

# Encrypted Traffic Analysis for Detecting Malicious Insider



By

Muhammad Zain ul Abideen

Reg No. 277706

Supervisor

Dr Hasan Tahir

Department of Information Security

School of Electrical Engineering and Computer Sciences

National University of Sciences and Technology (NUST), Islamabad, Pakistan

April 2021

## Approval

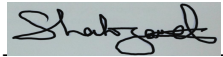
It is certified that the contents and form of the thesis entitled "Encrypted Traffic Analysis for Detecting Malicious Insider" submitted by MUHAMMAD ZAIN ABIDEEN have been found satisfactory for the requirement of the degree

Advisor : Dr. Dr Hasan Tahir

Signature: 

Date: 26-Mar-2021

Committee Member 1:Dr. Shahzad Saleem

Signature: 

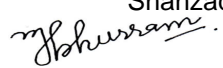
Date: 30-Mar-2021

Committee Member 2:Dr. Mehdi Hussain

Signature: 

Date: 28-Mar-2021

Committee Member 3:Dr. Muhammad Khuram  
Shahzad

Signature: 

Date: 30-Mar-2021

## THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS/MPhil thesis entitled "Encrypted Traffic Analysis for Detecting Malicious Insider" written by MUHAMMAD ZAIN ABIDEEN, (Registration No 00000277706), of SEECs has been vetted by the undersigned, found complete in all respects as per NUST Statutes/Regulations, is free of plagiarism, errors and mistakes and is accepted as partial fulfillment for award of MS/M Phil degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in the said thesis.

Signature: \_\_\_\_\_  \_\_\_\_\_

Name of Advisor: Dr. Dr Hasan Tahir \_\_\_\_\_

Date: \_\_\_\_\_ **26-Mar-2021** \_\_\_\_\_

Signature (HOD): \_\_\_\_\_

Date: \_\_\_\_\_

Signature (Dean/Principal): \_\_\_\_\_

Date: \_\_\_\_\_


# Dedication

To my parents, my wife, my son, my supervisor and the GEC members who have always been supportive.

## Certificate of Originality

I hereby declare that this submission titled "Encrypted Traffic Analysis for Detecting Malicious Insider" is my own work. To the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at NUST SEECS or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at NUST SEECS or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation and linguistics, which has been acknowledged. I also verified the originality of contents through plagiarism software.

Student Name: MUHAMMAD ZAIN ABIDEEN

Student Signature: \_\_\_\_\_

## **Acknowledgements**

I am extremely thankful to Allah Almighty. I would like to thank my family and friends who encouraged me and supported me. I express my enormous gratitude to my supervisor Dr. Hasan Tahir and Dr. Shahzad Saleem and my GEC members. I was very lucky that I could work on my thesis in KTH-AIS lab at SEECS-NUST. I want to thank Dr. Shahzad Saleem and Dr. Mehdi Hussain on their generosity in sharing precious thoughts and knowledge with me. Last but not least, I want to thank Dr. Muhammad Khurram Shahzad at SEECS-NUST for sharing his valuable experiences and improving overall presentation of my work.

# CONTENTS

Chapter 1 Introduction .....	1
1.1 Research Problem .....	2
1.2 Proposed Solution .....	4
1.3 Summary .....	5
Chapter 2 Literature Review .....	6
2.1 Overview of challenges faced in IDS Design .....	7
2.2 Anomaly based IDS for HTTP protocol in SSL channel .....	10
2.3 Hybrid Approach to IDS Design .....	11
2.4 Pre-shared Key Central IDS Design .....	13
2.5 Domain Reputation based IDS .....	15
2.6 IDS Design for Detecting APT .....	17
2.7 Summary .....	20
Chapter 3 Research Methodology .....	21
3.1 Fundamentals of our Research .....	22
3.2 Our Research Process .....	23
3.2.1 Construction of Conceptual Framework .....	24
3.2.2 Develop a System Design .....	25
3.2.3 Analyze and Design .....	26
3.2.4 System Prototype .....	26
3.2.5 Observe and Evaluate .....	27
3.3 Summary .....	27
Chapter 4 Forensic Analysis of VPN Services .....	28
4.1 ZenMate VPN .....	29
4.2 Hotspot Shield VPN .....	33
4.3 Browsec VPN .....	35
4.4 uVPN .....	39
4.5 Summary .....	41
Chapter 5 System Design and Evaluation .....	43
5.1 Traffic Information Extraction .....	44
5.2 5-Tuple Extraction .....	45

5.2.1 DNS based Information Extraction.....	46
5.2.2 HTTPS/TLS Protocol Detection .....	46
5.2.3 SSL Based Feature Extraction .....	46
5.3 Traffic Classification .....	47
5.3.1 IP Based Classification .....	47
5.3.2 Name Server Classification .....	49
5.3.3 DNS based Classification.....	49
5.4 System Evaluation .....	50
5.4.1 Traffic Generation.....	52
5.4.2 Traffic Classification Alert.....	53
5.5 Summary.....	54
Chapter 6 Conclusion .....	55
6.1 System Features .....	56
6.2 Future Work .....	56
Bibliography.....	58



# List of Figures

Figure 2-1 General Architecture of Software Based IDS.....	8
Figure 2-2 Proposed Solution for Message transfer and inspection.....	13
Figure 2-3 Malicious Domains Detection through DNS Data Analysis[29].....	15
Figure 3-1 Sub Processes for DSR.....	23
Figure 4-1 Fiddler-ZenMate-Server IP list.....	30
Figure 4-2 Wireshark-ZenMate-Non-secure traffic on 9002.....	31
Figure 4-3 NetworkMiner-ZenMate-DNS request Response for Server .....	31
Figure 4-4 Wireshark-ZenMate-Nonstandard port for HTTPS.....	32
Figure 4-5 NetworkMiner-ZenMate-DNS request Response for Server-2 .....	32
Figure 4-6 Web Response-Hotspot Shield- Server Name list .....	33
Figure 4-7 NetworkMiner-Hotspot Shield-DNS request Response for Server .....	34
Figure 4-8 Wireshark-HotspotShield-Standard port for HTTPS .....	34
Figure 4-9 Wireshark-Hotspot Shield-DNS request Response for multiple Servers	35
Figure 4-10 Fiddler-Browsec- Domain Name list.....	36
Figure 4-11 NetworkMiner-Browsec-DNS activity .....	37
Figure 4-12 Wireshark-Browsec-Service Status Plain.....	37
Figure 4-13 Wireshark-Browsec-Standard port for HTTPS .....	38
Figure 4-14 Wireshark-Browsec-Standard port for HTTPS Server-2 .....	38
Figure 4-15 Fiddler-uVPN- Server Name vs IP vs Ports .....	39
Figure 4-16 NetworkMiner-uVPN-DNS Activity .....	40
Figure 4-17 Wireshark-uVPN-Standard port for HTTPS Server.....	40
Figure 4-18 Wireshark-uVPN-Standard port for HTTPS Server-2.....	41
Figure 5-1 Feature Extraction .....	45
Figure 5-2 Traffic Classification .....	48
Figure 5-3 Deployment Model.....	51

## List of Table

Table 2-1 Classification results legitimate vs Malicious Users .....	11
Table 3-1 DSR Guidelines .....	23
Table 5-1 VPN Services Analyzed.....	50
Table 5-2 Bird Eye view of VPN Services Analysis.....	52
Table 5-3 Summary of User activity .....	53

# Abstract

Recently the use of secure protocols on web such as HTTPS (Hyper Text Transfer Protocol Secure), instead of HTTP (Hyper Text Transfer Protocol), has increased widely. HTTPS provides confidentiality of information between the two parties. This increase in encrypted traffic has forced organizations to use network firewalls along with Intrusion Detection and Prevention Systems (IDPS) to analyze the network traffic for detecting attacks and vulnerabilities inside the network.

Generally to inspect or govern HTTPS or encrypted traffic inside the network, the organization relies on the unencrypted traffic to be inspected by firewalls and intrusion detection system (IDS). A Virtual Private Network (VPN) is a service which hides even the unencrypted traffic of the user by creating a secure tunnel, generally protected by HTTPS, between the service provider and customer. This allows any VPN service to bypass the filters or signatures applied on any network security appliances. In addition to this, these services may be used to leak any sensitive information or an entry point for any new threat for the network.

In this study we have proposed a novel approach to safeguard the network from such VPN activity. The communication between the client and the server is analyzed and multiple features are extracted from network (IP), transport (TCP, UDP) and application layer (HTTPS, DNS). These extracted features are not encrypted and helps the system in classifying the network traffic. By analyzing DNS (Domain Name System) packets and HTTPS (Hyper Text Transfer Protocol Secure) based traffic the traffic is classified. Once the traffic is classified, server's IP, TCP port connected, domain name of each connection is analyzed. Based on the analysis the system is able to differentiate between legitimate and VPN-based connection. Our proposed system has no added overhead in terms of network traffic and is light weight due to the analysis on plain traffic only.

## Related Publication

M. Zain Ul Abideen, S. Saleem, and M. Ejaz, “VPN Traffic Detection in SSL-Protected Channel,” *Secur. Commun. Networks*, vol. 2019, 2019, doi: 10.1155/2019/7924690.[1]

DOI: <https://dl.acm.org/doi/10.1155/2019/7924690>

# CHAPTER 1

## INTRODUCTION

To enable access to a webpage or any resource available online, standard protocols of TCP/IP suite work alongside[2]. These protocols enable a user to be able to access resources over the network or internet. Accessing a web page hosted at some website, following standard activity may be performed at network and application stack level for the resource to be available to the user.

- 1) DNS request /response
- 2) TCP based connection
- 3) HTTPS based connection request
- 4) Data being transferred

Initially the Domain name of the website is translated to its corresponding Internet Protocol (IP) address. The protocol that is responsible for translation of Domain name to its relevant IP address is known as Domain Name System (DNS). Every internet connected device has its unique IP address. This IP address enables the system to be accessible over the internet. Once the IP address of the specific server/domain is available the system tries to connect to servers via Transmission Control Protocol (TCP). Once the TCP connection is established with the server, HTTPS connection request is initiated. Hypertext Transfer Protocol Secure (HTTPS) is one of the most commonly used protocols in today's internet. HTTPS provides transport layer security, which prevents any one for eavesdropping on the communication as it is encrypted. HTTPS uses Transport Layer Security (TLS) protocol which uses Public Key Infrastructure (PKI) ensuring confidentiality and integrity of data being communicated using HTTPS.

## **1.1 Research Problem**

HTTPS is the successor of Hypertext Terminal Protocol (HTTP). HTTP was the standard protocol for web access in early 2000s. Attacks like eavesdropping, man in the middle, replay and masquerading were threat to HTTP based traffic. The TCP/IP stack was implemented with no real concern to confidentiality and integrity of the data being transferred[3]. These issues raised a lot of concerns in the security and research domain of networks. These threats are now managed through different services[4]. To ensure source authentication, freshness of data and confidentiality of

data/information being transferred HTTPS was introduced[5]. Such trends in internet traffic have increased the encrypted content over the network[6].

An organization may want to monitor and analyze the traffic coming in and going out of its network to better govern it[7]. The organization may need to inspect or manage the traffic to ensure that no data leak is happening from inside the network. To be able to stop any insider attack, organization manages its network traffic using different solutions. One of these solutions is to deploy network-based firewalls and Intrusion Detection Systems (IDS). Firewalls are used to monitor any incoming or outgoing traffic based on the pre-defined set of rules. These rules help shape the network traffic and prevent any attack from the internet. Most firewall deployed in medium level organizations do not have SSL inspection and SSL off-loading which may allow the encrypted traffic to bypass any policing[8].

Firewalls are generally used alongside with Intrusion Detection Systems (IDS) or Intrusion Detection and Prevention Systems (IDPS). With the increase of encrypted traffic inside the network, it is very difficult for an organization to be able to analyze the data being transferred[9]. The encrypted traffic is generally being managed by the help of lower layered protocols like IP and TCP/UDP protocol information. These rules may be easily bypassed by using some Virtual Private Network (VPN) server making a covert channel that is not being checked by the firewall[10].

A VPN service allows a user to connect to a server present outside the organization. This server then communicates on behalf of the client and may connect to another

server which was otherwise not allowed by the organization due to its policies[11]. A VPN service consists of 2 main parts, VPN client and VPN server. VPN client is installed in user's computer while VPN server sits outside the organization over the internet. Once the connection is established, the client requests the internet resources from VPN server[12].

The traffic between VPN client and VPN server is encrypted and generally HTTPS based. This helps VPN service to go through the rules of firewall or IDPS untraced[13]. Some organizations, big enough, may install a monitoring solution which is able to inspect SSL traffic and detect the traffic even inside the VPN tunnel. These solutions are costly and may add latency in the network.

## **1.2 Proposed Solution**

We have proposed a solution which is lightweight, cost effective and does not add considerable delay in the network. Our proposed solution[1] uses the plain information present in different OSI layers and is able to co-relate between different connections to detect the presence of VPN traffic inside the organizational network.

Our solution may be further divided in 2 main components:

- a) Feature Extraction
- b) Traffic Classification

For each incoming connection features like 5Tuple, DNS query related information and SSL handshake is extracted. Once the features are extracted these features are co-related to identify and classify each new incoming connection.



We have worked on 6 of the top freely available VPN services and successfully detected their traffic. More services may be added using similar approach for increasing the detection coverage of the proposed solution.

## **1.3 Summary**

In this chapter we have discussed the general motivation and research problem related to detecting malicious insider. We have also presented the issues present in basic TCP/IP stack which led to the issues like data confidentiality and integrity. In order to analyze encrypted payload a lot of resources are required. We have come up with a solution which is able to analyze and classify network encrypted traffic to detect any VPN activity inside the network. With the help of our solution the communication in encrypted channel can be monitored using plain network protocols. In the upcoming chapter a detailed study of the latest research in Deep Packet Inspection (DPI) techniques and Intrusion Detection Systems (IDS) design.

# **CHAPTER 2**

## **LITERATURE REVIEW**

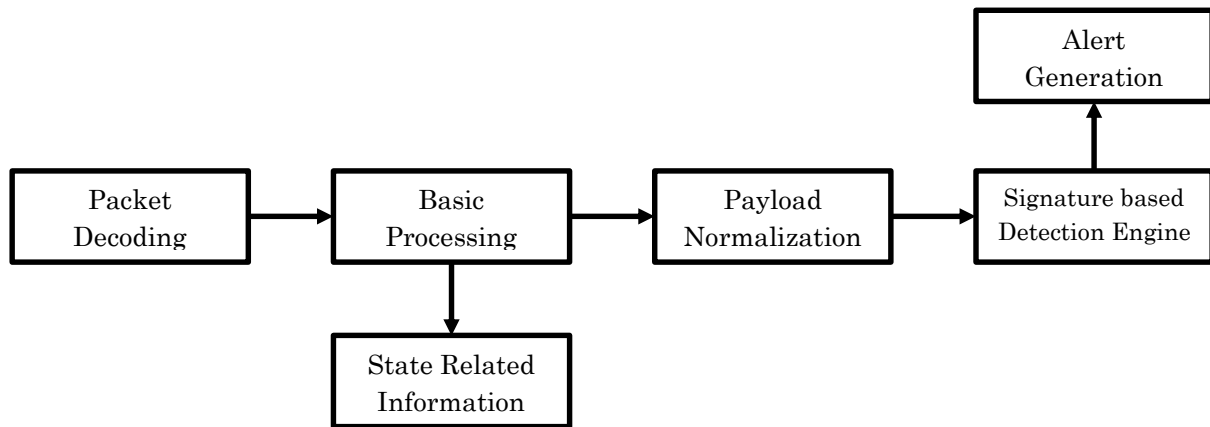
In the previous chapter we discussed about the issues due to unauthorized use of VPN inside the network. This chapter will explore the latest research in field of Deep Packet Inspection (DPI) techniques and Intrusion Detection Systems (IDS) design. Various algorithms and design schemes have also been discussed in this chapter to highlight the presence of the problem at hand. We have done an extensive review of the techniques and ideas proposed to be able to understand the requirements and challenges faced by the research community when dealing with encrypted traffic.

