Internet Usage Control System using IoT

[1]Priyansh Bajaj, [2]Shivani Rohilla

SRM Institute of Science and Technology
Ghaziabad – 201204, India

***Abstract*- This paper proposes an automated system for internet usage control within organizations. With internet-of-things emerging, there's a growing need for a dedicated system that can optimize access control of these devices. The system utilizes Raspberry Pi as a DNS server to manage internet access in a network. The aim is to develop the system with flexible template-based configurations, a user-friendly interface for efficient administration for internet usage control. Since the proposed system is implemented using Raspberry Pi, the system is cost-effective, quick to implement and secure. The expected results are an efficient and secure system for internet usage control.**

***Keywords*- Internet Usage Control, Internet of Things, Raspberry Pi, Security, DNS Server.**

I. INTRODUCTION

In today's digital world, where all our devices are networked and have internet access, there is an emerging need to regulate internet access throughout an organization. In critical situations, this can prove useful as the attack surface of the organization is reduced when limiting connections to remote systems. This project aims to provide flexibility to dynamically change configurations in real-time, acting as an automated solution to restrict access to specific websites organization-wide. It promotes the idea of responsible usage as, in case of incident response, the cert team can get access logs in real time, thus complying with the regulations more efficiently. For the demonstration of this paper, a Raspberry Pi is used due to its compact form factor, cost-effectiveness and respectable processing power.

Many organizations provide wireless network access to their employees, some also follow BYOD policy, which opens another door that can affect their day-to-day operations and the reliability of their management. In order for system administrators to manage the network with more flexibility, this system can be configured in the existing network infrastructure, which will limit web domains to restrict access to any objectionable content, thus regulating the internet usage of the users in a very effective manner. Making a system that works with the current existing system of the as-is, this program can be configured in the existing network infrastructure, enabling small businesses to implement the solution with minimal changes. Thus increasing the reliability and also meeting the organization needs by blocking any malicious web domains to restrict access to any objectionable content, thus regulating internet usage of users effectively.

For more efficient management, a web application is designed to manage the DNS server deployed in the network. This web application provides a user interface for network administrators to enable and disable specific configurations pre-loaded in the system. Additionally, this interface will also display important system information that can be useful for debugging any unknown issues that might arise in the future. With new organizations emerging, this system can be an extremely beneficial addition to an organization's infrastructure, adding another layer of security to protect digital assets from hackers who aim to alter them for unethical advantage.

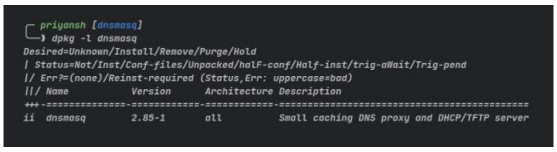
II. BACKGROUND

Raspberry Pi is a low-cost, Linux-based device that can be used for managing internet access and increasing security in organizations. Its compact form factor, availability of ports and respectable performance make it an ideal choice for this work. In this work, Raspberry Pi 4, 4gb variant is used. Below, figure 1 shows architecture of the device.

Figure 1 Raspberry Pi 4 Board

Ideally, Raspberry Pi can be configured with ARM OS, which in this case is Raspberry Pi OS Lite 64 bit which is an operating system based on Debian with minimal programs bundled. Like a standard Linux server, this raspberry Pi can be used to host a web application or run a DHCP server, or paired with a keyboard and mouse for better accessibility. The web interface hosted be this Raspberry Pi can be accessed by browsers in the same network. However, it's important to note that while Raspberry Pi can be a powerful tool for network management, it does have limitations. Like certain security measures should be taken to protect the Raspberry Pi from attacks. Regular updates and backups should be taken so that the system is more reliable.

