# Malicious/Phishing URL Detection System in A Network with Raspberry Pi (NETBITS)

Mohamad Aniq Fakhrul Mohamad Fauzi, Lili Marziana Abdullah

Department of Information Systems, Kulliyyah of Information and Communication Technology, International Islamic University Malaysia, Kuala Lumpur, Malaysia.

lmarziana@iium.edu.my

***Abstract***— Phishing is an online fraudulent act that has affected many organisations. Organisation and home networks are becoming more susceptible to attacks with the emergence of more and more malicious and phishing URLs, and the increase significance of communication and data storage on the internet. Firewall and antivirus may not be enough against such intrusion. Without a secure and efficient detection system, a local network can easily be penetrated by the attackers or unauthorized individuals. Several existing network intrusion detection products are available out there, but those systems are usually integrated with large-scale hardware and difficult for the public to set up, use readily and carry everywhere. This paper reports on a system development project to design and develop a cross-platform malicious/phishing URL detection system, referred to as Netbits. Netbits, built using Django framework based on Python language and installed on Raspberry Pi, can monitor network traffic flow for any malicious activities in real-time. The simulation carried out in a network environment and testing performed showed that Netbits system can detect malicious activities and display the infected scans. Nevertheless, more enhancements may be required before rolling out Netbits to the public.

***Keywords***— Phishing, Network Intrusion, URL Detection, Raspberry Pi.

## I. Introduction

Phishing is an online fraudulent act that employs technical scheme to deceive internet users into revealing their sensitive data or credentials [1]. The schemes used by attackers to manipulate people include spoofed emails, fake websites, and fake SMS from service providers. The common targets are big corporations and small companies that have employees. A 2021 study conducted by Proofpoint, a leading cybersecurity and compliance company in the US, revealed that 83% of the survey respondents, consisting of professionals from U.S., Australia, France, Germany, Japan, Spain and UK, said that their organisations experienced at least one successful email-based phishing attack in 2021 [2]. Some of the techniques used by the attackers are through sending malicious emails to trick user to provide credentials and sending infected URLs to employees in companies that enabled ransomware to be deployed. According to the study, 78% of organisations experienced a ransomware attack in which a phishing email was the initial infection vector. Ransomware executes a malicious code that denies user or organisation access to files in their computer and the attacker then demands a ransom payment to restore the files to the initial state. Most victims tend to pay quickly because of the sensitive nature of their resources [3].

Firewall and antivirus may not be enough to protect users or organisations against such intrusion. Without a secure and efficient detection system, a local network can easily be penetrated by the attackers or unauthorized individuals. Unauthorized entry, especially to big networking systems in organisations might leave a huge impact to the organisations since it can cause data breach or personal data being leaked outside of the organisations. Attackers commonly attack the weakest part of an organization which is the employees [4]. They will target the employees by sending them malicious phishing emails. A poll by GetApp found that, out of 714 employees polled, almost half said someone from their organization clicked a phishing link [5]. A person tends to download malicious file from the email without having it scanned first so the file can potentially bypass the antivirus. This commonly happens when the IT department of an organization did not update the detection software in the environment.

Without adding an extra layer of security, the malicious file received through emails or infected URLs will easily bypass the network and get downloaded on the local computer. This could be prevented by installing an intrusion detection system on the local network to filter out malicious files or links going in and out of the building. Organisation and home networks are becoming more susceptible to attacks with the emergence of more and more malicious and phishing URLs, and the increase significance of communication and data storage on the internet. Developing malicious URL detection is now a vital network security task in a network environment [6]. On this premise, work was carried out to design a mobile and affordable system that any organisation and home can set up easily and use to detect malicious and phishing URLs, and receive alerts and notifications to prevent users from clicking the URLs.

This paper reports on the design of a cross-platform malicious/phishing URL detection system (for Windows, Unix and Macintosh) using Raspberry Pi, referred to as Netbits. The objectives are to:

1) monitor the traffic flow for any malicious activities of a network in real-time,
2) reduce the phishing activity in an organization or home network, and
3) notify users on their daily drivers once they are infected.

