# Raspberry Pi-Based Intelligent Cyber Defense Systems for SMEs: An Exploratory Study

Sreenivas Sremath Tirumala[1](✉), Narayan Nepal[2], and Sayan Kumar Ray[1]

[1] School of Digital Technologies, Manukau Institute of Technology, Auckland, New Zealand
{sreenivas.tirumala,sayan.ray}@manukau.ac.nz
[2] Technology and Innovation Research Group, School of Information Technology, Whitecliffe, Christchurch, New Zealand
narayann@whitecliffe.ac.nz

**Abstract.** Ongoing ransomware attacks have forced business to think about security of their resources. Recently, small-to-medium enterprises (SMEs) have become easy targets for attackers since they don't have cyber defense mechanism in place other than simple firewall systems which are quite vulnerable. Cyber defense systems are costly and often not within the budget of SMEs which inspired to think about low cost yet highly efficient cyber defense solutions. This research explores the prospects of implementing a Raspberry Pi (Raspberry Pi)-based intelligent cyber-defense system (iCDS) for SME networks and Smart-homes to filter malicious contents from incoming traffic. Primarily, the work presented in this paper tries to evaluate the hardware capability of network interfaces (both internal, and attached) of Raspberry Pi for handle high volumes of incoming traffic. For this, we measure the network performance of the Raspberry Pi using the speed test software. The results show that the built in Ethernet interface outperforms the built in WiFi and external attached USB to Ethernet Adapter in terms of latency, download and upload throughput.

**Keywords:** Cyber defense · Raspberry-Pi · Intelligent cyber-defense system

## 1 Introduction

The internet revolution had notable impact on day-to-day activities of small-to-medium enterprises (SMEs). The cloudification (moving the software and other operations services to cloud) of SMEs partially impacted the dependence on local hardware and networking configurations. In recent times high speed internet has become a mandatory requirement for SMEs since majority of services are operated through cloud. This reliance on internet made SMEs exposed to the rest of the world, particularly for hackers as soft targets for exploitation particularly through ransomware attacks. The operational implications of providing a secure environment for SMEs is costly due to demanding resource requirements like manpower and technology. With limited operational budget, majority of SMEs rely on internet service providers (ISPs) and local firewall or antivirus software for providing IT security. In countries like New Zealand, where majority of the


S. Paiva et al. (Eds.): SmartCity360° 2021, LNICST 442, pp. 3–14, 2022.
https://doi.org/10.1007/978-3-031-06371-8_1

business are SMEs, impose budget and resource constraints and are not be able to afford operational costs for providing cyber defense systems. According to a survey conducted by InternetNZ, about 48% of computers in SMEs are used by hackers for testing new malware and/or as bots to simulate Denial of Service (DoS) attacks. Also, considering the recent events where gaming devices are usd for mining bitcoins, there is a high chance for Smart-Homes being easy targets by hackers. Hence, the internal networks of SMEs (and Smart-Homes) have to be secured enough to prevent such external attacks.

Simple rule-based firewalls (i.e., based on administrator defined policies) of SMEs have failed to prevent attacks from random malware. Rule-based intruder detection systems (IDS) have managed to counter the attacks to some extent but not fully capable to provide complete security to the organization's network. The rule-based systems simply monitor and filter incoming network traffic based on set of predefined rules (malware signatures) stored in the repository. From the literature and implementation documents [1] it can be concluded that highly efficient IDS is more powerful and assertive in identifying malicious packets entering a network. However, traditional IDS requires special equipment and manpower and thus are resource savvy and costly to install and maintain. Also, it requires regular upgrades to identify and respond to new threats. Thus, majority of the SMEs with limited budget find it difficult to implement and maintain an effective IDS. Implementing low cost IDS solution that can operate as Security as a Service (SECaaS) and can be offered as subscription-based service, is another option for SMEs to consider. However, SECaaS still relies on rule-based systems and incurs all drawbacks of cloud-based and other remote service offerings. Moreover, the fact that SECaaS is expensive, a major concern for SMEs and are not effective for networks with IoT based devices [2, 3]. With the rapid integration of IoT with traditional networks, SECaaS may become a burden as the subscriptions needs to be paid in spite of them being used few times, purging less resources or bandwidth.

