

# Argos VPN



By

**Muhammad Faheem, Hamza Shaukat, Syed Muhammad Ahmad Ali Gilani, Iqra  
Bibi and Zaryab Sohail**

Submitted to the Faculty of Department of Electrical Engineering,  
Military College of Signals, National University of Science and Technology, Islamabad

in partial fulfillment for the requirement of a B.E Degree in

Telecom Engineering

JULY 2018

## CERTIFICATE

This is to certify that this Thesis Report entitled “**Argos VPN**” by **Muhammad Faheem, Hamza Shaukat, Syed Muhammad Ahmad Ali Gilani, Iqra Bibi** and **Zaryab Sohail** is submitted in partial fulfillment of the requirement for the degree of BETE in Military College of Signals (NUST) during the academic year 2017-2018, is a bona fide record work carried out under my guidance and supervision.

**Name (Supervisor):**

**Asst. Prof. Mian Muhammad Waseem Iqbal**

**Signature:**

\_\_\_\_\_

**Date:**

\_\_\_\_\_

## ABSTRACT

### **Project Name:**

Argos VPN

### **Skills Required:**

- Network Security
- Penetration testing
- Offline Username password cracking
- Traffic Monitoring

### **Description:**

The project will first implement different methods to sniff client data from different VPN servers to check vulnerabilities like anonymity, IP leakages and DNS hijacking etc. Using the data acquired from the results, we will analyze the security lapses and vulnerabilities in any particular VPNs, and then this project will further lead to a design of a customized VPN, secure in all parameters.

### **Project Deliverables:**

1. Design of the VPN Analyzer.
2. Hardware implementation.
3. Source code.
4. Project report.

We dedicate this to our family, friends and above all our amazing teachers,  
without their prayers, support, encouragement, guidance  
and appreciation we couldn't have achieved  
such a milestone.

## **DECLARATION**

We declare, No portion of this work presented in this thesis has been submitted in support of another award or qualification either at this institution or elsewhere.

## **ACKNOWLEDGEMENTS**

Due extension of our humble gratitude to the most magnificent Allah Almighty, without His will it would never have been possible to have this attainment. We thank Him for providing us with all the knowledge, intelligence, sagacity and understanding which was needed for the successful completion of this thesis research work.

We thank our project supervisor Asst. Prof. Mian Muhammad Waseem Iqbal, who supported us whole heartedly and stimulated our intellect during our work. Without his interest, involvement and assistance it wouldn't have been possible to carry out the research and complete the project work.

We are also thankful to our mentors and colleagues for helping us in the development of this project and rendering us their support whenever it was needed.

Last but not the least, we are more than thankful to our parents for all the prayers, understanding and massive support. Their encouragement helped us a lot in achieving all our project tasks and their words kept us motivated and dedicated throughout our project work.

## Table of Contents

<b>ABSTRACT</b> .....	ii
<b>DEDICATION</b> .....	iii
<b>DECLARATION</b> .....	iv
<b>ACKNOWLEDGMENTS</b> .....	v
<b>LIST OF FIGURES</b> .....	viii
<b>LIST OF TABLES</b> .....	viii
<b>CHAPTER 1 INTRODUCTION</b> .....	1
1.1 Overview: .....	1
1.2 Problem Statement: .....	1
1.3 Approach: .....	2
1.4 Objective: .....	2
<b>CHAPTER 2 BACKGROUND STUDY</b> .....	3
2.1 Existing Literature: .....	3
2.1.1 VPN Tunneling: .....	3
2.1.2 User Anonymity: .....	4
2.1.3 IP leakages: .....	4
2.1.4 Offline Password Cracking: .....	5
2.1.5 DNS Hijacking: .....	5
2.1.6 Man-in-the-Middle Attacks: .....	5
2.1.7 VPN Encryption: .....	6
2.1.8 IPSec Based VPNs: .....	7
2.2 Problem Formulation: .....	8
<b>CHAPTER 3 DESIGN SPECIFICATIONS</b> .....	8
3.1 Technical specifications .....	8
3.1.1 Raspberry pi 3b .....	9
3.1.2 DumpIt.....	9
3.1.3 Volatility.....	9
3.1.4 Nmap 7.60. ....	10
3.1.5 IKE-scan 1.9. ....	10
3.2 Design Requirements.....	10
3.3 Detailed design with justification.....	11

<b>CHAPTER 4 VPN Testing</b> .....	11
4.1 Information Gathering: .....	11
4.2 Prerequisite of the Zero knowledge test .....	15
4.2.1 Memory Forensics.....	15
4.3 Zero Knowledge Test.....	18
4.3.1 Port Scanning.....	18
4.3.2 Packet Tracing.....	19
4.3.3 Finger Printing.....	19
4.3.4 Back off Strategy.....	20
4.4 Username Enumeration Vulnerabilities.....	21
4.4.1 IKE Aggressive mode.....	21
4.5 Countermeasures: .....	22
4.5.1 Defense against IPv6 Leakage: .....	22
4.5.2 Authentication Vulnerabilities: .....	23
4.5.3 Configuration Issues Management: .....	23
4.6 Top 10 Free VPNs Comparison.....	24
 <b>CHAPTER 5 CUSTOMIZED OPEN VPN</b> .....	 24
5.1 Network Configuration.....	24
5.2 OpenVPN Configuration Files .....	28
 <b>CHAPTER 6 IMPLEMENTATION</b> .....	 30
6.1 VPN GUI (Graphical User Interface) .....	30
 <b>CHAPTER 7 RESULTS</b> .....	 31
7.1 Comparison b/w Top 10 Free VPNs with ArgosVPN: .....	32
7.2 IP Leak Test: .....	33
7.3 WebRTC leak Test: .....	33
7.4 DNS leak Test: .....	34
7.5 Internet Speed Test: .....	34
 <b>FUTURE WORK</b> .....	 35
<b>CONCLUSION</b> .....	36
<b>GLOSSARY</b> .....	37
<b>APPENDIX A: GUI Code</b> .....	39
<b>APPENDIX B: MEMORY FORENSICS</b> .....	41
<b>BIBLIOGRAPHY</b> .....	44