## II. Related Work

Open-source intrusion detection systems that analyse network traffic for malicious links are readily available. Among them are Open-Source HIDS SECurity (OSSEC), Snort, Suricata, Zeek, Samhain, Fail2ban and Security Onion [7]. These systems are cost-effective in monitoring network environments and detecting cyber attacks, however, review of related works for this paper focuses on three popular, intelligent detection systems which are Suricata, Snort and Fail2Ban [8]. They are used extensively in the industries and even houses. These systems allow users to secure their network by detecting malicious or brute forcing attack within the network in real time, 24/7.

Suricata is an open-source intrusion detection system that was developed by Open Security Foundation (OISF) [9]. Suricata combines intrusion detection (IDS), intrusion prevention (IPS), network security monitoring (NSM) and PCAP processing. It can do offline PCAP processing which means, users can manually sniff the network with tools like tcpdump or wireshark and scan the PCAP file with Suricata. Suricata inspects the network traffic using powerful and extensive rules and signature language, and it uses Lua scripting support to detect complex threats. It also provides a github repository that allows contributors to enhance the system [9].

Snort is a free and open-source network intrusion detection system that was created in 1998 by Martin Roesch, founder and former CTO of Sourcefire [10]. Snort uses a series of rules that help define malicious network activity and uses those rules to find packets that match against them and generate alerts for users. Snort can also be deployed inline to stop these packets. Snort has three primary uses which are as a packet sniffer, as a packet logger (useful for network traffic debugging) and as a full-scale network intrusion prevention system. Snort can be downloaded and configured for personal and business use [10].

Fail2Ban is another open-source system that scans log files *(e.g., /var/log/apache2/error.log)* and bans IP addresses that show malicious signs such as too many password failures and suspicious actions within a given time period [11]. It has an interesting feature that prevents attackers from performing brute force attacks such as directory brute forcing or password spraying. Fail2Ban can reduce the rate of incorrect authentication attempts by blocking them from reconnecting to the network. However, it cannot eliminate the risk that weak authentication presents as attackers can use random user agents or IP spoofing to bypass the restrictions [11].

Several existing network intrusion detection products are available out there, but those systems are usually integrated with large-scale hardware and difficult for the public to set up, use readily and carry everywhere. People with average computer skills may find it difficult to execute the numerous set-up steps required by the existing systems. Hence, Netbits was designed for public use such as home or organization that has internet connection. It is installed and set up on a Raspberry Pi device, a low cost, bank--card sized computer that can be plugged into a monitor or TV, and normal keyboard and mouse can be used on the device because it provides multiple USB Type-A ports like many other laptops. Raspberry Pi was chosen for its portability, inexpensive cost and decent processing power [12]. Raspberry Pi can do almost everything we would expect a huge pricy computer to do, from playing high-definition video, browsing the internet and also playing games.

## III. Methodology

Continuous research and development of network intrusion detection systems are important in preventing networks from being infected by the growing numbers and sophistication of attacks that occur daily. The development of Netbits followed several phases as shown in Fig. 1. The rest of the paper briefly describes the work carried out in the development phases and the output which is Netbits.

In the analysis phase, two (2) people experienced in networking were interviewed. The feedback gathered highlighted a few features and designs to be considered in the development. These include use of authentication, inspection of network activity and use of honeypots, implementation of dashboard for summarizing data and detections, and manual scanning.

Once features were prioritised, the design phase started with the design of use cases and narratives for receiving incoming network packets, parsing and examining the received packets, generating reports of infected packets and notifying users if infected. Fig. 2 illustrates the proposed architectural flow of Netbits. The main tasks in the architectural flow are described in the ensuing sub-sections. Corresponding data stored would include parsed packets before scanning, scan results, device information in the network (i.e. IP addresses), reports received from VirusTotal, and activities log.