DNSMASQ is an open source utility for Linux which can be used to create a DHCP server which can listen and respond to DNS requests. It is lightweight and easy to set-up, many Linux distributions and Android use this in their core system to speed up the resolution of DNS names, thus improving the overall network performance.



```
╭─ priyansh [dnsmasq]
╰─❯ dpkg -l dnsmasq
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-aWait/Trig-pend
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||/ Name           Version      Architecture Description
+++-==============-============-============-=================================
ii  dnsmasq        2.85-1       all          Small caching DNS proxy and DHCP/TFTP server
```

Figure 2 DNSMASQ Program

Above, figure 2 shows it configured on Raspberry Pi to later give the flexibility to make the router point the DNS requests to this DNSMASQ's server, this particularly is the arrangement that is used to have control over internet access network-wide. Configurations saved by web user interface can be applied by converting them to configurations compatible to this tool.

Apache Server is also open source cross-platform software, released under terms of the Apache License. Apache is developed and maintained by an open source developers of the Apache Software Foundation. As of February 2023, over 45\% of all web servers use it to host their web applications. Apache allows hosting both static and dynamic websites, which in this case will be dynamic and written in PHP. Therefore, it is used in this paper to develop the intranet web application which will be user interface for network administrators to interact with the Raspberry Pi. Good documentation of Apache allows reliable development of the application.

Integration of all these tool-set can be used to address the problem of internet usage control in an organization network. In the current as-is network configuration, router also provides its own firewall rules, these rules aren't equally intuitive as they don't allow wildcard(*) and other regex based pattern matching for domain names. Having Raspberry Pi set-up as a DNS server using DNSMASQ and running Apache Web Server for the user interface is the approach that will be used. Since the internet is always changing, such systems should have up-to-date definitions of list of offending or malicious domains. In order to protect the on-site infrastructure and block objectionable content, this system can effectively reduce the security risk which helps troubleshooting such incidents.

III. Proposed System

The solution involves with the development of an automatically managed internet usage control system that is autonomously updated with definitions of new threats on a regular basis. Using the internet-of-things technology, a solution is suggested which can help keep an eye on blocking and limiting access to malicious content. Regulating employee's internet usage and the time they spend online as an additional plus with this solution, since a flexible configuration is available in the system, implementing the solution in existing infrastructure is easy automating and assisting the organization to keep their employees safer in an efficient manner with the help of automation. The architecture of network to build the system which involves the Raspberry Pi Configured as the DNS server of the router. So, for any domain queries the Raspberry Pi can answer according to rule-set configured in dnsmasq's configuration. Below figure 3 shows a high level architecture to set-up the system in the network.

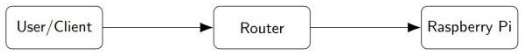


Figure 3 High Level Network Architecture

This network architecture involves building an automated system using a scripting language like python to automate the configuration generation in the Raspberry Pi. Since the router is set-up to forward all DNS requests to the Raspberry Pi by use of built-in LAN configuration function of the router. The system proposes any DNS requests made by User/Client to be answered by Router with the help of Raspberry Pi. In case of there being multiple clients or switches in the network, the main router is the one which uses the DHCP server to assigns all the clients IP addresses from the pool. This allows this configuration to be flexible and work even in the case of multiple repeaters being used.

The user interface for the system is proposed in PHP, which will allow the Raspberry Pi to serve dynamic web pages to the administrator. Below figure 4 shows how a user, admin in this case can access the configuration user interface with the help of web browser which is served by Apache Web Server which fetches all our programs from the WWW/Project Directory. For authentication, a basic login mechanism is integrated using SQL to restrict login access from other users.