## LIST OF FIGURES

Figure 1: Project Approach. ....	2
Figure 2: VPN Tunneling .....	3
Figure 3: User Anonymity .....	4
Figure 4: DNS hijacking/leaking through DNS Server .....	5
Figure 5: MITM Man-in-The-Middle Attack .....	6
Figure 6: VPN Encryption over the Internet .....	6
Figure 7: IPSec Composition and Architecture .....	7
Figure 8: Raspberry Pi 3b .....	9
Figure 9: Design Requirements .....	10
Figure 10: Project Design .....	11
Figure 11: ArgosVPN GUI .....	31

## LIST OF TABLES

Table 4.3.1 Mapping of Open ports to VPN type.....	17
--	----

# CHAPTER 1

## INTRODUCTION

### 1.1 Overview:

Virtual Private Network (VPN) is a common and an effective means to communicate securely over an insecure public network. VPN applications secure clients by encrypting the traffic of an online user and then routing that traffic through remote servers in different countries, which makes it difficult for hackers trying intercept your internet traffic. To establish a secure tunneling connection, the user establishes a connection to the public internet via their ISP and then starts a VPN connection with a remote VPN server through client software provided by the company. With numerous VPNs claiming online privacy and anonymity, they are still open to attacks like Offline Password Cracking, IP leakages, DNS hijacking, Man-in-the-Middle Attacks and Malware Infections etc. By VPN hijacking, the attacker cannot only access the sensitive data being transmitted but can also get an access to the internal network resources.

### 1.2 Problem Statement:

VPNs are useful for encrypting your web traffic or getting around regional restrictions. A study by researchers from Data61/CSIRO, UC Berkeley, UNSW Sydney, and UCSI found that among 283 different apps many of them inject Adware, Trojans, Malware or Spyware. What they found was not great.

Following are the findings:

- 18% do not encrypt traffic
- 84% leak user data
- 38% reveal malware
- 80% request access to sensitive data like user accounts or text messages

For different consumers and organizations who want their internet traffic to be safe from online snooping, be encrypted and not vulnerable to be exploited by harmful attacks. It is hard for them to choose any VPN that would provide total encryption, anonymity and

online security, as they probably do not have enough technical knowledge to analyze different parameters of a VPN they want to choose.

### 1.3 Approach:

Initially, this project will fetch VPN's server IP and then will apply Zero Knowledge Test. Here it will first determine open ports, fingerprint the VPN to check its configuration, vendor and version number by using Nmap or Ike-scan.

Then this project will exploit inherent vulnerabilities in the process of establishing a VPN connection by forcing server to use aggressive mode authentication and sniff PSK using PSK-crack/TCP-dump or Cain and Abel. After cracking, use PGP-Net to connect with vulnerable VPN server.



Figure 1: Project Approach

### 1.4 Objective:

The basic aims behind this project are:

- To find out different VPN configurations.
- To discover and fingerprint the VPN servers.
- To identify the potential security risks involved.
- To analyze IP leakages.
- To check DNS hijacking.
- To identify countermeasures and evaluate the performance and security impacts of counter measures
- To make an open source VPN providing more security and more privacy.

## CHAPTER 2

### BACKGROUND STUDY

#### 2.1 Existing Literature:

The increasing use of the VPN services by students, businesses and organizations is the main motivation of our project. The problem is where people use VPN services for anonymity and security from government monitoring, Only a few know that they are exposed to the ISP. Following security parameters, according to the literature review, are potentially vulnerable to the attackers while hijacking the VPN server.

##### 2.1.1 VPN Tunneling:

Tunneling, a networking means that allows the encapsulation of a protocol packet inside the datagram of an entirely different protocol. Point-to-Point Tunneling Protocol (PPTP) can be used by windows VPN connections for encapsulation and sending private network information, like TCP/IP traffic in a public network.

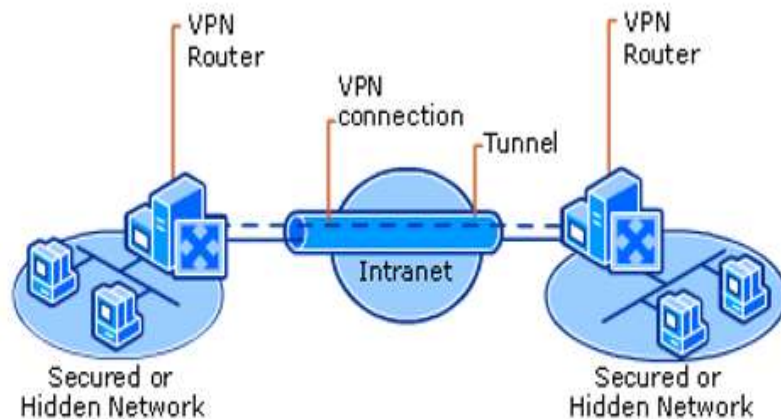


Figure 2: VPN tunneling

For PPTP and Layer-2 Tunneling Protocol (L2TP) is

like a session, both of the tunnel end devices require an acknowledgement to establish a tunnel and require to exchange configuration variables, for instance, assignment of address, compression or encryption parameters. Most of the cases, the information sent

through the tunnel is directed using any datagram-based protocol, using a tunnel management protocol as a tool to establish, uphold, and terminate the tunnel.

