International Journal of Engineering & Technology, 7 (2.5) (2018) 58-61

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research Paper

Ads Block Management System Using Open Virtual Private Network on Ubuntu Operating System

Edy Rahman Syahputra[1*], Boni Oktaviana Sembing[2], Arie Rafika Dewi[1], H Hasdiana[1], Halim Maulana[1]

[1]Deparment of Information System, Universitas Harapan Medan, Medan, Indonesia
[2]Department of Informatics, Universitas Harapan Medan, Medan, Indonesia
**Corresponding author E-mail: ydeaja@yahoo.com*

Abstract

In internet, an ad is something that able to be avoided when opening a web page. At this time many internet based businesses rely on advertising for their income. However, users are dissatisfied with the presence of ads on websites they visit, since the size of the ad data is often proportional to the actual content ads. This impacts not only on loading web pages and may lead to a file that may potentially be a carrier of the virus and may damage the operating system of the visitor. A system that can overcome the annoying ads will be needed by the user for the convenience of users in using the internet media. To overcome this will be built a system using the pi-hole and Ubuntu operating system

Keywords: *OpenVpn, Ubuntu, Pi-Hole, Dns, Adevertisement*

1. Introduction

Online advertising is not only very annoying, but also a potential source of malware[1]–[6] in digital devices such as smartphones, tablets, laptops or computers. While there are plugins designed to block ads based on per-app or per device, stopping ads at the DNS level provides a much more complete solution across all your apps and devices at once. Pi-hole - the DNS server originally created for use on Raspberry Pi single board computers - filters the request (demmand) to the ad serving domain, blocks ads and improves network performance[7]. With Pi-hole, you can actively monitor every DNS request performed on your network and block requests by phone. This functionality also goes beyond web browser, allowing you to filter ads in other apps by targeting appropriate DNS queries. Pi-hole is very effective when you use it in conjunction with Virtual Private Network (VPN). The VPN builds and maintains connections through the tunnel, which is a logical network connection between the client and the server. In addition, if your VPN supports Secure Socket Layer (SSL), all transactions are encrypted, providing secure links for data transmission. In this study, the authors will install and configure OpenVPN and Pi-hole to act as private, network-based, DNS-based ad storage filters for all devices connected to the Internet. So with the existence of this blocking system of the site, users can feel more secure from the ads that can potentially as a virus and can damage the system.

2. Literature Review

A. OSI MODEL

OSI open network reference model or OSI Reference Model for open networking is a network architectural model developed by the International Organization for Standardization (ISO) in Europe in 1977. OSI itself stands for Open System Interconnection. This model is also called the "OSI seven layer model" model (OSI seven layer model)[8]–[10].

B. OPERATING SYSTEM

Operating system (English: operating system) is a system software that manages the resources of hardware and software, and as a daemon for computer programs[11]. Without the operating system, users can not run application programs on their computers, except boot programs [12], [13]. The operating system has systematic scheduling including calculation of memory usage, data processing, data storage, and other resources. For hardware functions such as input and output and memory allocation, the operating system acts as an intermediary between the application program and the computer hardware, even though the application code is usually executed directly by the hardware and will often contact the OS or disconnect by that [14], [15]. The operating system is found on almost any device that contains computers-from mobile phones and video game consoles to supercomputers and web servers. Examples of modern operating systems are Linux, Android, iOS, Mac OS X, and Microsoft Windows.

C. UBUNTU

Ubuntu is one of the Debian-based distributions of Linux and is distributed as free software. The name Ubuntu comes from a philosophy of South Africa which means "humanity to others" [16]. Ubuntu is designed for personal use, but the Ubuntu server version is also available, and has been widely use [17]. The official Ubuntu project is sponsored by Canonical Ltd. which is a company owned by South African businessman Mark Shuttleworth. The purpose of the Ubuntu Linux distribution is to bring the spirit embodied in the Ubuntu philosophy into the software world. Ubuntu is a complete Linux-based operating system, freely available, and has good support from community and professional experts.

D. OPENVPN

OpenVPN is an open-source software released in 2002. It has become a popular choice as it can easily traverse wireless access points, firewalls, NAT-based routers, and HTTP proxy servers. OpenVPN allows you to use different ports to cut both firewalls and throttling. Remote internet routing has mainly minimal impact on the transfer speed with this protocol. Up to 256 bits, OpenVPN has strong encryption from PPTP, regardless of your VPN service. Some argue that with PPTP, it is possible for someone to retrieve your password when connecting, but the likelihood of it happening is impossible. So OpenVPN has a bit more to offer those who prioritize maximum security [18], [19].