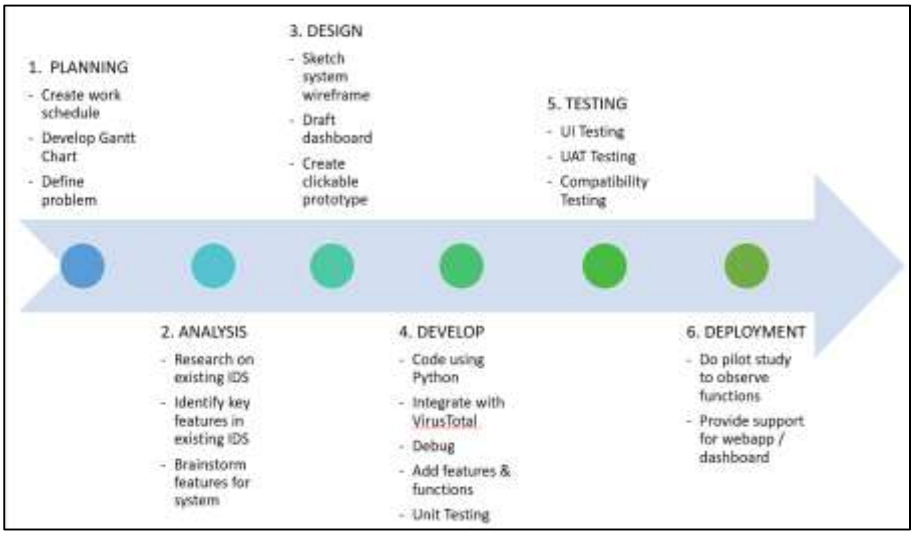

1. PLANNING
- Create work schedule
- Develop Gantt Chart
- Define problem
2. ANALYSIS
- Research on existing IDS
- Identify key features in existing IDS
- Brainstorm features for system
3. DESIGN
- Sketch system wireframe
- Draft dashboard
- Create clickable prototype
4. DEVELOP
- Code using Python
- Integrate with VirusTotal
- Debug
- Add features & functions
- Unit Testing
5. TESTING
- UI Testing
- UAT Testing
- Compatibility Testing
6. DEPLOYMENT
- Do pilot study to observe functions
- Provide support for webapp / dashboard

Fig. 1 Development Phases

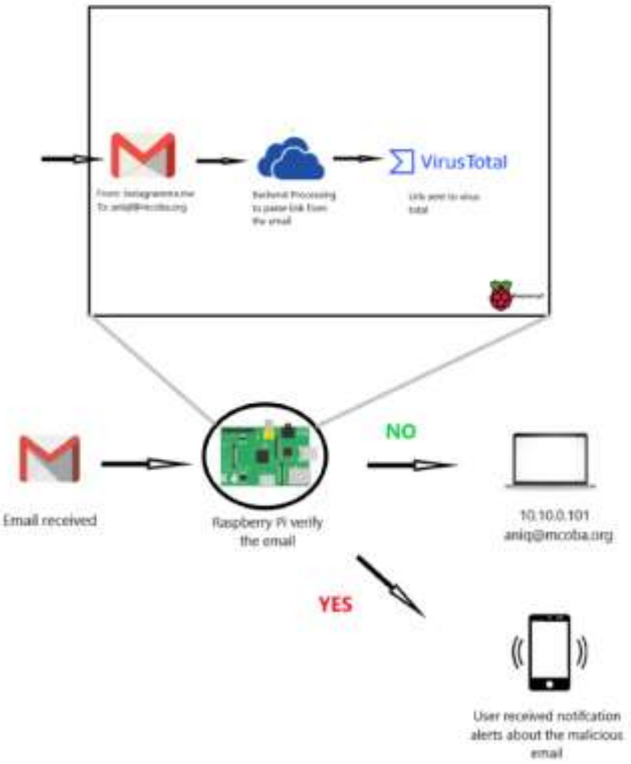

VirusTotal
Email received
Raspberry Pi verify the email
NO
YES
10.10.0.101
aniq@mcoba.org
User received notification alerts about the malicious email

Fig. 2 Netbits Architecture

## A. Packet Capturing

New packets are generated by sniffing or capturing the network traffic by using a Linux tool called “tcpdump” which allows us to capture packets transmitted through the network interface and to export the packets in a PCAP file. This tool captures the packets received from the network and converts those packets to the user readable form.