### 2.1.2 User Anonymity:

User information not hidden from their VPN service provider who may also retain this information. Many VPN services prompt user to enter private data, credentials or a valid phone number at registration time.

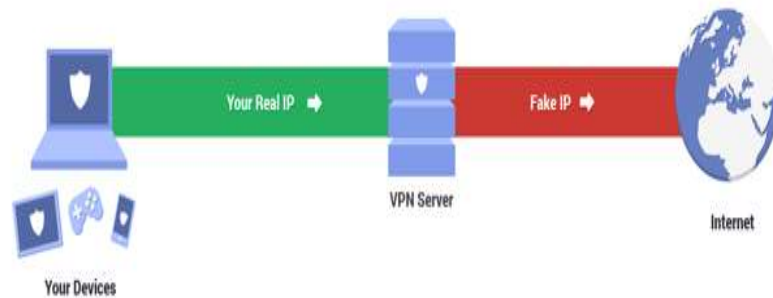


Figure 3: User Anonymity

VPN service providers have to be trusted blindly not to be malicious and keep your user traffic unexposed to the

third parties. A number of cases have occurred where the VPN service providers have disclosed the user traffic.

### 2.1.3 IP leakages:

The VPN client programs only exploit the IPv4 routing table and not the IPv6 routing table. In the result of this all IPv6 traffic bypassing the VPN's virtual interface.

The study revealed that a small leakage of IPv6 traffic can depict the complete browsing history of the user even on IPv4 only. [16]

#### 2.1.4 Offline Password Cracking:

A valid username can be obtained and an attacker can crack the associated password by using the hash from the VPN Server. The offline cracking use PSK-crack that supports the dictionary cracking mode, brute-force cracking mode and hybrid cracking mode. [18]

#### 2.1.5 DNS Hijacking:

DNS Hijacking or DNS redirection is a more concentrated to attack transparently capture all DNS queries, both IPv4 and IPv6, from the VPN Client machine. [7]

There are two types of DNS configurations for VPN Clients

Namely Default Configuration where the VPN Client keeps using its Existing DNS server as the default, and VPN-Managed Configuration where the VPN Client overrides the DNS server settings during setup to a third-party DNS of the VPN Service provider. [8][14]

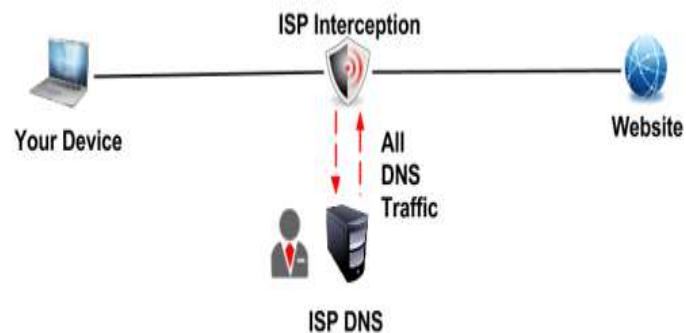


Figure 4: DNS hijacking/leaking through DNS server

#### 2.1.6 Man-in-the-Middle Attacks:

Before the connection is established between client and server, it is ARP cache poisoned to launch a Man-in-the-Middle Attack. The client fooled into believing that the attacker is the VPN server, and the server that the attacker is the VPN client, forcing both to relay messages through the attacker. [13]