Formerly, computer networks are protected by firewall from the external attacks which is not different ffor SMEs. However, the usage of algorithms to create malware with no standard structure or pattern challenged the capabilities of simple rule-based firewalls and IDS. Majority of the firewall systems as well as IDS are based on administrator defined policies, or in simple terms, rule based. At present, the traffic is monitored and 'filtered' based on a set of rules (malware signatures) present in the repository. The limitations of firewalls, IDS and SECaaS discussed above, indicate an immediate necessity of introducing a low-cost, low-resourced yet advanced network security solution for SMEs particularly for stopping, as much as possible, the malicious network traffic from entering the SME networks.

This inspired to undertake an exploratory study on designing an intelligent intruder detection systems (iids) that can be implemented on a low-cost device to provide a small budget solution to SMEs and smart-homes. There has been some background work on non-rule based (pattern recognition based) solution for detecting malware [4]. This paper explores the prospects of implementing a low-cost intelligent cyber defense system (iCDS), in form of a filtering device, to protect the SMEs from malicious traffic. The proposal considers the plausibility of using Raspberry Pi device as a commercial IDS with the purpose of filtering malicious network traffic from entering SME networks. Primarily, through a systematic experimental evaluation this work tries to explore the capability of

network interfaces of Raspberry Pi device to understand their competence in handling high volumes of incoming traffic similar to commercial IDS systems. A comparative study of the performance of the inbuilt network interfaces, namely Ethernet (wired) and WiFi on the Raspberry Pi device, as well as an externally connected USB adapter interface (USB to Ethernet interface) are carried out in context to network parameters like latency, download throughput and upload throughput.

The remainder of the paper is structured as follows. Section 2 provides a review of the different filtering approaches and an introduction of the usage of Raspberry Pi-based IDS. While, Sect. 3 explores the prospects of using Raspberry Pi device as an iCDS, Sect. 4 discusses the evaluation results and Sect. 5 concludes the paper.

## 2 Review of Traditional IDS and Raspberry Pi-Based IDS

Traditional firewalls and IDSs use packet inspection for filtering traffic based on malware impressions [5, 6], and [7]. The workable solution proposed in [6] used a conceptual 'trust' based filtering that only allowed 'useful' packets to pass through. False positive results are often produced by the trust-based approach (similar to traditional fuzzy rule-based approach) and hence it was inconsistent in nature [6]. However, the proposed approach was successful in detecting malicious contents resulting from insider attacks in an organization. Since, iCDS mostly deals with identifying and filtering malicious contents from external network traffic trying to penetrate inside an SME network, insider attacks at this stage of the research is not considered. The filtering approach presented in [5] consisted of a restriction and access policy working as a traditional gateway. However, no evidence of experimental evaluation of the approach is proposed. An interesting machine learning based filtering model using Support Vector Machines (SVM) and Naïve Bayes is presented in [7], which also provides a good practical implementation scenario. However, due to its resource heavy and computationally complex nature, this proposed approach is unsuitable for SMEs. All these discussed research work provide an overview of important methods proposed for malicious network traffic filtering based on purpose and relevance. However, these implementations are generic in nature, not cost effective, and demand high configuration hardware for implementation. The next subsection discusses the implementation of Raspberry Pi-based low-cost IDS systems.

### 2.1 Background Study of Raspberry Pi-Based IDS

There is a lack of systematic literature review on implementing low cost IDS solutions for SMEs. Furthermore, very few research projects have been done on the feasibility of implementing Raspberry Pi (or a similar device)-based low-cost IDS for SMEs. This research gap provides an immediate necessity of such a research study to start with. A standard case of identifying low cost IDS solution for SME networks (containing different IoT devices), particularly using Raspberry Pi-based implementation, is relevant to the current research. IoT-based IDS implementations proposed by the research fraternity are mostly for non-commercial purpose and are either policy-based or graph-based. Policy-based approaches [8, 9] depend on a fixed predefined policy based on a specific domain or problem-based scenario similar to traditional network traffic packet

