Soft Computing in Intrusion Detection – Research Article

International Journal of Distributed Sensor Networks

International Journal of Distributed Sensor Networks
2019, Vol. 15(11)

DOI: 10.1177/1550147719888109
journals.sagepub.com/home/dsn
SAGE

An adaptive intrusion detection and prevention system for Internet of Things

Sheikh Tahir Bakhsh[1], Saleh Alghamdi[1], Rayan A Alsemmeari[1] and Syed Raheel Hassan[2]

Abstract

The revolution of computer network technologies and telecommunication technologies increases the number of Internet users enormously around the world. Thus, many companies nowadays produce various devices having network chips, each device becomes part of the Internet of Things and can run on the Internet to achieve various services for its users. This led to the increase in security threats and attacks on these devices. Due to the increased number of devices connected to the Internet, the attackers have more opportunities to perform their attacks in such an environment. Therefore, security has become a big challenge more than before. In addition, confidentiality, integrity, and availability are required components to assure the security of Internet of Things. In this article, an adaptive intrusion detection and prevention system is proposed for Internet of Things (IDPIoT) to enhance security along with the growth of the devices connected to the Internet. The proposed IDPIoT enhances the security including host-based and network-based functionality by examining the existing intrusion detection systems. Once the proposed IDPIoT receives the packet, it examines the behavior, the packet is suspected, and it blocks or drops the packet. The main goal is accomplished by implementing one essential part of security, which is intrusion detection and prevention system.

Keywords

Intrusion detection, networks security, IoT security, efficient

Date received: 28 June 2019; accepted: 10 October 2019

Handling Editor: Aneel Rahim

Introduction

The Internet of Things (IoT) devices are growing rapidly but these devices have limited memory, computation, and processing power in which they are based on low-end microcontroller.[1–5] There is no user interface in some of the devices that are made by the original equipment manufacturers (OEMs) that do not concern more about the security. The main issue these days is how to enable strong and secure low-end devices. In addition, it is important to make the implementation easier for OEMs. More than 360 IoT platforms use more than 100 protocols.[6] These varieties present several threats such as threats related to anomalies and intrusions in the network. Traffic in the network is monitored to report unusual activities like anomalies behavior that produced malicious attacks, for instance, viruses, denial-of-service attack (DoS), and distributed denial-of-service attack (DDoS), other attacks can cause accidental outages and fail in the equipment.[7] In

[1]Information Technology Department, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia
[2]Computer Science Department, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia

Corresponding author:
Sheikh Tahir Bakhsh, Information Technology Department, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia.
Email: stbakhsh@kau.edu.sa

order for ensuring the security of network infrastructure and communications through the Internet, several approaches and techniques have been developed. Intrusion detection and prevention systems, anti-virus software packages, and firewalls are examples of that method, and techniques have been wildly used to achieve security requirement. However, firewalls alone cannot defend against all types of intrusions and attacks, where intrusions try to break network security by taking advantage of vulnerabilities in the network.[8,9] The detection of abnormal behaviors in the networks such as penetrations, break-ins, or any other form of suspicious activity is called intrusion detection. An intrusion detection system (IDS)[10] is responsible to monitor all of the activities in the network and user behaviors to check if there are any suspicious activities or any violations in the specified policy. In addition, IDS can provide a report to the management station. Moreover, IDS is considered as an added wall that provides extra security to the network.

The IDS is a method that determines if there are any threats caused by intrusions on the system throughout the observations of the network traffic.[11] It is available around the clock to generate information regarding the state of the system, monitor the activities of the users, and provide reports to a management station. The classifications of IDS are network-based, host-based, and hybrid-based. The classification depends on the source and type of information for identifying security breaches.[12,13] There is no standard definition for IDS which we consider as any breach to the system; however, this also does not report the issues properly. Governments sectors, private sectors, companies, small business establishments, health sectors, and even individual users need to implement the IDS for identifying attacks and prevent in both host-based systems and network-based systems.[14]

The operation contains set of rules and policies to identify any type of threats, attacks, or intrusions to gain unauthorized access to any source of data or intercept a package on its way to the destination. IoT devices that connect to the Internet directly can be subjected to several threats and can be attacked easily. Although there are several techniques that have been applied to protect such environment, for instance, safe configuration, up-to-date patching, and firewalls, all of them are not easy to maintain and cannot ensure that the system can be secure form different types of attacks. IDS provides protection in which it monitors network or systems for policy violations or malicious activity. An IDS works like a "guard" which monitors the network and provides better security than other measures. The main objective of this article is to propose a solution for agent-based IDS for IoT environment that can enhance security measures including both host-based and network-based by examining the existing IDSs used in this field.