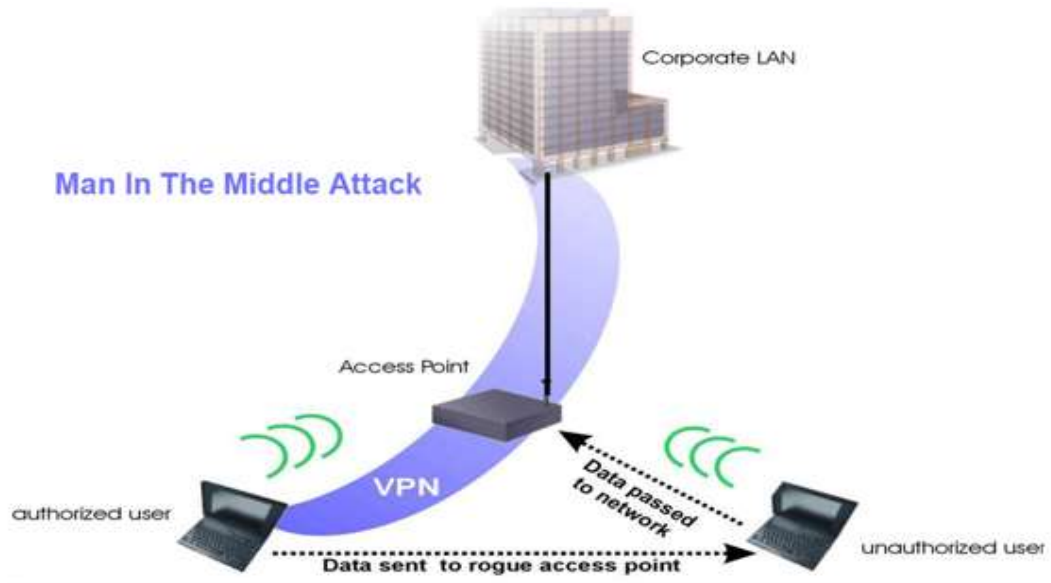


Figure 5: MITM Man-in-The-Middle Attack

### 2.1.7 VPN Encryption:

To guarantee the privacy of user data traversing through the shared or public web of networks, VPNs must keep it encrypted on the sender side and decrypted on the receiver side. As data encryption is a method done between the client and server, we do not require using encryption on the link of communication among the dial-up client and the Internet service provider (ISP).

For instance, a mobile phone client uses a dial-up connection to establish a connection to a local ISP. Once the connection is made, the user connects to the VPN server by a VPN connection. If there is

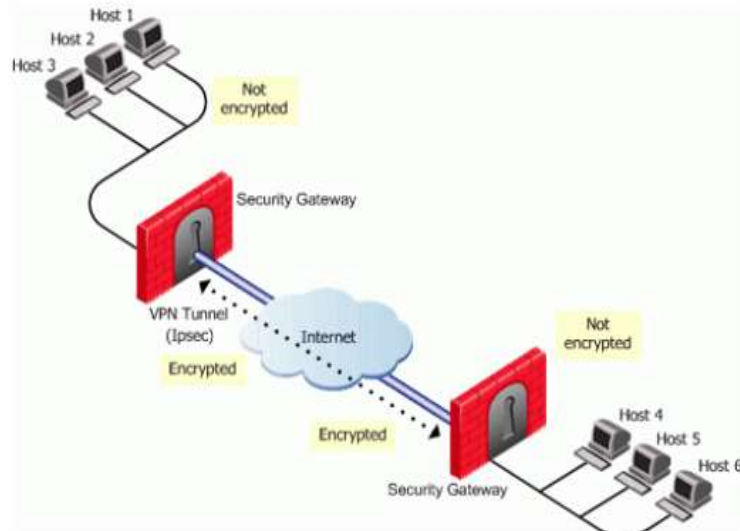


Figure 6: VPN Encryption over the Internet

encryption in a VPN connection, then we do not need to use encryption on the dial-up connection amongst the client and the ISP.

### 2.1.8 IPsec Based VPNs:

The IPsec based VPN use software at client end to connect to VPN server contrary to SSL VPN configuration where only browser is used. In second step a method of pentesting the VPN setup is described, and then moving forward to review the structure and its classified configuration is advocated. A complete VPN assessment should comprise of all probable attack courses for it to be a convenient instrument of security provision. A VPN is categorized into two different categories, Site-to-Site VPN, that virtually spreads the commercial LAN to that organization's satellite offices.

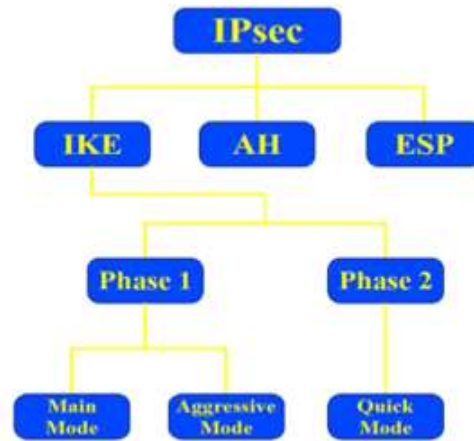


Figure 7: IPsec Composition and Architecture

IPsec VPNs necessitate an impenetrable VPN client at layer-3, contrary to SSL-based VPNs which only require to set up connectivity to its internal resources using a browser on the client end.

IPsec employs symmetric-key encryption and involves the following key security mechanisms:

- **AH (Authentication Header):** It is essential for legitimacy of message, an attached checksum to every packet ensures the authenticity of the message and keeps its integrity as the packet traverses over the internet.
- **ESP (Encapsulating Security Payload):** It is the encryption method made to protect the confidentiality and integrity of communication among the negotiating entities.
- **IKE (Internet Key Exchange):** It is the protocol that delivers the means to exchange the secret authentication key securely, which is necessary to effectively operate the AH and ESP among the connected subjects. Although exchanging the secret authentication keys is not achievable manually, the keys must be updated intermittently to decrease the probability of them being compromised.



IKE Main Mode employs the Diffie-Hellman key exchange to produce a pre-shared key among the server and the client. In contrast to IKE Main Mode, IKE Aggressive Mode does not actually use the Diffie-Hellman key exchange to shield the authentication credentials. Hence, it is possible to sniff these authentication credentials using a sniffer, and then it can be cracked offline.

## **2.2 Problem Formulation:**

VPNs use a secure channel and encryption to make a secure tunnel, and a key exchange phase to generate a symmetric Session Key. In key Exchange phase while establishing a VPN connection, the inherent vulnerability of a VPN can be exploited by forcing server to use aggressive mode authentication, where Pre-Shared Key is usually sent in plain text and can be sniffed by using PSK-crack/TCP-dump or Cain and Abel.

