Hindawi
Security and Communication Networks
Volume 2019, Article ID 7924690, 17 pages
https://doi.org/10.1155/2019/7924690

Research Article

VPN Traffic Detection in SSL-Protected Channel

Muhammad Zain ul Abideen, Shahzad Saleem, and Madiha Ejaz

School of Electrical Engineering and Computer Science (SEECS), National University of Sciences and Technology (NUST), Islamabad, Pakistan

Correspondence should be addressed to Muhammad Zain ul Abideen; mabideen.msis18seecs@seecs.edu.pk

Received 24 May 2019; Revised 4 September 2019; Accepted 16 September 2019; Published 29 October 2019

Guest Editor: Rajkumar Soundrapandiyan

In recent times, secure communication protocols over web such as HTTPS (Hypertext Transfer Protocol Secure) are being widely used instead of plain web communication protocols like HTTP (Hypertext Transfer Protocol). HTTPS provides end-to-end encryption between the user and service. Nowadays, organizations use network firewalls and/or intrusion detection and prevention systems (IDPS) to analyze the network traffic to detect and protect against attacks and vulnerabilities. Depending on the size of organization, these devices may differ in their capabilities. Simple network intrusion detection system (NIDS) and firewalls generally have no feature to inspect HTTPS or encrypted traffic, so they rely on unencrypted traffic to manage the encrypted payload of the network. Recent and powerful next-generation firewalls have Secure Sockets Layer (SSL) inspection feature which are expensive and may not be suitable for every organizations. A virtual private network (VPN) is a service which hides real traffic by creating SSL-protected channel between the user and server. Every Internet activity is then performed under the established SSL tunnel. The user inside the network with malicious intent or to hide his activity from the network security administration of the organization may use VPN services. Any VPN service may be used by users to bypass the filters or signatures applied on network security devices. These services may be the source of new virus or worm injected inside the network or a gateway to facilitate information leakage. In this paper, we have proposed a novel approach to detect VPN activity inside the network. The proposed system analyzes the communication between user and the server to analyze and extract features from network, transport, and application layer which are not encrypted and classify the incoming traffic as malicious, i.e., VPN traffic or standard traffic. Network traffic is analyzed and classified using DNS (Domain Name System) packets and HTTPS- (Hypertext Transfer Protocol Secure-) based traffic. Once traffic is classified, the connection based on the server's IP, TCP port connected, domain name, and server name inside the HTTPS connection is analyzed. This helps in verifying legitimate connection and flags the VPN-based traffic. We worked on top five freely available VPN services and analyzed their traffic patterns; the results show successful detection of the VPN activity performed by the user. We analyzed the activity of five users, using some sort of VPN service in their Internet activity, inside the network. Out of total 729 connections made by different users, 329 connections were classified as legitimate activity, marking 400 remaining connections as VPN-based connections. The proposed system is lightweight enough to keep minimal overhead, both in network and resource utilization and requires no specialized hardware.

1. Introduction

To enable the communication between the computers, TCP/IP stack was implemented. The stack was implemented without the consideration of security of information being transferred in the communication [1]. This issue raised a lot of security concerns which are constantly managed by different security services [2]. Secure Sockets Layer (SSL) is commonly used to provide authentication and encryption security service in TCP/IP stack [3].

The trend of encrypted traffic in the network has largely increased in the last decade due to security concerns in general and privacy concerns in specific [4]. The encryption has provided a lot of benefits for the user ensuring end-to-end secrecy and data confidentiality. The need to inspect the traffic originating or destined for the organization's network

has immensely increased for many security reasons. One of the reasons may be to simply validate parties involved in the communication [5].

Simple firewalls are generally not equipped with SSL inspection or off-loading which allows encrypted traffic to pass without any inspection [6]. This allows malicious traffic inside the network over covert channels that are not inspected by the firewall [7]. There is a dire need to detect legitimate and illegitimate traffic with minimal network overhead and overall system cost. This will allow any scale organization to better govern their organizational policies.

Virtual private network (VPN) service may be used to hide the real traffic in the network which may be otherwise not allowed or may be monitored [8]. A user using VPN service connects to a VPN server using normal Transport Layer Security (TLS) connection outside the network. Once connected, it requests the website or service from the server [9, 10]. The VPN server originates the request on behalf of the user to the server requested. The encrypted response is sent to the user on already established channel; as a result, the whole activity passes any filter on the network firewall.

Such techniques may be used by the users which aim to hide from or deceive the organization of their Internet activity [9]. This paper proposes a novel technique to detect VPN traffic inside a network. The proposed technique extracts the network traffic features and classifies the traffic to indicate if the traffic is legitimate or not. Key features are extracted from the network traffic and are compared against the already identified features of traffic found to be illegitimate or VPN traffic.

The system is also able to classify the traffic which is not following the pattern of normal traffic or normal user activity and flags that particular traffic stream to be invalid. We tested our system against five well-known freely available web-based VPN service providers; the proposed system was able to classify all of them correctly. More traffic-characterizing features may be added to identify more applications.

2. Related Work and Comparison

Multiple VPN services like TOR [11], Hotspot Shield, and other services have unique fingerprints, and not all the services can be distinguished using a similar criterion. Yamada et al. discussed a technique that uses statistical analysis on the encrypted traffic [12]. The scheme discussed, uses data size of network packets and performs timing analysis on the received packets to detect malicious traffic inside an encrypted channel. This technique is very useful for Web service providers to analyze the traffic coming to their servers and detect any malicious activity coming from outside the network.

A study on android-based applications which use VPN services [13] to show that these VPN services may use third-party trackers to track user behavior, and some may be used to bypass android sandbox environment. Once a malware or virus is delivered to the device inside the network, the whole network is vulnerable to attacks [14].

VPN clients inside the network act as a proxy, which connect to the respective VPN server. Once the connection is established, the VPN service provider is able to change or eavesdrop on the information and network traffic as required [15, 16]. This attracts many third-party advertisement or tracking entities [17, 18]. Any malicious entity can read, save, and/or modify our request and the related information to and from the destined service.

VPN services can change the data as they are in control of incoming and outgoing traffic from network to device. VPN services are also able to perform TLS interception [19] by using their own certificates which is trusted locally by the system, for VPN service to work properly. This leads to a more potentially risky situation when the device connected contains sensitive data [13, 20]. One of the countermeasures to this issue is certificate pinning [13, 21]. So, detecting such VPN services inside your network can save you from huge losses in terms of the information lost.

Goh et al. [22] proposes a man-in-the-middle approach to detect VPN traffic in the network. The article puts forward a solution that uses *secret-sharing* scheme which involves a massive key management overhead using public key infrastructure (PKI) technique. The paper assumes that the traffic coming to the system is unencrypted and the data are available in plain form for the system to analyze and detect VPN traffic. This is achieved by using application layer proxy which generates the copy of unencrypted traffic against each connection which is then sent to the system for further analysis. This technique approximately doubles the network traffic and computational resources of existing system while increasing the memory requirements to decrypt and re-encrypt the web traffic.

Another solution that uses *Deep Packet Inspection* technique [23] uses multiple sensors throughout the network to get the unencrypted traffic from the end hosts and send it back to snort-based IDS [24] to detect unusual behavior in traffic. It increases the overall network traffic because a sensor is to be installed on each network machine to be able to detect any unusual activity. Another technique is to copy the entire connection traffic and use preshared secret to analyze any malicious traffic [25].

To identify applications being run inside the network, network analysis is used extensively. The work discussed by He et al. [26] uses basic yet one of the most effective and used techniques in network traffic analysis for traffic classification. Based on *five-tuple connection classification*, the technique uses connection characteristics like packet size, their interarrival time, and the direction and order of the packets to identify the network signature of any android application. The scheme provides basic understanding of traffic classification. However, network traffic generated by web-based VPN services will have no major difference or identifying characteristics, different to a standard HTTPS connection.

The use of unencrypted traffic to manage, analyze, and categorize encrypted traffic is an exciting concept, discussed by Niu et al. [27]. The schemes use labelled *DNS-based data set* to identify malicious command and control traffic and label the traffic as suspicious or normal. The concept provides a unique prospective to analyze the network traffic beyond five-tuple/ current connection technique discussed

previously [26]. Table 1 provides basic attributes of already discussed techniques. The techniques discussed pave the path of our proposed scheme.

Our proposed system analyzes *DNS records* to identify malicious or illegitimate VPN server names. Connection features are extracted using *five-tuple approach*. Five-tuple approach classifies each new connection by five attributes listed below:

(i) Source IP
(ii) Destination IP
(iii) Protocol (TCP/UDP)
(iv) Source port
(v) Destination port

DNS-based traffic analysis and connection management were done using five-tuple techniques; our proposed system goes a step further to analyze *HTTPS handshake*. This is done to verify the server name used in the connection with the DNS activity which the user has generated by his network activity. Using this novel approach of managing a connection by using the activity preceding the current connection, we are able to detect and identify VPN traffic inside the network.

3. Forensic Analysis of VPN Services Client

To detect the network activity of VPN services, we carried out the forensic analysis of VPN services. For this purpose, we choose top five freely available web-based VPN services listed below:

(i) TOR browser
(ii) Hotspot shield free
(iii) Browsec VPN
(iv) ZenMate VPN
(v) Hoxx VPN

For each of these VPN services, we analyzed the network traffic, generated by their clients, installed on a user PC. The initial analysis was performed using Wireshark [28] and NetworkMiner [29]. Detailed analysis of each VPN service is discussed below.

3.1. Hotspot Shield. Hotspot shield [30] developed by AnchorFree is one of the leading free VPN services used. We tested its two versions:

(i) Client application for windows desktop
(ii) Firefox add-on

3.1.1. Client Application for Windows Desktop. In client version of the abovementioned VPN service, it was observed that once enabled, the service uses standard port 443 for HTTPS connections but generally connects to only one server. All the traffic may it be multisite traffic uses the same active connection. Figure 1 shows the connection details for current user activity against Hotspot Shield. Hotspot Shield uses fake well-known server name in SSL certificate to bypass the traffic from server name-based filters over the network, if any, as shown in Figure 2 below.