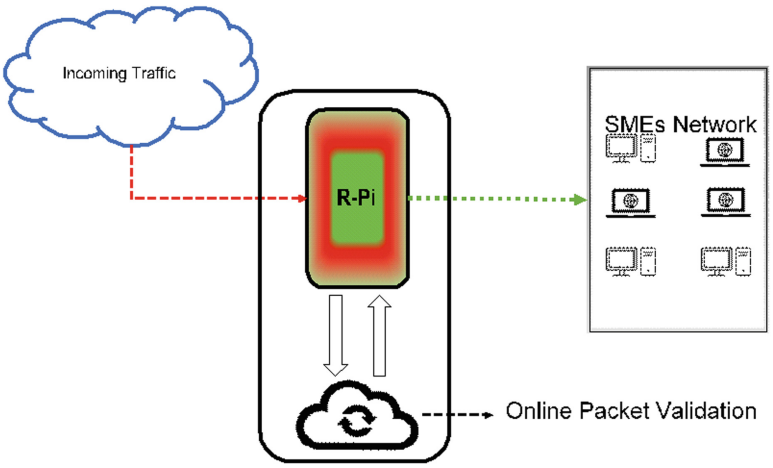


**Fig. 1.** The block diagram of iCDS representing various components

filtering approaches. The graph-based approaches [10] implement polices stored in a repository, which can be updated periodically (follows a dynamic rule). Such updates, however, lead to latency. A Raspberry Pi based firewall proposed by [11] to secure home networks, uses a remote cloud database with set of predefined rules. It uses on-board Ethernet interface for incoming network traffic and WiFi for outgoing traffic. The proposed approach is prone to delays and when applied for SME networks may incur significant latency. Another non-commercial implementation named as Pi- IDS is a Raspberry Pi 2.0-based standalone firewall implemented to filter websites in a school network. Although, an interesting concept, it has significant limitations in context to operation time and network traffic filtering capability.

Few research also proposed installing open source IDSs on Raspberry Pi so that it can replace a regular computer and can operate as a complete IDS of its own. For example, NetGaurd, proposed for traffic monitoring to track man-in-the-middle attacks, installs an open VPN and IDS software on Raspberry Pi to implement a complete IDS [12]. However, NetGaurd is nothing different to a traditional IDS and just provides privacy by hiding the IP of the monitoring source, as an extra feature. There are few other similar implementations like [3, 14]. The mere purpose of these implementation is to install and test IDS software on Raspberry Pi for various purposes. Two other research proposed by [15] and [16], used classification techniques for detection malicious contents in incoming network traffic. However, not only these two proposals lacked the technical details of hardware and software limitations of Raspberry Pi when experimenting it as an IDS, but also, they considered limited traffic with known malicious variants during the experiments. So, previous research mostly focused on studying how Raspberry Pi-based IDS can be implemented and if it can replace the traditional rule-based IDS implemented on normal computers. These implementations, knowingly or unknowingly overlooked the different challenges, including hardware limitations, to make Raspberry Pi operate as a fully commercial and real-world implementation of IDS. Furthermore,

such implementations are vertically divided into cloud based and non-cloud based and do not emphasize the need of a mixed model or fail- over model.

### 2.2 Key Challenges to Consider in Raspberry Pi-Based IDS

On a practical note, the following challenges need to be considered if implementing a Raspberry Pi- based iCDS for filtering malicious network traffic contents from entering SME networks.

*Handling High Volumes of Traffic:* Raspberry Pi has one on-board Ethernet port, which limits and delays the flow of incoming (from the internet) and outgoing traffic (after filtering). How to handle such latency? If external Ethernet adapter is used, what are its implications in terms of power, cost and heat?

*Processing Capabilities:* Rasberry Pi, being an embedded system has a low end processer and its processing capabilities may create some issue while handling the traffic and may effect a significant increase in processing and serialization delay too.

*Heat and Power Source:* Is the hardware of Raspberry Pi capable enough to run continuously and uninterrupted for a week?

*Storage and Real-Time Updates of Repository:* Efficient mechanism to store and update the repository (for rule based, signature based or any other approach).

*Hardware Capability for Storage and Execution of Algorithms:* There is a need for AI-based IDS for packet level monitoring of network traffic to filter malicious content. Therefore, is it possible to store and efficiently execute powerful AI algorithms on Raspberry Pi?

The overall research consists of various plausibility studies for hardware, software and algorithms. The AI-based algorithmic evaluation is been initiated and published [4]. This systematic experimental evaluation presented in this paper is confined to understand the capability of input network interface(s) of Raspberry-Pi.