There are many ways to consider the key exchange phase and the actual data protection. The tunneling is just an implementation detail, but there can be security problems with IPv6/IPv4 leakage, User Anonymity and DNS leakage by eccentric combinations of AH/ESP when using IPsec or combining UDP with TLS, or information leakage due to compatibility/performance requirements of the tunneling process.

## **CHAPTER 3**

### **DESIGN SPECIFICATIONS**

#### **3.1 Technical specifications:**

This project is technically design on different software and hardware tools. The list of some of the hardware and software used in this project to give the desired output is given below, describing the features of the desired material required for this project.

### 3.1.1 Raspberry Pi 3b:

The Raspberry Pi 3 is the third-generation Raspberry Pi.

- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
- 1GB RAM
- 4 USB 2 ports
- DSI display port for connecting a Raspberry Pi touchscreen display
- Micro SD port for loading your operating system and storing data
- Upgraded switched Micro USB power source up to 2.5A

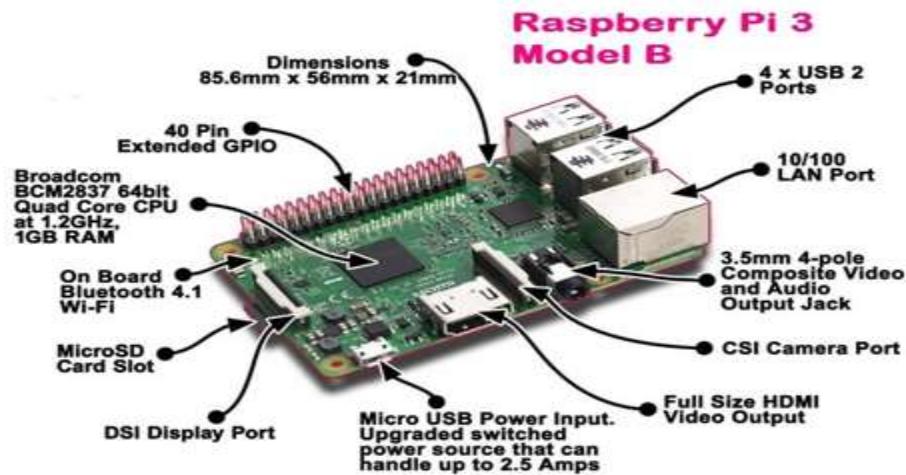


Figure 8: Raspberry Pi 3b

### 3.1.2 DumpIt:

DumpIt is comprised of two trusted tools, win32dd and win64dd, combined into one executable. DumpIt will take the snapshot of the host's physical memory and save it to the folder where the DumpIt executable file was located.

### 3.1.3 Volatility:

Volatility, an open-source memory-forensics framework to perform digital investigations, incident responses and malware analysis. It can also be used to monitor the connections of windows applications to which IP they are connected to and from which port information is being sent.

### 3.1.4 Nmap (Network Mapper) 7.60:

It is the open source network mapping tool used for the security scanner of the network and for the port scanning of the VPN server to find the open ports.

### 3.1.5 IKE-Scan 1.9:

For IPsec VPN we use tool called IKE-Scan that can Fingerprint many VPN vendors and Models.