It can be seen that the used server name is *twitter.com*. It does not generate any DNS entry for such server name. The NetworkMiner tool shows us the connection details in Figure 3. We can see that eight unique connections were made; in this case, it generally means eight unique web pages were open. Requests of all these web pages were managed by the server whose IP is *136.0.99.219*. Certificate details can also be seen against this server IP which were received. Total 20,708 packets were sent in this activity, and *116,84* packets were received.

Figure 4 shows that no DNS activity for such host name was found during the communication. We can see all the DNS generated by the user while using Hotspot Shield client.

3.1.2. Firefox Add-On. Hotspot Shield in add-on uses standard https port along with standard DNS queries. The only way to detect Hotspot Shield inside the network is to identify the domain names used by Hotspot Shield. Shown below in Figure 5 is the network traffic generated by Hotspot Shield captured using Wireshark.

It can be seen in Figure 6 that the domain name is *ext-mi-ex-nl-ams-pr-p-1.northghost.com* for which the connection is established.

We observed that Hotspot Shield domain name consists of two main parts:

(i) Server identifier
(ii) Domain name

This can also be seen in certificate details in Figure 7, analyzed by NetworkMiner tool:

It is clearly observed that the domain name is $*.northghost.com$ and the other part is some server identifier as it may change once you reinitate the connection. It can be seen that the connections for Hotspot Shield were established against only one server with IP address *216.162.47.67*. Total connections established were 35, and a total of 207,08 packets were sent in this activity, and 11,684 packets were received.

The add-on also generates standard DNS activity as shown in Figure 8.

Changing the VPN locations from add-on's option has no effect on the server being connected by the client as the server identifier in the same activity does not change.

3.2. ZenMate. ZenMate [31] developed by ZenGuard is also very popular free VPN service used. We analyzed the chrome-based add-on of ZenMate. It uses standard https port along with standard DNS queries. The only way to detect ZenMate inside the network is to identify the domain names used by ZenMate VPN. Shown below in Figure 9 is the network traffic generated by ZenMate VPN captured using Wireshark.

It can be seen in Figure 10 that the domain name is *63.ayala-maroon.ga* for which the connection is established.

Table 1: Attributes of related techniques.

Research techniques	Strengths	Limitations
NIDS-based technique [22]	(1) Complete architecture to handle encrypted traffic-based intrusion detection (2) Protection against remote access and evasion techniques	(1) Multiple devices to be added in the network (2) Increased bandwidth inside the network due to traffic duplication
DNS-based technique [27]	(1) Introduces the concept of DNS scoring and analysis. Helpful in detecting malicious CNC based on DNS	(1) All CNC may not use only DNS based implementation
Connection-based technique [26]	(1) Five-tuple-based connection management. Helpful in identifying different protocol and application behavior	(1) Traffic generated by HTTPS based VPN will generally look like standard HTTPS streams

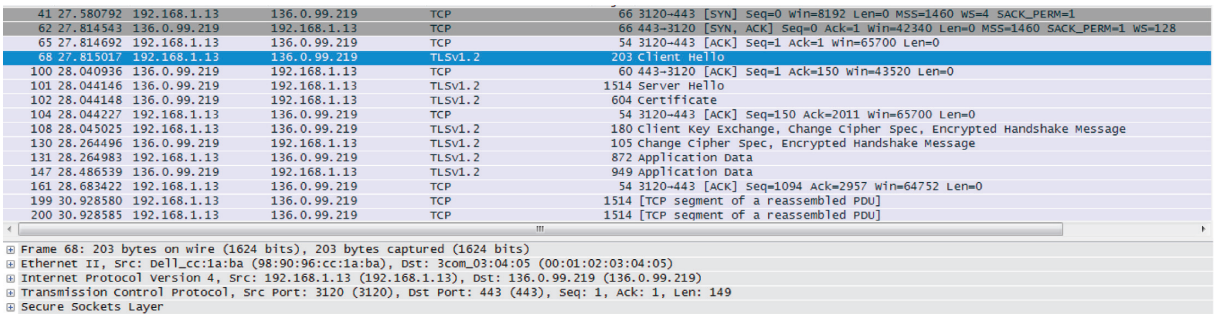

Figure 1: Wireshark: Hotspot Shield client.

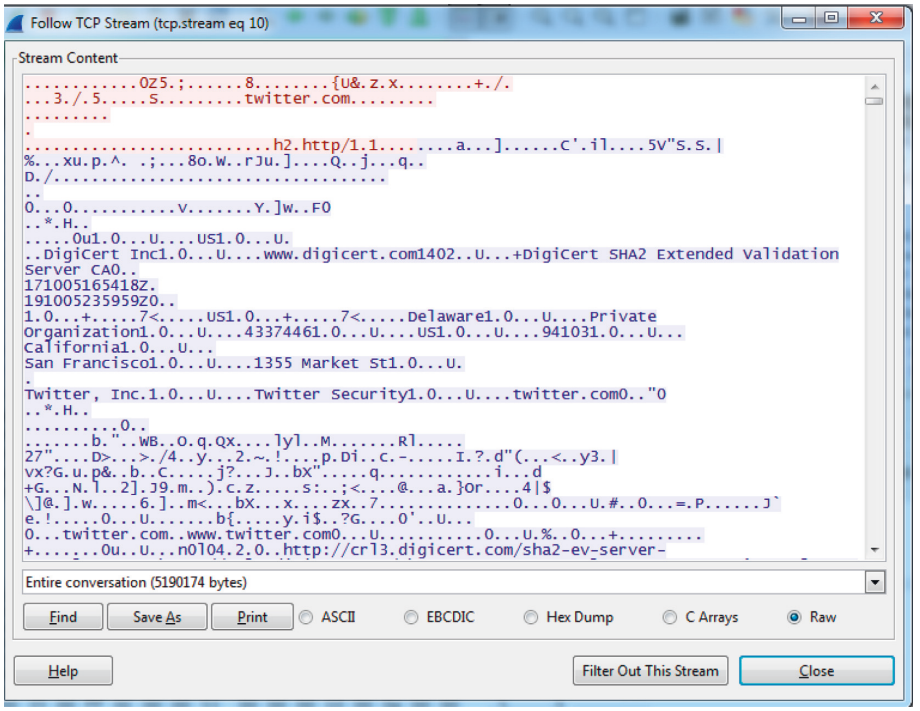

Figure 2: Wireshark: Hotspot Shield TCP stream.

Like Hotspot Shield, ZenMate's domain name also consists of two main parts:

(i) Server identifier

(ii) Domain name

This can also be seen in certificate details in Figure 11, analyzed by NetworkMiner tool:

It is clearly observed that the domain name is $*.ayala\text{-}maroon.ga$, and the number part is some server identifier. ZenMate is unique from other VPN services as it constantly

Figure 3: NetworkMiner: Hotspot Shield connection details.

Frame nr.	Timestamp	Client	Client Port	Server	Server Port	IP TTL	DNS TTL (time)	Transaction ID	Type	DNS Query	DNS Answer
3	2018-09-26 10:14:28 UTC	192....	49283	192....	53	61	00:01:00	0xD25A	0x0005 (CNAME)	geo.hotspotshield.com	us.hotspotshield.com
3	2018-09-26 10:14:28 UTC	192....	49283	192....	53	61	00:01:00	0xD25A	0x0001 (Host Address)	us.hotspotshield.com	74.115.0.53
32524	2018-09-26 10:15:31 UTC	192....	51671	192....	53	61	00:01:00	0x0337	0x0001 (Host Address)	djh90c9110sn8.cloudfront.net	143.204.208.138
32524	2018-09-26 10:15:31 UTC	192....	51671	192....	53	61	00:01:00	0x0337	0x0001 (Host Address)	djh90c9110sn8.cloudfront.net	143.204.208.6
32524	2018-09-26 10:15:31 UTC	192....	51671	192....	53	61	00:01:00	0x0337	0x0001 (Host Address)	djh90c9110sn8.cloudfront.net	143.204.208.122
32524	2018-09-26 10:15:31 UTC	192....	51671	192....	53	61	00:01:00	0x0337	0x0001 (Host Address)	djh90c9110sn8.cloudfront.net	143.204.208.156

Figure 4: NetworkMiner: no DNS information found for 136.0.99.219.

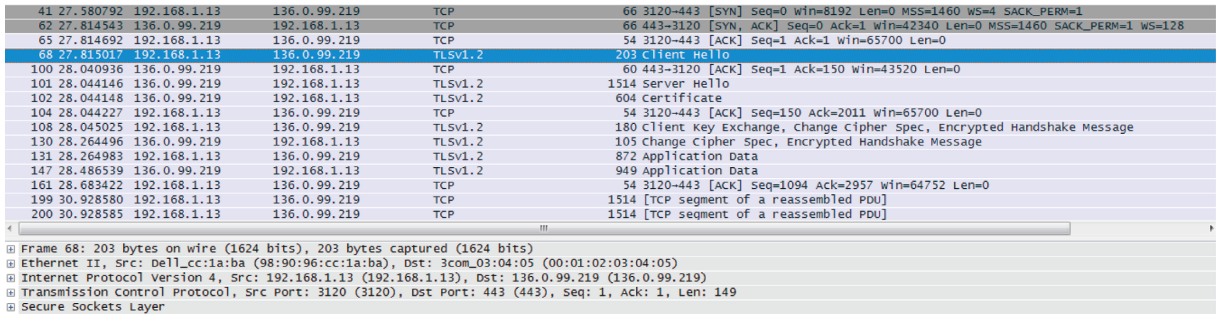

Figure 5: Wireshark: Hotspot Shield add-on.

changes the servers being connected by a user. So, any suspicious or long activity with one server cannot be identified by automated tools. As seen in Figure 11, multiple host names against the same domains are listed in SSL certificate provided by the VPN server. These servers/hosts may be used randomly to request multiple resources over the Internet. It is clearly shown in the figure that the number of connections against this server is only five, which is less than other VPN servers' connection discussed in the paper.

Another unique feature that ZenMate offers is that it changes the domain name as well once the location of the VPN server is changed from the settings of add-on. As shown in Figure 12, the server name is changed to *34.lutz-obrien-olive.ga* once the user has changed the location.