Intrusion detection methods and techniques

IDS can be classified into three main categories host-based, network-based, and hybrid-based.

i. *Host-based* IDS monitors a single system. In most cases, the IDS software runs on the host. It looks for logs and activities occurring on the system and tries to find anomalies.
ii. *Network-based* IDS system monitors a network segment in which IDS is sampling all the packets that pass through a specific point on the network. The network interface card listens to all the packets.
iii. *Hybrid-based* IDS, both host-based and network-based IDSs, can be used at the same time.

IDS

i. Misuse or signature-based detection model: The IDS has knowledge of suspicious behavior in which it looks for a recognized attack in its database by comparing the current activities with a signature attack in which if the system discovers a pattern it will send an alarm.
ii. Anomaly detection model: The IDS has knowledge of normal behavior, it looks for usage anomalies by sampling normal activities and an alarm of abnormal behaviors. However, it might result in several false-positive alarms.

The rest of the article is organized as follows. Section "Related work" discusses the related work and gaps in the existing study. The proposed methodology is presented in section "The proposed an adaptive intrusion detection and prevention system for IoT." Finally, the conclusion and future directions are drawn in section "Conclusion and future work."

Related work

Intrusion detection is an active field of research for about more than three decades. The interest in network intrusion detection has increased among the researchers along with the needs of security. Using automated tools and exploit scripts for the attacks, experienced intruders have performed large numbers of attacks 1980s in order to affect sites on the Internet. However, anybody can intrude using different tools.[15] Figure 1 illustrates the statistics of federal agencies in the United States, which shows that the number of cybersecurity incident reports increased dramatically from 2006 to 2015. However, due to some changes in the federal guidelines, it decreased by 60% in 20016.

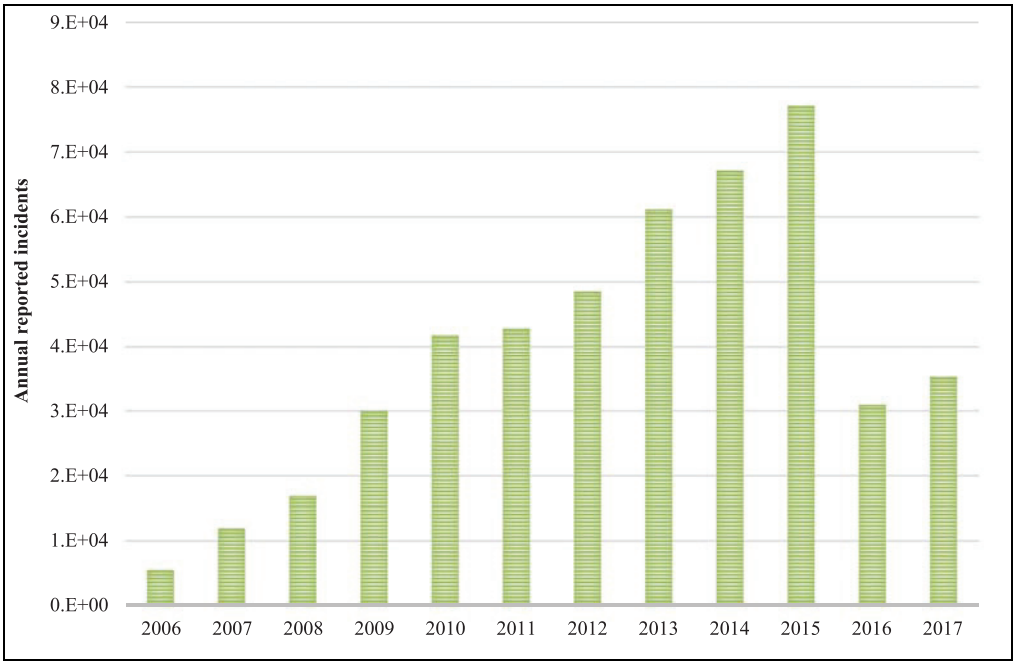

Figure 1. Number of cybersecurity incidents between 2006 and 2017.

Snort[3] is described as an open-source cross-platform lightweight network intrusion detection tool. It is considered as one of the most popular IDSs.[4] Snort is a signature-based detection model that is developed to observe and monitor the network traffic packets and detect any suspicious activity, intrusion, or threats on packets using pre-defined detection rules. It keeps a database of pre-defined rules and policies, which are used to describe different types of attacks, signature, and patterns of those attacks. In addition, a database can be updated by adding new rules to detect new discovery of any anomalous behavior or pattern of attacks. Snort has the ability to analyze the header and the payload of the network packet to detect any possible threats or attacks. The authors Razak et al.[16] used the idea of a friend in a small world phenomenon to propose an IDS framework for mobile ad hoc network (MANET) platforms. It is two tiers in which it is designed with the help of friend nodes to control long mechanisms of detection. In addition, it can overcome detection suffering from false accusations and the potential for blackmail attackers. The article shows that the impacts of the IDS issues can be reduced using their method of getting the advantage of friend nodes. However, it is analyzed that their proposed framework cannot work on several MANET platforms.