### 3.2 Design Requirements:

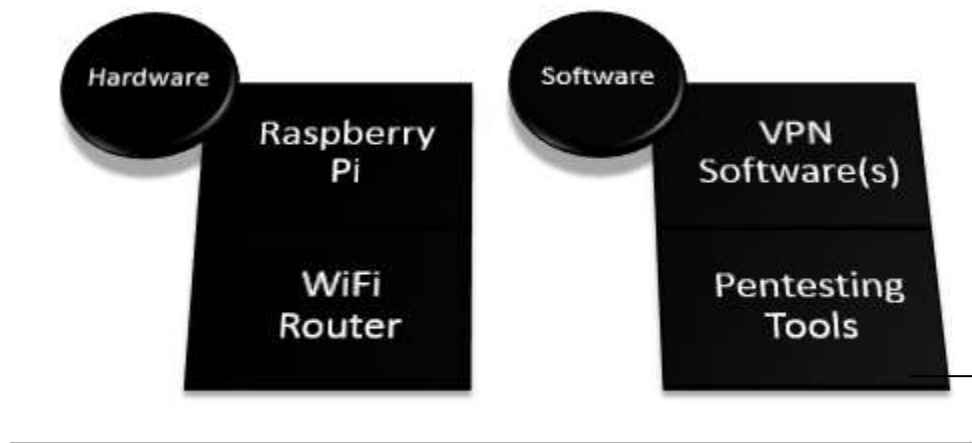


Figure 9: Design Requirements

### 3.4 Detailed Design with justification:

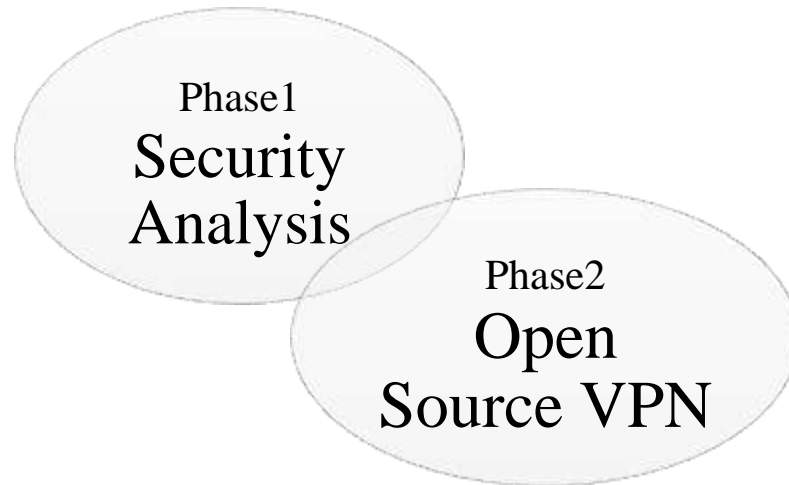


Figure 10: Project Design

#### Phase 1:

The project will first analyze the different VPN services using Pentesting tools like IKE-Scan, Volatility and Evil Foca and using the same tools, the project will perform several types of attacks to check IP leakage and DNS hijacking vulnerability which can potentially jeopardize the user's anonymity.

#### Phase 2:

Using the same data acquired from the previous analysis, the project will customize open source VPNs accordingly and create a new and secure customized open source VPN, which will then be installed in a raspberry pi device, and will act as a server.

## CHAPTER 4

### VPN Testing

#### 4.1 Information Gathering:

In order to get a understanding what the VPN app is sending the info to the VPN server, we have analysed the dump file of the VPN app by using the winhex application.

These are following steps we have followed

1. In order to acquire the dump of hide.me VPN we will access the detail tab of the task manager and locate the hide me.exe file. [Appendix B(i)]
2. We will right click on the hide me.exe and select the “create the dump file” option. [Appendix B(ii)]
3. After we have selected the dump file option, we have initiated the process of dumping process and when it is completed, it will instruct us where it is located. [Appendix B(iii)]
4. This is where the dump file is located and now we will open it using the winhex application. [Appendix B(iv)]
5. Winhex is helpful in the area of computer forensics, data recovery, low-level data processing, and IT security. An advanced tool to inspect and edit all kinds of files, recover deleted files or lost data from hard drives with corrupt file systems or from digital camera cards. [Appendix B(v)]
6. Now we will start the process of finding the relevant info in the dump file. [Appendix B(vi)]

Here we can see that what type of user data sent by a VPN app.

1. Common Variable Folders:

Folder variables refer to folders that are located differently on various Windows platforms and even individual computers. These variables are used by both legitimate applications and malware when accessing, changing, or creating files or folders. This page provides a reference for the default locations referenced by common folder variables.

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
02545530	50	61	63	6B	61	72	64	5C	53	69	6D	70	6C	65	50	61
02545540	73	73	5C	3B	43	3A	5C	50	72	6F	67	72	61	6D	20	46
02545550	69	6C	65	73	5C	49	6E	74	65	6C	5C	49	6E	74	65	6C
02545560	28	52	29	20	4D	61	6E	61	67	65	6D	65	6E	74	20	45
02545570	6E	67	69	6E	65	20	43	6F	6D	70	6F	6E	65	6E	74	73
02545580	5C	44	41	4C	3B	43	3A	5C	50	72	6F	67	72	61	6D	20
02545590	46	69	6C	65	73	5C	49	6E	74	65	6C	5C	49	6E	74	65
025455A0	6C	28	52	29	20	4D	61	6E	61	67	65	6D	65	6E	74	20
025455B0	45	6E	67	69	6E	65	20	43	6F	6D	70	6F	6E	65	6E	74
025455C0	73	5C	49	50	54	3B	43	3A	5C	50	72	6F	67	72	61	6D
025455D0	20	46	69	6C	65	73	20	28	78	38	36	29	5C	49	6E	74
025455E0	65	6C	5C	49	6E	74	65	6C	28	52	29	20	4D	61	6E	61
025455F0	67	65	6D	65	6E	74	20	45	6E	67	69	6E	65	20	43	6F
02545600	6D	70	6F	6E	65	6E	74	73	5C	44	41	4C	3B	43	3A	5C

### 2. Public Key token:

The public key token is the hash of public key which is 64-bit, corresponds to the private key used to sign the assembly. It is used to make an assembly name unique, so two strongly named assemblies can have the same filename, but .NET will treat them as different assemblies.

09AA0E30	9E	2E	01	80	84	53	79	73	74	65	6D	2E	53	65	63	75
09AA0E40	72	69	74	79	2E	50	65	72	6D	69	73	73	69	6F	6E	73
09AA0E50	2E	53	65	63	75	72	69	74	79	50	65	72	6D	69	73	73
09AA0E60	69	6F	6E	41	74	74	72	69	62	75	74	65	2C	20	6D	73
09AA0E70	63	6F	72	6C	69	62	2C	20	56	65	72	73	69	6F	6E	3D
09AA0E80	34	2E	30	2E	30	2E	30	2C	20	43	75	6C	74	75	72	65
09AA0E90	3D	6E	65	75	74	72	61	6C	2C	20	50	75	62	6C	69	63
09AA0EA0	4B	65	79	54	6F	6B	65	6E	3D	62	37	37	61	35	63	35
09AA0EB0	36	31	39	33	34	65	30	38	39	15	01	54	02	10	53	6B
09AA0EC0	69	70	56	65	72	69	66	69	63	61	74	69	6F	6E	01	80
09AA0ED0	9B	2E	01	80	84	53	79	73	74	65	6D	2E	53	65	63	75
09AA0EE0	72	69	74	79	2E	50	65	72	6D	69	73	73	69	6F	6E	73
09AA0EF0	2E	53	65	63	75	72	69	74	79	50	65	72	6D	69	73	73
09AA0F00	69	6F	6E	41	74	74	72	69	62	75	74	65	2C	20	6D	73
09AA0F10	63	6F	72	6C	69	62	2C	20	56	65	72	73	69	6F	6E	3D
09AA0F20	34	2E	30	2E	30	2E	30	2C	20	43	75	6C	74	75	72	65
09AA0F30	3D	6E	65	75	74	72	61	6C	2C	20	50	75	62	6C	69	63
09AA0F40	4B	65	79	54	6F	6B	65	6E	3D	62	37	37	61	35	63	35