ZenMate changes domain names against region selected by the user, but for the same region, the server identifier of domain name may change but domain remains the same. If a user is constantly changing the locations, after some time when all locations available are exhausted, the domains for each location could be identified. As shown is Figure 13, multiple domains for ZenMate service used by this user are as follows:

(i) lutz-obrien-olive.ga
(ii) ayala-maroon.ga
(iii) hall-silver.ga
(iv) young-purple.ga

This information can now be used to prepare a filter to identify ZenMate VPN inside the network. One can also notice that the last part of domain is always a *color* and ends with *.ga*. So, if we received DNS request or response and the domain name ends with *.ga* with "-" *(dash)* in the query, it could be separated on "-." Once separated, if the last string

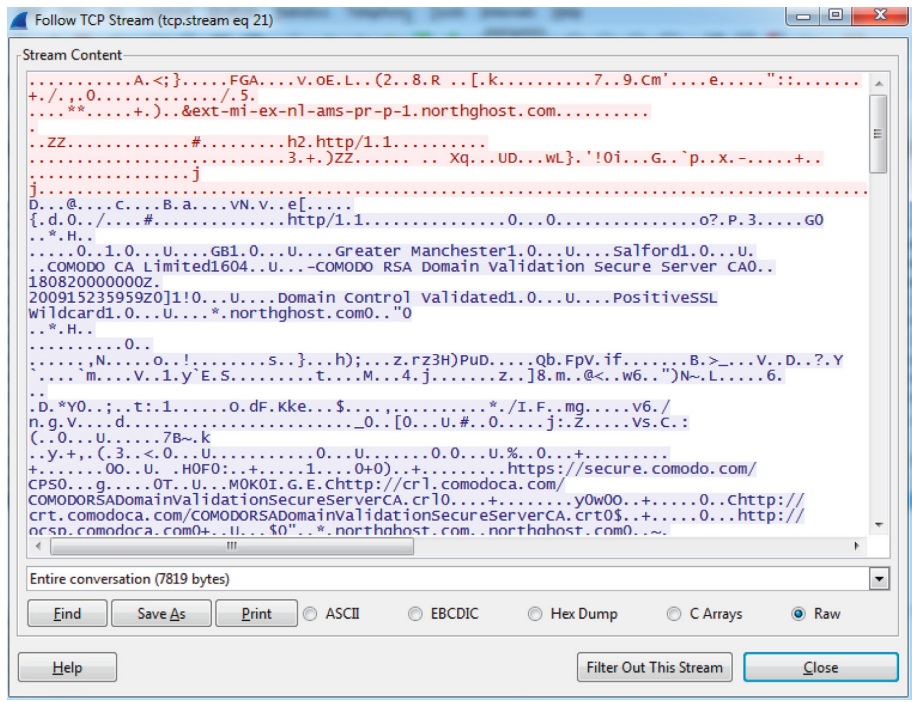

Figure 6: Wireshark: Hotspot Shield add-on TCP stream.

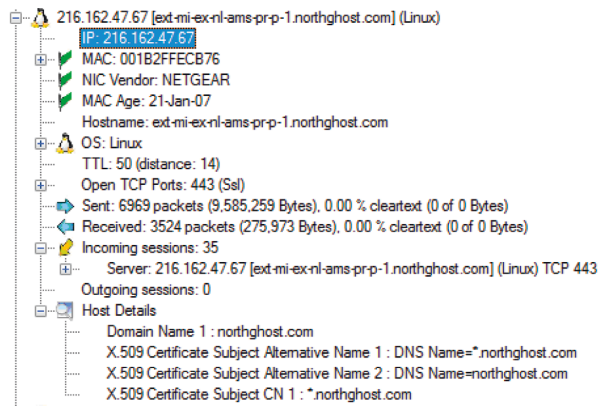

Figure 7: NetworkMiner: Hotspot Shield add-on connection details.

2983	2019-05-13 07:13:26 UTC	192....	60552	192....	53	64	00:04:17	0x35...	0x0005 (CNAME)	fonts.gstatic.com	gstaticadssl.l.google.com
2983	2019-05-13 07:13:26 UTC	192....	60552	192....	53	64	00:04:17	0x35...	0x0001 (Host Address)	gstaticadssl.l.google.com	172.217.19.3
3201	2019-05-13 07:13:26 UTC	192....	62215	192....	53	64	02:49:29	0x3A...	0x0005 (CNAME)	www.google-analytics.com	www-google-analytics.l.google.com
3201	2019-05-13 07:13:26 UTC	192....	62215	192....	53	64	00:04:39	0x3A...	0x0001 (Host Address)	www-google-analytics.l.google.com	172.217.19.14
5200	2019-05-13 07:13:30 UTC	192....	61388	192....	53	64	00:05:00	0xE9...	0x0001 (Host Address)	ext-mi-ex-nl-ams-pr-p-1.northghost.com	216.162.47.67
14297	2019-05-13 07:13:43 UTC	192....	59456	192....	53	64	00:24:24	0x25...	0x0005 (CNAME)	edge-chat.facebook.com	star.c10r.facebook.com
14297	2019-05-13 07:13:43 UTC	192....	59456	192....	53	64	00:00:07	0x25...	0x0001 (Host Address)	star.c10r.facebook.com	157.240.24.20
14927	2019-05-13 07:13:44 UTC	192....	51326	192....	53	64	00:00:47	0xA2...	0x0001 (Host Address)	star-mini.c10r.facebook.com	157.240.24.35
14927	2019-05-13 07:13:44 UTC	192....	51326	192....	53	64	00:04:26	0xA2...	0x0005 (CNAME)	www.facebook.com	star-mini.c10r.facebook.com

Figure 8: NetworkMiner: DNS information for 136.0.99.219.

contains any well-known color name, we can classify it as ZenMate DNS server. As shown in Figure 14, the domain name analysis was done by NetworkMiner, we can see the same pattern discussed above.

3.3. TOR Browser. TOR Browser [11] is used generally by users to hide their Internet activity and to access resources on dark web. TOR browser uses a concept of onion routing to hide user's activity. We installed TOR browser to analyze the network traffic generated by the browser. It uses a nonstandard port for communication over Internet. It uses HTTPS over 9001 TCP Port initially for circuit connection. After the circuit connection is established, TOR may use 443 for normal Internet or any other port as configured. TOR

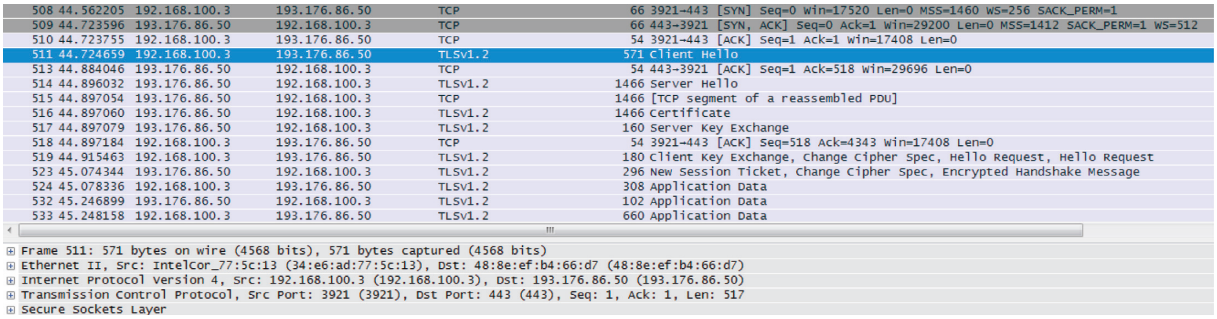

Figure 9: Wireshark: ZenMate add-on.

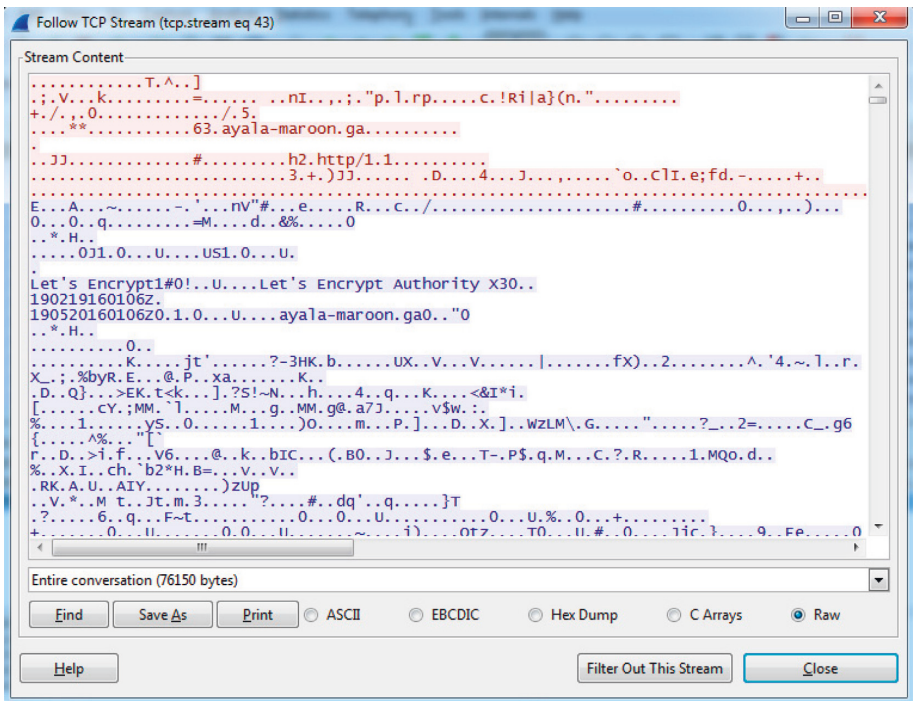

Figure 10: Wireshark: ZenMate add-on TCP stream.

will generally not generate any DNS traffic. A normal TOR stream viewed in Wireshark is shown in Figure 15.

Opening of each website may create new connection to server and server name along with their IP addresses which are communicated to TOR browser during circuit establishment process and are encrypted. Figure 16 shows a TOR-based TCP stream analyzed in Wireshark.