## B. Packet Parsing

Parsing phase takes place after the PCAP file has been generated by the tcpdump tool. Parsing consists of few techniques, and they are all parsed by a script written in Python programming language. The packet files are being filtered by protocols which are UDP and TCP. This system will only accept TCP protocol because malware and internet traffic are mostly transferred through TCP protocol.

After it is filtered to only accept TCP protocol, it will then be further parsed through different subcategories - link/URL, source IP and destination IP. Parsing phase converts the byte sequence from the packet to a human readable form which it will then parse the PCAP file into source IP address, destination addresses and URLs.

## C. POST and GET to VirusTotal API

After the packet is ready for analysis, the URLs are sent to VirusTotal via public API to be scanned by 70 antivirus scanners and URL/domain blocklisting services. The URLs are matched with existing URLs in their databases. VirusTotal is part of Google Cloud and is a free online scanning service that analyses suspicious files, URLs, domains, and IP addresses to detect cybersecurity threats [13]. If it detects or returns any matches in their databases, it will then return the number of matches into a JSON key called positives. All the reports will then be compiled with other JSON keys such as metadata, scanned data and time. VirusTotal will then provide Netbits with a JSON report that contains full URL, domain and positive results.

## D. Notify Users

Netbits, also, will enable a pop-up notification through a third-party app called Notify17 as it is free and easy to use. Notify17 lets you generate notifications with simple web requests, and you can receive them either on Android, IOS and desktop/laptop. The system will use the information reported by VirusTotal and extract the URL and source IP to identify specific users from the IP address. This will be easier for the system to recognize and push the notification to the targeted user/employee. Users in the network will be notified together with the website URL they visited.

### *E. Block Unwanted Threats*

Additionally, Netbits has also made provision to block traffic from a specific IP address which has been recognized to be malicious or troublesome. There is also the provision to allow traffic from specific IP addresses for some thruster systems, from which traffic is not monitored. Traffic from unknown hosts is monitored and any potential attacks are informed to the user such as phishing or brute force attacks. This is implemented based on the detection mechanism used by the Fail2ban system mentioned earlier.

## IV. Development

The Netbits system was developed as a set of interdependent subsystems built to achieve stated objectives by performing specific functions such as scanning, parsing, fetching reports, device ping and others. These functions correspond to the tasks mentioned in Section III earlier. Within the context of the project, the integration activities were carried out at different levels of components, subsystems to full system level. Additionally, this project requires the set up of a network environment to simulate the real environment in an organization's network as shown in Fig. 3.

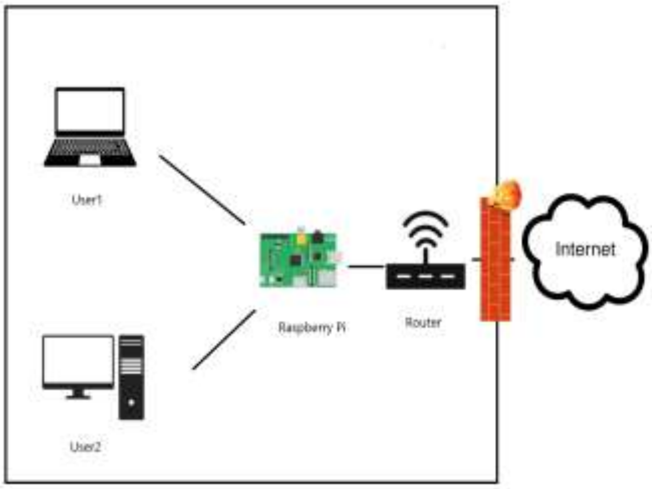


Fig. 3 Netbits Environment

The integration of components in this network environment includes the following:

### *A. Router*

A small, isolated network infrastructure was created that can easily be monitored and managed by the network administrator.

### *B. Raspberry Pi*

The device acted as a proxy server which is also the main hardware to execute script and analyse the network. Proxy server is an intermediary server that sits in between the users and the internet to monitor, for example, browsing history. Fig. 4 shows the Raspberry Pi used in the implementation.