E. DOMAIN NAME SYSTEM

Domain name system (DNS) is a primary identification mechanism for Internet applications. However, DNS resolutions often take an unbearably long time, and this could seriously impair the consistency of the service quality of Internet applications based on DNS such as World Wide Web. Several approaches reduce DNS resolution time by proactively refreshing expired cached records or prefetching available records beforehand, but these approaches have an inherent problem in that they cause additional DNS traffic. In this paper, we propose a DNS resolution time reduction scheme, named renewal using piggyback (RUP), which refreshes expired cached records by piggybacking them onto solicited DNS queries instead of by issuing additional DNS queries. This method decreases both DNS resolution time and DNS traffic since it reduces the number of queries generated to handle a given DNS resolution without generating additional DNS messages. Simulation results based on two large independent DNS traces show that our proposed approach much reduces not only the DNS resolution time but also the DNS traffic[20], [21].

F. BUSINESS OF DNS

As the Internet develops into multiple knowledge repositories, social networks, e-businesses, virtual educational institutions and a myriad of other tools for personal, business, and educational use, DNS issues must be contended with, for example, those related to the global expansion of the Internet as illustrated by the global IP address distribution, as well as the numerous new issues that continue to surface. As the Internet expands, becoming integrated into our daily lives and increasingly more critical to the livelihood of organizations and individuals, so it becomes not only a tool or a resource but a business in itself. As a result, networks are becoming increasingly complex, requiring the multiple IP resources being used by organizations to be managed. This critical need is being met by tools such as IP address management (IPAM) software, and by registrars like Oversee.net providing services beyond the simple purchase of a domain name. For example, Oversee.net offers brokerage services for the buying and selling of domain names, comparing their service to that of the brick-and-mortar real estate brokering that has been taking place for centuries. Much like a sophisticated advertising firm, Oversee.net also assists with attracting customers to websites through their "monetizing direct navigation traffic" services. Even those who wish to build their e-business on the Internet itself can do so with Oversee.net's Emerging Business Division [22], [23].

G. TUNNELING

DNS Tunneling is a method of cyber-attack that encodes the data of other programs or protocols in DNS queries and responses. DNS tunneling often includes data payloads that can be added to an attacked DNS server and used to control a remote server and applications. Typically, DNS tunneling requires the compromised system to have external network connectivity, as DNS tunneling requires access to an internal DNS server with network access [24]. Hackers must also control a domain and a server that can act as an authoritative server in order to execute the server-side tunneling and data payload executable programs. A 2016 Infoblox Security Assessment Report found that 40 percent early half of files tested by Infoblox show evidence of DNS tunneling. Cybercriminals know that DNS is a well-established and trusted protocol, and have figured out that many organizations do not examine their DNS traffic for malicious activity. DNS tunneling enables these cybercriminals to insert malware or pass stolen information into DNS queries, creating a covert communication channel that bypasses most firewalls. While there are quasi-legitimate uses of DNS tunneling, many instances of tunneling are malicious. There are also several off-the-shelf tunneling toolkits readily available on the Internet, so hackers don't always need technical sophistication to mount DNS tunneling attacks. At the same time, DNS tunneling is often part of very sophisticated attacks, including those sponsored or directly managed by nation states [25].

3. Methods

In this phase describe approach for identifying users from passive traces. The output is that ad-blocker users access fewer ads than non-ad-blocker users. Thus, we need a methodology for identifying Web objects and separating them into ad or non-ad objects. This is where we rely on Pi-Hole as we see in figure 1.

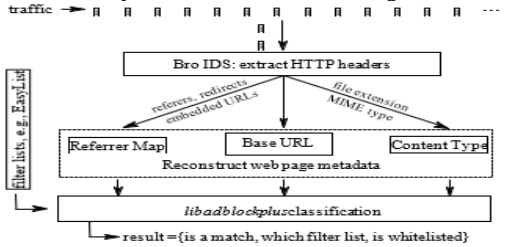

Fig.1: Methodology Approach to classify ad requests

The purpose of our methodology is to classify ad-traffic in header traces captured in HTTP/TCP via passive measurements. For this classification we rely on *pi-Hole.* This allows us to classify Web objects from the traces in an off-line fashion without the need to operate a full browser. A summary of the classification methodology is shown in Figure 1. To identify if an end user has requests than a browser without any ad-blocker. Which is, a low installed an pi-Hole System we use a following observation such as: a browser with such an extension should issue less ad number of ad requests is a strong indicator for the presence of an ad-blocker. Although, we have to identify all requests of a particular end user and compute the ratio of ad requests. The ratio for an end user depends on *i)* the browser configuration, i.e., whether there is an ad-blocker installed and, if so which ad-blocker with which configuration, and *ii)* which sites the user visits. Given that many of the most popular sites do indeed have extraneous content.