Connection details of a TOR connection analyzed by NetworkMiner are shown in Figure 17. It shows that, against server IP *5.9.42.230*, a total of 639 packets were sent and 586 packets were received by the user.

Complete activity of the user for the session being discussed is also shown in Figure 18. It is interesting to mention here that no DNS activity was found for TOR browser.

3.4. Browsec VPN. Browsec VPN [32] is another freely available VPN. We used it as Firefox add-on. It uses standard HTTPS port along with standard DNS queries. The only way to detect Browsec VPN inside the network is to identify the domain names used by it. Shown below in Figure 19 is the network traffic generated by Browsec VPN captured using Wireshark.

It can be seen in Figure 20 that the domain name is *nl30.tcdn.me* for which the connection was established. Like other VPN services, the domain name of Browsec VPN can also be further divided for better analysis. It consists of three main parts; it can also be seen in certificate details in Figure 21, analyzed by NetworkMiner tool:

(i) Country code

(ii) Server identifier

(iii) Domain name

It is clearly observed that the domain name is $*.tcdn.me$ and the other part consists of some server identifier and location identifier. In Figure 21, the location identifier is *nl*,

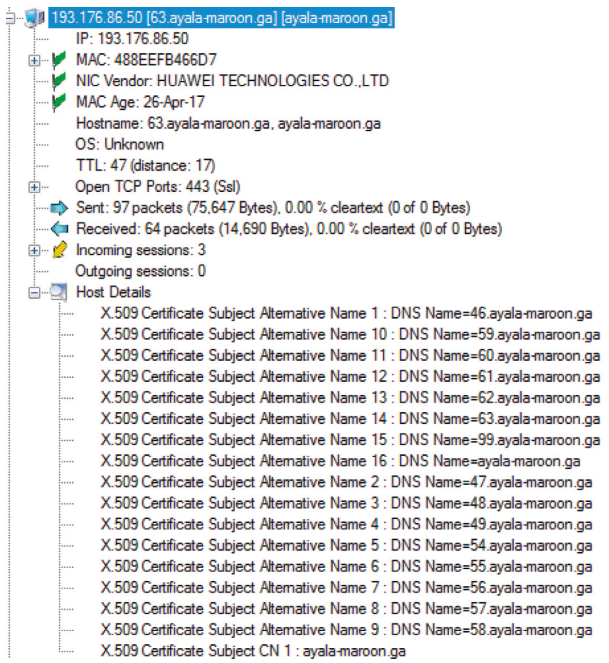

Figure 11: NetworkMiner: ZenMate VPN add-on connection details.

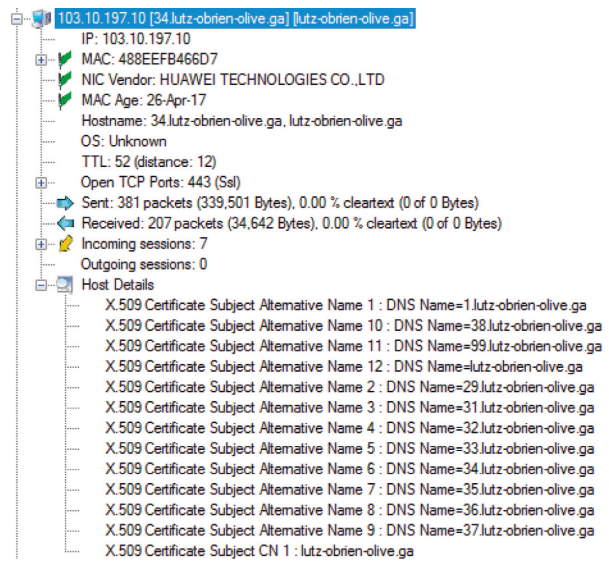

Figure 12: NetworkMiner: ZenMate VPN add-on connection details—changed location.

which means Netherlands, and in Figure 22, we can see the country is United Kingdom.

Like ZenMate VPN, Browsec VPN also changes its DNS information when changing the location, but unlike ZenMate, the domain name is not changed rather only the server qualifier is changed. Figure 23 shows the DNS traffic generated by user's activity.

3.5. Hoxx VPN. Hoxx VPN [33] is another freely available VPN. We used it as Firefox add-on. It uses standard HTTPS port along with standard DNS queries. We can detect Hoxx

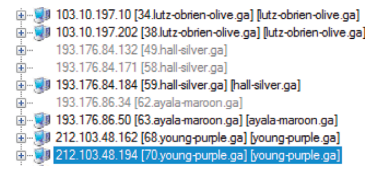

Figure 13: NetworkMiner: ZenMate VPN add-on connection details—all locations.

VPN inside the network by identifying the domain names used by the VPN service. Shown below in Figure 24 is the network traffic generated by Hoxx VPN captured using Wireshark.

It can be seen in Figure 25 that the domain name is *dyn-146-185-141-219-5871-b377a.klafive.com* for which the connection is established. Like other VPN services, the domain name of Hoxx VPN server can also be further divided for better analysis. It consists of two main parts:

(i) Server identifier

(ii) Domain name

This division can also be seen in certificate details in Figure 26, analyzed by NetworkMiner tool. It is clearly observed that domain name is $*.klafive.com$ and the other part consists of some server identifier. Figure 27 shows the DNS traffic generated by user's activity.

4. Proposed System

The proposed system distinguishes the normal flow of an Internet activity or session from an abnormal one. Normally, when a user wants to connect to a website a DNS request is made to translate the web name to IP address [34]. After successful name resolution, against the IP, a TCP (Transmission Control Protocol) session is initiated and required security associations are established. This behavior may be used to monitor and analyze different features of network traffic. [35–37].

The proposed system classifies any incoming data into multiple categories depending on the current state of connection; in addition to that, Internet activity preceding the connection is also monitored to identify the traffic as VPN or simple Internet traffic. The process of detecting any illegitimate traffic is further classified into two main processes:

(i) Feature extraction

(ii) Traffic classification

4.1. Feature Extraction. To classify traffic as normal or VPN, we have to extract different traits of the network traffic. Now, most of these traits can be found in current traffic stream while some of them are collected before the actual stream starts. Figure 28 shows the basic flow of network traffic feature extraction module of the system. The analyzer extracts the following information to be used for traffic categorization.

5438	2019-04-01 19:32:38 UTC	192....	53872	192....	53	64	00:01:08	0x5F28	0x0001 (Host Address)	34.lutz-obrien-olive.ga	103.10.197.10
634	2019-04-01 19:30:33 UTC	192....	62279	192....	53	64	00:00:19	0xC61C	0x0001 (Host Address)	38.lutz-obrien-olive.ga	103.10.197.202
4705	2019-04-01 19:31:16 UTC	192....	63787	192....	53	64	00:05:00	0x1FCC	0x0001 (Host Address)	49.hall-silver.ga	193.176.84.132
4592	2019-04-01 19:31:00 UTC	192....	60082	192....	53	64	00:05:00	0x743A	0x0001 (Host Address)	58.hall-silver.ga	193.176.84.171
4980	2019-04-01 19:31:53 UTC	192....	55896	192....	53	64	00:05:00	0xCE24	0x0001 (Host Address)	59.hall-silver.ga	193.176.84.184
356	2019-04-01 19:30:06 UTC	192....	61995	192....	53	64	00:04:39	0xE773	0x0001 (Host Address)	62.ayala-maroon.ga	193.176.86.34
507	2019-04-01 19:30:30 UTC	192....	49346	192....	53	64	00:04:21	0xF496	0x0001 (Host Address)	63.ayala-maroon.ga	193.176.86.50
5270	2019-04-01 19:32:18 UTC	192....	60309	192....	53	64	00:03:43	0xC40C	0x0001 (Host Address)	68.young-purple.ga	212.103.48.162
5153	2019-04-01 19:32:07 UTC	192....	65171	192....	53	64	00:00:09	0x713C	0x0001 (Host Address)	70.young-purple.ga	212.103.48.194

Figure 14: NetworkMiner: DNS information for ZenMate servers.

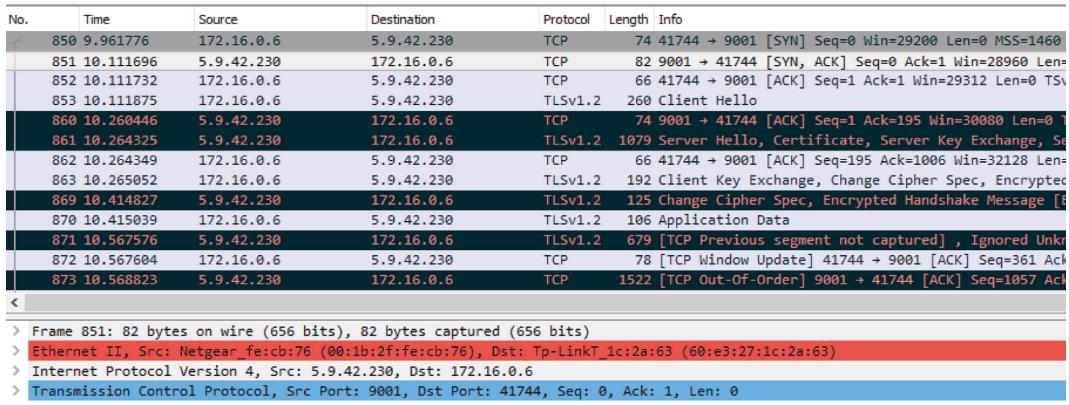

Figure 15: Wireshark: TOR traffic.