## 2.1 Overview of challenges faced in IDS Design

AbuHmed et al. [14] studies multiple Deep Packet Inspection (DPI) techniques, implementation and algorithms. There are multiple challenges to be considered while working with any DPI technique. The paper[14] discusses a few, like the searching capability of the algorithm to find relevant signature in case of IDS. The ever increasing number of signatures and the overlap between them asks for an efficient lookup pre-processor. The proposed study also recognizes the challenges faced by the research community for DPI due to encrypted traffic in the network. The study[14] suggests that a good DPI system should have following characteristics:

- a) The technique should have a deterministic performance, independent of signature or traffic characteristics.
- b) Due to ever increasing number of signatures, memory access and efficiency is a key component while designing any DPI technique.
- c) There should be an easy way to dynamically update the signatures especially in hardware-based DPI techniques.
- d) A good DPI engine must support regular expressions and signatures and should be scalable especially in hardware-based systems.

Mainly there are two types of DPI system implementations, software and hardware. In software-based implementations we have systems like SNORT, Bro, Suricata like IDS that perform DPI. Generally, a software-based IDS may have a following architecture as shown in Figure 2-1:



**Figure 2-1 General Architecture of Software Based IDS**

High speed packet processing and string/expression-based signature matching at line rate is also a requirement. To achieve line rate DPI software-based IDS have followed multi-threaded architecture in recent years. Hardware-based implementations have always surpass software-based implementation in terms of line rate signature matching. This is due to the fact that hardware-based solutions have pipeline architecture which supports multi-packet processing. The network interface cards are 10Gbps supported and leveraged fully to get maximum output. Apart from these benefits' hardware-based techniques need to be able to dynamically update signature and be scalable in different network environments. The study also discusses multiple

string-matching techniques based on their complexity to design and also the throughput these techniques provide. These techniques are discussed below:

- a) KMP Algorithm[15]: Knuth-Morris-Pratt is a technique which skips the next characters in string matching if a mismatch occurs. This helps reduce the time to recheck the mismatched bytes and gives better performance versus simple string lookup.
- b) Bloom Filter[16]: The technique suggests to compress the string to be looked up as hash values. These values are then compared with the incoming data.
- c) Content Addressable Memory[17]: In Hardware-based components CAM type matching is highly used in form of TCAM or BitWiseCAM. The main idea is to search expressions parallelly inside the payload and generate results with single operations. This algorithm is highly used in hardware-based implementation but it has high memory requirement and high-power requirements too.
- d) Finite State Machine[18]: There are further two type of finite state machines, nondeterministic finite automata and deterministic finite automata. Aho-Corasik algorithm is one of the leading hardware implantations used for signature machine. It was able to provide line rate 10Gbps signature matching.

## **2.2 Anomaly based IDS for HTTP protocol in SSL channel**

The work done by Yamada et al.[19] answers the basic question why common IDS are unable to protect attacks coming via encrypted channel. Common IDS need to inspect HTTP header and body by reconstructing the payload to identify any attacks with the signature database. In case of encrypted payload, IDS are not able to do that because of the encrypted layer above HTTP. In order to provide data integrity and data confidentiality, protocols like SSL/TLS were designed. In order to secure the web server, the paper presents a novel approach based on data size and timing analysis to detect covert channel, if any, inside the communication.

The paper[19] distinguishes an attack from normal traffic using statistical data. Typical traffic that is, legitimate traffics characteristics are compared with any new incoming traffic pattern. The traffic targeted for the web server is passed through an anomaly detector designed for web-based requests, to distinguish between normal and rare events. Normally request packets from client to servers are small in size and less in occurrences and the response from the server is large in size. This helps the anomaly detector to detect if a web server is under scan attacks. In scan attacks multiple addresses and options methods are used to identify the directory listing of the web server. In this case the requests from client will follow normal behavior as they will be short but server will also respond with small responses as the resources are not available at server. So, by using these stats the paper was able to identify scanning attacks on web servers. The paper[19] shares some results as classified by the system shown in Table 2-1.

Data Sets	Classification	Traffic Size (MBs) (Approx)	Total Client Requests	Connections Established
User Generated Data Set	Legitimate	750	81K	12K
	Illegitimate	2	500	463
DARPA IDS Data Set	Legitimate	2667	428K	56K
	Illegitimate	172	3K	481

**Table 2-1 Classification results legitimate vs Malicious Users**

### 2.3 Hybrid Approach to IDS Design

Aydin et al. [20] discusses that due to the increase in secure traffic simple firewall or IDS are not able to fully protect the network and the user is vulnerable to Remote to Local (R2L) and User to Root (U2R) attacks. In order to overcome this issue, the paper suggests a hybrid mechanism for Intrusion Detection System. Two types of IDS are mainly discussed misuse-based IDS and anomaly-based IDS. In misuse-based IDS, the IDS is responsible to report the events that violates the system policies enforced[21]. In Anomaly-based IDS, the IDS is responsible to report the events that do not follow a standard behavior of a user[22].

SNORT IDS is used as a signature-based network IDS[23]. Along with SNORT protocol based modeling and time-based modeling of network in short-time techniques are used as anomaly-based IDS. For misuse-based IDS and open-source

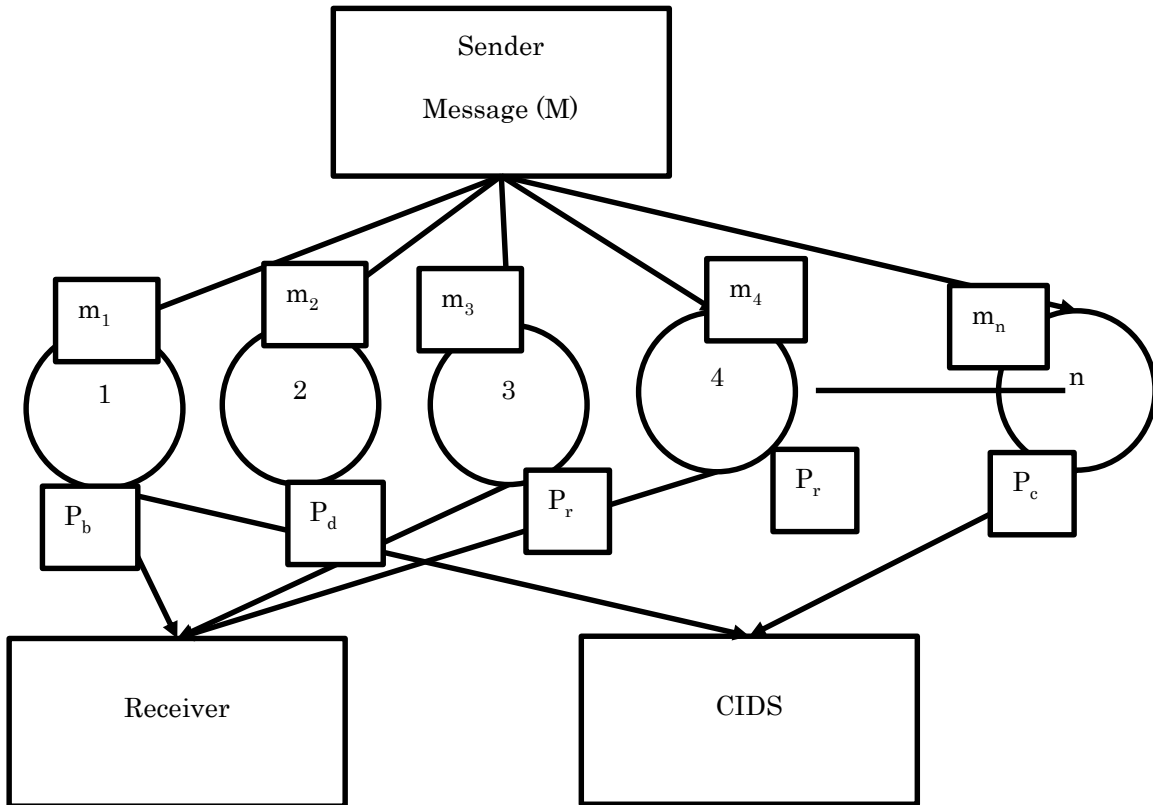
client OSSEC is used to send OS logs to detect any policy violations[24]. The paper [20] also discusses that based on network applications and topologies the IDS may be updated and discusses few use cases:

- a) IDS in MANET[25]: MANET is Mobile Adhoc Network. Multiple network mobile devices communicate with each other without a central node. The IDS used in such network is different from tradition networks. A watchdog node is responsible for eavesdropping the communication of next hop and identify any malicious nodes. A path rater tries to find the routes advertised where the nodes are not available and preserve the devices power.
- b) IDS for Cloud Computing[26]: In Cloud computing there is a high request vs response rate as the users are using multiple virtualized services. In order to achieve confidentiality and integrity of the data the data is end to end encrypted. Intrusion Detection Message Exchange Format (IDMEF) enabled IDS nodes are used to exchange messages between software and network-based IDS nodes for Cloud based IDS.
- c) IDS in Data Mining[27]: In data mining process, extracting the hidden knowledge, any set of malicious action that compromises the integrity and availability of the resource needs to be detected. For this purpose, anomaly and misuse-based IDS techniques may be used to verify the authenticity of labelled data.



## 2.4 Pre-shared Key Central IDS Design

Paper by Goh et al.[28] focuses on Network Intrusion Detection system (NIDS) ability to be effective in analyzing and detecting intrusion in encrypted traffic. For NIDS to be able to examine encrypted traffic going through it, the encrypted tunnel from



**Figure 2-2 Proposed Solution for Message transfer and inspection**

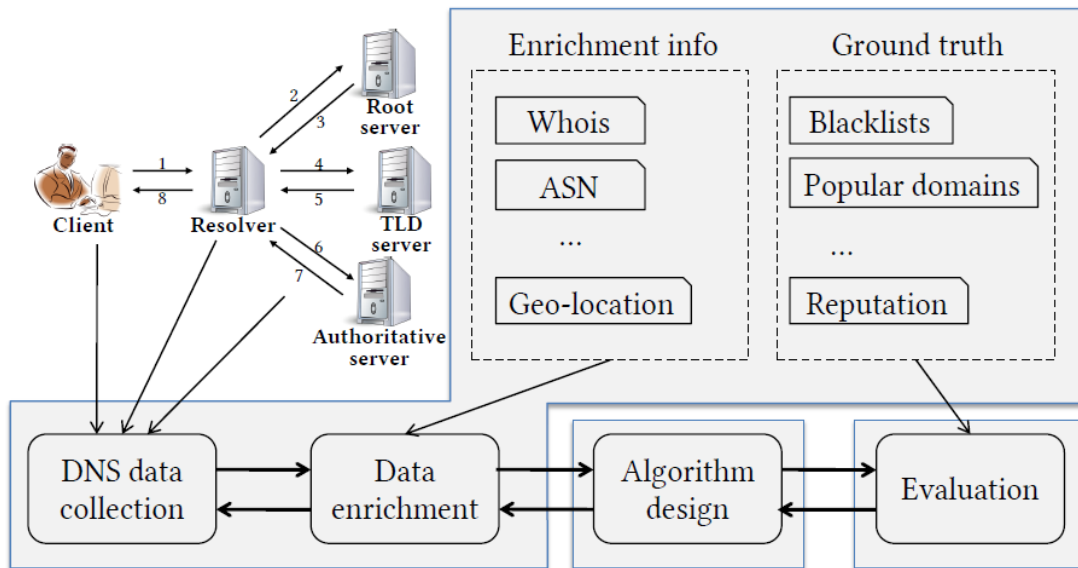
outside the network needs to be terminated at NIDS and the data is transferred to client in plain. Terminating traffic at NIDS is not possible when the data confidentiality is also required end to end. This paper suggests to add another hope in the network named as Central Intrusion Detection System (CIDS). Multiple IDS sensors to be installed at each network hope. These sensors are responsible for

duplicating the unencrypted traffic for sender and sending it to a network of proxies inside the network and to the receiver too.

The proxies have pre-shared keys with the IDS sensor and CIDS. The message which needs to be send is received by IDS sensor. The message is broken into multiple parts. Each part is sent to a randomly chosen proxy inside the network. Minimum 2 parts are needed to recover a message at each end as shown in Figure 2-2. These proxy network ensures that the sender/attacker cannot evade or falsify the message received at CIDS being sent to the sender. The communication between IDS sensor, proxy network and CIDS is secure using pre-shared key. This is done so that no additional burden of PKI in the scheme. The CIDS where any NIDS may be running then uses messages parts to re-construct the message and evaluate it using Intrusion Detection System. The paper[28] has used SNORT and shows that CIDS was able to detect any remote access attacks that was encrypted sent from sender to receiver.

## 2.5 Domain Reputation based IDS

Another study[29] focuses on identifying malicious DNS server and what challenges are faced to research community for generating a malicious DNS database which may be used around the world. In order for malwares to be disseminated and data to be communicated to Command and Control (C&C) server name of any C&C would need to resolve for the malicious program to reach it. In order to detect the origin and C&C many approaches are used widely. These include analysis of network traffic,



**Figure 2-3 Malicious Domains Detection through DNS Data Analysis[29]**

inspection and filtering of web content or the combination of both.

The proposed framework[29] for DNS data enrichment is shown in Figure 2-3. DNS data is not a centrally managed repository. DNS is a core component in today's TCP/IP communication and possess a lot of security threat[30]. The authors[29] propose to collect data, from multiple DNS resolvers. This may be done by either

actively collecting data from live network within an organization or passively via any ISP server. Next the system purposes to enrich the DNS data with multiple information like:

- a) Geo-location of the IPs and domains commonly used[31].
- b) Verifying Autonomous System Number (ASN) few to many, as to evade detection the malicious domains may use multiple ASN hops.
- c) Registration records are usually not maintained but if available may be used to further by authorities in case of any incident.
- d) IP domain blacklist and whitelist lists are maintained by different sources, may also be associated with domain name data[32].
- e) Associative resources records may also be used to identify what other associated services are being provided using MX records.
- f) Network traffic coming from the domain may also be used to analyze what type of ports or traffic is being managed by the servers[33].