4. Results and Discussion

When both OpenVPN and Pi-hole really prepared and work together, each DNS request that is done on your network are routed to the Pi-hole, which will then check to see if the requested domain matches the domain of others both in the block list or black list in this case, the filter will drop the domain at all; If not, the filter will allow the domain to pass through. Although the Pi hole has not been configured to work with OpenVPN, you can still verify the current installation by testing the ability of the Pi-hole to filter ad serving domains directly from your server. To perform the test, use the host command to perform a DNS lookup on google.com, specify the default gateway, name servers, as 10.8.0.1 for query. This will cause the request to go through Pi-filter hole.

host google.com 10.8.0.1

Because the output includes the IP address of the public domain, you know that google.com does not match the domain anywhere in the block list or black list. Pi-thread holes with a set of default block list that is maintained by a team of application development pi-hole. However this list alone is not always used. Ideally, the block must adjust the block list to suit custom applications used. We can arrange the block list and even more with the web interface admin Pi-hole. To manage Pi-hole through its web interface, you must be connected to a network of OpenVPN in advance. Once connected, navigate your web browser to the default home

page in the web interface http://10.8.0.1/admin. Next will be greeted by a page that reports the number of Queries that were blocked last 24 hours, the number of Queries last 24 Hours, the percentage of power that Blocked the last 24 hours, and the number of domains on Blocklists. You will also see a graph Query during the last 24 hours, the Status indicator Pi-hole, and the navigation options to display the Dashboard, log in, and donations at PayPal.

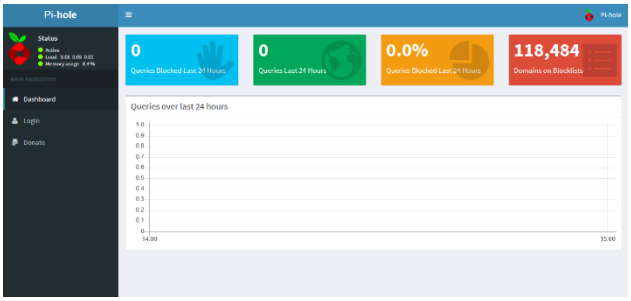

Fig.2: Main Page of Pi-Hole

Click Login to access the full interface. When prompted, enter the password that you received in the installation screen Pi-hole late in the previous step. After you sign in, the General layout of the interface will remain the same, but now will include more of the menu options on the left side of the screen and additional widgets for this type of Query from the time and purpose of Forward from time to time.

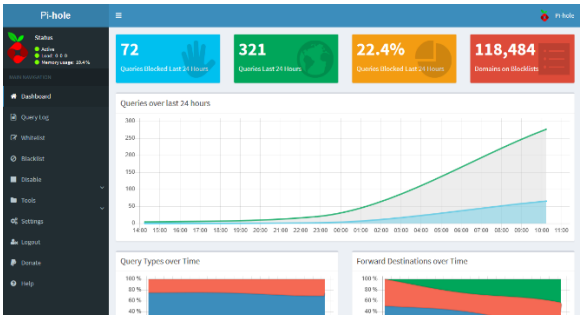

Fig.3: Dashboard Admin Page

Before adding a list of additional blocks to Pi-hole, you must first update the block list data from Project Manager, because it may just be the latest updates include some or all of the data sources that you will add it manually. On the left side of the screen, click Tools to extend the navigation menu and then select Update Lists. On the next screen, click the blue List in the middle of the screen to retrieve the last version of the official bloc sources list.