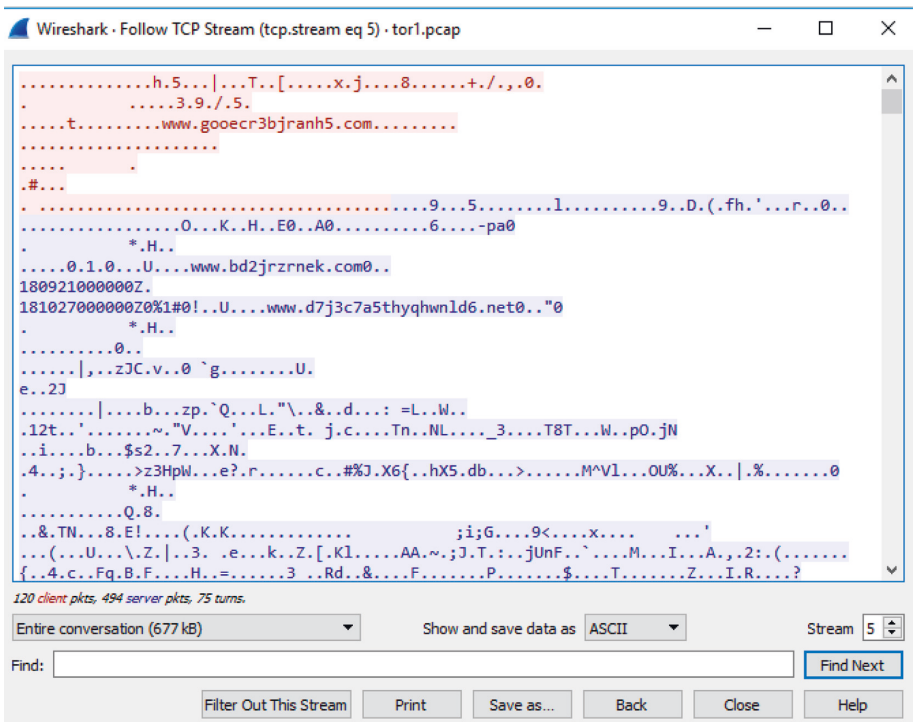

Figure 16: Wireshark: TOR browser TCP stream.

4.1.1. Basic Feature Extraction. Server IP of the server and user is extracted at the first step. This information is extracted from IPv4 Protocol fields, source IP and destination IP [38]. Depending upon the transport layer protocol, the source port and destination ports are also extracted [39].

4.1.2. Domain Name Server Analysis. Unencrypted traffic information is as important in traffic characterization and behavior analysis of users as the encrypted traffic. For any web request, generated by a user, a *DNS* request is initiated by the user's browser to request the IP information of the

Figure 17: NetworkMiner: TOR browser connection details.

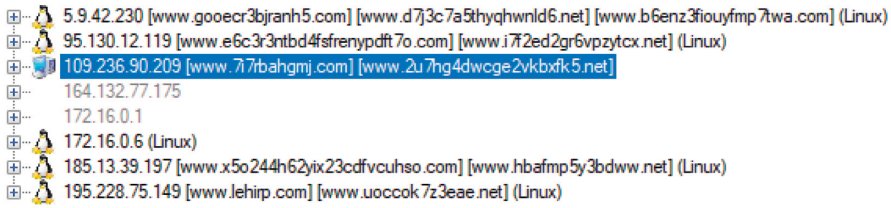

Figure 18: NetworkMiner: TOR browser user activity details.

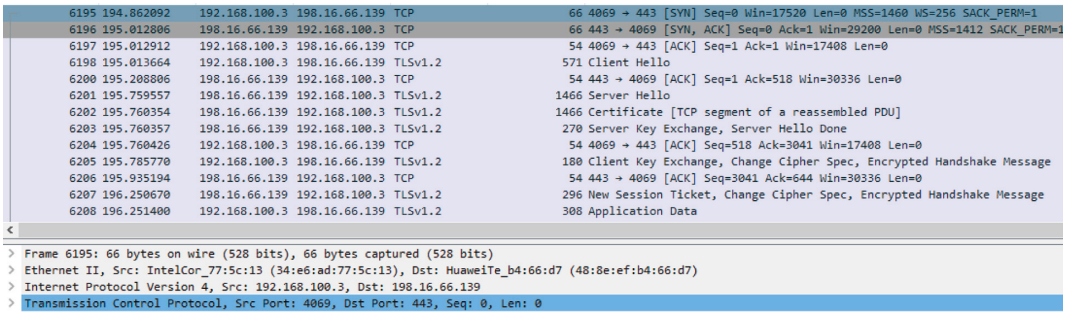

Figure 19: Wireshark: Browsec VPN add-on.

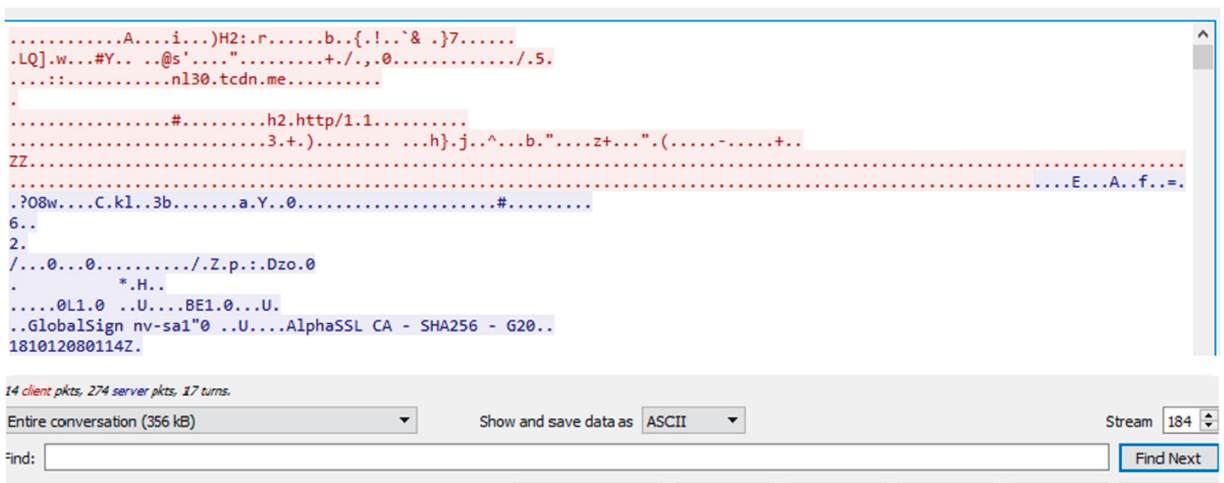

Figure 20: Wireshark: Browsec VPN TCP stream.

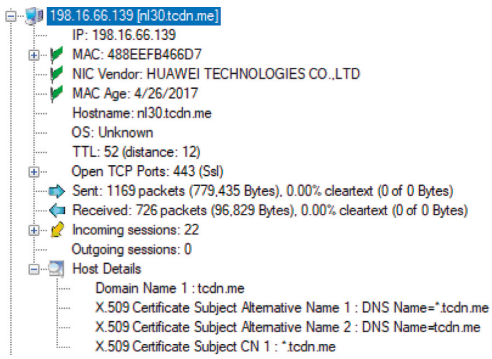

Figure 21: NetworkMiner: Browsec VPN connection details—NL.

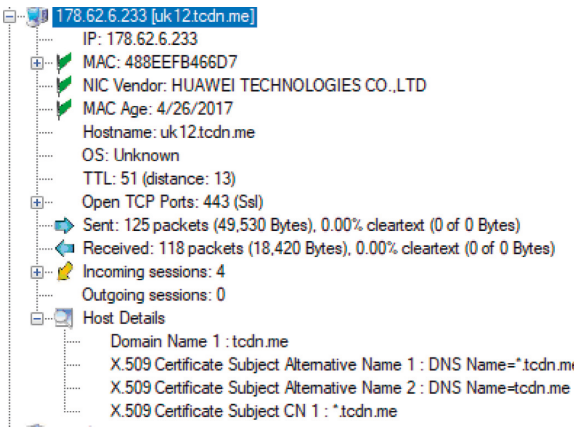

Figure 22: NetworkMiner: Browsec VPN connection details—UK.

Frame nr.	Timestamp	Client	Client Port	Ser...	Server Port	IP TTL	DNS TTL (time)	Transaction ID	Type	DNS Query	DNS Answer
6194	2019-04-01 19:33:01 UTC	192....	50050	192....	53	64	17:24:48	0x3D78	0x0001 (Host Address)	nl30.tcdn.me	198.16.66.139
7425	2019-04-01 19:33:15 UTC	192....	59982	192....	53	64	17:20:27	0xDA3C	0x0001 (Host Address)	nl10.tcdn.me	198.16.66.123
8497	2019-04-01 19:33:39 UTC	192....	53446	192....	53	64	11:18:17	0x7234	0x0001 (Host Address)	sg17.tcdn.me	178.128.57.177
8770	2019-04-01 19:34:00 UTC	192....	61604	192....	53	64	12:19:49	0x7957	0x0001 (Host Address)	sg25.tcdn.me	178.128.117.77
8980	2019-04-01 19:34:02 UTC	192....	64325	192....	53	64	17:14:48	0x173F	0x0001 (Host Address)	uk1.tcdn.me	178.62.34.82
9282	2019-04-01 19:34:23 UTC	192....	56276	192....	53	64	17:19:22	0x19B4	0x0001 (Host Address)	uk9.tcdn.me	46.101.16.229
9762	2019-04-01 19:34:44 UTC	192....	49190	192....	53	64	17:24:24	0x86B9	0x0001 (Host Address)	uk12.tcdn.me	178.62.6.233

Figure 23: NetworkMiner: DNS information for Browsec VPN.

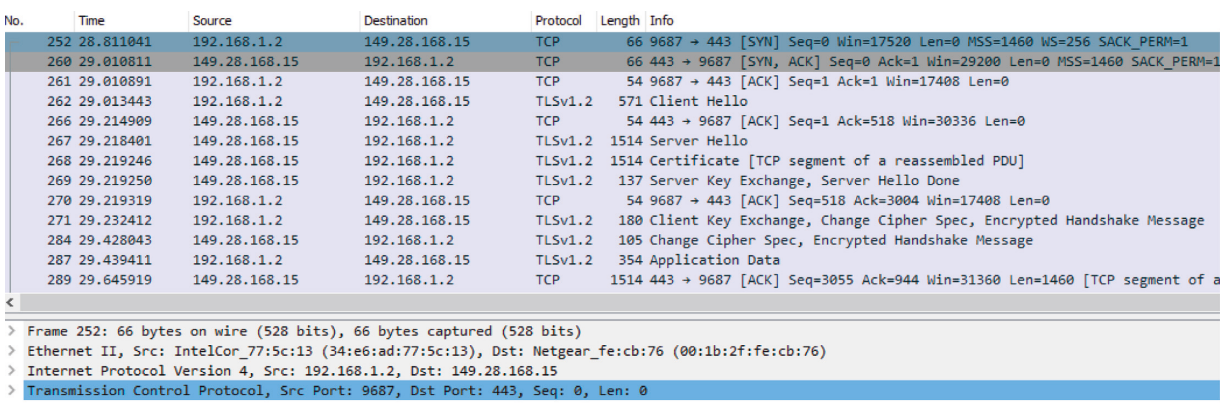

Figure 24: Wireshark: Hoxx VPN add-on.