The paper[29] proposes to further assign ground truth to multiple domains, for example multiple anti-virus sites like virus total provide malicious domain information so these domains may be marked as malicious while Alexa top ranked domains may be used for tagging high ranking popular domain.

## 2.6 IDS Design for Detecting APT

Another study[34] proposes a novel solution to detect Command and Control (C&C) servers for Advanced Persistent Threat (APT) using DNS and network-based traffic. The proposed solution puts forwards 14 features extracted from network traffic to classify it as malicious or legitimate traffic. In APTs attack the malware needs to be in a constant and persistent connection with its control server. DNS is widely used in these types of attacks to locate the C&C IP addresses. A rather new concept of dynamic DNS is used in case of Remote Access Trojan (RAT). This is done by obtaining a lot of 2LD and 3LD domains. The proposed system is further sub categorized into 3 main classes based on network traffic behavior:

- a) Malicious DNS Detector
- b) Signature based Detector
- c) Anomaly based Detector

The 14 features classified by the systems can be classified into 5 main categories:

- a) Domain Name
  - a. Famous domain: such as yahoo, google part of domain name
  - b. Particular domain: such as mail, web or news as third level domain
  - c. Phishing domain: such as gooogle.com instead of google.com
- b) DNS Answer
  - a. Silent IP: such as local IPs like 127.0.0.1 or 192.168.1.xx

- b. Distinct IP addresses: when domains are not being used and only IPs are used to communicate
  - c. Distinct Countries: Usually C&C IP resolves in different countries than victims
  - d. Domains vs IP address: A single attacker can only maintain so much domains at a time. Paper suggests that around 30 domains may be registered against single IP address
  - e. IP in same class as known C&C: If an attacker uses VPS services it is more likely that the VPS has more than 1 IP and these IPs are available in DNS record
- c) Time Value
- a. Similarity in data sent: It was observed that some APTs follow a behavior that they change the domain daily in morning and later shift it to silent mode for the night.
  - b. Query number vs Time: Within a same time-window, essentially the DNS query amount will be same. Some infected PCs may send large amount of DNS requests if the response is not received in timely manner.
  - c. Low Frequency: To keep the network presence minimum some APTs send the DNS queries after a long time to refresh the IP address of C&C.
- d) TTL Value

- a. TTL Value: Average TTL value for a resolver to retain a DNS record is usually set to 30, 31, 60 and 300 seconds. Low frequency APTs may set this to 84600 seconds which is approximately a day.
- e) Active lookup
- a. Web Server or not: Actively send a HTTP request respond to port 80 of C&C server where the infected PC is communicating if it serves a web page of some sort that means it is not collecting information and working as a normal server if not then its malicious
  - b. Whois Information: Openly available domain name data to be pulled off the internet to be able to ascertain as to what this domain is intended to do.

These features extracted from the traffic coming inside the network are passed through 3 behavior detectors discussed above. The output from these 3 is then consolidated and the traffic is classified as malicious or legitimate.

Recently, android-based VPN application were studied [35], it concluded that multiple third party trackers were tracking user behavior. Some VPN based applications may be used to bypass the sandbox environment of android. Once a malware or virus bypasses these security and infects the device, the whole network is compromised to the attack [36].

VPN clients when connected act as a gateway for the VPN server inside the network. After the connection is established, the VPN services are able to alter, delete or listen

to the information being passed through the network [37], [38]. This type of service attract many advertisements or user-tracking services [39], [40].

VPN service is actually performing TLS interception[41] by using its locally trusted certificate. This may lead to data leakage, breach of confidentiality, especially when the device connected contains sensitive data [35], [42]. One of the way to prevent these issue is by certificate pinning for specific applications [35], [43]. So, the detection of VPN services inside the organizational network can save organization from huge losses.

## **2.7 Summary**

In this chapter we discussed multiple IDS design and approached presented by the community. We started our research by studying about basic DPI techniques implemented in open-source IDS. We also learnt the importance of domain name analysis in order to better classify the encrypted content. We have also discussed the protocol based behavior analysis used in recent IDS. The proposed system was designed keeping these key considerations which the literature focuses on. In the upcoming chapter the research expansion plan has been presented. Incremental development has been used to construct an idea and then eventually a working system.



# CHAPTER 3

## RESEARCH METHODOLOGY

The research on VPN traffic detection in SSL protected channel is holistic in nature. The problem domain of searching artifacts in encrypted traffic to facilitate forensic analysis of VPN applications have been presented. This research can be termed as multi-domain as it falls in domain like network security[44], digital forensic[45] and information security and assurance[46]. Research is a comprehensive process and set of activities to understand an event. This event, when found interesting by the relevant research community marks the research as genuine and worthy.

### 3.1 Fundamentals of our Research

Our work lies under the research domain of Design Science Research (DSR). The DSR is aimed to solve real world problems by developing a viable design and contributing to the scientific knowledge of the domain. We consider our work is novel and falls under problem solving domain; which is one of the most important characteristics of DSR. Our research methodology follows the following guidelines outlined for Design Science Research[47] as shown in Table 3-1.

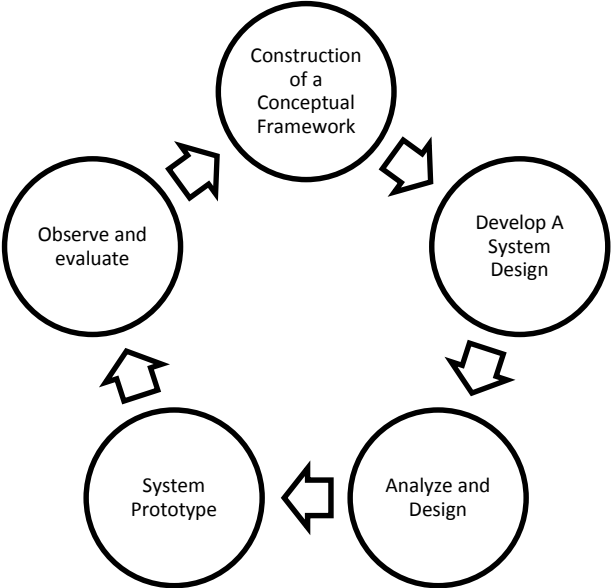
S. No.	DSR Guidelines	Description
1	Design An artifact	DSR must produce a viable artifact it may be one of these:  a) Construct  b) Model  c) Method  d) Instantiation
2	Problem Relevance	The objective of DSR is to develop solutions to relevant domain problems
3	Design Evaluation	The utility and quality of the proposed design be evaluated
4	Research Contribution	DSR research must provide clear and valuable solutions. It may be in design artifacts, foundation or methodologies.

5	Research Rigor	Rigorous methods to be used in both construct and evaluation of the artifact
6	Design as a search process	Using available means to reach desired end in search of artifacts and satisfying laws in the problem environment
7	Communication of Research	DSR must be presented effectively to both managerial and technical peers.

**Table 3-1 DSR Guidelines**

### 3.2 Our Research Process

We focused on Design Research Cycle research process and used it to navigate research process. The sub processes of DSR[48] are presented in Figure 3-1:



**Figure 3-1 Sub Processes for DSR**

### **3.2.1 Construction of Conceptual Framework**

The first step in DSR is to identify the research problem. The research problem may be fresh and evolve from a rich literature review from multiple resources which include industrial requirements, continuation of an ongoing research or an innovative idea based on some technological gap.

During our literature review of the issues being faced by the network security industry, we stumbled upon the problem that is very fresh and is being actively researched upon. The problem is how to analyze the encrypted traffic inside the network for intrusion detection. The intruder can be both insider and an outsider. We analyzed that mostly researchers focused on securing inside network from outsiders. The evasion techniques used by insider are generally difficult to detect and require a lot of resources. We were able to find some techniques related to these but these techniques add overhead in terms of additional processing, nodes or increase in network traffic.

The network and forensic analysis of VPN being used by insiders to bypass the organizational policies come under misuse-based IDS. It is important for organizations to keep their private data or internal information inside the network and any evasive technique, in our case the use of VPN clients, may be detected and reported. To be able to solve this problem we devised series of experimental setups and were able to analyze VPN traffic patterns and with the use of iterative process of DSR were able to purpose the framework which is able to detect the presence of well-known VPN clients inside the network.

### 3.2.2 Develop a System Design

After doing an extensive literature review of the problem and applying existing techniques to see the results against the VPN traffic, we then started to design our own system to solve the research problem. As there are many VPN services available, we chose the one's whose browser extension were freely available and highly downloaded.

The task was broken down further to 3 main tasks.

- a) We carefully analyzed the traffic of multiple users using VPN. We captured the traffic using Wireshark. Wireshark is an open-source tool used widely in research community for capturing and analyzing the network traffic. We captured the traffic when VPN was activated versus when VPN was disconnected. This gave us the difference between connection stream when connecting to the same website with or without VPN.
- b) Next, we use this network traffic to analyze the out of bound communication for this network. This is a novel approach in our design that we have targeted not only the current connection stream but are also using the history of network traffic preceding the connection. In this analysis we focused on domain names. We categorized and analyzed the traffic behavior of each VPN based on the DNS activity it produces. Later on, we also used fiddler proxy[49] to better understand the communication between the VPN client and the server and extracted the geo-location-based server information and in cooperated in our analysis.

- c) Lastly, we analyzed the secure traffic originating from client machine. The user is supposed to use TCP port 443 for SSL based connection. If the VPN client is active as an evasion technique, they use non-standard port to transfer the encrypted payload between the client and server.

With the help of above knowledge, we moved to next phase of our DSR process.

### **3.2.3 Analyze and Design**

With all the knowledge and input from the previous phase we started developing the proposed solution. We started by extracting the basic information from the connection i.e., 5 tuple information. We also analyzed the SSL certificates of servers within the connection to see what services these certificate offer and are they consistent with the request for connection generated by client. We also analyzed the DNS request response generated by user internet activity and associated each new connection with the DNS requests and its response generated by the user inside the network. This iterative process was constantly improved and results were evaluated to mitigate any false reporting.

### **3.2.4 System Prototype**

After the above iterative step we were able to purpose and design a prototype which is capable of detecting top 5 freely available VPN clients with minimal false positives. This prototype can also be used to identify R2L Trojans if the C&C information is silent IP. If C&C is domain based this prototype is able to update the signature to identify the malicious C&C domain.

### **3.2.5 Observe and Evaluate**

Over a period of time, we observed that the domains and behavior of the said VPN clients kept on changing. So in order for the system to be able to detect the activities we actively observed the changes and updated our system with recent trends and were able to get satisfactory results when the system was evaluated.

## **3.3 Summary**

The chapter discusses the research methodology science behind our research work. The outcome of our research in conclusion stage resulted in a journal publication which could be taken as a stepping stone for network security community to explore other VPN clients on similar lines:

**Related Journal Publication:** M. Zain Ul Abideen, S. Saleem, and M. Ejaz, “VPN Traffic Detection in SSL-Protected Channel,” *Secur. Commun. Networks*, vol. 2019, 2019, doi: 10.1155/2019/7924690.[1].

DOI: <https://dl.acm.org/doi/10.1155/2019/7924690>

# CHAPTER 4

## FORENSIC ANALYSIS OF VPN SERVICES

In this chapter we discuss the science and analysis done behind each VPN service in order to understand the unique behavior presented by each service. VPN services working on HTTPS like protocol are difficult to detect as they follow a similar behavior as of common HTTPS connection. Analyzing the traffic for these services require in depth analysis of the traffic behavior and pattern for the specific service. Analyzing traffic of any VPN service generally consisted of following steps:

- 1) Install and configure VPN client on the user machine.
- 2) Perform internet activity using VPN clients while capturing the network traffic using Wireshark.



- 3) Analyze the traffic behavior and pattern using Wireshark[50] and NetworkMiner[51] tools.
- 4) Extract relevant information for classification of the VPN service.
- 5) Update classifier of proposed system to enable the detection of the VPN service.

We have focused on 5 of the top freely available VPN services namely:

- 1) Hotspot Shield[52]
- 2) ZenMate VPN[53]
- 3) Browsec VPN[54]
- 4) Hoxx VPN[55]
- 5) uVPN[56]

Network traffic for every VPN service, provided above, was analyzed. Analysis and findings are presented and discussed below.

## **4.1 ZenMate VPN**

ZenMate VPN is one of the most used VPN services. It currently has over 45 million users all across the globe and over 35 global server locations. It is also highly updated VPN service. This means that the signatures of ZenMate need constant updating. During this research ZenMate completely changed its traffic behavior twice. The most recent is discussed below:

- 1) Initially ZenMate authenticates the user using its public domain i.e. “zenguard.biz”.

- 2) A server IP list is also communicated inside the SSL tunnel. It can be observed in the Figure 4-1 below:

```
HTTP/1.1 200 OK
Date: Mon, 02 Dec 2019 06:56:05 GMT
Content-Type: application/json; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/7.2.24
X-FRAME-OPTIONS: SAMEORIGIN
X-Record-Count: 5
X-Status: SUCCESS
Expires: Mon, 02 Dec 2019 06:56:05 GMT
Cache-Control: max-age=0
CF-Cache-Status: DYNAMIC
Expect-CT: max-age=604800, report-uri="https://report-uri.cloudflare.com/cdn-cgi/beacon/expect-ct"
Server: cloudflare
CF-RAY: 53eb66207f55c619-KHI
Content-Length: 626

[
  {
    "serverLookup": "http://103.10.197.11:81",
    "countrycode": "HK",
    "servers_id": "1864"
  },
  {
    "serverLookup": "http://103.10.197.156:81",
    "countrycode": "HK",
    "servers_id": "1865"
  }
]
```

**Figure 4-1 Fiddler-ZenMate-Server IP list**

- 3) The IP list shared is then used to connect to TCP Port 81 using HTTP protocol to get the server name for the above shared regions. The servers on the IPs communicated may only be responsible for managing domain name for the regions.
- 4) Along with the domain name, TCP port of the server it is listening to is also communicated to client. As shown in Figure 4-2 the domain name for Hong Kong based server is “hongkong-s56-i01.cg-dialup.net” and is listening to port no “9002”:

```

GET / HTTP/1.1
Host: 103.10.197.133:81
Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.0.3904.108 Safari/537.36
Accept: */*
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9

HTTP/1.1 200 OK
Server: nginx
Date: Mon, 02 Dec 2019 06:27:30 GMT
Content-Type: text/html
Content-Length: 36
Last-Modified: Mon, 02 Dec 2019 06:22:05 GMT
Connection: close
ETag: "5de4ad8d-24"
Expires: Wed, 01 Jan 2020 06:27:30 GMT
Cache-Control: max-age=2592000
Accept-Ranges: bytes

hongkong-s56-i01.cg-dialup.net:9002

```

**Figure 4-2 Wireshark-ZenMate-Non-secure traffic on 9002**

5) These domain names are used to resolve DNS query to get the VPN server’s IP responsible for managing client’s user data i.e. hiding the user generated network traffic. The DNS response seen by network miner is also provided in Figure 4-3 for server name ““hongkong-s56-i01.cg-dialup.net””.