## 3 Raspberry Pi as an iCDS: From Perspective of Hardware Capability

This current research explores the prospects of implementing a Raspberry Pi (Raspberry Pi)-based low cost and intelligent cyber-defense system (iCDS) for SME networks, the architecture of which is presented in Fig. 1. In the iCDS, all incoming traffic to the network of the SME will go through the Raspberry Pi device that will scan the traffic for any malicious contents. The traffic will be monitored and filtered through a cloud-based filtering system and all malicious traffic will be quarantined for further actions by the SME. A deep learning-based signature verification system will be used for filtering the traffic in the next phase of this work. The primary focus of the work presented in this paper is to explore (a) the feasibility of using Raspberry Pi device to develop an iCDS,

and (b) if the hardware components present in the latest Raspberry Pi devices are capable and compatible enough to support the use of Raspberry Pi-based iCDS for commercial SME networks. Use of Raspberry Pi as a low-cost device is becoming common in various IoT-based systems due to its simple operation, cost effective usage and support of open source software and operating systems. From the literature study presented in Sect. 2, it can be concluded that, although, research has shown the effectiveness of using Raspberry Pi-based commercial IDSs, previous work done on this aspect (i.e., use of Raspberry Pi as a commercial IDS) have not evaluated the efficiency and capabilities of the hardware components, especially, the Ethernet and WiFi modules on the Raspberry Pi board when handling input and output traffic. Also, typically, a commercially available IDS/Firewall will need gigabyte Ethernet-based connections for its input and output interfaces depending on the network requirements but each iCDS, on the other hand, need to have at least two physical interfaces with high end throughput to segregate the internal and external network traffic from each other. Thus, to explore whether it is possible to develop a Raspberry Pi-based iCDS, monitoring the performance of the different hardware interfaces on the Pi device when handling high volume of real traffic, is necessary. The following sub-sections will discuss these in detail.

### 3.1 Use of Raspberry Pi as an iCDS

Raspberry Pi is a low-cost computer that is commonly finding its usage in IoT and cyber-physical systems. Currently, Raspberry Pi 4 is the latest version and it has built in Ethernet interface and WiFi module. Owing to its tiny size, negligible power consumption and low cost, Raspberry Pi 4 can ideally be used as a commercial iCDS for filtering of malicious traffic entering the SME networks. However, traffic filtering using Raspberry Pi device will not be a straight forward process since Raspberry Pi can use only one network interface at any given time even if it may have multiple network interface connections (i.e., internet traffic only goes through the particular interface connection). For traffic filtering purpose an IDS needs at least two network interfaces, one for incoming traffic and the other for outgoing traffic. When connected to an external network, incoming and outgoing internet traffic to and from the network only flows through the particular interface of the Raspberry Pi that is directly connected to the external network, be it the Ethernet interface or the WiFi interface. Even if multiple USB adapters are connected to the different available ports in the Raspberry Pi device, internet traffic from the external network will only flow through one of these connections and that is an issue with the use of Raspberry Pi as an iCDS.

Using some channel bonding technology, however, it is possible to channelize the network traffic to flow through two separate network interface connections, one for incoming traffic entering the Raspberry Pi device from external network and the other for outgoing traffic from the Raspberry Pi device. This will need two network interface connections (e.g., network adaptors or network interface cards) in the Raspberry Pi 4.0 board and such connections can be in any form, like, the on-board Ethernet interface, on-board WiFi interface, USB Ethernet, and USB WiFi. For traffic filtering purpose, the Raspberry Pi device connected to a SME network, will require incoming and outgoing network traffic flowing through any of the two separate network interfaces. Based on such flow of network traffic, the following combinations are possible:

*Option1:* Network traffic entering the Raspberry Pi device through the on-board Ethernet interface and flowing out through the on-board WiFi interface.

*Option 2:* Network traffic entering the Raspberry Pi device through the on-board Ethernet interface and flowing out through USB Ethernet interface.

*Option 3:* Network traffic entering the Raspberry Pi device through the on-board WiFi interface and flowing out through the USB WiFi interface.

*Option 4:* Network traffic entering the Raspberry Pi device through the on-board WiFi interface and flowing out through the USB Ethernet interface.