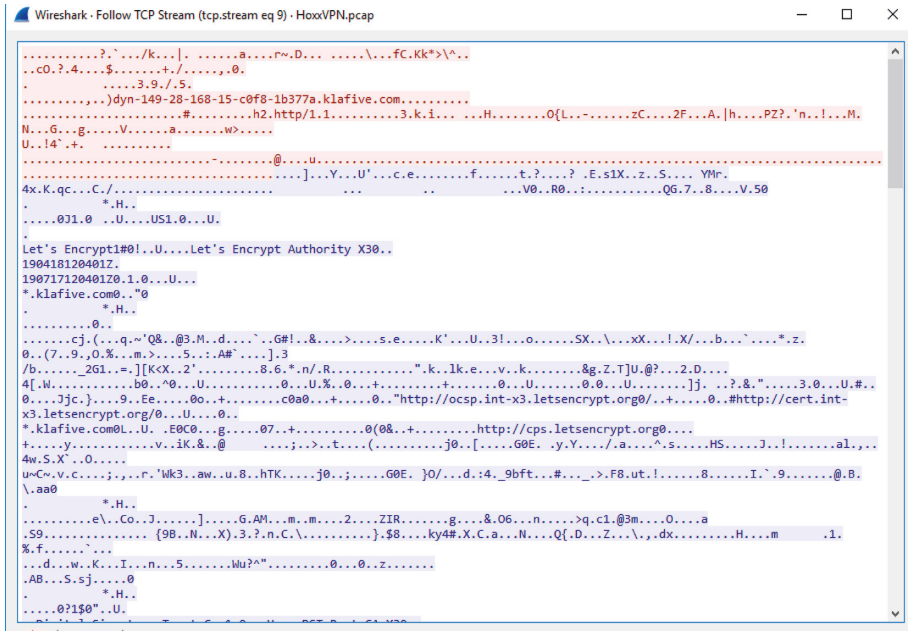

Figure 25: Wireshark: Hoxx VPN TCP stream.

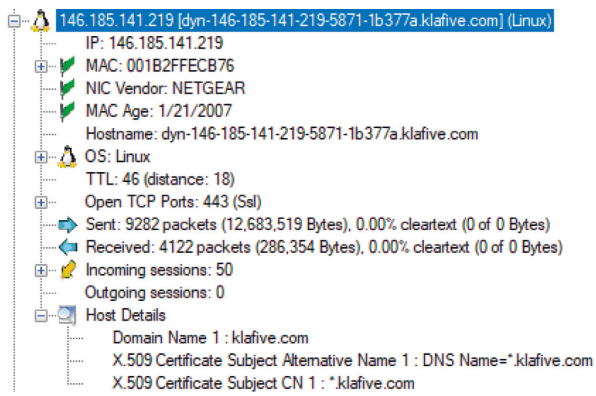

Figure 26: NetworkMiner: Hoxx VPN connection details.

250	2019-05-13 07:37:01 UTC	192....	51103	192....	53	64	1.00:00:00	0xCFD8	0x0001 (Host ...	dyn-149-28-168-15-c0f8-1b377a.klafive.com	149.28.168.15
251	2019-05-13 07:37:01 UTC	192....	51103	192....	53	64	1.00:00:00	0xCFD8	0x0001 (Host ...	dyn-149-28-168-15-c0f8-1b377a.klafive.com	149.28.168.15
254	2019-05-13 07:37:01 UTC	192....	62867	192....	53	64	1.00:00:00	0xC6BD	0x0001 (Host ...	dyn-149-28-168-15-c0f8-1b377a.klafive.com	149.28.168.15
263	2019-05-13 07:37:01 UTC	192....	61758	192....	53	64	00:00:00	0x621C	0x0000	dyn-149-28-168-15-c0f8-1b377a.klafive.com	No error condition (flags 0x8180)
264	2019-05-13 07:37:01 UTC	192....	61758	192....	53	64	00:00:00	0x621C	0x0000	dyn-149-28-168-15-c0f8-1b377a.klafive.com	No error condition (flags 0x8180)
895	2019-05-13 07:37:14 UTC	192....	64210	192....	53	64	00:00:00	0x6FAE	0x0000	dyn-149-28-168-15-c0f8-1b377a.klafive.com	No error condition (flags 0x8180)
18562	2019-05-13 07:37:48 UTC	192....	54745	192....	53	64	1.00:00:00	0xCE67	0x0001 (Host ...	dyn-146-185-141-219-5871-1b377a.klafive.com	146.185.141.219
18563	2019-05-13 07:37:48 UTC	192....	54745	192....	53	64	1.00:00:00	0xCE67	0x0001 (Host ...	dyn-146-185-141-219-5871-1b377a.klafive.com	146.185.141.219
18572	2019-05-13 07:37:48 UTC	192....	50164	192....	53	64	23:59:59	0x5EB6	0x0001 (Host ...	dyn-146-185-141-219-5871-1b377a.klafive.com	146.185.141.219
19170	2019-05-13 07:37:49 UTC	192....	55650	192....	53	64	00:00:00	0xCC57	0x0000	dyn-146-185-141-219-5871-1b377a.klafive.com	No error condition (flags 0x8180)
19173	2019-05-13 07:37:49 UTC	192....	55650	192....	53	64	00:00:00	0xCC57	0x0000	dyn-146-185-141-219-5871-1b377a.klafive.com	No error condition (flags 0x8180)

Figure 27: NetworkMiner: DNS information for Hoxx VPN.

server name. A response is sent to the user from DNS server containing IP information of the server [34]. This information is stored by our system to verify the DNS server name vs. HTTPS certificate's server name to see for any inconsistencies.

4.1.3. HTTPS Protocol Detection. Incoming traffic is then passed to HTTPS detection module. The system looks for HTTPS other than port 443. This is done by looking for HTTPS headers on streams which are TCP-based connections but the server port number is other than 443. A lot of

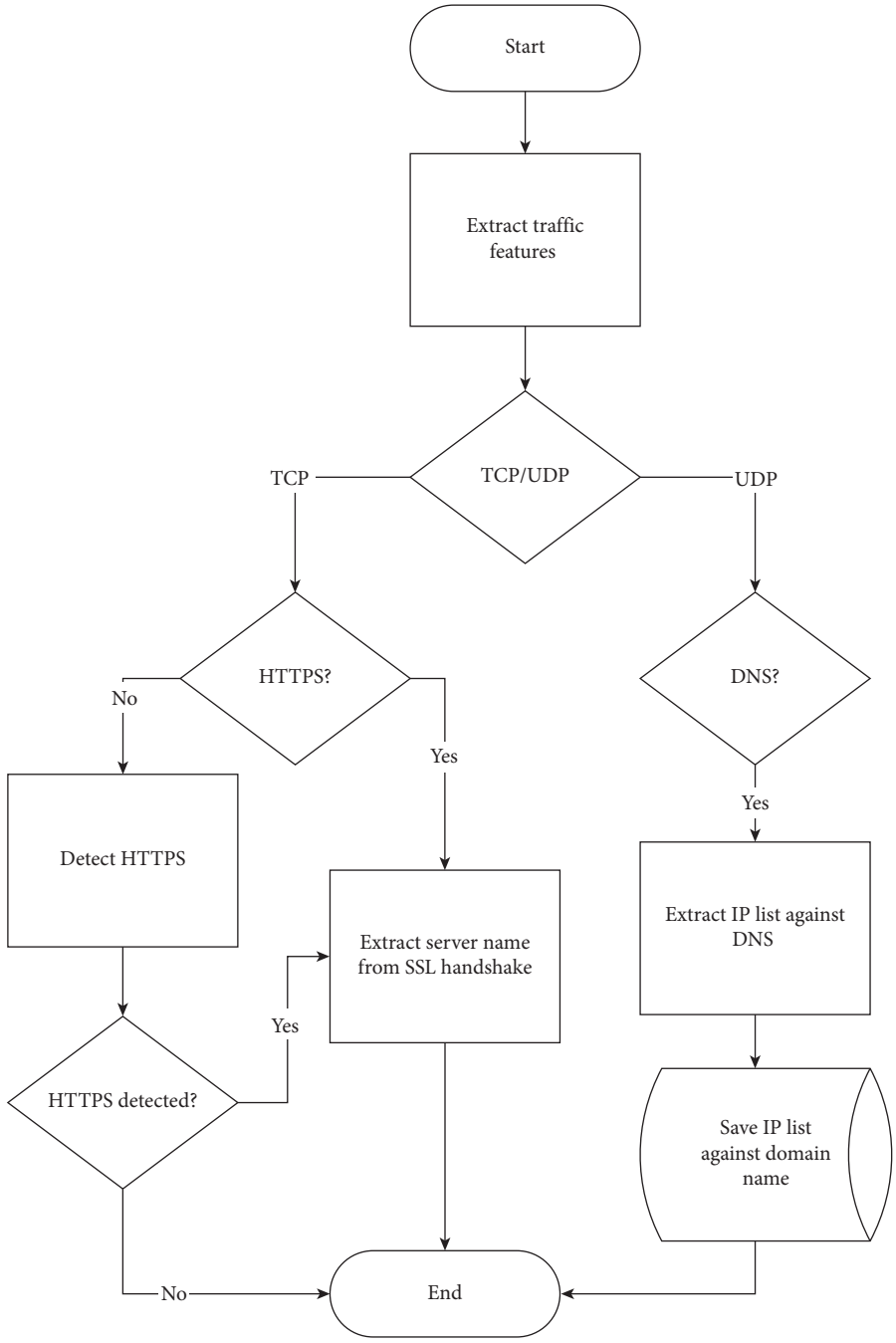

Figure 28: Feature extraction.

applications and services use the technique to change the server port. This allows them to pass through network firewall and is not labelled as encrypted payload.