NetworkMiner 2.5

File Tools Help

-- Select a network adapter in the list --

Hosts (74) Files (4) Images Messages Credentials Sessions (53) DNS (7) Parameters (106) Keywords Anomalies

Filter keyword:   Case sensitive ExactPhrase Any column Clear Apply

Frame nr.	Timestamp	Client	Client Port	Server	S..	IP TTL	DNS TTL (time)	Transaction ID	Type	DNS Query	DNS Answer
6	2019-12-02 06:27:19 UTC	192.168.1.101...	57817	192.168.1.1	53	60	00:05:00	0x6DE4	0x000...	apiv2.zenguard.biz	104.19.155.59
469	2019-12-02 06:27:30 UTC	224.0.0.251	5353	192.168.1.101 [...]	5..	1	00:01:00	0x0000	0x001C	DESKTOP-NNPI2IP.local	fe80::882a:9a80:3c8d:c8bf
469	2019-12-02 06:27:30 UTC	224.0.0.251	5353	192.168.1.101 [...]	5..	1	00:01:00	0x0000	0x000...	DESKTOP-NNPI2IP.local	192.168.1.101
471	2019-12-02 06:27:30 UTC	#02.fb	5353	fe80::882a:9a80...	5..	N/A	00:01:00	0x0000	0x001C	DESKTOP-NNPI2IP.local	fe80::882a:9a80:3c8d:c8bf
471	2019-12-02 06:27:30 UTC	#02.fb	5353	fe80::882a:9a80...	5..	N/A	00:01:00	0x0000	0x000...	DESKTOP-NNPI2IP.local	192.168.1.101
477	2019-12-02 06:27:30 UTC	192.168.1.101...	54869	192.168.1.1 (Oth...	53	60	00:01:00	0x8B01	0x000...	cm.zenguard.biz	52.214.251.109
481	2019-12-02 06:27:30 UTC	192.168.1.101...	55056	192.168.1.1 (Oth...	53	60	01:00:00	0x02C2	0x000...	hongkong-s56-i01.cg-dialup.net	103.10.197.131

**Figure 4-3 NetworkMiner-ZenMate-DNS request Response for Server**

6) Once the DNS server responds to the query the IP received in the reply is used to establish a TLS based connection over the port advertised before. The Wireshark stream for such connection is provided in Figure 4-4.

Time	Source	Destination	Protocol	Length	Info
482	13.219166	192.168.1.101	103.10.197.131	TCP	66 7215 → 9002 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
498	13.567346	103.10.197.131	192.168.1.101	TCP	68 9002 → 7215 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
499	13.567564	192.168.1.101	103.10.197.131	TCP	54 7215 → 9002 [ACK] Seq=1 Ack=1 Win=131328 Len=0
500	13.568315	192.168.1.101	103.10.197.131	TLSv1.2	571 Client Hello
520	13.905268	103.10.197.131	192.168.1.101	TCP	56 9002 → 7215 [ACK] Seq=1 Ack=518 Win=30336 Len=0
521	13.905447	103.10.197.131	192.168.1.101	TLSv1.2	1514 Server Hello
522	13.905449	103.10.197.131	192.168.1.101	TCP	1514 9002 → 7215 [ACK] Seq=1461 Ack=518 Win=30336 Len=1460 [TCP segment of a reassembled PDU]
523	13.905611	192.168.1.101	103.10.197.131	TCP	54 7215 → 9002 [ACK] Seq=518 Ack=2921 Win=131328 Len=0
524	13.905854	103.10.197.131	192.168.1.101	TCP	1230 9002 → 7215 [PSH, ACK] Seq=2921 Ack=518 Win=30336 Len=1176 [TCP segment of a reassembled PDU]
525	13.905860	103.10.197.131	192.168.1.101	TCP	1514 9002 → 7215 [ACK] Seq=4097 Ack=518 Win=30336 Len=1460 [TCP segment of a reassembled PDU]
526	13.905862	103.10.197.131	192.168.1.101	TLSv1.2	423 Certificate, Server Hello Done
527	13.906062	192.168.1.101	103.10.197.131	TCP	54 7215 → 9002 [ACK] Seq=518 Ack=5926 Win=131328 Len=0
530	13.943307	192.168.1.101	103.10.197.131	TLSv1.2	372 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
549	14.285741	103.10.197.131	192.168.1.101	TLSv1.2	312 New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
550	14.286488	192.168.1.101	103.10.197.131	TLSv1.2	327 Application Data
575	14.620716	103.10.197.131	192.168.1.101	TLSv1.2	122 Application Data
576	14.621564	192.168.1.101	103.10.197.131	TLSv1.2	669 Application Data
587	14.977201	103.10.197.131	192.168.1.101	TLSv1.2	295 Application Data
588	14.980232	192.168.1.101	103.10.197.131	TLSv1.2	147 Application Data
589	14.980750	192.168.1.101	103.10.197.131	TLSv1.2	169 Application Data

**Figure 4-4 Wireshark-ZenMate-Nonstandard port for HTTPS**

- 7) As seen in Figure 4-4 that over port 9002 ZenMate uses TLS based communication. For general web browsing inside an organization HTTPS or QUIC (UDP based HTTPS) is used which is standard protocol. Any traffic not explicitly allowed should be flagged and identified.

Frame nr.	Timestamp	Client	Client Port	Server	Server Port	IP TTL	DNS TTL (time)	Transaction ID	Type	DNS Query	DNS Answer
171197	2019-12-02 06:36:54 UTC	192.168.1.101 ...	54899	192.168.1.1 ...	53	60	00:00:57	0xBA53	0x0001 (Host Address)	watson.telemetry.microsoft...	20.44.86.43
136731	2019-12-02 06:36:17 UTC	192.168.1.101 ...	63593	192.168.1.1 ...	53	60	01:00:00	0x4392	0x0001 (Host Address)	ipv4-c654-gjc002-dev-ix.1...	23.246.58.145
136867	2019-12-02 06:36:17 UTC	192.168.1.101 ...	63593	192.168.1.1 ...	53	60	00:59:59	0x4392	0x0001 (Host Address)	ipv4-c654-gjc002-dev-ix.1...	23.246.58.145
86110	2019-12-02 06:35:35 UTC	192.168.1.101 ...	56389	192.168.1.1 ...	53	60	01:00:00	0x774F	0x0001 (Host Address)	seoul-s52-i01.cg-dialup.net	27.255.75.243
86413	2019-12-02 06:35:35 UTC	192.168.1.101 ...	56389	192.168.1.1 ...	53	60	01:00:00	0x774F	0x0001 (Host Address)	seoul-s52-i01.cg-dialup.net	27.255.75.243

**Figure 4-5 NetworkMiner-ZenMate-DNS request Response for Server-2**

- 8) If the client wishes to change its location. The server associated with the location is also changed. As seen in Figure 4-5 client has asked for DNS of the server located at “Seoul” with server name of “seoul-s52-i01.cg-dialup.net”.

- 9) With the help of server domain names, non-standard port usage for the web traffic. We can classify this traffic as VPN traffic and alert may be generated.

## 4.2 Hotspot Shield VPN

Hotspot Shield is one of the leading VPN services with and was declared the fastest in VPN speed in 2019 OOKLA[57] Speed test award. This VPN uses TCP port 443 and standard DNS activity. This behavior bypasses any firewall which has internet allowed for internal network. Detecting such traffic is a matter of high importance in order to detect any VPN activity originating from the network. Detailed traffic analysis of hotspot shield VPN is discussed below.

- 1) Initially hotspot shield using the server name “hsselite.com” creates a new session for the user.

```
{
  "api_url": "https://heike.northghost.com",
  "servers": [
    {
      "address": "do-ex-us-sf-stage-1.northghost.com",
      "country": "us",
      "name": "do-ex-us-sf-stage-1.northghost.com",
      "password": "{id}.h78239hd",
      "port": 443,
      "priority": 100,
      "scheme": "https",
      "title": "US Server",
      "username": "{id}.h783ohaw09jdf0"
    },
    {
      "address": "ext-dc-ex-ru-1.northghost.com",
      "country": "ru",
      "port": 443,
      "priority": 1000,
      "scheme": "https"
    },
    {
      "address": "ext-dc-ex-ru-mow-pr-p-1.northghost.com",
      "country": "ru",
      "port": 443,
      "priority": 1000,
      "scheme": "https"
    },
    {
      "address": "ext-dc-ex-ru-mow-pr-p-2.northghost.com",

```

Figure 4-6 Web Response-Hotspot Shield- Server Name list

- 2) Using the URL “<https://s3-us-west-2.amazonaws.com/hssext/hss-free.json>.” server name list is fetched for hotspot shield client to use as shown in Figure 4-6.
- 3) The list has the server name, country identifier, TCP Port to be used for the server and protocol to be used.
- 4) Once connected, the client sends a DNS query for the selected server based on the country selected. As shown in the DNS analysis provided in .

56051	2	192.168.100.8...	62111	192.168.100.1	53	64	00:03:26	0x3657	0x0001 (Host Addr...	ext-zx-ex-nl-ams-prp-9.northghost.com	2.58.194.141	N/A
56054	2	192.168.100.8...	65351	192.168.100.1	53	64	00:03:26	0x8993	0x0001 (Host Addr...	ext-zx-ex-nl-ams-prp-9.northghost.com	2.58.194.141	N/A
56101	2	192.168.100.8...	58274	192.168.100.1	53	64	00:00:00	0xC90D	0x0000	ext-zx-ex-nl-ams-prp-9.northghost.com	No error conditio...	N/A
59586	2	192.168.100.8...	59556	192.168.100.1	53	64	00:00:00	0xFD60	0x0000	ext-zx-ex-nl-ams-prp-9.northghost.com	No error conditio...	N/A
61692	2	192.168.100.8...	64193	192.168.100.1	53	64	00:03:33	0x2844	0x0001 (Host Addr...	ext-zx-ex-de-fra-prp-15.northghost.com	178.162.198.118	N/A
61695	2	192.168.100.8...	49411	192.168.100.1	53	64	00:03:33	0x4A3D	0x0001 (Host Addr...	ext-zx-ex-de-fra-prp-15.northghost.com	178.162.198.118	N/A
61698	2	192.168.100.8...	55127	192.168.100.1	53	64	00:00:00	0x6A8D	0x0000	ext-zx-ex-de-fra-prp-15.northghost.com	No error conditio...	N/A
63935	2	192.168.100.8...	61418	192.168.100.1	53	64	00:05:00	0x78C2	0x0001 (Host Addr...	ext-tb-ex-ru-ams-prp-2.northghost.com	185.80.221.235	N/A
63939	2	192.168.100.8...	62660	192.168.100.1	53	64	00:05:00	0x6D73	0x0001 (Host Addr...	ext-tb-ex-ru-ams-prp-2.northghost.com	185.80.221.235	N/A
63954	2	192.168.100.8...	56376	192.168.100.1	53	64	00:00:00	0x1A79	0x0000	ext-tb-ex-ru-ams-prp-2.northghost.com	No error conditio...	N/A
64575	2	192.168.100.8...	52593	192.168.100.1	53	64	00:00:00	0xAAFE	0x0000	ext-tb-ex-ru-ams-prp-2.northghost.com	No error conditio...	N/A
65517	2	192.168.100.8...	61520	192.168.100.1	53	64	00:00:00	0xF83F	0x0000	ext-tb-ex-nl-ams-prp-2.northghost.com	No error conditio...	N/A

**Figure 4-7 NetworkMiner-Hotspot Shield-DNS request Response for Server**

No.	Time	Source	Destination	Protocol	Length	Info
61714	64.112862	192.168.100.8	178.162.198.118	TCP	66	14579 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
61722	64.258475	178.162.198.118	192.168.100.8	TCP	66	443 → 14579 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1412 SACK_PERM=1 WS=128
61723	64.258579	192.168.100.8	178.162.198.118	TCP	54	14579 → 443 [ACK] Seq=1 Ack=1 Win=66384 Len=0
61724	64.260785	192.168.100.8	178.162.198.118	TLSv1.3	604	Client Hello
61728	64.406094	178.162.198.118	192.168.100.8	TCP	54	443 → 14579 [ACK] Seq=1 Ack=551 Win=30336 Len=0
61729	64.412447	178.162.198.118	192.168.100.8	TLSv1.3	446	Server Hello, Change Cipher Spec, Application Data, Application Data
61730	64.415012	192.168.100.8	178.162.198.118	TLSv1.3	118	Change Cipher Spec, Application Data
61732	64.418318	192.168.100.8	178.162.198.118	TLSv1.3	634	Application Data
61735	64.567159	178.162.198.118	192.168.100.8	TCP	54	443 → 14579 [ACK] Seq=393 Ack=1195 Win=31488 Len=0
61738	64.675398	178.162.198.118	192.168.100.8	TLSv1.3	304	Application Data
61739	64.716379	192.168.100.8	178.162.198.118	TCP	54	14579 → 443 [ACK] Seq=1195 Ack=643 Win=65536 Len=0
61740	64.950748	192.168.100.8	178.162.198.118	TLSv1.3	634	Application Data
61742	65.108801	178.162.198.118	192.168.100.8	TLSv1.3	304	Application Data
61743	65.153866	192.168.100.8	178.162.198.118	TCP	54	14579 → 443 [ACK] Seq=1775 Ack=893 Win=65280 Len=0
61811	69.627361	192.168.100.8	178.162.198.118	TLSv1.3	560	Application Data

**Figure 4-8 Wireshark-HotspotShield-Standard port for HTTPS**

- 5) The traffic of hotspot shield follows the basic HTTPS traffic pattern, as shown in Figure 4-8.