An anomaly-based IDS approach[17] is proposed to incorporate between a multivariate statistical process control (MSPC) which is called Hotelling's T^2 and radio frequency fingerprinting (RFF) in order to detect the attack. Depending on the generated signal, RFF is responsible for distinctively identifying a transceiver based on the transceiver print. We can achieve through wireless device MAC (media access control) address. However, still there is an issue because MAC address could be attacked, the transceiver prints would not match the profile with the claimed MAC address. Wormhole Geographic Distributed Detection (WGDD)[18] algorithm is proposed for distributed wormhole detection. The main task of this algorithm is to find a disorder of network produced by a wormhole. The passive nature of this kind of attack, a hop counting method, is used in the algorithm for detecting wormhole attacks. The local maps are reconstructed in every node. The algorithm can detect the abnormal behavior produced by wormhole attacks using a feature named diameter. A key benefit of applying the algorithm is that it can detect the position of wormhole that can help in the future to secure against these attacks.

Payload-based anomaly (PAYL)[19] detector builds a profile for the normal application payload of the network traffic in the training phase and uses that profile later for comparing detected intrusions. In the training phase, the profile of the application payload is built automatically in an unsupervised way. The profile consists of the centroids and the standard deviation of the byte frequency distribution of the network traffic payload for the flows based on the network hosts and ports. The byte frequency is computed by calculating the number of existences of every byte in the traffic

payload and then dividing it by the total number of bytes. For each different payload length, a different byte frequency distribution model is calculated. To detect intrusions, the byte frequency distribution of the network traffic payload is calculated. After that, the distance between the byte frequency distribution of the network payload and the profile is calculated based on the centroids and the standard deviation. If the distance is larger than a specified threshold, then an alarm is activated. Moreover, incremental learning is supported by PAYL, where the profile can be updated using new data without the need to recreate the whole profile again. As a result of the dependency on the payload length to build the models of the profile, a huge number of models are required. Therefore, to satisfy this requirement, the clustering technique is used to reduce the number of required models.

Hierarchal Intrusion Detection (HIDE)[20] developed as a distributed hierarchal system based on anomaly network intrusion detection system (NIDS). HIDE depends on statistical modeling, preprocessing, and classification of a neural network to detect network-based attacks. The network traffic information is observed to build the network statistical model. HIDE contains many intrusion detection agents, which are gathered in different hierarchal tiers. HIDE divides the network into zones. For each zone, a set of tier-1 agents is used to monitor the activities of the servers and the network bridges of that zone, to build the traffic statistical model, generate the monitoring reports periodically, and send the reports to an agent in tier 2. A tier-2 agent is used in each zone to receive the periodical reports of tier-1 agents of that zone, monitor and analyze the performance of the zone based on the received reports, and generate and send the report to an agent in tier 3. In addition, to receive the reports of tier-2 agents, tier-3 agents receive the reports of the tier-1 agents that are deployed in the network firewalls and routers. The network statistical model is built up by all agents participated in all different tiers to provide the neural network classifier. The neural network classifier's main objective is to decide whether the provided statistical model is normal or not.

HIDE has different components, a probe component monitors the network traffic to collect and extract a set of statistical variables based on the collected data for network traffic to reflect the network situation and generate periodical reports to the event preprocessor. Event preprocessor receives the reports generated from both the probe component and the reports of the agents in the lower tier, and construct the statistical model based on the received reports. The statistical processor compares the reports generated by the even preprocessor to the reference model and creates the stimulus vector which is provided to the neural network classifier. The neural network classifier receives the stimulus vector generated by the statistical processor, analyzes it, and classifies the network traffic whether it is normal or not. Postprocessor the neural network classifier to generate a report to the agents in the upper tier by the classifier. A neural network classifier needs time for training to learn before it can be used for detection. In the training phase, the neural network classifier is learned using learning data.

Flow-Based Statistical Aggregation Schemes (FSAS)[21] produces 22 statistical features for every network flow. The neural network classifier receives those features extracted by FSAS. The network flow can be modeled to be classified into two modes, safe and unsafe flows. This modeling is basically built in the training phase as a set of probability density functions of the 22 features values. The model contains two profiles, normal and attack profiles. In addition, FSAS consists of two main processes, which are a feature generator and a flow-based detector. An event preprocessor collects the network traffic from hosts or networks. Flow management module decides if each received packet is a part of existing network flow, or if it is the first packet in a new network flow. Afterward, it updates the records of the corresponding flow based on the received packet. The probe receives the information from the network flow coming from the flow management module and then extracting a set of statistical components to introduce the network status. Neural Network Classifier classified every network flow based on its score vector to be a safe or malicious flow. Feature analyzer identifies the type of attack based on the network's major behavior changes.