There are, however, few issues with the selection of the different interfaces on the Raspberry Pi 4.0 board for incoming and outgoing network traffic unless proper channel bonding is used. One such issue, for example, when choosing option 1 (on-board Ethernet interface for incoming traffic and WiFi interface for outgoing traffic), the configuration will face an issue with the assigned IP addresses for the two interfaces. Generally, individual IP addresses will be assigned to the Ethernet interface and WiFi interface, respectively, for incoming packets entering the Raspberry Pi board to identify the particular entry interface's IP address and filtered outgoing packets (i.e., network traffic packets leaving the Raspberry Pi board to enter the SME network gateway) to identify the exit interface's IP address. Since, network traffic flows through only one connection (at a time) on the Raspberry Pi board, in absence of channel bonding technique, all traffic will just identify the Ethernet interface's IP address and flow through that, whereas, the other WiFi interface connection will remain unnoticed. This implies, that traffic will not enter the gateway of the SME network. Also, in case of option 3, when choosing two WiFi interfaces for incoming and outgoing traffic there can be an issue with the Raspberry Pi board not properly identifying the particular WiFi interface after every reboot operation (i.e., which interface is for incoming and which one is for outgoing traffic). There is a possibility that Raspberry Pi may not identify the WiFi interfaces correctly when rebooted and that may lead to incorrect communication of the network traffic. Thus, from these discussions it can be concluded that it is feasible to use Raspberry Pi device to develop an iCDS but proper channel bonding needs to be used for tracking the network traffic entering and exiting the different interfaces on board. In the following sections we study the performance of different interfaces on the Raspberry Pi device in handling high volume of real traffic entering the device.

### 3.2 Proposed Testbed for the Experiments

Figure 2 shows the proposed system model for using Raspberry Pi 4 as the iCDS in order to filter malicious packets from entering SME networks. Ideally, Raspberry Pi 4.0 device with 4 GB of RAM and 1.5 GHz 64-bit quad-core Arm Cortex-A72 processor will be used. It has built in Ethernet and WiFi interfaces. The Gigabit Ethernet interface in Raspberry Pi 4.0 can reduce communication latency and provide faster network connectivity. The device also has USB 3.0 and 2.0 ports. USB 3.0 ports can enable transfer of data up to ten times faster than USB 2.0. Based on the discussion provided in the previous sub-section, the proposed model will likely opt for option 2, where the on-board Ethernet interface will be used for incoming network traffic from external networks trying to enter the SME network through the Raspberry Pi 4-based iCDS and an USB Ethernet interface (in form of an adaptor) will be used as the exit for the filtered outgoing traffic

from the Raspberry Pi device to the gateway of the connected SME network (refer to Fig. 2). There is also an issue with choosing USB Ethernet for communication (option 2) as it may slow down the transfer of outgoing network traffic from the Raspberry Pi device to the gateway of the SME network, however, with the choice of proper USB Ethernet adaptor this shortcoming can be overcome. USBs are rated at speeds different to Ethernet, for instance, USB 3.0 is rated at 5 gigabits per second whereas USB 2.0 is rated at 54 megabits per second. For our proposed experimental testbed in this research, a Raspberry Pi 4.0 device is used that has a Gigabit Ethernet interface. Also, to ensure that the network communication on the Raspberry Pi 4.0 board does not slow down, a USB 3.0 Gigabit Ethernet interface (adaptor) is used so that communication between the two Gigabit Ethernet interfaces (the on-board one and the USB one) can happen. All the incoming internet traffic meant for the SME network will first enter the Raspberry Pi based iCDS acting as a protective shield for the SME network.

This entire research work will be carried out in two phases. In the first phase, as mentioned before, the aim is to study the feasibility of using Raspberry device to develop the iCDS and to explore if hardware interfaces on the Pi device are capable of handling high volume of real traffic to support its usage as a commercial iCDS, which is what this paper will discuss. In the following phase, the incoming traffic on the Raspberry Pi device will be sent through a cloud-based validation system where the signatures of the packets will be thoroughly checked to identify malicious contents (e.g., malwares). Such checking will be done at the signature-based detection online module (shown as cloud) of the proposed model where a lightweight AI-based pattern recognition and deep learning algorithm will inspect every packet to filter the malicious contents before letting the outgoing packets pass through the exit USB Ethernet interface to safely enter the SME network's gateway. These second phase activities are kept for future work and hence are not discussed in this paper. An important point that needs mentioning here is how the Raspberry Pi device can capture and track the network traffic flowing between its incoming and outgoing interfaces. This can be done in the following way. On starting, the Raspberry Pi device will load two scripts, the first of which is a shell script that will set up a software bridge connection between the incoming and outgoing interfaces. The bridge interface will have its own unique IP address assigned and will allow for network connectivity. The second Python script will tcpdump the network packets (flowing between the input and output interfaces) on the Raspberry Pi 4.0 device so that they can be captured and assessed.