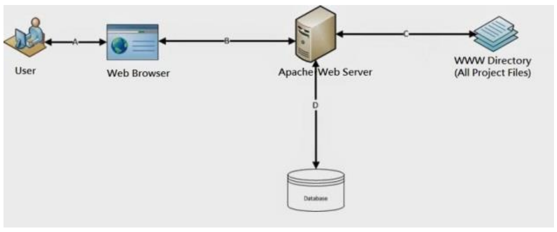


Figure 4 User Interface Architecture

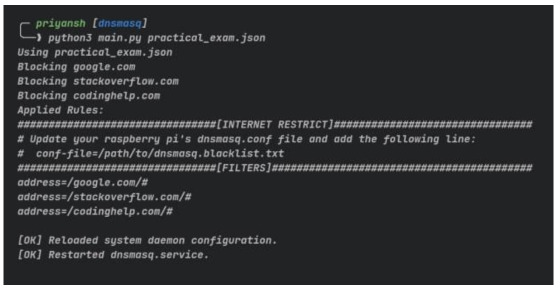
IV. IMPLEMENTATION

The methodology to implement the solution involves the use of Raspberry Pi which will be set-up inside the organization network through the help of Wi-Fi or Ethernet. Raspberry pi is then configured with Raspberry Pi OS Lite 64 bit as the operating system. With dependencies like Apache, dnsmasq, PHP, SQLite, python installed. The raspberry pi is connected to the Wi-Fi router and a connection is established. Writing code in python to load configuration, this can be executed in the existing system architecture to manage the internet usage in the organization. The raspberry Pi is configured to handle the request based on rules, for this dnsmasq, which is the DHCP server in this case needs to be configured. IPtables configuration and DNS query logging is implemented in raspberry Pi. Thus, improving the network performance, since these functions will limit the domain queries by DNS query events from the router.

The following methodology can be followed:

1) Use Raspberry Pi Imager to prepare an SD card for the Raspberry Pi by setting up Raspberry Pi OS Lite 64 bit.
2) Set up Wi-Fi router's LAN/WLAN configuration such that the Raspberry Pi gets a reserved static IP.
3) Configure the router configuration such that the primary DNS server points to the Raspberry Pi's static IP.
4) Install and configure all dependencies for this system on the Raspberry Pi.
5) Deploy the system in network and test it using a client.

When implementing the system, a network is first set-up in a way that whenever an employee or user in a network tries reaching a domain the query is resolved by the raspberry pi. The router in the main network is configured to resolve the DNS to a local intranet device, which is raspberry pi in this case, all devices in the network are configured so that raspberry pi has control over which requests to allow to resolve using the actual DNS provider. For handling these DNS requests use the python script to convert configurations to dnsmasq's compatible format. Below figure 5 shows how such python script can generate dnsmasq configurations using input JSON files and reload the system daemons to reload the configurations dynamically, for this sudoers file is configured to allow the core script for this action.



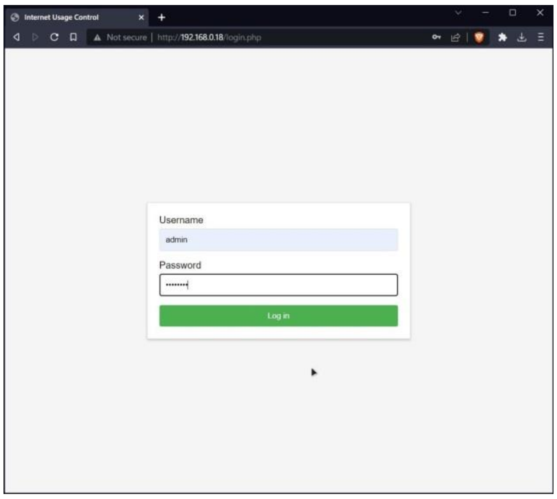
```
┌─ priyansh [dnsmasq]
└─❯ python3 main.py practical_exam.json
Using practical_exam.json
Blocking google.com
Blocking stackoverflow.com
Blocking codinghelp.com
Applied Rules:
################################[INTERNET RESTRICT]################################
# Update your raspberry pi's dnsmasq.conf file and add the following line:
#  conf-file=/path/to/dnsmasq.blacklist.txt
################################[FILTERS]##########################################
address=/google.com/#
address=/stackoverflow.com/#
address=/codinghelp.com/#

[OK] Reloaded system daemon configuration.
[OK] Restarted dnsmasq.service.
```

Figure 5 Core Script in Python

The internet usage control is managed by software that is running as a server on the raspberry pi, to restrict the access to websites this server has a live list of whitelisted and blacklisted domains which are allowed to be reached in the network. Any inappropriate or malicious domains can be blocked here. The logging of user activity can also be implemented here. An easy-to-configure file is created which can be reused in different conditions, employers or network administrators in a small business can also configure these and reuse it across different incidence response systems according to the requirements.