KMNP (k-means clustering based intrusion detection protocol)[22] detects intrusions efficiently using a clustering technique and a classification technique in two phases. In the first phase, KMNP uses the K-means clustering technique, the second phase uses the Naïve Bayes classifier. K-means technique is used to cluster and classify data into malicious and non-malicious groups in the first phase. In the second phase, Naïve Bayes classifier classifies data into its potential group. In addition, KMNP, K-means technique clusters data into three groups. The first group contains all the attack data such as a probe, R2L, and U2R. The second group contains the DoS attacks data. The third group contains normal network traffic data. K-means technique grouped data into K clusters/groups, where the centroid (mean value) of each cluster is considered as the seed point of that cluster. After that, based on the value of the squared distance between the data input and the centroids of the clusters, each data input is assigned to the nearest cluster. In the second phase, the Naïve Bayes technique is used which is considered as popular learning techniques. Naïve Bayes technique analyzes the relationship between the independent variable and

Table 1. Anomaly-based detection techniques.

System	Host-based	Network-based	Hybrid	Anomaly	Signature	Hybrid
Snort	×	√	×	×	√	×
MANET	×	√	×	×	√	×
RFF	×	√	×	×	√	×
WGDD	×	√	×	×	√	×
PAYL	×	√	×	√	×	×
Hierarchal	×	√	×	√	×	×
K-means	×	√	×	×	×	√
MINDS	×	√	×	√	×	×
GPGPU	×	√	×	×	√	×
Multi-agent	×	√	×	√	×	×

MANET: mobile ad hoc network; RFF: radio frequency fingerprinting; WGDD: Wormhole Geographic Distributed Detection; PAYL: payload-based anomaly; MINDS: Minnesota Intrusion Detection System; GPGPU: general-purpose graphics processing unit.

the dependent variable to identify a conditional probability for that relationship. Therefore, the Naïve Bayes technique classifies the network data into five classes: normal, DoS, probe, R2L, and U2R.

Minnesota Intrusion Detection System (MINDS)[23] is a data mining technique for intrusion detection. Each network connection is assigned with a score based on the probability of that connection to be an intrusion. MINDS detects the intrusions by using the packet's header information to construct the flow information. Flow information consists of IP addresses and ports of the source and destination, protocol, flags, number of bytes and number of packets of that flow. Based on time-window derived features, they are generated for the network flows with similar characteristics in the last "T" seconds. The local outlier factor (LOF) of the network flow is calculated based on the flow information and extracted features. LOF measures the degree of a network flow of being an outlier for its neighbors. To calculate the LOF, the density of the neighborhood is calculated. LOF is then computed as the average of the ratios of the density of the network flow and the density of its neighbors.

Graphics processor unit (GPU)-based hybrid multi-pattern algorithm (HMA)[24] is an IDS that has the computational capabilities power of a modern GPU. Network traffic throughput needs high-performance processors to handle high network traffic. Many network packets can be dropped and not examined while using CPUs with IDS overhead. In addition, those dropped packets may contain the intrusion and not recorded. The motivation behind using GPU is to provide IDS with real-time performance and has the ability to process network traffic by supporting parallelism. The authors Ashraf et al.[25] proposed a multi-agent artificial immune system for IDS. The system is proposed to implement multi-layers detection and classification for each agent in each host. An artificial immune system method is used based on the negative selection methodology. For classification, Best First Tree, Naive Bayes, and classifiers are used. The system has two categories of agents, which are the main agent and detector agent. The main agent is running in a centralized server and the detector agents installed and distributed in all machines in the network. The main agent generates the required information for the detection process and then distributes that information to the detector agents. The main agent generates and produces a set of anomaly detectors, which distributed to all detector agents. The detector agents evaluate each network connection using the anomaly detectors. If the evaluated network connection is matched with one of the anomaly detectors, an intrusion is detected and an alarm is generated. Many papers have been focused on discussing signature-based techniques. However, the researchers should contribute more to studying anomaly-based detection techniques, particularly for WLAN as shown in Table 1.

The proposed an adaptive intrusion detection and prevention system for IoT

This proposed IDPIoT is based on agent technology to support mobility, rigidness, and self-started attributes. Due to IoT limitations, the proposed solution is implemented in the middle, between IoT devices and the router that can be installed in a gateway. The proposed IDPIoT is a hybrid solution as it is based on misuse and anomaly. The prevention agent instance sent to perform prevention on IoT devices in case of attack or intrusion to isolate the IoT from the protected network until it is cured. Figure 2 shows the monitor agent is responsible for receiving the packet from the network and passing it to the detector agent, where the detector agent is responsible for detecting any suspicious activity