## 4 Performance of the Raspberry Pi Interfaces

### 4.1 Measurement of Latency

Latency is a significant aspect in determining the efficiency of any network interface. In the experiments conducted, latency of each interface on the Pi board (i.e., Ethernet, WiFi, and USB interfaces) is measured individually based on the incoming unfiltered real network traffic entering each interface separately over a time interval of t to $t + 1$. Figure 3 depicts the latency comparison of the three interfaces on the Raspberry Pi 4.0 device based on separate measurements of the incoming network traffic.

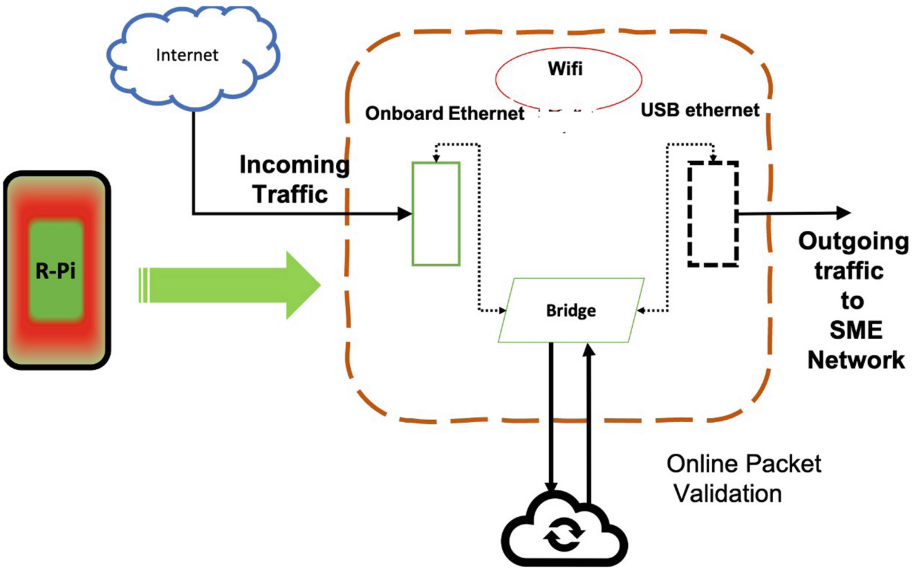


**Fig. 2.** The hardware architecture for the proposed Raspberry Pi-based iCDS

As can be seen in Fig. 3, the latency of the built-in WiFi interface on the Raspberry Pi device is considerably high in comparison to the latencies of the built-in Ethernet and USB Adapter interfaces. Apart from the fact that Ethernet (wired) connections usually offers better network speed and significantly lower latency compared to WiFi (wireless) connections, the other reason can be that the built-in WiFi on the Pi device has a single antenna and not a MIMO, so lower speed and more latency anyway. On the other hand, the Ethernet interface also offers lower latency than the USB adapter interface.

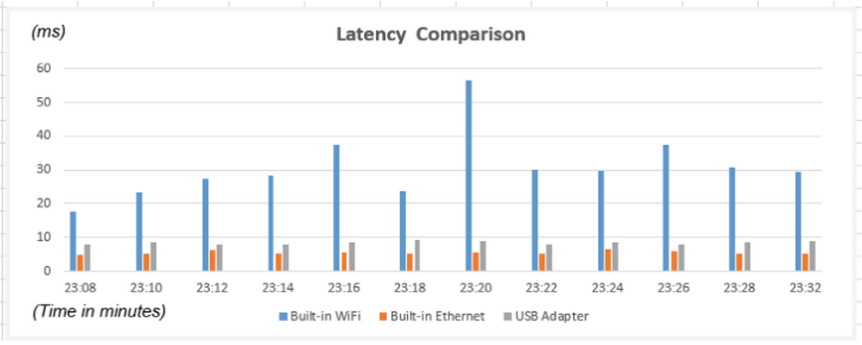


**Fig. 3.** Latency comparison of Raspberry-pi interfaces when handling external traffic

### 4.2 Measurement of Download Traffic Throughput

Similar to latency, download and upload throughput of network traffic are other important aspects of determining the efficiency of a communication interface. The download traffic for each interface on the Raspberry Pi 4.0 device is measured separately over the t to t + 1 time interval and the comparison results for the three interfaces are shown in Fig. 4.