4.1.4. SSL Analysis. The proposed system decodes SSL certificates [40] once HTTPS is detected. There are 4 basic types of messages in SSL:

(i) Handshake
(ii) Change Cipher Spec
(iii) Application data
(iv) Alert

From the Handshake messages, we extract the server information such as name of the server to which the connection is made. This is used to verify or detect the DNS activity versus server name.

These features once extracted are used by traffic classifier to classify each connection to VPN or normal traffic.

4.2. Traffic Classification. After features are extracted, we can classify the incoming traffic as normal traffic or VPN traffic only for the TCP-based connections. TCP connection states

are stored for every new connection. Once the connection is established, it is classified as legitimate or VPN traffic based on extracted features of previous network traffic and new connection. This classification may be as legitimate traffic or VPN traffic. The proposed scheme classifies the incoming connections as shown in Figure 29 and is discussed below.

4.2.1. IP-Based Classification. Server IP of each new connection is looked up in an already populated IP-based hash table. This hash table contains the IP list of TOR's exit nodes [11] along with the server IP that were previously classified by the system as VPN servers. This is done to minimize the *resource utilization* against already classified VPN server. If server IP of the current connection is found in this IP-based hash, then the traffic is classified as VPN traffic.

4.2.2. Server Name-Based Classification. If the connection is not classified by VPN IP-based hash table, the server name specified in HTTPS *Client Hello* message is used to classify the connection. In a normal TCP/IP-based communication, whenever a service or website needs to be accessed, first its domain name is converted into IP address. This is done to access the resources over the Internet [41]. An IP address at a given time is bound to a specific domain. Using this technique, we classify the normal domains against the domains responsible for VPN Services. This classification can be further divided into two steps.

4.2.3. No Server Name Analysis. Against the current server name extracted from the connection, we look up our self-maintained DNS list, populated by network traffic. If no DNS entry is present for that server name in the list or the server IP of the connection is not associated against the given server name, such traffic is classified as VPN traffic. Mostly, inside the initial connection to VPN server, these IPs against DNS are shared with the client's application in SSL-protected channel as to avoid any DNS-based filtering.

4.2.4. Server Name Analysis. The server name or the domain name of the current connection is looked up against the well-known VPN server's domain names. The list is maintained to look up the server name; if found, the connection is classified as VPN-based connection. The list is generated by the traffic analysis of these VPN servers, and some unique strings are extracted specific to that VPN service as discussed previously is Section 3.

5. System Evaluation

The deployment of our proposed solution, if used only for detection, can be passive as well. Passive deployment will result in *lower latency* as the traffic is being mirrored by the switch or gateway itself. For passive deployment, all the traffic destined outside the network and DNS traffic must pass through the tapped interface as shown in Figure 30.

We analyzed the traffic pattern of well-known available VPN services which use HTTPS protocol for communication. These servers are listed below:

(i) TOR browser
(ii) Hotspot Shield free
(iii) Browsec VPN
(iv) ZenMate VPN
(v) Hoxx VPN

The traffic of these VPN services was analyzed, and a selection criterion was build based on the pattern emerging from the analysis. The key features for each VPN service are shown in Table 2. In case of TOR, we see nonstandard HTTPS behavior which means that it may not be on default port 443. We can also detect TOR by *TOR nodes* list populated and updated by community.

In case of Hotspot Shield, we tested two variants of its client. One was the add-on of Firefox web browser, and the other client was desktop application. In case of web browser extension or add-on, Hotspot Shield uses special domain names which are used to uniquely classify the service. In case of desktop application, the client uses nonstandard port for HTTPS with no DNS activity. *Browsec and Hoxx* VPNs both were tested as add-on to the browser, and they are uniquely classified using the domain names the servers use.

All three services discussed above use the same type of domain names across multiple geolocations, e.g., any traffic may be classified as traffic of Hoxx VPN if its domain name contains $*.klafive.com$. This is not the case for ZenMate VPN. It changes domain names with respect to geolocations chosen by the user. The list of these domain names is communicated during initial connection setup and is updated frequently. This allows VPN services like ZenMate and others to work over a network which uses DNS-based filters, if these filters are not updated frequently.

5.1. Traffic Generation. Across multiple systems inside the network, multiple clients of the abovementioned VPN services were installed and configured. These clients were enabled, and network activity was generated by surfing the Internet. The activity was monitored by VPN detector, and alerts were generated once the VPN activity was detected.

5.2. Traffic Classification Alert. The alerts generated above for different VPN services were of different types depending upon the activities performed by the users. The generated alerts by five of these users are shown in Table 3.

The alerts shown in Table 3 show the traffic classification of each type of VPN service used with respect to its unique characteristics as discussed in Table 2. Mostly, VPNs may be classified with the help of DNS activity which enable the user to access such services.

The results shown in Table 3 show that the system classified 400 out of 729 active connections as potential VPN connections. Once the system is deployed, any new connection activity in the network is monitored. Each system connected to Internet manages its on DNS cache to reuse

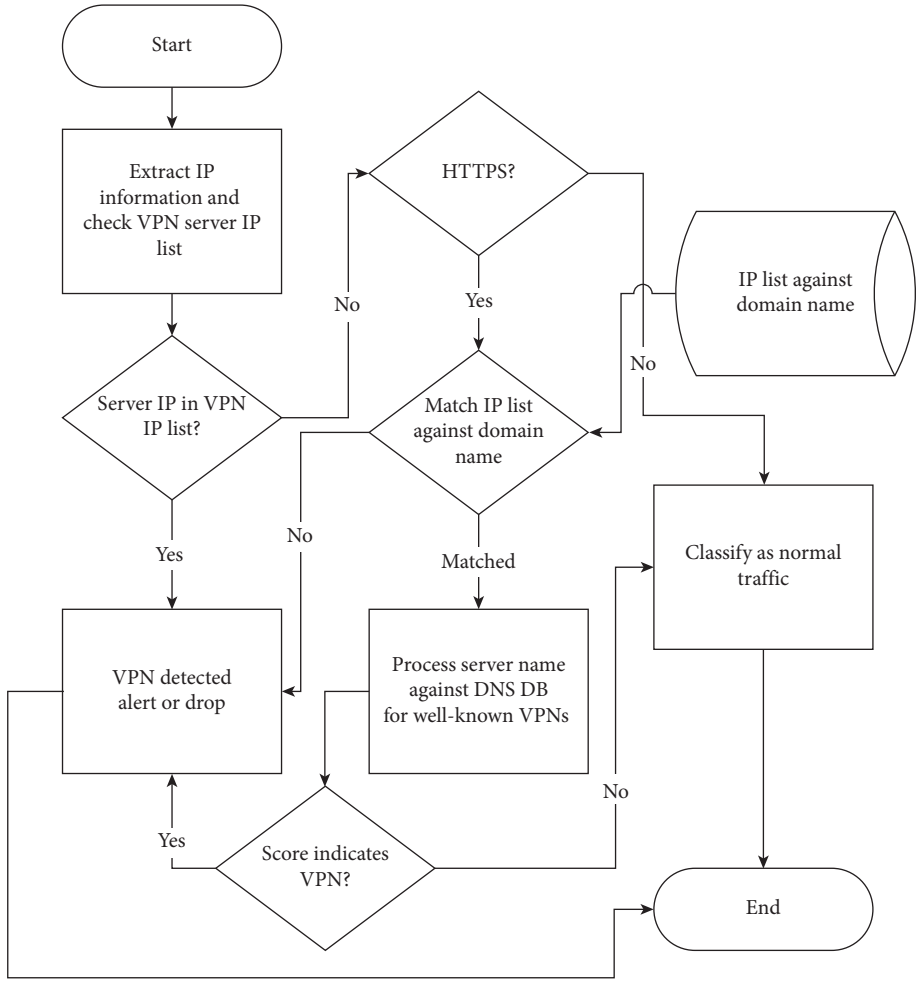

Figure 29: Traffic classification.

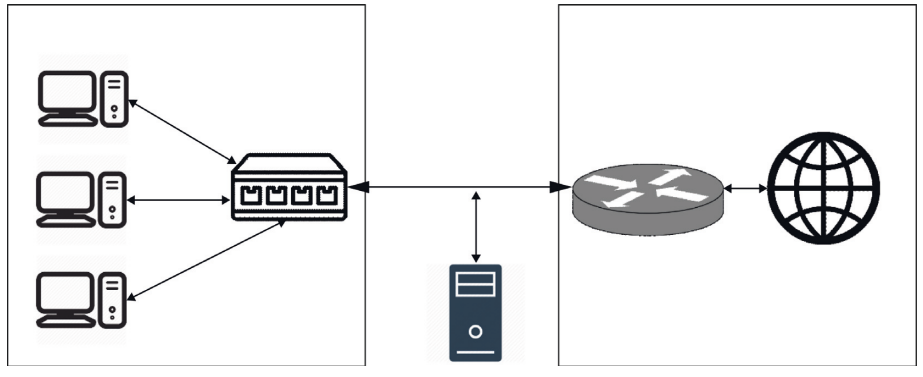

Figure 30: Deployment model.

DNS information. If a new connection is made and no DNS activity is present in the system for the server, the system will flag it as potential VPN traffic. To improve system's precision, the system ignores the already established connections.

VPN classification based on IP and DNS activity may need periodic updates to the lists maintained by the system. Updating this information will increase the overall accuracy of the system and result in less false positives and negatives. Our test shows that, in case of TOR IP analysis, the IP

Table 2: Forensic analysis of freely available VPN services.

VPN services	Classifiers for forensic analysis			
	IP	Host name	Nonstandard HTTPS	DNS activity
TOR browser	✓	×	✓	✓
Hotspot Shield free	×	✓	✓	✓
Browsec VPN	×	✓	×	×
ZenMate VPN	×	✓	×	×
Hoxx VPN	×	✓	×	×

Table 3: Alerts generated for the user activity.