55939	32.747987	192.168.100.8	192.168.100.1	DNS	98 Standard query 0x3657 A ext-zx-ex-nl-ams-pr-p-9.northghost.com
56051	32.774585	192.168.100.1	192.168.100.8	DNS	343 Standard query response 0x3657 A ext-zx-ex-nl-ams-pr-p-9.northghost.com A 2.58.194.141 NS ns-146.awsdns-18.
56053	32.775795	192.168.100.8	192.168.100.1	DNS	98 Standard query 0x8993 A ext-zx-ex-nl-ams-pr-p-9.northghost.com
56054	32.780404	192.168.100.1	192.168.100.8	DNS	114 Standard query response 0x8993 A ext-zx-ex-nl-ams-pr-p-9.northghost.com A 2.58.194.141
56055	32.781562	192.168.100.8	192.168.100.1	DNS	98 Standard query 0xc90d AAAA ext-zx-ex-nl-ams-pr-p-9.northghost.com
56056	32.810120	192.168.100.8	192.168.100.1	DNS	98 Standard query 0xc90d AAAA ext-zx-ex-nl-ams-pr-p-9.northghost.com
56101	33.052692	192.168.100.1	192.168.100.8	DNS	180 Standard query response 0xc90d AAAA ext-zx-ex-nl-ams-pr-p-9.northghost.com SOA ns-1370.awsdns-43.org
59585	38.885784	192.168.100.8	192.168.100.1	DNS	98 Standard query 0xfd60 AAAA ext-zx-ex-nl-ams-pr-p-9.northghost.com
59586	38.888313	192.168.100.1	192.168.100.8	DNS	98 Standard query response 0xfd60 AAAA ext-zx-ex-nl-ams-pr-p-9.northghost.com
61690	63.569318	192.168.100.8	192.168.100.1	DNS	99 Standard query 0x2844 A ext-zx-ex-de-fra-pr-p-15.northghost.com
61692	63.579456	192.168.100.1	192.168.100.8	DNS	428 Standard query response 0x2844 A ext-zx-ex-de-fra-pr-p-15.northghost.com A 178.162.198.118 NS ns-146.awsdns
61694	63.581046	192.168.100.8	192.168.100.1	DNS	99 Standard query 0x4a3d A ext-zx-ex-de-fra-pr-p-15.northghost.com
61695	63.585990	192.168.100.1	192.168.100.8	DNS	115 Standard query response 0x4a3d A ext-zx-ex-de-fra-pr-p-15.northghost.com A 178.162.198.118
61696	63.586523	192.168.100.8	192.168.100.1	DNS	99 Standard query 0x6a8d AAAA ext-zx-ex-de-fra-pr-p-15.northghost.com
61697	63.607070	192.168.100.8	192.168.100.1	DNS	99 Standard query 0x6a8d AAAA ext-zx-ex-de-fra-pr-p-15.northghost.com
61698	63.604701	192.168.100.1	192.168.100.8	DNS	181 Standard query response 0x6a8d AAAA ext-zx-ex-de-fra-pr-p-15.northghost.com SOA ns-1370.awsdns-43.org
63901	82.309396	192.168.100.8	192.168.100.1	DNS	98 Standard query 0x7bc2 A ext-tb-ex-ru-ams-pr-p-2.northghost.com
63907	82.326114	192.168.100.8	192.168.100.1	DNS	98 Standard query 0x7bc2 A ext-tb-ex-ru-ams-pr-p-2.northghost.com
63935	82.415475	192.168.100.1	192.168.100.8	DNS	427 Standard query response 0x7bc2 A ext-tb-ex-ru-ams-pr-p-2.northghost.com A 185.80.221.235 NS ns-146.awsdns-1
63937	82.417035	192.168.100.8	192.168.100.1	DNS	98 Standard query 0x6d73 A ext-tb-ex-ru-ams-pr-p-2.northghost.com
63938	82.431703	192.168.100.1	192.168.100.8	DNS	114 Standard query response 0x6d73 A ext-tb-ex-ru-ams-pr-p-2.northghost.com A 185.80.221.235

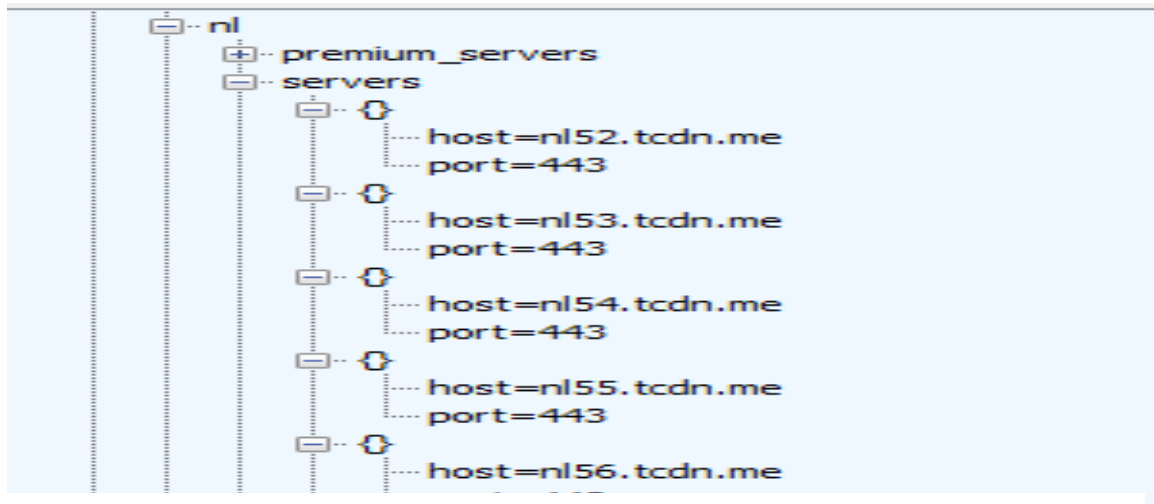
**Figure 4-9 Wireshark-Hotspot Shield-DNS request Response for multiple Servers**

- 6) It is also important to mention that by changing the location the server to be connected may also be changed as shown in DNS requests in Figure 4-9.
- 7) The change in the server IP for VPN service means that we cannot only filter a set of IP Addresses for the server to identify or block it. We need to actively monitor DNS traffic to enlist the IP addresses pertaining to the VPN service.

## 4.3 Browsec VPN

Browsec VPN is another popular VPN service available for free. Browsec VPN offers high speeds to its user of up to 100Mbps. It has over 40 locations to anonymize user data. Browsec VPN has over 4 million users worldwide. Browsec VPN uses both HTTP and HTTPS ports i.e. 443 and 80. The activity is standard internet activity which does not raise any alarms as the ports used are standard. Traffic analysis of Browsec is discussed below:

- 1) Browsec VPN initially establishes a session with server name “browsec.com”.  
The same server name is used to communicate server names inside the SSL tunnel established. The snippet of these server names is shown in Figure 4-10.
- 2) On these server name received irrespective of region chosen. Figure 4-11 shows the DNS request response for these servers:



**Figure 4-10 Fiddler-Browsec- Domain Name list**

- 3) Now for each response received the client asks the server for its service status.  
This is done using port 80 i.e. HTTP. The traffic seen by wireshark can be seen in Figure 4-12.
- 4) The information in Figure 4-12 shows that the service on the server is live and is waiting for new connections.



Hosts (170)	Files (527)	Images	Messages	Credentials	Sessions (854)	DNS (154)	Parameters (14701)	Keywords	Anomalies		
Filter keyword: <input type="text"/> <input type="checkbox"/> Case sensitive ExactPhrase Any column Clear											
Fra...	Timestamp	Client	Client Port	Server	Server Port	IP TTL	DNS TTL (time)	Transaction ID	Type	DNS Query	DNS Answer
6883	2019-12-11 08:32:40 UTC	192.168.1.100 ...	64644	192.168.1.1	53	60	03:19:00	0x3820	0x0001 ...	br1.lunrac.com	191.96.70.45
6900	2019-12-11 08:32:41 UTC	192.168.1.100 ...	56939	192.168.1.1	53	60	03:18:59	0xF7FB	0x0001 ...	ca1.lunrac.com	149.56.99.163
6917	2019-12-11 08:32:41 UTC	192.168.1.100 ...	55180	192.168.1.1	53	60	03:19:00	0xC2C9	0x0001 ...	ch1.lunrac.com	46.231.204.119
6920	2019-12-11 08:32:41 UTC	192.168.1.100 ...	65530	192.168.1.1	53	60	03:19:00	0x011A	0x0001 ...	cl1.lunrac.com	37.235.52.76
6922	2019-12-11 08:32:41 UTC	192.168.1.100 ...	64711	192.168.1.1	53	60	23:17:26	0xCA27	0x0001 ...	cz2.lunrac.com	81.2.247.9
6937	2019-12-11 08:32:41 UTC	192.168.1.100 ...	64318	192.168.1.1	53	60	07:10:32	0x131A	0x0001 ...	cz4.lunrac.com	85.255.4.182
6955	2019-12-11 08:32:41 UTC	192.168.1.100 ...	50484	192.168.1.1	53	60	00:51:47	0xDDD4	0x0001 ...	cz3.lunrac.com	195.181.211.129
6972	2019-12-11 08:32:41 UTC	192.168.1.100 ...	63609	192.168.1.1	53	60	07:59:04	0xD23F	0x0001 ...	de1.lunrac.com	46.101.247.242
6974	2019-12-11 08:32:41 UTC	192.168.1.100 ...	52876	192.168.1.1	53	60	19:17:53	0xEF94	0x0001 ...	de3.lunrac.com	207.154.215.197
6986	2019-12-11 08:32:42 UTC	192.168.1.100 ...	52898	192.168.1.1	53	60	20:14:18	0x1455	0x0001 ...	dk7.lunrac.com	185.186.79.250
6991	2019-12-11 08:32:42 UTC	192.168.1.100 ...	64231	192.168.1.1	53	60	03:19:06	0x681D	0x0001 ...	nl68.todn.me	198.16.66.196
7054	2019-12-11 08:32:44 UTC	192.168.1.100 ...	54598	192.168.1.1	53	60	03:19:04	0x2A6D	0x0001 ...	nl68.todn.me	50.7.93.85
7075	2019-12-11 08:32:44 UTC	192.168.1.100 ...	49499	192.168.1.1	53	60	00:00:00	0x1373	0x0000	nl68.todn.me	No error condition (flags 0x818)
7617	2019-12-11 08:32:51 UTC	192.168.1.100 ...	54286	192.168.1.1	53	60	03:18:59	0x3AE3	0x0001 ...	nl58.todn.me	198.16.66.156
7620	2019-12-11 08:32:51 UTC	192.168.1.100 ...	51780	192.168.1.1	53	60	07:10:28	0x17C7	0x0001 ...	nl71.todn.me	198.16.70.29
7622	2019-12-11 08:32:51 UTC	192.168.1.100 ...	63517	192.168.1.1	53	60	20:14:15	0x9668	0x0001 ...	nl86.todn.me	50.7.93.85
7640	2019-12-11 08:32:51 UTC	192.168.1.100 ...	57156	192.168.1.1	53	60	1:00:00:00	0x194D	0x0001 ...	nl72.todn.me	198.16.70.51
7658	2019-12-11 08:32:52 UTC	192.168.1.100 ...	62013	192.168.1.1	53	60	10:14:01	0x15C3	0x0001 ...	nl93.todn.me	198.16.76.27
7668	2019-12-11 08:32:52 UTC	192.168.1.100 ...	52503	192.168.1.1	53	60	20:46:20	0x43AB	0x0001 ...	nl69.todn.me	198.16.70.27
7680	2019-12-11 08:32:52 UTC	192.168.1.100 ...	51665	192.168.1.1	53	60	07:10:28	0x79A4	0x0001 ...	nl77.todn.me	198.16.66.197
7699	2019-12-11 08:32:52 UTC	192.168.1.100 ...	60113	192.168.1.1	53	60	20:14:13	0xC863	0x0001 ...	nl64.todn.me	50.7.93.28
7708	2019-12-11 08:32:52 UTC	192.168.1.100 ...	58529	192.168.1.1	53	60	23:17:31	0xB84A	0x0001 ...	nl63.todn.me	50.7.93.27
7724	2019-12-11 08:32:52 UTC	192.168.1.100 ...	51793	192.168.1.1	53	60	07:59:00	0xA3D0	0x0001 ...	nl59.todn.me	198.16.66.157
7740	2019-12-11 08:32:52 UTC	192.168.1.100 ...	51692	192.168.1.1	53	60	03:19:32	0x5780	0x0001 ...	nl55.todn.me	50.7.142.180
7748	2019-12-11 08:32:53 UTC	192.168.1.100 ...	56925	192.168.1.1	53	60	00:51:41	0x75B3	0x0001 ...	nl88.todn.me	198.16.74.44
7761	2019-12-11 08:32:53 UTC	192.168.1.100 ...	56180	192.168.1.1	53	60	23:37:08	0xD714	0x0001 ...	nl52.todn.me	198.16.74.204
7763	2019-12-11 08:32:53 UTC	192.168.1.100 ...	59141	192.168.1.1	53	60	23:24:32	0x013A	0x0001 ...	nl57.todn.me	198.16.66.155
7779	2019-12-11 08:32:53 UTC	192.168.1.100 ...	52487	192.168.1.1	53	60	07:59:01	0x9503	0x0001 ...	nl85.todn.me	50.7.93.84
7806	2019-12-11 08:32:53 UTC	192.168.1.100 ...	60500	192.168.1.1	53	60	22:11:58	0x6DA4	0x0001 ...	nl90.todn.me	198.16.78.43
7808	2019-12-11 08:32:53 UTC	192.168.1.100 ...	57826	192.168.1.1	53	60	20:14:12	0x4E46	0x0001 ...	nl95.todn.me	198.16.76.29
7836	2019-12-11 08:32:54 UTC	192.168.1.100 ...	64951	192.168.1.1	53	60	07:59:01	0xA8F9	0x0001 ...	nl60.todn.me	198.16.66.123

Figure 4-11 NetworkMiner-Browsec-DNS activity

```

GET /api/test?232111810663 HTTP/1.1
Host: at11.lunrac.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:70.0) Gecko/20100101 Firefox/70.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive

HTTP/1.0 200 Found
Cache-Control: no-cache
Connection: close
Content-Type: application/json

{ "ok" : true }

```

Figure 4-12 Wireshark-Browsec-Service Status Plain

- 5) Once the status of service across the server is received. VPN Client connects to a server based on user configuration. This connection is established on port 443 and follows the similar behavior as of HTTPS based traffic. This behavior can also be seen in Figure 4-13.

Time	Source	Destination	Protocol	Length	Info
7550	55.096464	192.168.1.100	198.16.66.196	TCP	66 1980 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
7568	55.256550	198.16.66.196	192.168.1.100	TCP	68 443 → 1980 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
7571	55.256684	192.168.1.100	198.16.66.196	TCP	54 1980 → 443 [ACK] Seq=1 Ack=1 Win=131328 Len=0
7574	55.258078	192.168.1.100	198.16.66.196	TLSv1.2	597 Client Hello
7581	55.416268	198.16.66.196	192.168.1.100	TLSv1.2	195 Server Hello, Change Cipher Spec, Encrypted Handshake Message
7582	55.417150	192.168.1.100	198.16.66.196	TLSv1.2	105 Change Cipher Spec, Encrypted Handshake Message
7587	55.431149	192.168.1.100	198.16.66.196	TLSv1.2	344 Application Data
7595	55.600425	198.16.66.196	192.168.1.100	TCP	56 443 → 1980 [ACK] Seq=142 Ack=885 Win=29312 Len=0
7781	57.948370	198.16.66.196	192.168.1.100	TLSv1.2	122 Application Data
7783	57.950511	192.168.1.100	198.16.66.196	TLSv1.2	600 Application Data
7804	58.156619	198.16.66.196	192.168.1.100	TCP	56 443 → 1980 [ACK] Seq=210 Ack=1431 Win=29312 Len=0
7818	58.278060	198.16.66.196	192.168.1.100	TCP	1514 443 → 1980 [ACK] Seq=210 Ack=1431 Win=29312 Len=1460 [TCP segment of a reassembled PDU]
7819	58.278063	198.16.66.196	192.168.1.100	TCP	1514 443 → 1980 [ACK] Seq=1670 Ack=1431 Win=29312 Len=1460 [TCP segment of a reassembled PDU]
7820	58.278066	198.16.66.196	192.168.1.100	TLSv1.2	371 Application Data
7822	58.278286	192.168.1.100	198.16.66.196	TCP	54 1980 → 443 [ACK] Seq=1431 Ack=3447 Win=131328 Len=0
7827	58.293024	192.168.1.100	198.16.66.196	TLSv1.2	176 Application Data
7845	58.556944	198.16.66.196	192.168.1.100	TCP	56 443 → 1980 [ACK] Seq=3447 Ack=1553 Win=29312 Len=0

**Figure 4-13 Wireshark-Browsec-Standard port for HTTPS**

- 6) Once the server location is changed the IP of the server connected is also changed and a new connection is established as shown in Figure 4-14.