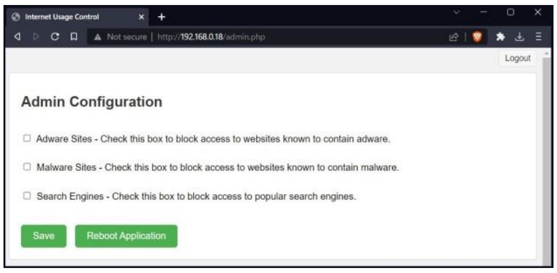
The user interface is implemented in HTML and CSS which is dynamically inserted into client's browser using PHP. For proper security, the first page the admin visits will be the login page which will ask for authentication before any actions can be performed. This page is optimized to run well on most screens which provides efficient user experience. Below figure 6 shows the login page implemented for the same.



Username
admin
Password
••••••••

Figure 6 GUI Login Page for Administrators

This login page takes user's credentials in a form and evaluates them by the help of SQLite which is a minimal engine implementation of SQL for primarily ARM based devices. SQLite is used as a library as the dependencies are installed from Raspberry Pi OS's built in repositories. Once the user is logged in a session id is generated to keep the user logged in for the session. This login page can be accessed by entering the IP of Raspberry Pi in browser, similar to how router's configuration pages are accessed. Since the user interface is implemented in HTML and CSS which is dynamically inserted into client's browser, a dashboard is created in similar way. For proper security, if the user isn't logged in and trying to browse this page is redirected back to login. When an admin visits this page after logging in, a dashboard is presented with multiple configuration toggles for the internet usage control server. This page is optimized to remember previous configurations made by admin. Below figure 7 shows this implementation.



Admin Configuration

☐ Adware Sites - Check this box to block access to websites known to contain adware.

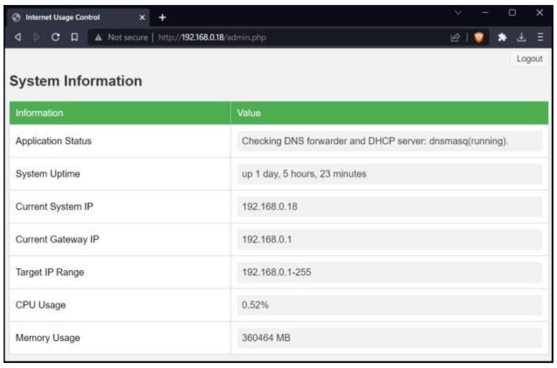
☐ Malware Sites - Check this box to block access to websites known to contain malware.

☐ Search Engines - Check this box to block access to popular search engines.

Figure 7 GUI Administrator Configurations Page

This configuration page takes user's configuration in a form and evaluates them in the back-end and creates a configuration in JSON format which is compatible by the core python script which was created. Once the save button is clicked a shell script updates the core configuration. The dnsmasq core code matches these configuration options with list of available configurations and dynamically creates a dnsmasq compatible configuration which is saved in /etc/dnsmasq.conf. The reboot application button can also be accessed by the administrator which can be used to reload the system and dnsmasq daemons. For this, sudoers file is modified to explicitly grant these privileges to the apache web server without changing ownership of any file.

In the admin dashboard, additionally an information section created in which all the fields are dynamically update. This enables the administrator to properly debug and track the status of this internet usage control server. Information like current IPs from the interfaces that the Raspberry Pi is configured on, the CPU and memory states of Raspberry Pi. Below figure 8 demonstrates the same.



