mechanism for anomaly detection in IoT network.	It improves security of IoT devices and the whole network as well.	It is not efficient security protocol for many IoT devices.
[11]	It proposes a protocol (CollabDict) for a collaborative anomaly detection based on blockchain and Gaussian mixture learning algorithm.	Performance of CollabDict is better than fuses multitask learning algorithm.	Collaborative learning has three main challenges, namely: (i) validation, (ii) consensus building, and (iii) data security.
[12]	It relies on the use of the kmeans algorithm to distinguish between malicious nodes and normal nodes through the analysis of pattern behavior for each node in a blockchain network.	It manages nodes and transactions in the network efficiently, also, it classifies nodes correctly.	It uses mean value for each cluster; thus, inaccurate cluster head might be selected, besides, it uses a static distance measure rather than a dynamic one.
[13]	It detects anomaly behaviours of participates in a blockchain network based on a game theory and a supervised machine learning algorithms.	It provides probability for each attack based on value of the transaction.	It requires improvements to strengthen its defense mechanism.
[14]	It builds BAD model to detect malicious transactions and prevent spreading them over the network.	It prevents malicious software from modifying the transactions' trace. Furthermore, data behavior should be verified from all participants in the network; thus, network security is increased.	It cannot detect the malicious transactions efficiently.

References

1 Tank D Aggarwal A, Chaubey N (2019) Virtualization vulnerabilities, security issues, and solutions: a critical study and comparison. Int J Inf Technol:1–16

2 Parul Singh, Virender Raga (2021) Attack and intrusion detection in cloud computing using an ensemble learning approach.

3 Modi CN, Patel DR, Patel A, Muttukrishnan R (2012) Bayesian classifier and snort based network intrusion detection system in cloud computing. IEEE, pp 1–7

4 Tony Bradley (2018) What is an IDS and Why Do You Need It? https://www.alertlogic.com/blog/what-is-a-network-ids-and-why-do-you-need/

5 Shruti Singh, Swedel Viola Fernandes, Vaibhav Padmanabha, Rubini PE (2021) MCIDS-Multi Classifier Intrusion Detection system for IoT Cyber Attack using Deep Learning algorithm.

6 Ly Vu, Quang Uy Nguyen , Diep N. Nguyen, Dinh Thai Hoang, Eryk Dutkiewicz (2022) Deep Generative Learning Models for Cloud Intrusion Detection Systems.

7 Dr. Manish Kumar, Ashish Kumar Singh (2022) Distributed Intrusion Detection System using Blockchain and Cloud Computing Infrastructure.

8 K. Devi,B. Muthusenthil (2022) Intrusion detection framework for securing privacy attack in cloud computing environment using DCCGAN-RFOA

9 Santhosh Parampottupadam, and Arghir-Nicolae Moldovann (2022) Cloud-based Real-time Network Intrusion Detection Using Deep Learning.

10 T. Golomb, Y. Mirsky and Y. Elovici, "CIoTA: Collaborative IoT anomaly detection via Blockchain," 2018. [Online]. Available at: https://arxiv.org/abs/1803.03807.

11 T. Idé, "Collaborative Anomaly Detection on Blockchain from Noisy Sensor Data," in 2018 IEEE Int. Conf. on Data Mining Workshops (ICDMW), pp. 120–127, 2018.

12 R. Kumari and M. Catherine, "Anomaly detection in Blockchain using clustering protocol," International Journal of Pure and Applied Mathematics, vol. 118, no. 20, pp. 391–396, 2018.

13 S. Dey, "Securing majority-attack in blockchain using machine learning and algorithmic game theory: A proof of work," in 2018 10th Computer Science and Electronic Engineering (CEEC), pp. 7–10, 2018.

14 M. Signorini, M. Pontecorvi, W. Kanoun and R. Di-Pietro, "BAD: Blockchain anomaly detection," IEEE Access, vol. 8, pp. 173481–173490, 2020.

15 F. Dai, Y. Shi, N. Meng, L. Wei and Z. Ye, "From Bitcoin to cybersecurity: A comparative study of blockchain application and security issues," in 2017 4th Int. Conf. on Systems and Informatics (ICSAI), pp. 975–979, 2017.

16 C. Elsden, A. Manohar, J. Briggs, M. Harding, C. Speed et al., "Making sense of Blockchain applications: A typology for HCI," in Proc. of the 2018 CHI Conf. on Human Factors in Computing Systems, pp. 458, 2018.

17 A. Al Omar, M. S. Rahman, A. Basu and S. Kiyomoto, "Medibchain: A Blockchain based privacy preserving platform for healthcare data," in Int. conf. on security, privacy and anonymity in computation, communication and storage, pp. 534–543, 2017.