Time	Source	Destination	Protocol	Length	Info
14722	96.218043	192.168.1.100	46.101.16.229	TCP	66 2195 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
14795	96.412923	46.101.16.229	192.168.1.100	TCP	64 443 → 2195 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 WS=256
14796	96.412997	192.168.1.100	46.101.16.229	TCP	54 2195 → 443 [ACK] Seq=1 Ack=1 Win=131328 Len=0
14801	96.414793	192.168.1.100	46.101.16.229	TLSv1.2	571 Client Hello
14905	96.617309	46.101.16.229	192.168.1.100	TLSv1.2	1514 Server Hello
14906	96.617312	46.101.16.229	192.168.1.100	TLSv1.2	1514 Certificate [TCP segment of a reassembled PDU]
14907	96.617313	46.101.16.229	192.168.1.100	TLSv1.2	321 Server Key Exchange, Server Hello Done
14908	96.617405	192.168.1.100	46.101.16.229	TCP	54 2195 → 443 [ACK] Seq=518 Ack=3188 Win=131328 Len=0
14909	96.619714	192.168.1.100	46.101.16.229	TLSv1.2	180 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
14921	96.645358	192.168.1.100	46.101.16.229	TLSv1.2	324 Application Data
15032	96.809052	46.101.16.229	192.168.1.100	TLSv1.2	105 Change Cipher Spec, Encrypted Handshake Message
15043	96.832907	46.101.16.229	192.168.1.100	TLSv1.2	122 Application Data
15045	96.833023	192.168.1.100	46.101.16.229	TCP	54 2195 → 443 [ACK] Seq=914 Ack=3307 Win=131072 Len=0
15048	96.836661	192.168.1.100	46.101.16.229	TLSv1.2	600 Application Data
15209	97.037353	46.101.16.229	192.168.1.100	TCP	1514 443 → 2195 [ACK] Seq=3307 Ack=1460 Win=17920 Len=1460 [TCP segment of a reassembled PDU]
15210	97.037353	46.101.16.229	192.168.1.100	TLSv1.2	1458 Application Data

**Figure 4-14 Wireshark-Browsec-Standard port for HTTPS Server-2**

- 7) Continuous monitoring of the system will be required to update server name entries with respect to the browser.

## 4.4 uVPN

uVPN is another highly used freely available VPN service. The VPN has several servers spread across 20 countries. The network traffic analysis used for detecting uVPN is discussed below.

- 1) uVPN initially authenticates the user based on the email provided in registration step using server name “api.uvpn.me”.
- 2) Server lists with IP and locations are shared along with the valid server name inside an SSL tunnel as shown in Figure 4-15.



**Figure 4-15 Fiddler-uVPN- Server Name vs IP vs Ports**

3) Once authenticated with the server the client, based on selected location sends a DNS request at \*.uvpn.me. Depending on the region the server name changes. As shown in DNS analysis in Figure 4-16:

6367	2019-12-11...	192.168.1.100...	55006	192.168.1.1	53	60	00:03:11	0x5823	0x0001 ...	ajax.cloudflare.com	104.17.64.4
6835	2019-12-11...	192.168.1.100...	58598	192.168.1.1	53	60	00:03:22	0xDECA	0x0001 ...	www.googletagmanager.com	172.217.19.168
6841	2019-12-11...	192.168.1.100...	63334	192.168.1.1	53	60	00:05:00	0x6A26	0x0001 ...	cdn.paddle.com	104.20.59.238
6843	2019-12-11...	192.168.1.100...	63334	192.168.1.1	53	60	00:05:00	0x6A26	0x0001 ...	cdn.paddle.com	104.20.60.238
7560	2019-12-11...	192.168.1.100...	63684	192.168.1.1	53	60	00:02:48	0x28E7	0x0001 ...	stats.g.doubleclick.net	172.253.120.156
7791	2019-12-11...	192.168.1.100...	55653	192.168.1.1	53	60	00:03:02	0x8832	0x0001 ...	www.google.com	172.217.19.164
7816	2019-12-11...	192.168.1.100...	62396	192.168.1.1	53	60	00:02:31	0xEA7A	0x0001 ...	www.google.com.pk	216.58.208.227
7882	2019-12-11...	192.168.1.100...	61508	192.168.1.1	53	60	00:05:00	0x424D	0x0001 ...	fr102.uvpn.me	51.83.3.220
7886	2019-12-11...	192.168.1.100...	50687	192.168.1.1	53	60	00:04:59	0xCC6D	0x0001 ...	fr102.uvpn.me	51.83.3.220
7892	2019-12-11...	192.168.1.100...	54580	192.168.1.1	53	60	00:00:00	0x4BB0	0x0000	fr102.uvpn.me	No error condition flags 0x8180
8367	2019-12-11...	192.168.1.100...	59855	192.168.1.1	53	60	00:05:00	0xF728	0x0001 ...	fr102.uvpn.me	91.134.2.190
8373	2019-12-11...	192.168.1.100...	49772	192.168.1.1	53	60	00:05:00	0x7F95	0x0001 ...	fr102.uvpn.me	91.134.2.190
8376	2019-12-11...	192.168.1.100...	59998	192.168.1.1	53	60	00:00:00	0xEA01	0x0000	fr102.uvpn.me	No error condition flags 0x8180
9787	2019-12-11...	192.168.1.100...	54140	192.168.1.1	53	60	00:00:00	0xC6A6	0x0000	fr102.uvpn.me	No error condition flags 0x8180

**Figure 4-16 NetworkMiner-uVPN-DNS Activity**

4) After the DNS response is received HTTPS based connection is established with the VPN service. It can be seen the connection is established on port 443 and has standard HTTPS traffic pattern as seen in Figure 4-17.

7946	125.994823	192.168.1.100	51.83.3.220	TCP	66	14541 → 443	[SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
7955	126.173257	51.83.3.220	192.168.1.100	TCP	68	443 → 14541	[SYN, ACK] Seq=0 Ack=1 Win=7300 Len=0 MSS=1460 SACK_PERM=1 WS=1024
7956	126.173440	192.168.1.100	51.83.3.220	TCP	54	14541 → 443	[ACK] Seq=1 Ack=1 Win=131328 Len=0
7957	126.180388	192.168.1.100	51.83.3.220	TLSv1.2	571		Client Hello
7963	126.350127	51.83.3.220	192.168.1.100	TCP	56	443 → 14541	[ACK] Seq=1 Ack=518 Win=9216 Len=0
7964	126.350442	51.83.3.220	192.168.1.100	TLSv1.2	1514		Server Hello
7965	126.350444	51.83.3.220	192.168.1.100	TLSv1.2	1514		Certificate [TCP segment of a reassembled PDU]
7966	126.350615	192.168.1.100	51.83.3.220	TCP	54	14541 → 443	[ACK] Seq=518 Ack=2921 Win=131328 Len=0
7967	126.351549	51.83.3.220	192.168.1.100	TLSv1.2	592		Server Key Exchange, Server Hello Done
7968	126.357803	192.168.1.100	51.83.3.220	TLSv1.2	396		Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
7976	126.518093	51.83.3.220	192.168.1.100	TLSv1.2	320		New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
7977	126.571019	192.168.1.100	51.83.3.220	TCP	54	14541 → 443	[ACK] Seq=860 Ack=3725 Win=130560 Len=0
7984	126.937790	192.168.1.100	51.83.3.220	TLSv1.2	587		Application Data
7990	127.149545	51.83.3.220	192.168.1.100	TCP	56	443 → 14541	[ACK] Seq=3725 Ack=1393 Win=11264 Len=0
8017	128.147163	51.83.3.220	192.168.1.100	TCP	1514	443 → 14541	[ACK] Seq=3725 Ack=1393 Win=11264 Len=1460 [TCP segment of a reassembled PDU]
8018	128.147164	51.83.3.220	192.168.1.100	TCP	1514	443 → 14541	[ACK] Seq=5185 Ack=1393 Win=11264 Len=1460 [TCP segment of a reassembled PDU]
8019	128.147165	51.83.3.220	192.168.1.100	TLSv1.2	1411		Application Data
8020	128.147339	192.168.1.100	51.83.3.220	TCP	54	14541 → 443	[ACK] Seq=1393 Ack=8002 Win=131328 Len=0
.....	.....	.....	.....	.....	.....	.....	.....

**Figure 4-17 Wireshark-uVPN-Standard port for HTTPS Server**

5) On changing the country location from which the VPN server is from, server name and IP also changes as shown in Figure 4-18.

8555	136.933789	192.168.1.100	91.134.2.190	TCP	66 14557 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
8579	137.106194	91.134.2.190	192.168.1.100	TCP	68 443 → 14557 [SYN, ACK] Seq=0 Ack=1 Win=7300 Len=0 MSS=1460 SACK_PERM=1 WS=1024
8580	137.106348	192.168.1.100	91.134.2.190	TCP	54 14557 → 443 [ACK] Seq=1 Ack=1 Win=131328 Len=0
8582	137.112369	192.168.1.100	91.134.2.190	TLsv1.2	571 Client Hello
8597	137.274700	91.134.2.190	192.168.1.100	TCP	56 443 → 14557 [ACK] Seq=1 Ack=518 Win=9216 Len=0
8598	137.275135	91.134.2.190	192.168.1.100	TLsv1.2	1514 Server Hello
8599	137.275137	91.134.2.190	192.168.1.100	TLsv1.2	1514 Certificate [TCP segment of a reassembled PDU]
8600	137.275274	192.168.1.100	91.134.2.190	TCP	54 14557 → 443 [ACK] Seq=518 Ack=2921 Win=131328 Len=0
8601	137.275424	91.134.2.190	192.168.1.100	TLsv1.2	592 Server Key Exchange, Server Hello Done
8603	137.284185	192.168.1.100	91.134.2.190	TLsv1.2	396 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
8611	137.430596	91.134.2.190	192.168.1.100	TLsv1.2	320 New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
8616	137.479319	192.168.1.100	91.134.2.190	TCP	54 14557 → 443 [ACK] Seq=860 Ack=3725 Win=130560 Len=0
8715	139.293181	192.168.1.100	91.134.2.190	TLsv1.2	683 Application Data
8730	139.532054	91.134.2.190	192.168.1.100	TCP	56 443 → 14557 [ACK] Seq=3725 Ack=1489 Win=11264 Len=0
8795	140.475787	91.134.2.190	192.168.1.100	TCP	1514 443 → 14557 [ACK] Seq=3725 Ack=1489 Win=11264 Len=1460 [TCP segment of a reassembled PDU]
8796	140.475789	91.134.2.190	192.168.1.100	TCP	1514 443 → 14557 [ACK] Seq=5185 Ack=1489 Win=11264 Len=1460 [TCP segment of a reassembled PDU]
8798	140.476121	192.168.1.100	91.134.2.190	TCP	54 14557 → 443 [ACK] Seq=1489 Ack=6645 Win=131328 Len=0
8799	140.479051	91.134.2.190	192.168.1.100	TLsv1.2	1427 Application Data

**Figure 4-18 Wireshark-uVPN-Standard port for HTTPS Server-2**

6) IP addresses keep changing and server names also get updated periodically during our analysis on the VPN service.

## 4.5 Summary

In this chapter we have discussed in detail, the forensic analysis carried for each VPN service. Based on their traffic patterns, each service is different from other. Hotspot shield and ZenMate may not use standard ports for communications. The change of geo-location in VPN client is handled differently by each VPN service provider. These differences are because of the fact that no standard for VPN services is present. This means that each VPN service has its own unique signature and traffic pattern. We have tried to analyze the encrypted payload between the client and the server to

understand each service's connection behavior. In next chapter we will discuss the system design of our proposed system.

# **CHAPTER 5**

## **SYSTEM DESIGN AND EVALUATION**

In the previous chapter we discussed the forensics analysis for each VPN service. The analysis showed that there are multiple features that needs to be extracted from the network traffic. In this chapter we will see how these features are extracted and co-related in order to detect and categorize these VPN services. We also established a lab setup to evaluate our system. The setup and the results received are also discussed.

Our proposed system accepts incoming traffic from a pre-defined interface. The incoming traffic is categorized based on traffic features. This categorization is based on the current connection details and also the preceding activity. Designed system is able to co-relate between different OSI layers and multiple connections to be able to

classify the incoming traffic. This traffic may be VPN based traffic or non-VPN based traffic.

The process of detecting and classifying VPN connections can be further divided in two sub processes.

- 1) Traffic information extraction
- 2) Traffic classification

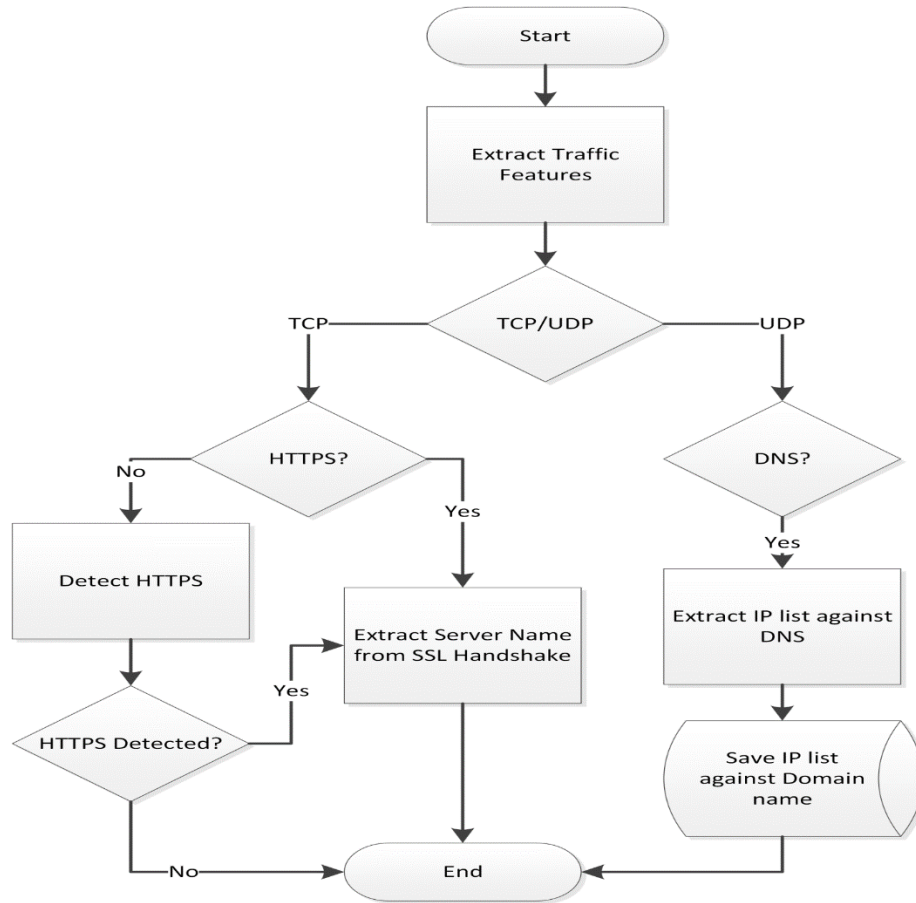
## **5.1 Traffic Information Extraction**

In order to distinguish between VPN and non-VPN based traffic we extracted multiple distinguishing features of the incoming network traffic, mostly the features are collected using the current connection details like:

- 1) Client IP
- 2) Server IP
- 3) Transport Layer Protocol
- 4) Client Port
- 5) Server Port
- 6) Application Protocol (HTTPS/HTTP)
- 7) Server Name (SSL Certificate)

DNS based information for Server IP is extracted before the TCP connection is established from the traffic. General flow of feature extraction is provided in the Figure 5-1. This sub-process is responsible for following characteristics extractions.