From the given figure, it is evident that the throughput of the built-in Ethernet interface on the Pi device is significantly higher than the throughput of the WiFi and USB Adapter interfaces. Again, this can be related to the fact that Ethernet connections generally offer better network speed and thus better (download) throughout in comparison to WiFi and the USB connections. Performance of the in-built WiFi and USB interfaces look somewhat similar.

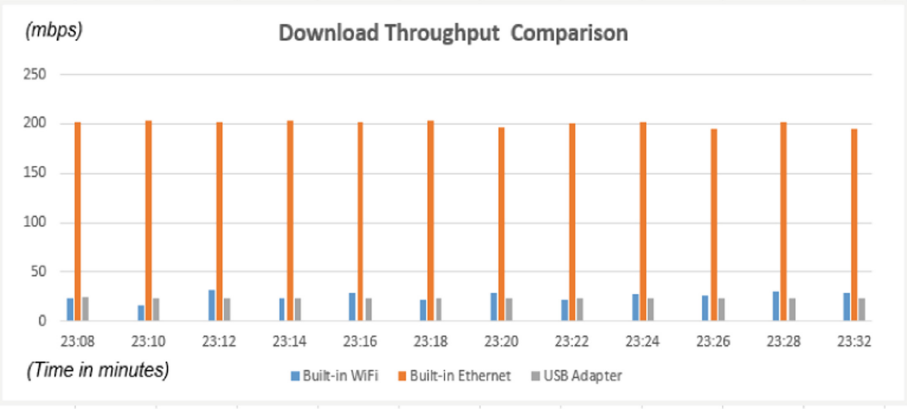


**Fig. 4.** Traffic comparison for download throughput

### 4.3 Measurement of Upload Traffic Throughput

Figure 5 compares the throughput of the upload traffic for the three network interfaces on the Raspberry Pi 4.0 device. The upload throughout performance of the built-in Ethernet interface has somewhat outperformed the other two interfaces. The USB Adapter on the Pi 4.0 device, unlike the Ethernet, shares a common bus and hence its bandwidth is also distributed among other ports, which is why it experiences some internal delays and has a low throughput.

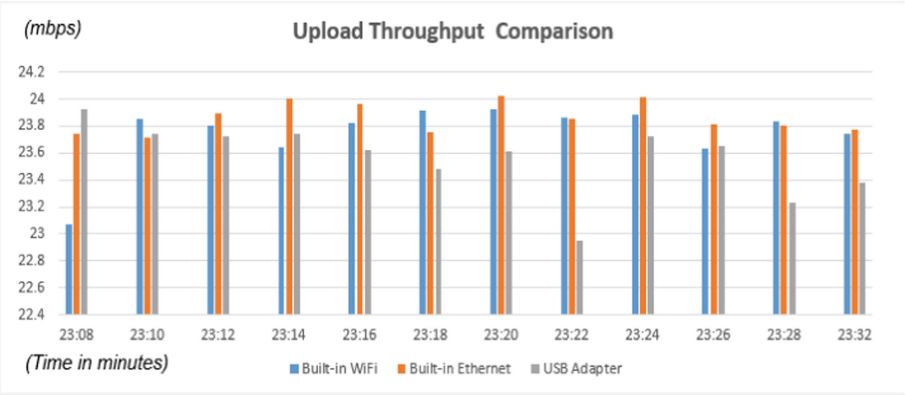


**Fig. 5.** Traffic comparison for upload throughput

## 5 Conclusions and Future Work

Primarily, the work presented in this paper has a two-fold focus: (a) to explore the feasibility of using Raspberry Pi device to develop a low-cost intelligent cyber-defense system or iCDS for commercial SME networks, and (b) to study if the hardware components present in the latest Raspberry Pi devices are capable and compatible enough to support the use of Raspberry Pi-based iCDS for SMEs. Based on the detailed discussions presented in the paper, it can be concluded that it is feasible to use Raspberry Pi device to develop a low-cost iCDS as an alternative to the traditional rule-based IDSs in use. Moreover, from the experimental results as discussed in Sect. 4, it is evident that the different interfaces on the Raspberry Pi 4.0 device, e.g., built-in Ethernet (wired) connection, WiFi and the external USB Adapter, studied in this research are capable of handling high volumes of traffic entering the Raspberry Pi device from outside networks. The evaluations also showed that in terms of network performance comparison carried out based on parameters, like, latency, downward traffic throughout and upward traffic throughput, the built-in Ethernet network interface has outperformed the other two interfaces and thus can be an ideal choice to use for handling external traffic.