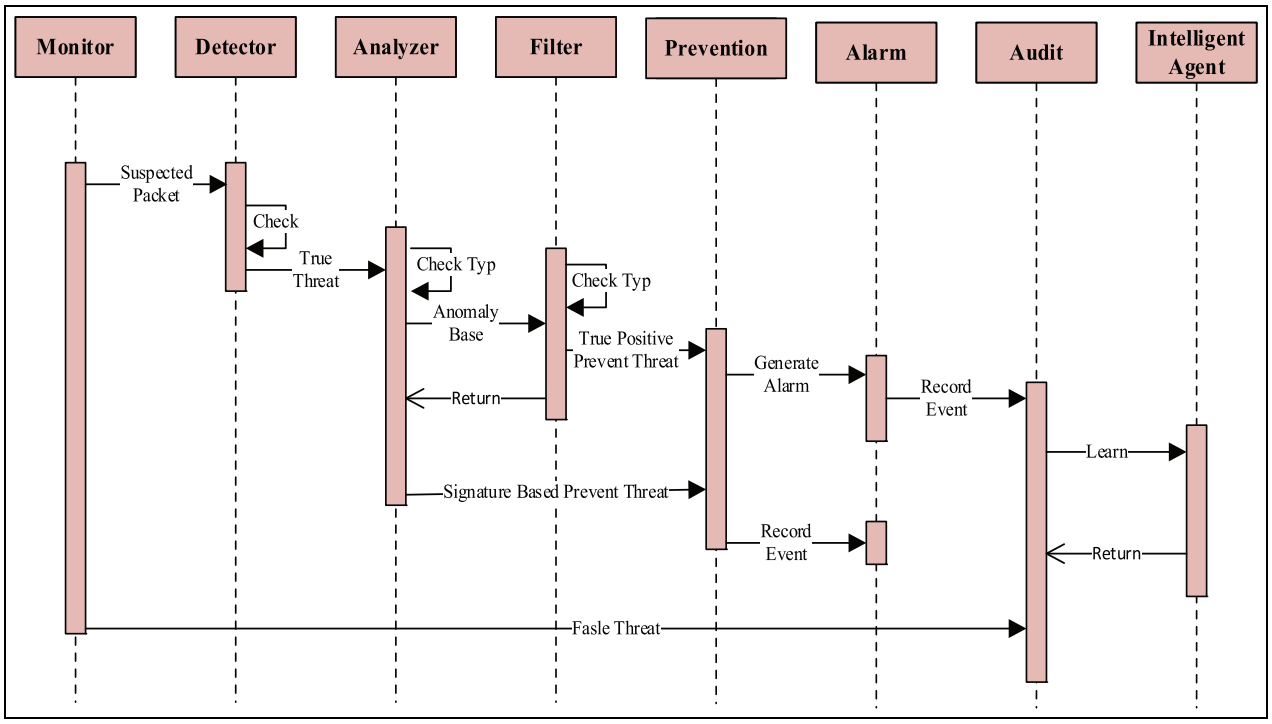

Figure 2. A sequence diagram shows system interactions.

and then passing it to the analyzer or filter based on the suspicion's activity (anomaly- or signature-based).

The analyzer agent runs in active (real-time) and passive mode, it is responsible to check the packet and decides whether it is a normal packet or threat based on the two approaches, signature-based or anomaly-based. If it decides it is up to normal packet or behavior, then it goes through the filter agent. The filtering agent decides whether it is false-positive or true-positive; if it is false-positive, no alarm will be generated; if it is true positive, an alarm will be generated, and prevention agent will take an action. If there is a suspicious intrusion or threat, an alarm generated to the administrative user, the packet is dropped, and the source of the packet is blocked by a prevention agent. The prevention manager is responsible for acting to protect the IoT devices, in case of DoS prevention manager, it sends an instance to the target IoT to drop the connection or packet of an intruder. An intelligent agent is where the agent can learn from the audit agent during the life of the system and can support later on for prevention or to update the analyzer and the filter agents. The data update agent is responsible for updating the filter and analyzer rules and policies, also it is getting updated by the intelligent agent. An audit agent is where all actions and events are registered in this agent. A report generator can generate periodic reports of the system based on user configuration.

Pseudo code for IDPIoT

```
Begin
  Foreach pkt ∈ PN do      //Packet analysis to dect the anomalies
        Alarm:= FALSE
        Detector_Agent (pkt)
        result_check = CheckThreat(Pkt)
        Audit_Agent(result_check)
        If result_check = "Known" then
              result_filter_agent = Filter_Agent(pkt)
              If result_filter_agent = "FP" then
                    Pass (pkt)
                    AuditAgent (pkt)
              Elseif (result_filter_agent = "TP" then
                    GO To I
              Endif
        Elseif result_check = "Unknow" then
              Normal:=Analyze(pkt)
              If ptk=Normal then
                    Pass(pkt)
                    AuditAgent(pkt)
              Else
        I:
                    Call Prevention_manager(pkt)
                    Alarm:=True
                    Drop(pkt)
                    Block(source)
                    AuditAgent(pkt)
              Endif
        Endif
        Intelligent agent     // Audi agent monitors the packets
              Audit_Data:=GetData from Audit_Agent()
              Update:=Learn (Audit_Data)
```

```
                Data_Update_Agent(updates)
                AuditAgent(updates)
    Data Update Agent              // Changes the packet status
                Update Filter_rules&polices()
                Update analyzer_rules&polices()
    Report                                  // Generates the report
                Check user_configuration()
                Generate_repot()
  End foreach
End
```