**Figure 5-1 Feature Extraction**

## 5.2 5-Tuple Extraction

Initially the system after receiving any new incoming connection extracts the 5-tuple information. These 5 tuples contain IP of Client and Server. Protocol is also extracted from IP header TCP/UDP[58]. Client port and server port of the connection are extracted. Based on these 5 traits, a unique identifier is generated for the new connection. This identifier is used for monitoring any incoming or outgoing traffic for this connection.

### **5.2.1 DNS based Information Extraction**

Before a user could actually connect to a web server, he needs a public IP address of the server. DNS based requests to the DNS server will in response get the server IP against the domain name sent by the user[58]. This activity is also monitored and a list is being maintained and constantly maintained for any addition/updating. This list is responsible to translate IP addresses to server name and vice versa.

### **5.2.2 HTTPS/TLS Protocol Detection**

After the 3-way handshake of a TCP Connection is completed, the traffic for the connection is forwarded to SSL detection Engine. HTTPS uses standard port of 443 over the internet. Some VPN may use different ports to send HTTPS/TLS based traffic over the network to avoid detection by obscuring the network port. All incoming traffic is passed through this module to detect for HTTPS/TLS based protocol in the traffic. This help identifying encrypted payload from normal traffic so that the encrypted traffic may be labeled as VPN or normal.

### **5.2.3 SSL Based Feature Extraction**

Once a connection is tagged as encrypted, the SSL certificate from client and server is decoded[59]. There are 4 types of message in SSL protocol

- i) Handshake Message
- ii) Alert Message
- iii) Change Cipher Spec Message
- iv) Application Data

Each type of message is used to transfer different information between server and client. Handshake messages are initially transferred to negotiate and establish the connection between the user and server. From these messages we are able to extract the server information like server name. This is used to verify or relate to the DNS activity present for that IP address.

Once these features are extracted from the connection and all the information required for the labeling of traffic as VPN or non-VPN is completed. The connection information is passed to traffic classification Module which classifies the connections as VPN or non-VPN.

## **5.3 Traffic Classification**

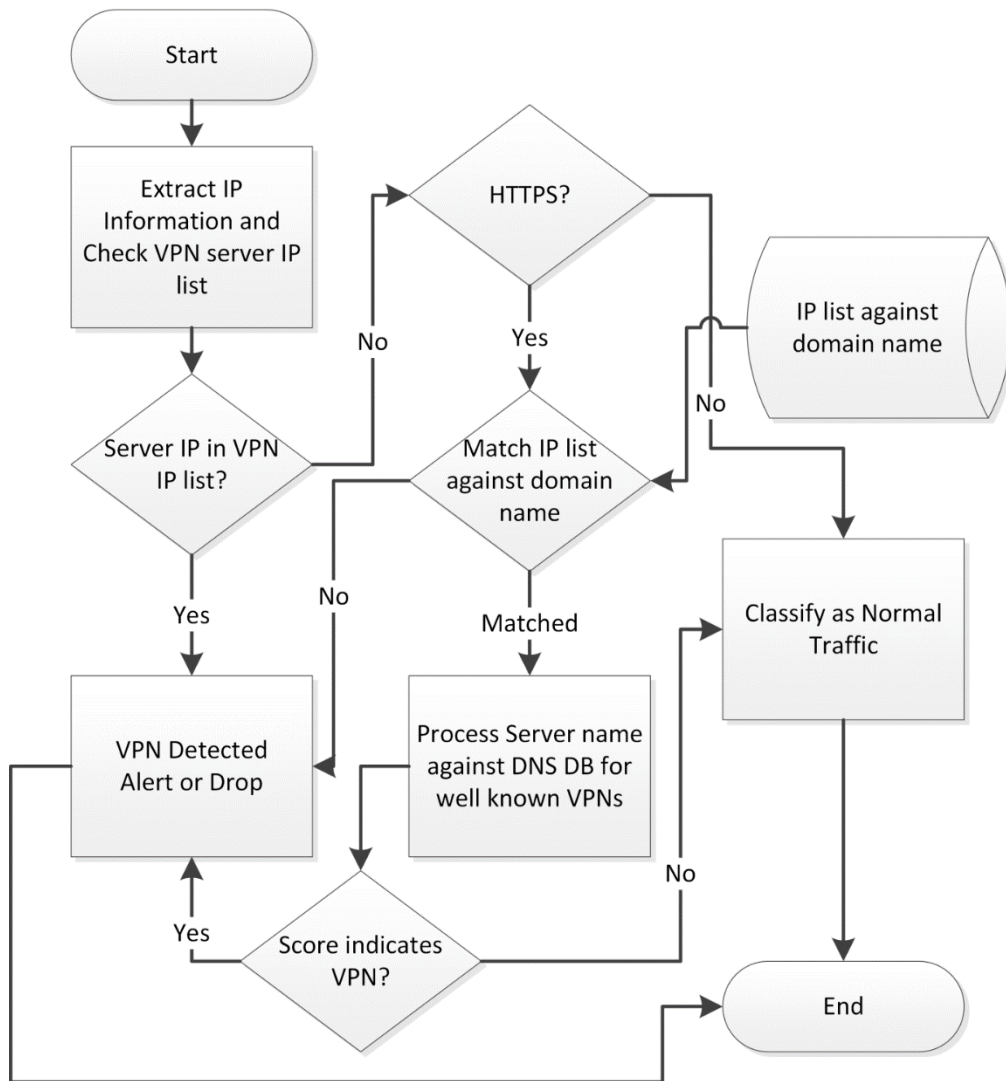
After feature extraction the TCP based connections are forwarded to classifier. The classifier is charged of marking the connection, Server IP as VPN or non-VPN. Classifier is further divided into 3 layers based on the level of analysis required by the system to label the incoming traffic. General flow for traffic classification is shown in Figure 5-2.

These layers help classify and separate normal traffic from VPN based traffic. Each layer is discussed below:

### **5.3.1 IP Based Classification**

This classifier tries to classify any incoming connection based on its server's IP. This is done at the very start of the connection as the server IP is known from the start of the connection and is present in 5Tuple. A list of IPs of already detected VPN server

is maintained by the system as the system identifies new VPN Servers. This is done to reduce processing and time overhead from re-identifying the VPN Server. After tagging a connection as VPN its server IP, if not already present in the list of VPN server IP, is added to the VPN server IP list maintained by the system.



**Figure 5-2 Traffic Classification**

### **5.3.2 Name Server Classification**

When the Server IP is not found in the VPN Server IP list, after SSL/HTTPS detection, previously described section, the domain name presented in SSL server certificate is verified against server IP vs server name list populated using DNS records. The Server IP being used by the client must be shared through standard domain name resolution. If no DNS record is present for the server IP, it is possible that this IP was communicated during initialization step as discussed and shown in forensic analysis of VPN services. Such behavior is not standard activity and is used to obscure the real traffic. Such behavior shows that the connection is VPN based. Once this connection is tagged as VPN, the server IP is added to VPN server IP list.

### **5.3.3 DNS based Classification**

Domain Name information related to each VPN was collected during the forensic analysis of these VPN services. Against these domain names a list of classifier is generated. These classifiers help in classifying incoming server name DNS request as VPN or non-VPN. Once a new connection is established and server name inside the SSL handshake is also identified as the same VPN server name, the connection is flagged as VPN connection. This step is the last step in classification chain. A connection classified here as VPN means that VPN is using a standard HTTPS protocol over port 443 and traffic characteristics follows the standard behavior of web traffic. Once the connection is flagged, its server IP is added in VPN server IP list. This is done to improve the time required to detect next connection from the same as IP as VPN.

## 5.4 System Evaluation

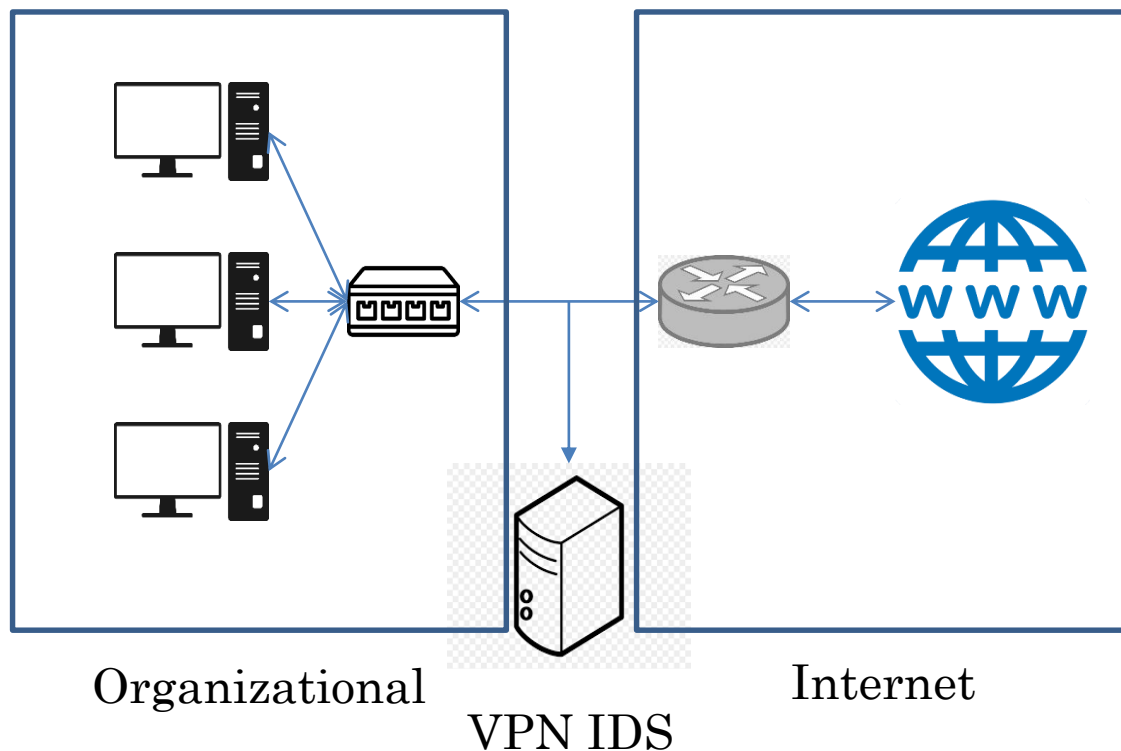
The system is designed as to be least intrusive in nature. This means that minimum level of network changes and system configuration be needed in order to deploy and use the system. Our system does not add any network traffic within the network as compared to some early work[20], [28]. Our system does not add many nodes to collect the data. A single node is configured to receive the incoming traffic and classify the traffic inside the network.

Generally there are 2 methods for network deployment active and passive. Our solution can also be used only for detection purposes. This means that the deployment can be passive as well. This means that no traffic latency be observed as the traffic is being mirrored by the switch or gateway itself. For such deployment DNS traffic and internet traffic should be made available on the same link being mirrored as shown in Figure 5-3. We analyzed the traffic pattern of well-known available VPN services which use HTTPS protocol for communication. These services are listed in Table 5-1:

S. No.	VPN Names
1	ZenMate VPN
2	Hotspot Shield
3	Browsesec VPN
4	uVPN
5	Hoxx VPN

**Table 5-1 VPN Services Analyzed**

These VPN were installed in multiple user accounts and the traffic was analyzed. A selection criterion was build based on the pattern emerging from the analysis. The key features for each VPN service are shown in Table 5-2 below.



**Figure 5-3 Deployment Model**

When analyzing hotspot shield, we tested both type of its client. One is the desktop application version and other browser add-on. Using web browser extension, VPN uses special domain names which were used to uniquely classify the VPN. In case of desktop version, the use of nonstandard port for HTTPS and no DNS requests was observed. When working with browsec, uVPN and hoxx VPNs the browser add-on were analyzed and we were able to classify them based on their server names.

The VPN discussed above use the same type of server names irrespective of geo-location selected. This means any traffic generated due to uVPN may be classified if its DNS request before the connection contains “\*. uvpn.me”. When discussing zenmate VPN this is not true. Zenmate changes server names as user choses geolocations. This list of server names and location is shared at start of the secure communication and may be updated when required. This type of behavior allows VPN services to work inside a network undetected which uses only DNS-based filters.

VPN Clients	Classifier for Forensic Analysis		
	Domain Names	Non-Standard HTTPS	No DNS Activity
Hotspot Shield	✓	✓	✓
Zen Mate	✓	✓	x
Browsec	✓	x	x
uVPN	✓	x	x
Hoxx	✓	x	x

**Table 5-2 Bird Eye view of VPN Services Analysis**

### 5.4.1 Traffic Generation

Multiple clients of these VPN services were used across multiple systems inside the network. Standard web activity was generated using these clients and saved in files. These files were passed on the system to be monitored. Alerts were generated once the said VPN activity was detected.



### 5.4.2 Traffic Classification Alert

Multiple types of alerts were generated for multiple type of activity detected by the system. Depending upon the activities performed by the users the alerts vary. Users had option to pick and choose between any VPN clients during the activity. The generated alerts by the system against each user are shown in Table 5-3.

User Details	Classification of Generated Alerts				
	Total Connections	Legitimate Activity	DNS Based VPN	NO DNS Activity	Non-Standard HTTPS
User 1	245	124	121	0	0
User 2	103	63	0	40	0
User 3	400	176	136	0	88
User 4	142	37	69	13	23
User 5	247	180	63	0	4

**Table 5-3 Summary of User activity**

User activity shown in Table 5-3 shows the traffic characteristics of each user. The alerts generated against illegitimate connections are based on the unique characteristics as discussed in the Table 5-2. We see that mostly VPN may be classified with the help of their domain names but we see other behavior too in user activity that is encouraging.

## **5.5 Summary**

The results discussed in Table 5-3 says that the system was able to distinguish 53% of 1035 user connections as VPN. After system deployment any incoming connection request is monitored by the system and is classified based on different criteria discussed above. Once the decision is established, the server IP is added to respected VPN or valid Server IP list. This is done to achieve low latency and minimize the instruction cycles to classify new incoming connection from the same IP address.

The proposed system is based on DNS and SSL analysis of the VPN services. The system is as effective as the domain name data incorporated in it. This means that the system must be managed and periodically updated for new DNS and traffic pattern to be effective in longer run.