Fig. 4 Image of Raspberry Pi

### *C. Proxy Tool*

Since the system is meant to intercept and log all users' activities over the network, a proxy software called Privoxy was installed on Raspberry Pi. Privoxy is an open-source tool that can be installed on any Unix-based computers. It acts as a proxy device that can log any users' activities over the network once they are connected to Privoxy. Netbits uses this functionality to parse the log file and get specific information such as source IP, domain, time and others. A proxy server can log every request over the network in real-time. This will save time and increase accuracy compared to manually capturing the network packets. Every single request from connected devices goes through the proxy before accessing the internet.

### *D. API*

Netbits scanning processes and stages would not be completed without the integration with VirusTotal public API that allow Netbits to send URLs and receive results in the form of JSON format. VirusTotal API performance and traffic bandwidth is very quick and smooth when transferring data and reports in parallel with the system that sends the URL in real time.

### *E. Python-based dashboard*

The dashboard is coded in Python because of its simplicity and fast execution time. The web app framework used is Django because it is a structured web application framework like Tomcat and Laravel, and it is based on Python language which suits the system backend process. The system main dashboard, shown in Fig. 5, is divided into four (4) sections. The dashboard sections are elaborated in the result section (Section V).

## V. Result

Netbits successfully detected malicious activities and displayed the infected scans when ran in the simulated network environment. When the system was booted up, the main dashboard as in Fig. 5 appeared with multiple sections, each with its own specific functionality.

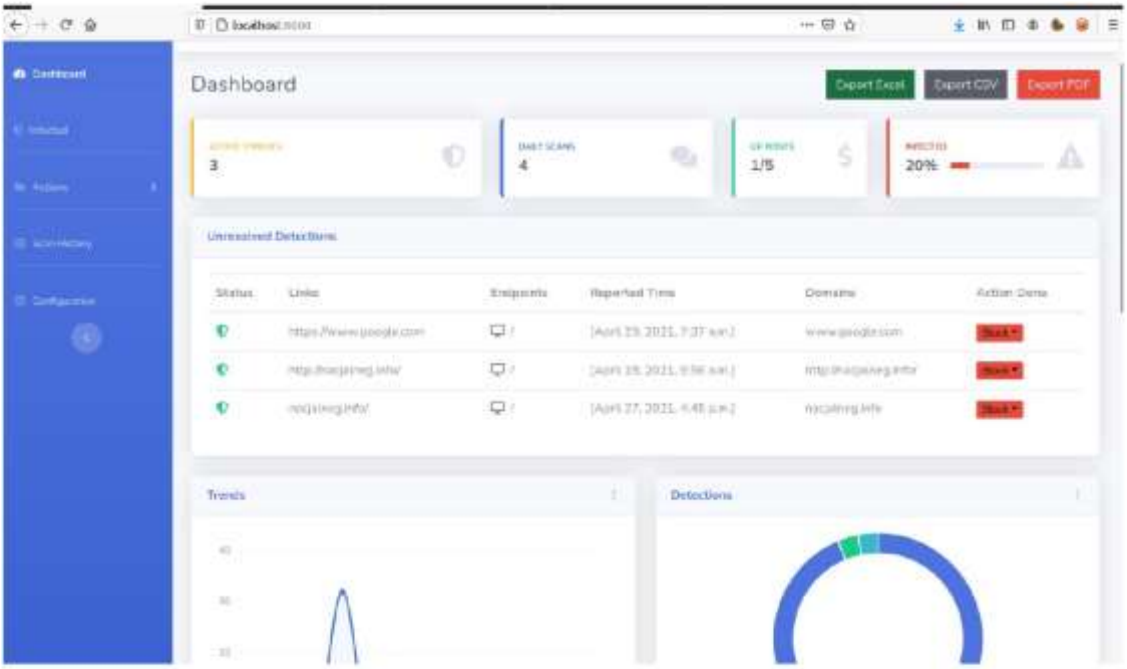
Referring to Fig. 5, the first box at the top-row section shows the number of *active threats* found for the day which is three (3), and the number will be refreshed daily to summarize the daily *active threats*. Four (4) in the second box is the number of *daily scans* performed by the system. System schedules the scans on intervals and system also parses the URL from Privoxy's log file during the scan interval. The back end is kept running, 24/7, non-stop to ensure that no malicious activity can bypass the checking. The third box shows the number of *up host* or active hosts (1 out 5 host is active currently) in the network environment while the percentage of computers (20%) infected for the day is shown in the last box. The system does a ping scan throughout the whole subnet to ensure that specific hosts are up when the system receives a response from their respective icmp server.