While, this paper explains the first phase of the research work on this topic, as mentioned in Subsect. 3.2, further work (second phase) will be focused on the following:

Incoming traffic on the Raspberry Pi device will be sent through a cloud-based validation system where the signatures of the packets will be thoroughly checked to identify malicious contents (e.g., malwares).

A lightweight AI-based pattern recognition and deep learning algorithm will be prepared to inspect the packets and filter out malicious contents before the packets can safely enter the SME networks.

A functional prototype of this low-cost iCDS will be developed, which will use the deep-learning based signature verification model for filtering malicious contents from incoming network traffic.

## References


1. Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J.: Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecurity **2**(1), 1–22 (2019)
2. Ali, B., Awad, A.I.: Cyber and Physical Security Vulnerability Assessment for iot-Based Smart Homes. Multidisciplinary Digital Publishing Institute (2018)
3. Granjal, J., Monteiro, E., Silva, J.S.: Security for the internet of things: a survey of existing protocols and open research issues. IEEE Commun. Surveys Tutorials **17**(3), 1294–1312 (2015)
4. Tirumala, S.S., Valluri, M.R., Nanadigam, D.: Evaluation of feature and signature based training approaches for malware classification using autoencoders. In: 2020 International Conference on COMmunication Systems & NETworkS (COMSNETS), pp. 1–5. IEEE (2020)
5. Malikovich, K.M., Rajaboevich, G.S., Karamatovich, Y.B.: Method of constucting packet filtering rules. In: International Conference On Information Science and Communications Technologies (icisct), IEEE (2017)
6. Meng, W., Li, W., Kwok, L.F.: Towards effective trust-based packet filtering in collaborative network environments. IEEE Trans. Netw. Serv. Manage. **14**(1), 233–245 (2017)
7. Serdechnyi, V., Barkovska, O., Rosinskiy, D., Axak, N., Korablyov, M.: Model of the internet traffic filtering system to ensure safe web surfing. In: International Scientific Conference "Intellectual Systems of Decision Making and Problem of Computational Intelligence" Springer, Cham (2019)
8. Kolias, C., Kambourakis, G., Stavrou, A., Voas, J.: DDoS in the IoT: Mirai and other botnets. IEEE Comput. **50**(7), 80–84 (2017)
9. Lu, D., Huang, D., Walenstein, A., Medhi, D.: A secure microservice framework for iot. In: Symposium on Service-Oriented System Engineering (SOSE), IEEE (2017)
10. Pahl, M.O., Aubet, F.X., Liebald, S.: Graph-based IoT microservice security. In: Network Operations and Management Symposium, (NOMS), IEEE (2018)
11. Gupta, N., Naik, V., Sengupta, S.: A firewall for internet of things. In: 9th International Conference on Communication Systems and Networks (COMSNETS), IEEE (2017)
12. Taib, A.M., Zabri, M.T., Radzi, N.A.M., Kadir, E.A.: NetGuard: Securing Network Environment Using Integrated OpenVPN, Pi-Hole, and IDS on Raspberry Pi. In: Charting the Sustainable Future of ASEAN in Science and Technology Springer, Singapore (2020)
13. Jesús, R.L.J., Cristhian, P.V.O., René, R.G.M., Heberto, F.M.: How to Improve the IoT Security Implementing IDS/IPS Tool using Raspberry Pi 3B. In: Editorial Preface from the Desk of Managing Editor (2019)
14. Tripathi, S., Kumar, R.: Raspberry pi as an intrusion detection system, a honeypot and a packet analyzer. In: International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS) IEEE (2018)
15. Soe, Y.N., Feng, Y., Santosa, P.I., Hartanto, R., Sakurai, K.: Implementing lightweight iot-ids on raspberry pi using correlation-based feature selection and its performance evaluation. In: International Conference on Advanced Information Networking and Applications, Springer, Cham (2019)
16. Sumanth, R., Bhanu, K.N.: Raspberry Pi based intrusion detection system using k-means clustering algorithm. In: Second International Conference on Inventive Research in Computing Applications (ICIRCA), IEEE (2020)