# CHAPTER 6

## CONCLUSION

In recent years we have seen increase in encrypted traffic over the internet to achieve confidentiality and integrity of the information being transferred. These services have evolved over the years to be effective against eavesdropping and data modification attacks. This increased encrypted traffic inside the network has caused a lot of traffic to go uninspected. Use of any type of VPN service which violates the policy of the organization may be used by any individual to evade such policies.

This misuse of organizational resources may result in data leakage and espionage. An organization may not be able to afford network monitoring tools to be able to stop such insider attacks. This work proposes a novel and effective approach to be able to detect such VPN services being used inside the network. This chapter presents a

summary of the system features and also presents a discussion on directions for future research.

## **6.1 System Features**

- a) The proposed solution works with unencrypted part of information inside the network communication.
- b) The analysis on plain traffic is used to minimize resource utilization thereby producing an optimized solution.
- c) The proposed solution differs from its predecessors in a way that it not only focuses on the current connection but also is aware of historical network traffic to be able to better understand the encrypted traffic.
- d) An anomaly-based IDS has also been designed that detects HTTPS traffic inside the network other than its standard port i.e. 443.

The results discussed above shows that our system is able to understand the network traffic of users and generate alerts. The overall analysis based on different VPN services is discussed in Table 5-2 which helps us understand the behavior of multiple VPN services. In order for the system to be effective the classifier needs to be updated with latest traffic trends of the network.

## **6.2 Future Work**

- a) In future more VPN services may be added using the same analysis techniques discussed above.
- b) This solution is also capable of detecting command and control (C&C) for malwares and APTs. Such, domain may also be added to improve its coverages.

- c) The proposed solution is a generic framework, it currently focuses only on VPN services as a use case. In future any type of service may be used to increase the breadth of intrusion detection.

# BIBLIOGRAPHY

- [1] M. Zain Ul Abideen, S. Saleem, and M. Ejaz, “VPN Traffic Detection in SSL-Protected Channel,” *Secur. Commun. Networks*, vol. 2019, 2019, doi: 10.1155/2019/7924690.
- [2] A. Sunyaev, *Internet Computing*. Springer International Publishing, 2020.
- [3] B. Harris, R. H.-C. communications, and undefined 1999, “TCP/IP security threats and attack methods,” *Elsevier*, Accessed: Feb. 13, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S014036649900064X>.
- [4] X. Li, M. Wang, H. Wang, Y. Yu, and C. Qian, “Toward Secure and Efficient Communication for the Internet of Things,” in *IEEE/ACM Transactions on Networking*, Apr. 2019, vol. 27, no. 2, pp. 621–634, doi: 10.1109/TNET.2019.2893249.
- [5] S. Chen, S. Jero, M. Jagielski, A. Boldyreva, and C. Nita-Rotaru, “Secure Communication Channel Establishment: TLS 1.3 (over TCP Fast Open) vs. QUIC,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Sep. 2019, vol. 11735 LNCS, pp. 404–426, doi: 10.1007/978-3-030-29959-0\_20.
- [6] A. Porter Felt, R. Barnes, A. King, C. Palmer, C. Bentzel, and P. Tabriz,

*Measuring HTTPS Adoption on the Web.* 2017.

- [7] J. Clark and P. C. Van Oorschot, “SoK: SSL and HTTPS: Revisiting past challenges and evaluating certificate trust model enhancements,” in *Proceedings - IEEE Symposium on Security and Privacy*, 2013, pp. 511–525, doi: 10.1109/SP.2013.41.
- [8] C Paya and O Dubrovsky, “Inspecting encrypted communications with end-to-end integrity,” 2009.
- [9] A. A. Abimbola, J. M. Munoz, and W. J. Buchanan, “NetHost-Sensor: Investigating the capture of end-to-end encrypted intrusive data,” *Comput. Secur.*, vol. 25, no. 6, pp. 445–451, Sep. 2006, doi: 10.1016/j.cose.2006.04.001.
- [10] Vladimir Lifliand, Avraham Michael Ben-Menahem, “Encrypted network traffic interception and inspection,” 2011.
- [11] N. Leavitt, “Anonymization Technology Takes a High Profile,” *Computer (Long. Beach. Calif.)*, vol. 42, no. 11, pp. 15–18, Nov. 2009, doi: 10.1109/mc.2009.340.
- [12] Z. Zhipeng, S. Chandel, S. Jingyao, Y. Shilin, Y. Yunnan, and Z. Jingji, “VPN: a Boon or Trap? : A Comparative Study of MPLS, IPSec, and SSL Virtual Private Networks,” Nov. 2018, pp. 510–515, doi: 10.1109/iccmc.2018.8487653.
- [13] A. Mileva and B. Panajotov, “Covert channels in TCP/IP protocol stack - Extended version-,” *Open Computer Science*, vol. 4, no. 2. Walter de Gruyter

- GmbH, pp. 45–66, 2014, doi: 10.2478/s13537-014-0205-6.
- [14] T. AbuHmed, A. Mohaisen, and D. Nyang, “A Survey on Deep Packet Inspection for Intrusion Detection Systems,” 2008, [Online]. Available: <http://arxiv.org/abs/0803.0037>.
- [15] P. Cao and S. Wu, “Parallel research on KMP algorithm,” in *2011 International Conference on Consumer Electronics, Communications and Networks, CECNet 2011 - Proceedings*, 2011, pp. 4252–4255, doi: 10.1109/CECNET.2011.5768201.
- [16] B. Debnath, S. Sengupta, J. Li, D. J. Lilja, and D. H. C. Du, “BloomFlash: Bloom filter on flash-based storage,” in *Proceedings - International Conference on Distributed Computing Systems*, 2011, pp. 635–644, doi: 10.1109/ICDCS.2011.44.
- [17] S. Baeg, “Low-power ternary content-addressable memory design using a segmented match line,” *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 55, no. 6, pp. 1485–1494, Jul. 2008, doi: 10.1109/TCSI.2008.916624.
- [18] M. Alicherry, M. Muthuprasanna, and V. Kumar, “High speed pattern matching for network IDS/IPS,” in *Proceedings - International Conference on Network Protocols, ICNP*, 2006, pp. 187–196, doi: 10.1109/ICNP.2006.320212.
- [19] A. Yamada, Y. Miyake, K. Takemori, A. Studer, and A. Perrig, “Intrusion detection for encrypted web accesses,” in *Proceedings - 21st International*



*Conference on Advanced Information Networking and Applications  
Workshops/Symposia, AINAW'07, 2007, vol. 2, pp. 569–576, doi:  
10.1109/AINAW.2007.212.*

- [20] M. A. Aydin, A. H. Zaim, and K. G. Ceylan, “A hybrid intrusion detection system design for computer network security,” *Comput. Electr. Eng.*, vol. 35, no. 3, pp. 517–526, May 2009, doi: 10.1016/j.compeleceng.2008.12.005.
- [21] D. Papamartzivanos, F. Gomez Marmol, and G. Kambourakis, “Introducing Deep Learning Self-Adaptive Misuse Network Intrusion Detection Systems,” *IEEE Access*, vol. 7, pp. 13546–13560, 2019, doi: 10.1109/ACCESS.2019.2893871.
- [22] D. Damopoulos, S. A. Menesidou, G. Kambourakis, M. Papadaki, N. Clarke, and S. Gritzalis, “Evaluation of anomaly-based IDS for mobile devices using machine learning classifiers,” *Secur. Commun. Networks*, vol. 5, no. 1, pp. 3–14, Jan. 2012, doi: 10.1002/sec.341.
- [23] K. Salah and A. Kahtani, “Performance evaluation comparison of Snort NIDS under Linux and Windows Server,” *J. Netw. Comput. Appl.*, vol. 33, no. 1, pp. 6–15, Jan. 2010, doi: 10.1016/j.jnca.2009.07.005.
- [24] R. Gassais, N. Ezzati-Jivan, J. M. Fernandez, D. Aloise, and M. R. Dagenais, “Multi-level host-based intrusion detection system for Internet of things,” *J. Cloud Comput.*, vol. 9, no. 1, pp. 1–16, Dec. 2020, doi: 10.1186/s13677-020-00206-6.

- [25] R. Chakravorty and J. Prakash, "A Review on Prevention and Detection Schemes for Black Hole Attacks in MANET," in *ICRITO 2020 - IEEE 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)*, Jun. 2020, pp. 801–806, doi: 10.1109/ICRITO48877.2020.9197810.
- [26] H. Tabrizchi and M. Kuchaki Rafsanjani, "A survey on security challenges in cloud computing: issues, threats, and solutions," *J. Supercomput.*, vol. 76, no. 12, pp. 9493–9532, Dec. 2020, doi: 10.1007/s11227-020-03213-1.
- [27] A. Sharma, G. Singh, and S. Rehman, "A review of big data challenges and preserving privacy in big data," in *Lecture Notes in Networks and Systems*, vol. 94, Springer, 2020, pp. 57–65.
- [28] V. T. Goh, J. Zimmermann, and M. Looi, "Towards intrusion detection for encrypted networks," in *Proceedings - International Conference on Availability, Reliability and Security, ARES 2009*, 2009, pp. 540–545, doi: 10.1109/ARES.2009.76.
- [29] Y. Zhauniarovich, I. Khalil, T. Yu, and M. Dacier, "A survey on malicious domains detection through DNS data analysis," *ACM Comput. Surv.*, vol. 51, no. 4, pp. 1–36, Jul. 2018, doi: 10.1145/3191329.
- [30] N. Wright, "DNS in Computer Forensics," *J. Digit. Forensics, Secur. Law*, vol. 7, no. 2, p. 2, Jan. 2012, doi: 10.15394/jdfsl.2012.1117.

- [31] M. Husák, M. Žádník, V. Bartoš, and P. Sokol, “Dataset of intrusion detection alerts from a sharing platform,” *Data Br.*, vol. 33, p. 106530, Dec. 2020, doi: 10.1016/j.dib.2020.106530.
- [32] J. L. Lewis, G. F. Tambaliuc, H. S. Narman, and W. S. Yoo, “IP Reputation Analysis of Public Databases and Machine Learning Techniques,” in *2020 International Conference on Computing, Networking and Communications, ICNC 2020*, Feb. 2020, pp. 181–186, doi: 10.1109/ICNC47757.2020.9049760.
- [33] H.-M. An, S.-K. Lee, J.-H. Ham, and M.-S. Kim, “Traffic Identification Based on Applications using Statistical Signature Free from Abnormal TCP Behavior \*,” 2015. Accessed: Feb. 13, 2021. [Online]. Available: [https://nmlab.korea.ac.kr/publication/published.papers/2015/2015.09\\_Abnormal\\_TCP\\_Behavior-JISE.Journal.pdf](https://nmlab.korea.ac.kr/publication/published.papers/2015/2015.09_Abnormal_TCP_Behavior-JISE.Journal.pdf).
- [34] G. Zhao, K. Xu, L. Xu, and B. Wu, “Detecting APT malware infections based on malicious DNS and traffic analysis,” *IEEE Access*, vol. 3, pp. 1132–1142, 2015, doi: 10.1109/ACCESS.2015.2458581.
- [35] M. Ikram, N. Vallina-Rodriguez, S. Seneviratne, M. Ali Kaafar, and V. Paxson, “An Analysis of the Privacy and Security Risks of Android VPN Permission-enabled Apps,” *dl.acm.org*, vol. 14-16-November-2016, pp. 349–364, Nov. 2016, doi: 10.1145/2987443.2987471.
- [36] S. Sudin, R. B. Ahmad, S. Zulkarnain, and S. Idrus, “A Model of Virus Infection Dynamics in Mobile Personal Area Network,” *journal.utem.edu.my*,

Accessed: Feb. 13, 2021. [Online]. Available:

<https://journal.utem.edu.my/index.php/jtec/article/view/4406>.

- [37] N. Weaver, C. Kreibich, M. Dam, and V. Paxson, “Here be web proxies,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014, vol. 8362 LNCS, pp. 183–192, doi: 10.1007/978-3-319-04918-2\_18.
- [38] C. Reis, S. D. Gribble, T. Kohno, and N. C. Weaver, “Detecting In-Flight Page Changes with Web Tripwires.,” in *NSDI*, 2008, vol. 8, pp. 31–44.
- [39] 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*, 2015, pp. 25–30.
- [40] N. Weaver, C. Kreibich, and V. Paxson, “Redirecting DNS for Ads and Profit.,” *FOCI*, vol. 2, pp. 2–3, 2011.
- [41] 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*, 2014, pp. 141–148.
- [42] 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*, 2015, pp. 15–26.
- [43] 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*, 2012, pp. 50–61.
- [44] M. Ciampa, *CompTIA security+ guide to network security fundamentals*. Cengage Learning, 2021.
- [45] S. L. Garfinkel, “Digital forensics research: The next 10 years,” *Digit. Investig.*, vol. 7, pp. S64–S73, 2010.
- [46] W. V. Maconachy, C. D. Schou, D. Ragsdale, and D. Welch, “A model for information assurance: An integrated approach,” in *Proceedings of the 2001 IEEE workshop on information assurance and security*, 2001, vol. 310, pp. 5–6.
- [47] A. R. Hevner, S. T. March, J. Park, and S. Ram, “Design science in information systems research,” *MIS Q.*, pp. 75–105, 2004.
- [48] J. F. Nunamaker Jr, A. R. Dennis, J. S. Valacich, and D. R. Vogel, “Information technology for negotiating groups: Generating options for mutual gain,” *Manage. Sci.*, vol. 37, no. 10, pp. 1325–1346, 1991.
- [49] “Fiddler - Web Debugging Proxy - Telerik.” <https://www.telerik.com/fiddler>

(accessed Feb. 13, 2021).

- [50] G. Sasi, P. Thanapal, V. S. Balaji, G. V. Babu, and V. Elamaran, “A Handy Approach for Teaching and Learning Computer Networks using Wireshark,” in *2020 Fourth International Conference on Inventive Systems and Control (ICISC)*, 2020, pp. 456–461.
- [51] M. A. Doshi and P. Sharma, “Digital Forensics Analysis for Network Related Data,” 2020.
- [52] “The Fastest Most Secure VPN Service | Hotspot Shield.”  
<https://www.hotspotshield.com/> (accessed Dec. 10, 2019).
- [53] “ZenMate VPN - Internet Security and Privacy VPN Service.”  
<https://zenmate.com/> (accessed Dec. 10, 2019).
- [54] “Browsec VPN your Personal Privacy and Security Online.”  
<https://browsec.com/en/> (accessed Dec. 11, 2019).
- [55] “Hoxx VPN Proxy - Free VPN Service.” <https://hoxx.com/> (accessed Feb. 13, 2021).
- [56] “uVPN - Unlimited encrypted VPN to Secure and Unblock content.”  
<https://uvpn.me/> (accessed Dec. 11, 2019).
- [57] “Speedtest by Ookla - The Global Broadband Speed Test.”  
<https://www.speedtest.net/> (accessed Feb. 13, 2021).

- [58] B. A. Forouzan and S. C. Fegan, *TCP/IP protocol suite*. McGraw-Hill Higher Education, 2010.
- [59] T. Dierks and E. Rescorla, “The transport layer security (TLS) protocol version 1.2,” 2008.