The second-row section lists the infected domains/links notified by the system. Every link is parsed into several smaller parts to differentiate between endpoints, domain, source and destination IP addresses. Action dropdown buttons located on the right side of the table allow users to update the action taken toward specific infected domain. By default, domains are blocked by the system.

The third-row section shows two types of charts. The line chart shows the monthly infected scans whereas the doughnut chart shows the domains that frequently infect the network environment. All these statistics can be exported to Excel, Word or PDF format. The fourth and final-row section of the page shows *occurrences* and *recent activities* (Fig. 6).

Fig. 6 Netbits Dashboard – Occurrences and Recent Section

*Occurrences* show the top five (5) domains that frequently infect the network, sorted in descending order. Logs are one of the most important parts of the system as this helps in the debugging process or determining whether the system is working properly or not. Every log or system functionalities working in the background are logged and displayed in *recent activities* box.



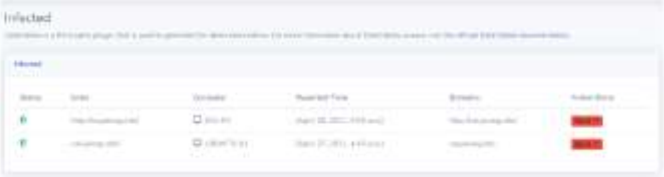
Dashboard

Unresolved Detections

| Status | Links | Endpoints | Reported Time | Domains | Action Done |
|---|---|---|---|---|---|
| | https://www.google.com | / | [April 29, 2021, 7:37 a.m.] | www.google.com | Block ▾ |
| | http://nacjalneg.info/ | / | [April 28, 2021, 9:56 a.m.] | http://nacjalneg.info | Block ▾ |
| | nacjalneg.info/ | / | [April 27, 2021, 4:45 p.m.] | nacjalneg.info | Block ▾ |

Fig. 5 Netbits - Main Dashboard

Three other pages in the system are *Infected* (Fig. 7), *Actions* (Fig. 8 and Fig. 9) and *Scan History* (Fig. 10). *Infected* page lists the infected scans and traces which computers browsed the infected scans. Action buttons are also displayed in this page to allow users to update status of any domains/links that are blocked by the system. *Actions* page is separated to two (2) pages which are *Allow List* and *Block List*. These pages allow users to specify exceptions to specific domains. *Allow List* page allows users to easily delete, add and update the row. *Block List* page shows how many blocked domains currently in the network. The list also provides the severity information that shows the most detected URL. *Scan History* page allows administrator to refer to old scans as scans are mapped to this page every time Netbits perform background scanning.



Infected

Fig. 7 Netbits - Infected Page

Fig. 8 Netbits – Allow List Page

Fig. 9 Netbits – Block List Page

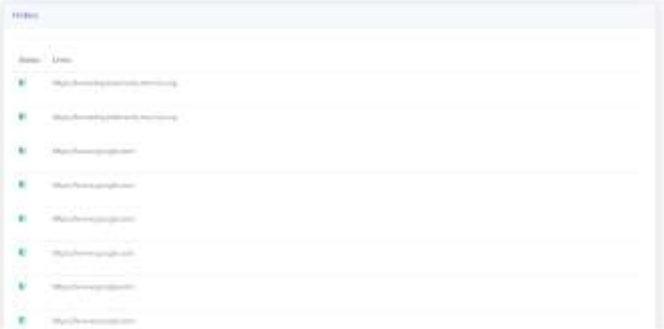
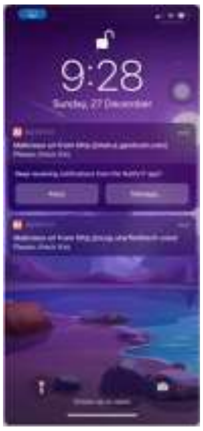


Fig. 10 Netbits – Scan History Page

With the integration of Notify17 app, user of the computer that browsed the infected scan are notified. The system maps all available computers in the network with their IP addresses and PC names in the database. The system runs a ping request to each computer each minute to identify which user is currently active. If the source IP address from the VirusTotal report matches either one in the database, it will report to the user (Fig. 11).