System Information

Information	Value
Application Status	Checking DNS forwarder and DHCP server: dnsmasq(running).
System Uptime	up 1 day, 5 hours, 23 minutes
Current System IP	192.168.0.18
Current Gateway IP	192.168.0.1
Target IP Range	192.168.0.1-255
CPU Usage	0.52%
Memory Usage	360464 MB

Figure 8 GUI System Information Page

This system information dashboard also provide application status which monitors dnsmasq's service daemon to get the current status of the application. This is configured by running a bash script through the PHP's shell function which allows getting real-time information on the dashboard page.

During implementation, all the resources like configuration that need to be written are protected by selinux so proper ownership needs to be defined for all the files that will be created by the apache application. This can be done using chown command and passing www-data as the owner. Additionally, all the shell scripts must be executable, this can be done using the chmod +x command, the resources that need to be accessed by web server need to have 755 permission for proper access. For debugging any errors during the deployment, apache logs at /var/log/apache2/error.log will be a good place to resolve them efficiently.

V. CONCLUSION

In this age of internet of things, where online assets have become a dependency for our economy it's important to have more control over our devices and their communication with remote systems on the web. Internet Usage Control System, when deployed can provide more control over the domains that are accessible network-wide. Implementing Raspberry Pi with such capability adds another layer to the protection of organization from the malicious actors.

In this paper, a Raspberry Pi is implemented as a DNS server to control and restrict access to systems according to the definition of configurations by the administrator. In future, features of this system can be extended by chaining the logs it produce with a machine learning model to detect abnormality in usual DNS patterns in a network. The overall security of the dashboard can be improved by implementing https based access with multi-factor authentication. The configuration templates can be standardized so they can be shared in community to provide easier future deployments. More features in the core system can be implemented like MAC-based access restrictions for better internet usage control in the future.

REFERENCES:

1 J. Schulz-Zander, C. Mayer, B. Ciobotaru, S. Schmid, and A. Feldmann, ‘OpenSDWN: Programmatic Control over Home and Enterprise WiFi’, in *Proceedings of the 1st ACM SIGCOMM Symposium on Software Defined Networking Research*, Santa Clara, California, 2015.

2 M. Coşar and S. Karasartova, "A firewall application on SOHO networks with Raspberry Pi and snort," *2017 International Conference on CSE (UBMK),* Antalya, Turkey, 2017, pp. 1000-1003, doi: 10.1109/UBMK.2017.8093414.

3 A. Mat Taib, ‘Securing network using raspberry Pi by implementing VPN, Pi-hole, and IPS (VPiSec)’, *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, pp. 457–464, 06 2020.

4 S. Ravidas, A. Lekidis, F. Paci, and N. Zannone, ‘Access control in Internet-of-Things: A survey’, *Journal of Network and Computer Applications*, vol. 144, pp. 79–101, 2019.

5 M. Catherine and A. Kumawat, ‘Securing a Small Network by using Raspberry Pi Honeypot’, 2020.

6 O. Karahan and B. Kaya , "Raspberry Pi Firewall and Intrusion Detection System", *Journal of Intelligent Systems: Theory and Applications*, vol. 3, no. 2, pp. 21-24, Sep. 2020, doi:10.38016/jista.653486.

7 C. Sheth and R. Thakker, "Performance Evaluation and Comparative Analysis of Network Firewalls," *2011 International Conference on Devices and Communications (ICDeCom)*, Mesra, India, 2011, pp. 1-5, doi: 10.1109/ICDECOM.2011.5738566.

8 P. Yadav, V. Safronov, and R. Mortier, ‘Enforcing Accountability in Smart Built-in IoT Environment Using MUD’, in *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, New York, NY, USA, 2019, pp. 368–369.

9 J. Melzer, J. Latour, M. Richardson, A. Ali and W. Almuhtadi, "Network Approaches to Improving Consumer IoT Security," *2020 IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, NV, USA, 2020, pp. 1-6, doi: 10.1109/ICCE46568.2020.9043121.