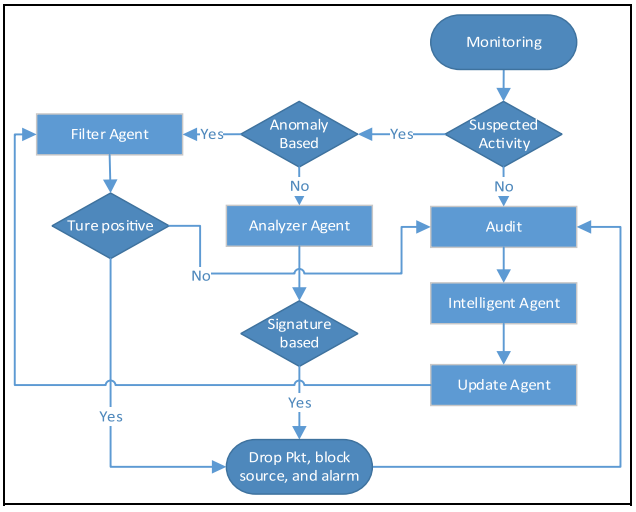

Figure 3. Activity diagram shows how to make decisions based on conditions.

In the proposed solution, the software is installed on an intrusion detection hardware and it is connected to the router and IoT devices to control the traffic and ensure all communication between IoT and the Internet is monitored. Its mediate between the router and the IoT devices connection only allowed from IDS. Therefore, an agent can travel to perform prevention to IoT devices and can isolate them from the network to prevent intrusion or attacks. The agent can also be installed on each IoT device as it can be run asynchronously. Users can access the IoT through the cloud, a middleware can be installed on the cloud. Figure 3 shows the system component, Radius/NAP, which is Remote Authentication Dial-In Users Server/Service. Network access protection can be used to authenticate the IoT devises. The firewall to add an extra layer of protection can be integrated with the proposed system. Intrusion detection and prevention system are based on a hybrid method for detection. IPSec is a secure network protocol suite that authenticates and encrypts the packets of data sent over an Internet protocol network to secure and encrypts the communication between IoT and end-user.

In the proposed solution, two possible scenarios are shown in Figure 4. First, an attacker may try to interfere with wireless to attack the IoT devices or the network or impersonate. Thus, in this solution, we implement RADIUS to authenticate the connected devices to the wireless network to ensure only legitimated devices are connected to the network. Second,

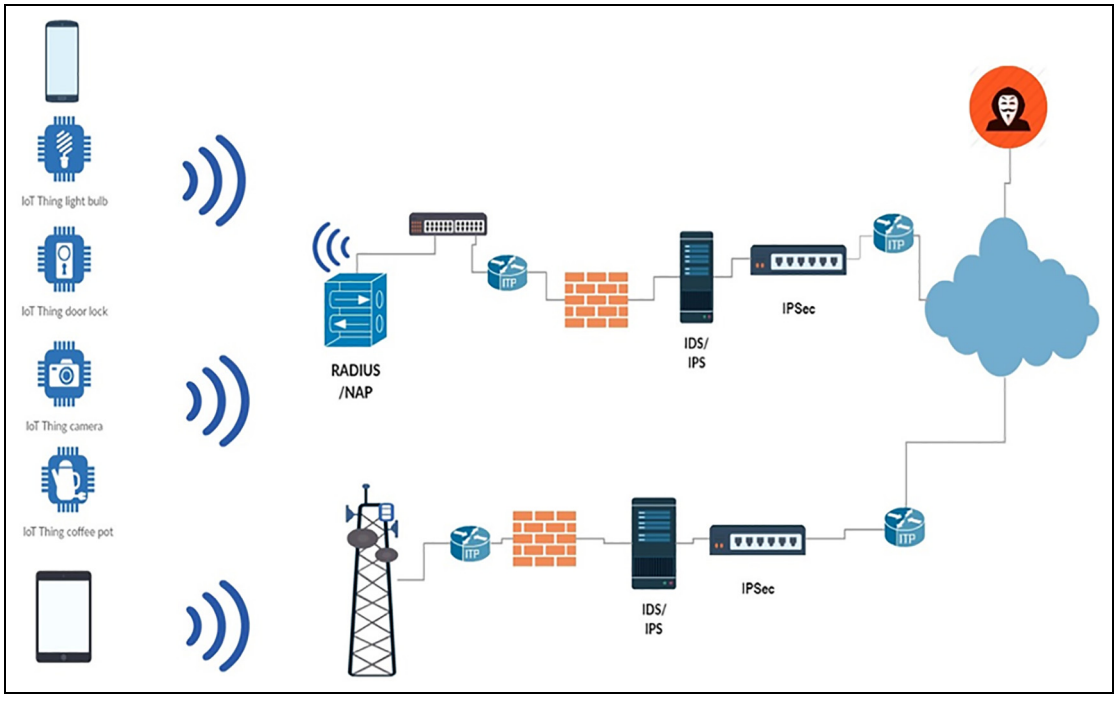

Figure 4. Deployment of the system.