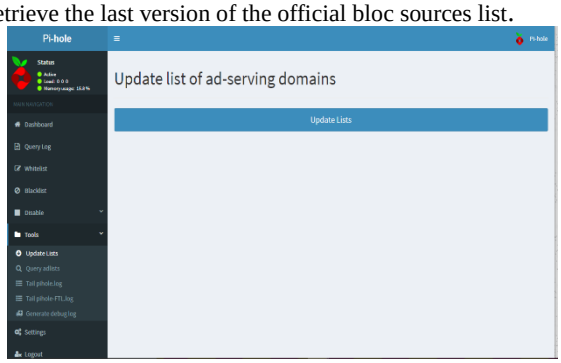

Fig.4: Updates the List of Domains

As hole-Pi an update, it will show the sources it draws data from the list, whether the source has been modified since your last update, and whether there is any data that you import into your installation. Once completed, the green bar at the top of the screen will be unreadable.

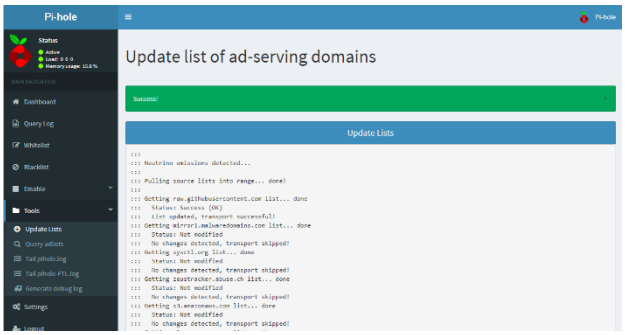

Fig.5: Update List Serving Domain

The official block list data is updated, you are ready to add your own additional block list. Click Settings on the navigation menu on the left side of the screen to the main configuration options Pi-hole. On the next screen, click the + symbol in the box labeled Pi-Hole's Block Lists to see the current list of data blocks.

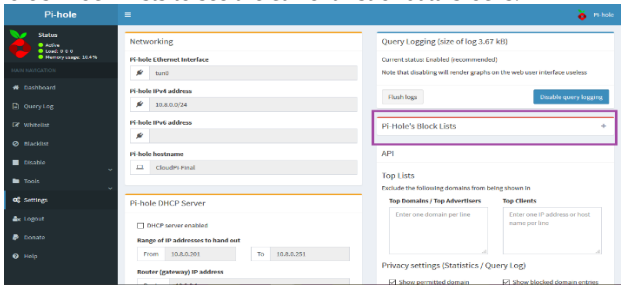

Fig.6: List of Block List

To add a new list to your installation, enter the URL in the input column list on the source at the bottom of the pane, and then press the Save button to save and update the addition and restart the function List update. It will automatically pull data associated with a list of the source block. For additional block list is broken down into several categories, such as Suspicious List, list, and list of Advertising Tracking Telemetry &, see Collection Big Blocklist.

5. Conclusion

Built system has a simple but effective way to filter DNS queries on the network Internet, but keep in mind that it may need to tweak the block list to suit personal browsing habits because the customs of each individual different browsing. To further enhance the security of your network, find out how to enable DNCrypt in Pi-hole installation your current intranet to make private and secure

References

[1] R. Rahim, "Man-in-the-middle-attack prevention using interlock protocol method," *ARPN J. Eng. Appl. Sci.*, vol. 12, no. 22, pp. 6483–6487, 2017.

[2] A. Putera, U. Siahaan, and R. Rahim, "Dynamic Key Matrix of Hill Cipher Using Genetic Algorithm," *Int. J. Secur. Its Appl.*, vol. 10, no. 8, pp. 173–180, Aug. 2016.

[3] R. Rahim, M. Dahria, M. Syahril, and B. Anwar, "Combination of the Blowfish and Lempel-Ziv-Welch algorithms for text compression," *World Trans. Eng. Technol. Educ.*, vol. 15, no. 3, pp. 292–297, 2017.

[4] R. Rahim, D. Hartama, H. Nurdiyanto, A. S. Ahmar, D. Abdullah, and D. Napitupulu, "Keylogger Application to Monitoring Users Activity with Exact String Matching Algorithm," *J. Phys. Conf. Ser.*, vol. 954, no. 1, p. 012008, 2018.

[5] H. Nurdiyanto and R. Rahim, "Enhanced pixel value differencing steganography with government standard algorithm," in *2017 3rd International Conference on Science in Information Technology (ICSITech)*, 2017, pp. 366–371.

[6] H. Nurdiyanto, R. Rahim, and N. Wulan, "Symmetric Stream Cipher using Triple Transposition Key Method and Base64 Algorithm for Security Improvement," *J. Phys. Conf. Ser.*, vol. 930, no. 1, p. 012005, Dec. 2017.