User details	Alerts classification (connection based)				
	Total	Legitimate activity	IP-based VPN	DNS-based VPN	NO DNS
User 1	178	59	4	109	6
User 2	85	50	0	35	0
User 3	250	114	0	135	1
User 4	71	24	2	41	4
User 5	145	82	0	63	0

information should be populated in real time to get better results.

6. Conclusion

A VPN service inside an organization may generally be used by an individual to hide the real communication. This communication may be harmful or damage the organization, and the organization may not allow such communication over its monitored network. An organization may not be able to invest heavily on SSL-based proxies to manage its network. This paper proposes a lightweight approach to detect and block unwanted VPN clients inside the organizational network responsible for some illegitimate activity.

Our proposed technique focuses on the information available in plain, which means there is no need to decrypt or decode any network communication. This helps in low *resource utilization*. The proposed solution not only focuses on the current connection but also keeps track of the network activity responsible for this communication, i.e., DNS activity. Such mapping of DNS with its next stream helps identify the normal behavior of the TCP/IP network stack. If no Domain Name information is available for current connection, it may not be normal traffic flow. The scheme also analyzes nonstandard use of HTTPS and detects this anomaly as it is largely used to hide such communication from HTTPS-based filters in firewall.

Results show that our proposed system is able to identify and classify such trends in network traffic and classify the network traffic. The analysis of the VPN services discussed in Table 2 is crucial to detect these services. These service providers keep changing the traffic characteristics for their service. Active analysis of these services must be carried out to keep VPN detector up to date with latest traffic trends.

Data Availability

The data used to support the findings of this study are provided within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] B. Harris and R. Hunt, "Tcp/ip security threats and attack methods," *Computer Communications*, vol. 22, no. 10, pp. 885–897, 1999.

[2] X. Li, M. Wang, H. Wang, Y. Ye, and C. Qian, "Toward secure and efficient communication for the internet of things," *IEEE/ACM Transactions on Networking*, vol. 27, no. 2, pp. 621–634, 2019.

[3] E. Rescorla, *SSL and TLS: Designing and Building Secure Systems*, vol. 1, Addison-Wesley, Boston, MA, USA, 2001.

[4] A. P. Felt, R. Barnes, A. King, C. Palmer, C. Bentzel, and P. Tabriz, "Measuring HTTPS adoption on the web," in *Proceedings of the 26th USENIX Security Symposium (USENIX Security 17)*, pp. 1323–1338, USENIX Association, Vancouver, BC, Canada, August 2017.

[5] J. Clark and P. C. Van Oorschot, "SoK: SSL and HTTPS: revisiting past challenges and evaluating certificate trust model enhancements," in *Proceedings of the 2013 IEEE Symposium on Security and Privacy*, pp. 511–525, IEEE, Berkeley, CA, USA, May 2013.

[6] C. Paya and O. Dubrovsky, "Inspecting encrypted communications with end-to-end integrity," US Patent 7562211, 2009.

[7] V. Lifliand and A. Michael Ben-Menahem, "Encrypted network traffic interception and inspection," US Patent 8578486, 2013.

[8] N. Leavitt, "Anonymization technology takes a high profile," *Computer*, vol. 42, no. 11, pp. 15–18, 2009.

[9] Z. Zhang, S. Chandel, J. Sun, S. Yan, Y. Yu, and J. Zang, "VPN: a boon or trap?: a comparative study of MPLs, IPSec, and SSL virtual private networks," in *Proceedings of the 2018 2nd International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 510–515, IEEE, Erode, India, February 2018.

[10] K. Karuna Jyothi and B. I. Reddy, "Study on virtual private network (VPN), VPN's protocols and security," *International*

Journal of Scientific Research in Computer Science, Engineering and Information Technology, vol. 3, no. 5, 2018.

[11] D. Roger, N. Mathewson, and S. Paul, "TOR: the second-generation onion router," Technical report, Naval Research Laboratory, Washington, DC, USA, 2004.

[12] A. Yamada, Y. Miyake, K. Takemori, A. Studer, and A. Perrig, "Intrusion detection for encrypted web accesses," in *Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07)*, vol. 1, pp. 569–576, Niagara Falls, Ont., Canada, May 2007.

[13] M. Ikram, N. Vallina-Rodriguez, S. Seneviratne, M. A. Kaafar, and V. Paxson, "An analysis of the privacy and security risks of android VPN permission-enabled apps," in *Proceedings of the 2016 Internet Measurement Conference*, pp. 349–364, ACM, Santa Monica, CA, USA, November 2016.

[14] S. Sudin, R. B. Ahmad, and S. Z. Syed Idrus, "A model of virus infection dynamics in mobile personal area network," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 10, no. 2–4, pp. 197–201, 2018.

[15] N. Weaver, C. Kreibich, M. Dam, and V. Paxson, "Here be web proxies," in *Proceedings of the International Conference on Passive and Active Network Measurement*, pp. 183–192, Springer, Los Angeles, CA, USA, March 2014.

[16] C. Reis, S. D. Gribble, T. Kohno, and N. C. Weaver, "Detecting in-flight page changes with web tripwires," in *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, vol. 8, pp. 31–44, San Francisco, CA, USA, April 2008.

[17] N. Vallina-Rodriguez, S. Sundaresan, C. Kreibich, and V. Paxson, "Header enrichment or ISP enrichment?: emerging privacy threats in mobile networks," in *Proceedings of the 2015 ACM SIGCOMM Workshop on Hot Topics in Middleboxes and Network Function Virtualization*, pp. 25–30, ACM, London, UK, August 2015.

[18] N. Weaver, C. Kreibich, and V. Paxson, "Redirecting DNS for ads and profit," in *Proceedings of the UNISEX Workshop on Free and Open Communications on the Internet 2011*, vol. 2, no. 2–3, San Francisco, CA, USA, August 2011.

[19] N. Vallina-Rodriguez, J. Amann, C. Kreibich, N. Weaver, and V. Paxson, "A tangled mass: the android root certificate stores," in *Proceedings of the 10th ACM International on Conference on Emerging Networking Experiments and Technologies*, pp. 141–148, ACM, Sydney, Australia, December 2014.

[20] Y. Song and U. Hengartner, "Privacyguard: a VPN-based platform to detect information leakage on android devices," in *Proceedings of the 5th Annual ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices*, pp. 15–26, ACM, Denver, CO, USA, October 2015.

[21] S. Fahl, M. Harbach, T. Muders, L. Baumgärtner, B. Freisleben, and M. Smith, "Why eve and mallory love android: an analysis of android ssl (in) security," in *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, pp. 50–61, ACM, Raleigh, NC, USA, October 2012.

[22] V. T. Goh, J. Zimmermann, and M. Looi, "Towards intrusion detection for encrypted networks," in *Proceedings of the 2009 International Conference on Availability, Reliability and Security*, pp. 540–545, IEEE, Fukuoka, Japan, March 2009.

[23] A. A. Abimbola, J. M. Munoz, and W. J. Buchanan, "Nethost-sensor: investigating the capture of end-to-end encrypted intrusive data," *Computers & Security*, vol. 25, no. 6, pp. 445–451, 2006.

[24] R.. Martin, "Snort—lightweight intrusion detection for networks," in *Proceedings of the 13th USENIX Conference on System Administration, LISA '99*, pp. 229–238, USENIX Association, Seattle, WA, USA, November 1999.

[25] X. Li, S. G. Karanvir, G. H. Cooper, and J. R. G., "Encrypted data inspection in a network environment," US Patent 9176838, 2013.

[26] G. He, B. Xu, and H. Zhu, "AppFA: a novel approach to detect malicious android applications on the network," *Security and Communication Networks*, vol. 2018, Article ID 2854728, 15 pages, 2018.

[27] W. Niu, X. Zhang, G. W. Yang, J. Zhu, and Z. Ren, "Identifying APT malware domain based on mobile DNS logging," *Mathematical Problems in Engineering*, vol. 2017, Article ID 4916953, 9 pages, 2017.

[28] A. Nath, *Packet Analysis with Wireshark*, Packt Publishing Ltd., Birmingham, UK, 2015.

[29] Netresec, Network miner.

[30] AnchorFree, Hoptspot Shield VPN.

[31] ZenGuard, ZenMate VPN.

[32] Browsec LLC, Browsec VPN Your Personal Privacy and Security Online.

[33] VPN1.com, Lightning Fast VPN Service | Hoxx VPN | VPN Service for Everyone.

[34] P. V. Mockapetris, "Domain names: implementation specification," Technical report, USC/Information Sciences Institute, Marina del Rey, CA, USA, 1983.

[35] L. Deri, R. Carbone, and S. Suin, "Monitoring networks using ntop," in *Proceedings of the 2001 IEEE/IFIP International Symposium on Integrated Network Management Proceedings. Integrated Network Management VII. Integrated Management Strategies for the New Millennium (Cat. No. 01EX470)*, pp. 199–212, IEEE, Seattle, WA, USA, May 2001.

[36] B. Paul, J. Kline, D. Plonka, and A. Ron, "A signal analysis of network traffic anomalies," in *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurment*, pp. 71–82, ACM, New York, NY, USA, 2002.

[37] Mohammed Abdul Qadeer, A. Iqbal, M. Zahid, and M. Rahman Siddiqui, "Network traffic analysis and intrusion detection using packet sniffer," in *Proceedings of the 2010 Second International Conference on Communication Software and Networks*, pp. 313–317, IEEE, Singapore, February 2010.

[38] J. Postel, "Internet protocol," Technical report, DARPA, Arlington County, VA, USA, 1981.

[39] J. Postel, "Transmission control protocol," Technical report, DARPA, Arlington County, VA, USA, 1981.

[40] T. Dierks and E. Rescorla, "The transport layer security (TLS) protocol version 1.2," Technical report, 2008.

[41] A. F. Behrouz, *TCP/IP Protocol Suite*, McGraw-Hill, Inc., New York, NY, USA, 2nd edition, 2002.