### 3. Public Key:

No methods or constants contained by this interface. It simply serves to groups (and provide type safety for) of all public key interfaces.

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0AB153C0	01	00	81	5E	50	70	6C	53	63	72	69	70	74	2E	50	72	^PplScript.Pr
0AB153D0	6F	74	6F	63	6F	6C	2C	20	50	75	62	6C	69	63	4B	65	otocol, PublicKe
0AB153E0	79	3D	30	30	32	34	30	30	30	30	30	34	38	30	30	30	y=00240000048000
0AB153F0	30	30	39	34	30	30	30	30	30	30	30	36	30	32	30	30	0094000000060200
0AB15400	30	30	30	30	32	34	30	30	30	30	35	32	35	33	34	31	0000240000525341
0AB15410	33	31	30	30	30	34	30	30	30	30	30	31	30	30	30	31	3100040000010001
0AB15420	30	30	30	37	64	31	66	61	35	37	63	34	61	65	64	39	0007d1fa57c4aed9
0AB15430	66	30	61	33	32	65	38	34	61	61	30	66	61	65	66	64	f0a32e84aa8taefd
0AB15440	30	64	65	39	65	38	66	64	36	61	65	63	38	66	38	37	0de9e8fd6aec8f87
0AB15450	66	62	30	33	37	36	36	63	38	33	34	63	39	39	39	32	fb03766c834c9992
0AB15460	31	65	62	32	33	62	65	37	39	61	64	39	64	35	64	63	1eb23be79ad9d5dc
0AB15470	63	31	64	64	39	61	64	32	33	36	31	33	32	31	30	32	cldd9ad236132102
0AB15480	39	30	30	62	37	32	33	63	66	39	38	30	39	35	37	66	900b723cf980957f
0AB15490	63	34	65	31	37	37	31	30	38	66	63	36	30	37	37	37	c4e177108fc60777
0AB154A0	34	66	32	39	65	38	33	32	30	65	39	32	65	61	30	35	4f29e8320e92ea05
0AB154B0	65	63	65	34	65	38	32	31	63	30	61	35	65	66	65	38	ece4e821c0a5efe8
0AB154C0	66	31	36	34	35	63	34	63	30	63	39	33	63	31	61	62	f1645c4c0c93clab
0AB154D0	39	39	32	38	35	64	36	32	32	63	61	61	36	35	32	63	99285d622caa652c

#### 4. Public IP:

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00E0AEE0	61	63	68	65	2C	20	70	72	69	76	61	74	65	0D	0A	58	ache, private X
00E0AEF0	2D	46	72	61	6D	65	2D	4F	70	74	69	6F	6E	73	3A	20	-Frame-Options:
00E0AF00	53	41	4D	45	4F	52	49	47	49	4E	0D	0A	0D	0A	38	38	SAMEORIGIN 88
00E0AF10	0D	0A	7B	22	6C	61	74	22	3A	33	33	2E	36	39	35	37	{"lat":33.6957
00E0AF20	2C	22	6C	6F	6E	22	3A	37	33	2E	30	31	31	33	2C	22	,"lon":73.0113,"
00E0AF30	63	69	74	79	4E	61	6D	65	22	3A	22	49	73	6C	61	6D	cityName":"Islam
00E0AF40	61	62	61	64	22	2C	22	63	6F	75	6E	74	72	79	4E	61	abad","countryNa
00E0AF50	6D	65	22	3A	22	50	61	6B	69	73	74	61	6E	22	2C	22	me":"Pakistan","
00E0AF60	63	6F	75	6E	74	72	79	43	6F	64	65	22	3A	22	50	4B	countryCode":"PK
00E0AF70	22	2C	22	69	70	22	3A	22	31	30	31	2E	35	30	2E	36	","ip":"101.50.6
00E0AF80	36	2E	38	30	22	2C	22	69	73	43	6F	6E	6E	65	63	74	6.80".isConnect
00E0AF90	65	64	22	3A	66	61	6C	73	65	7D	0D	0A	30	0D	0A	0D	ed":false} 0
00E0AFA0	0A	64	65	2E	6D	65	22	2C	22	66	6C	61	67	22	3A	22	de.me","flag":
00E0AFB0	63	61	22	2C	22	64	69	73	70	6C	61	79	4E	61	6D	65	ca","displayName
00E0AFC0	22	3A	22	43	61	6E	61	64	61	22	2C	22	69	73	50	61	":"Canada","isPa
00E0AFD0	69	64	22	3A	66	61	6C	73	65	2C	22	69	73	50	32	50	id":false,"isP2P
00E0AFE0	22	3A	74	72	75	65	7D	2C	7B	22	69	64	22	3A	31	35	":true},{"id":15
00E0AFF0	39	2C	22	68	6F	73	74	6E	61	6D	65	22	3A	22	66	72	9,"hostname":"fr