9:28

Sunday, 27 December

Fig. 11 Netbits – User Notification

## VI. Conclusions

The system performed with similar success during tests conducted with two testers. Though the system functioned well, several minor adjustments were made based on the feedback from testers. Alert or confirmation on the delete row was added in the *Allow List* page and icon status for the action button that did not update when clicked by the testers was rectified. The requirements as planned were successfully developed. More enhancements may be required before rolling out to the public, however, Netbits has the potential to become an open-source tool for everyone as it has its own functionalities and features that differentiate it from well-known tools such as Snort and Suricata.

The number of malicious network incidences is high. People are worried and feel unsafe when using technology. Netbits can be incorporated in the network of big companies or private home. The project is significant in:

1) *Protecting people from clicking malicious links*: The system provides another layer of security in a network. It protects users' personal data and credentials from being harvested by people outside the network. Additionally, it prevents users or companies from getting infected by the dangerous "ransomware".

2) *Alerting the people in the network about malicious links*: Users in the network are notified via their mobile phones if there are malicious activities happening in his/her surfing activity. The system identifies where the traffics are coming from and the destination address.

*3) Providing a cost-effective solution for securing a network*: Netbits aims for the public to use the system with minimal hardware requirement and skills needed to set up the system. It only requires one hardware which is Raspberry Pi as a platform for the system to run and filter network traffics.

## References

[1] A. A. Orunsolu, A. S. Sodiya & A. T. Akinwalre, "A predictive model for phishing detection." *Journal of King Saud University - Computer and Information Sciences*, vol. *34*(2), pp. 232–247, 2022.

[2] J. Vijayan. (2022) Dark Reading. [Online]. Available: https://www.darkreading.com/attacks-breaches/more-orgs-experienced-a-successful-phishing-attack-in-2021-than-year-before

[3] M. Korolov. (2016) CSO. [Online]. Available: https://www.csoonline.com/article/3154714/ransomware-took-in-1-billion-in-2016-improved-defenses-may-not-be-enough-to-stem-the-tide.html

[4] M. Jalali, M. Bruckes, D. Westmattelmann & G. Schewe, "Why Employees (Still) Click on Phishing Links: Investigation in Hospitals." *Journal of Medical Internet Research*, vol. 22(1), 2020.

[5] S. Fadilpašić. (2019) ITProPortal. [Online]. Available: https://www.itproportal.com/news/nearly-half-of-workers-have-clicked-on-a-phishing-link/

[6] J. Yuan, Y. Liu, and L. Yu, "A Novel Approach for Malicious URL Detection Based on the Joint Model," *Security and Communication Networks*, vol. 2021, p. 4917016, 2021.

[7] S. A. Sokolov, T. B. Iliev, and I. S. Stoyanov, "Analysis of Cybersecurity Threats in Cloud Applications Using Deep Learning Techniques," in *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2019, pp. 441–446.

[8] F. Alsakran, G. Bendiab, S. Shiaeles, and N. Kolokotronis, "Intrusion Detection Systems for Smart Home IoT Devices: Experimental Comparison Study BT - Security in Computing and Communications," *Communications in Computer and Information Science*, vol 1208, pp. 87–98, 2020.

[9] Open Information Security Foundation (OISF) (2021) Suricata (Version 6.0.3). [Online]. Available: https://suricata-ids.org

[10] Cisco (2021) Snort (Version 3.0). [Online]. Available: https://www.snort.org

[11] Fail2ban. (2019) Anti Brute Forcing System (Version 0.11.2) [Online]. Available: https://www.fail2ban.org/wiki/index.php/Main_Page

[12] S. Tripathi and R. Kumar, "Raspberry Pi as an Intrusion Detection System, a Honeypot and a Packet Analyzer," in *2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS)*, 2018, pp. 80–85.

[13] Google Chronical Security (2022) VirusTotal. [Online]. Available: https://www.virustotal.com/