[7] T. Listyorini and R. Rahim, "A prototype fire detection implemented using the Internet of Things and fuzzy logic," *World Trans. Eng. Technol. Educ.*, vol. 16, no. 1, pp. 42–46, 2018.
[8] M. Kayri, "a Proposed ' Osi Based ' Network," vol. 2, no. 3, pp. 12–18, 2010.
[9] G. Bora, S. Bora, S. Singh, and S. M. Arsalan, "OSI Reference Model Networking : An Overview," *Int. J. Comput. Trends Technol.*, vol. 7, no. 4, pp. 214–218, 2014.
[10] S. S. Ceballos and J. A. L. Leyva, "An optimized collaborative platform for educational cloud computing in application and presentation layers of OSI model," in *2015 International Conference on Computing Systems and Telematics (ICCSAT)*, 2015, pp. 1–5.
[11] R. Rahim *et al.*, "Block Architecture Problem with Depth First Search Solution and Its Application," *J. Phys. Conf. Ser.*, vol. 954, no. 1, p. 012006, 2018.
[12] C. Sabri, L. Kriaa, and S. L. Azzouz, "Comparison of IoT Constrained Devices Operating Systems: A Survey," in *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*, 2017, pp. 369–375.
[13] E. Baccelli *et al.*, "RIOT: an Open Source Operating System for Low-end Embedded Devices in the IoT," *IEEE Internet Things J.*, p. 1, 2018.
[14] H. Narayanan, V. Radhakrishnan, Shiju-Sathyadevan, and J. Poroor, "Architectural design for a secure Linux operating system," in *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, 2017, pp. 949–953.
[15] R. Ratnadewi, E. M. Sartika, R. Rahim, B. Anwar, M. Syahril, and H. Winata, "Crossing Rivers Problem Solution with Breadth-First Search Approach," in *IOP Conference Series: Materials Science and Engineering*, 2018, vol. 288, no. 1.
[16] S. D. Patel and T. M. Pattewar, "Static Detection of Shared Object Loadings on Linux (Ubuntu 14.10)," in *2015 International Conference on Computing Communication Control and Automation*, 2015, pp. 589–593.
[17] V. Boisselle and B. Adams, "The impact of cross-distribution bug duplicates, empirical study on Debian and Ubuntu," in *2015 IEEE 15th International Working Conference on Source Code Analysis and Manipulation (SCAM)*, 2015, pp. 131–140.
[18] D. Meng, "Implementation of a host-to-host VPN based on UDP tunnel and OpenVPN Tap interface in Java and its performance analysis," in *2013 8th International Conference on Computer Science & Education*, 2013, pp. 940–943.
[19] R. M. Pandurang and D. C. Karia, "Performance measurement of WEP and WPA2 on WLAN using OpenVPN," in *2015 International Conference on Nascent Technologies in the Engineering Field (ICNTE)*, 2015, pp. 1–4.
[20] F. Zou, S. Zhang, B. Pei, L. Pan, L. Li, and J. Li, "Survey on Domain Name System Security," in *2016 IEEE First International Conference on Data Science in Cyberspace (DSC)*, 2016, pp. 602–607.
[21] M. Dooley and T. Rooney, "Introduction to the Domain Name System (DNS)," in *DNS Security Management*, Wiley-IEEE Press, 2017, p. 324.
[22] S. Harhalakis, N. Samaras, and V. Vitsas, "The effects of Active Queue Management and Explicit Congestion Notification on DNS traffic," in *2011 IEEE Symposium on Computers and Communications (ISCC)*, 2011, pp. 906–911.
[23] C. Huang, D. A. Maltz, J. Li, and A. Greenberg, "Public DNS system and Global Traffic Management," in *2011 Proceedings IEEE INFOCOM*, 2011, pp. 2615–2623.
[24] I. Coonjah, P. C. Catherine, and K. M. S. Soyjaudah, "Performance evaluation and analysis of layer 3 tunneling between OpenSSH and OpenVPN in a wide area network environment," in *2015 International Conference on Computing, Communication and Security (ICCCS)*, 2015, pp. 1–4.
[25] I. Coonjah, P. C. Catherine, and K. M. S. Soyjaudah, "Experimental performance comparison between TCP vs UDP tunnel using OpenVPN," in *2015 International Conference on Computing, Communication and Security (ICCCS)*, 2015, pp. 1–5.