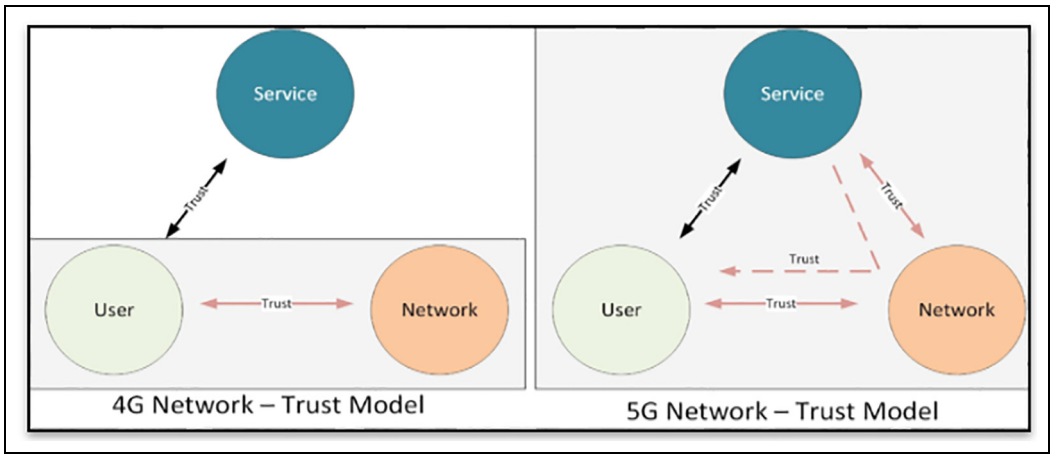

Figure 5. Network-based trust.

an intruder may try to impersonate the IoT device through using subscriber identity module SIM card to connect through cellular telephone subscribers to the network. Thus, our system is capable of detecting such intrusion and act to prevent this intruder by dropping the intruder packets and block the source of the packet. Thus, it can help in building the trust between service and network based on 5G networks as shown in Figure 5. In addition, it can be integrated with the firewall, so it can update the firewall rules and policies.

Conclusion and future work

The IoT is connecting more devices every day, with the current rate of IoT devices, utilization of security requirements is considered as the core component, as the attacker or intruder can misuse the devices to expose user confidentiality or disrupt services such as DoS and DDoS attacks. Thus, to satisfy the essential requirement, we need to implement and install intrusion detection and prevention system to keep IoT safe. IDSs can be categories into three types: signature-based, anomaly-based, and hybrid. In addition, IDS and IPS can be deployed as network-based, host-based, or hybrid-based. The proposed system provides a solution for intrusion detection to cover IoT security aspects. The proposed IDPIoT receives packets from the network interface and decodes the packets for processing to deliver to the detector agent. The detector agent checks each packet header for a certain type of behavior to detect any anomalies in the packet header. The system analyzer compares packet against pre-defined detection rules, such as matching the logging and alerting system is activated. It sounds alarms, log messages, and sends them to the output module. The system saves the output data and alert system to a pre-configured destination such as a log file or a database. Moreover, prevention agents drop the suspicious packet and block the source by providing real-time mitigation of attacks and isolation of the servers. In the future, the proposed work would be implemented and evaluated in the real systems. In addition, it may help in 5G networks to secure and build trust between service and network.

Acknowledgements

The authors acknowledge with thanks Deanship of Scientific Research for technical and financial support.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This project was funded by the Deanship of Scientific Research (DSR), at King Abdulaziz University, Jeddah, under grant no. D-078-611-1441.

ORCID iD

Sheikh Tahir Bakhsh https://orcid.org/0000-0001-8467-0375

References

1. Aldabbagh G, Bakhsh ST, Akkari N, et al. QoS-aware tethering in a heterogeneous wireless network using LTE and TV white spaces. *Comput Netw* 2015; 81: 136–146.
2. Imran M, Bakhsh ST, Tahir S, et al. A reconfigurable scatternet formation and maintenance scheme with heterogeneous services for smart Bluetooth devices. *Sustain Cities Soc* 2018; 40: 589–599.
3. Roesch M. Snort: lightweight intrusion detection for networks. In: *Proceedings of the 13th USENIX conference on system administration (LISA'99)*, Seattle, WA, 7–12

November 1999, pp.229–238. Berkeley, CA: USENIX Association.
4. Yin Y, Wang Y and Takahashi N. Set-based calculation of topological relations between snort rules. In: *2014 second international symposium on computing and networking (CANDAR)*, Shizuoka, Japan, 10–12 December 2014, pp.617–619. New York: IEEE.
5. Bakhsh ST, Sheikh MA and AlGhamdi R. Self-schedule and self-distributive MAC scheduling algorithms for next-generation sensor networks. *Int J Distrib Sens N*. Epub ahead of print 25 October 2015. DOI: 10.1155/2015/746216.
6. Xiao L, Wan X, Lu X, et al. IoT security techniques based on machine learning. arXiv preprint arXiv:1801.06275, 2018.
7. Bhanbhro H, Nizamani SZ, Hassan SR, et al. Enhanced textual password scheme for better security and memorability. *Int J Adv Comput Sci Appl* 2018; 9(7): 209–215.
8. Al-Turjman FM, Imran M, Bakhsh ST, et al. Energy efficiency perspectives of femtocells in Internet of Things: recent advances and challenges. *IEEE Access* 2017; 5: 26808–26818.
9. Zhang Z, Li J, Manikopoulos CN, et al. HIDE: a hierarchical network intrusion detection system using statistical preprocessing and neural network classification. In: *Proceedings of the 2001 IEEE workshop on information assurance and security*, West Point, NY, 5–6 June 2001, pp.85–90. New York: IEEE.
10. Bakhsh ST. Multi-tier mobile healthcare system using heterogeneous wireless sensor networks. *J Med Imag Health In* 2017; 7: 1372–1379.
11. Ertöz L, Eilertson E, Lazarevic A, et al. The MINDS—Minnesota Intrusion Detection System, 2004, https://www-users.cs.umn.edu/~kumar001/papers/minds_chapter.pdf
12. Aziz ASA, Hanafi SE and Hassanien AE. Multi-agent artificial immune system for network intrusion detection and classification. In: *Proceedings to international joint conference, advances in intelligent systems and computing*, Bilbao, 25–27 June 2014. Berlin, Heidelberg: Springer.
13. Hasan AA and Abdulrazzaq AA. GPGPU based hybrid multi-pattern algorithm design for high-speed intrusion detection system. In: *2014 IEEE international conference on control system, computing and engineering (ICCSCE)*, Batu Ferringhi, Malaysia, 28–30 November 2014, pp.141–146. New York: IEEE.
14. Tahir S, Bakhsh ST, AlGhamdi R, et al. Fog-based healthcare architecture for wearable body area network. *J Med Imag Health In* 2017; 7(6): 1409–1418.
15. Wang K and Stolfo SJ. Anomalous payload-based network intrusion detection. In: *International workshop on recent advances in intrusion detection*, Sophia Antipolis, 15–17 September 2004, pp.203–222. Berlin, Heidelberg: Springer.
16. Razak SA, Furnell S, Clarke N, et al. A two-tier intrusion detection system for mobile ad hoc networks —a friend approach. In: *International conference on intelligence and security informatics*, San Diego, CA, 23–24 May 2006, pp.590–595. Berlin, Heidelberg: Springer.
17. Hall J, Barbeau M and Kranakis E. Radio frequency fingerprinting for intrusion detection in wireless networks. *IEEE T Depend Secure* 2005; 12: 1–35.
18. Xu Y, Ford JC, Chen G, et al. Distributed wormhole attack detection in wireless sensor networks, 2010.
19. Aljawarneh S, Aldwairi M and Yassein MB. Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. *J Comput Sci* 2018; 25: 152–160.
20. Hamed T, Ernst JB and Kremer SC. A survey and taxonomy of classifiers of intrusion detection systems. In: Daimi K (ed.) *Computer and network security essentials*. Cham: Springer, 2018, pp.21–39.
21. Deng L, Li D, Yao X, et al. Mobile network intrusion detection for IoT system based on transfer learning algorithm. *Cluster Comput*. Epub ahead of print 31 January 2018. DOI: 10.1007/s10586-018-1847-2.
22. Kumar G, Saha R, Singh M, et al. Optimized packet filtering honeypot with snooping agents in intrusion detection system for WLAN. *Int J Inform Secur Priv* 2018; 12(1): 53–62.
23. Ahmed E, Yaqoob I, Hashem IAT, et al. Recent advances and challenges in mobile big data. *IEEE Commun Mag* 2018; 56(2): 102–108.
24. Hamed T, Ernst JB and Kremer SC. A survey and taxonomy on data and pre-processing techniques of intrusion detection systems. In: Daimi K (ed.) *Computer and network security essentials*. Cham: Springer, 2018, pp.113–134.
25. Ashraf N, Ahmad W and Ashraf R. A comparative study of data mining algorithms for high detection rate in intrusion detection system. *Ann Emerg Technol Comput* 2018; 2: 49–57.