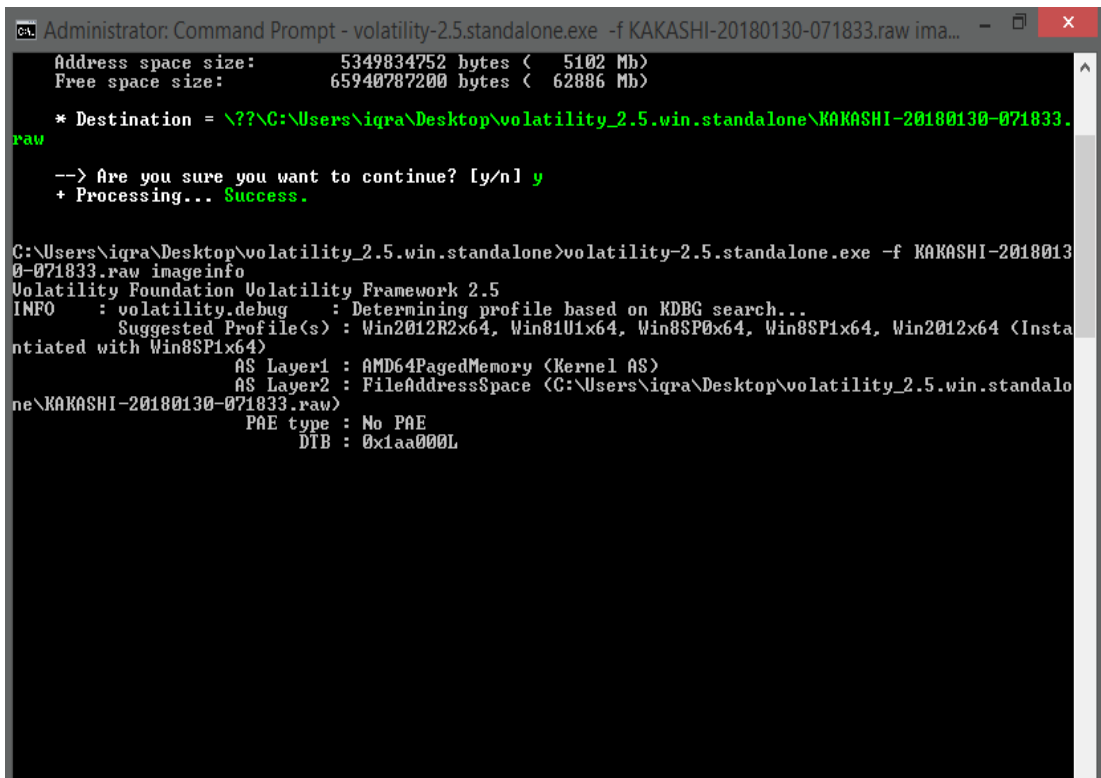


## 4.2 Prerequisite of the Zero knowledge test:

In order to complete the prerequisite of the zero-knowledge test, i.e. finding the VPN server IP, we use memory forensics tools to find out the client's connection to the VPN server. We use Volatility Framework and DumpIt tool to get the required information.

### 4.2.1 Memory Forensics:

The Volatility Framework consists of a list of tools developed in Python for the elimination of digital objects from volatile memory (RAM) sections. The data extraction methods are executed totally independent of the system under investigation, but also offer unparalleled perceptibility into the system in runtime state. This framework intends to present procedures that help us to excerpt digital artifacts from the samples of your virtual memory and deliver a platform for additional work in research area.



```
Administrator: Command Prompt - volatility-2.5.standalone.exe -f KAKASHI-20180130-071833.raw ima...
Address space size:      5349834752 bytes ( 5102 Mb)
Free space size:        65940787200 bytes ( 62886 Mb)

* Destination = \\??\C:\Users\iqra\Desktop\volatility_2.5.win.standalone\KAKASHI-20180130-071833.raw
--> Are you sure you want to continue? [y/n] y
+ Processing... Success.

C:\Users\iqra\Desktop\volatility_2.5.win.standalone>volatility-2.5.standalone.exe -f KAKASHI-20180130-071833.raw imageinfo
Volatility Foundation Volatility Framework 2.5
INFO : volatility.debug : Determining profile based on KDBG search...
Suggested Profile(s) : Win2012R2x64, Win81U1x64, Win8SP0x64, Win8SP1x64, Win2012x64 (Instantiated with Win8SP1x64)
AS Layer1 : AMD64PagedMemory (Kernel AS)
AS Layer2 : FileAddressSpace (C:\Users\iqra\Desktop\volatility_2.5.win.standalone\KAKASHI-20180130-071833.raw)
PAE type : No PAE
DTB : 0x1aa000L
```

We used the DumpIt tool in order to get the image of the whole memory.



```

C:\Users\iqra\Desktop\volatility_2.5.win.standalone>volatility-2.5.standalone.exe -f KAKASHI-20180130-071833.raw imageinfo
Volatility Foundation Volatility Framework 2.5
INFO : volatility.debug : Determining profile based on KDBG search...
      Suggested Profile(s) : Win2012R2x64, Win81U1x64, Win8SP0x64, Win8SP1x64, Win2012x64 (Instantiated with Win8SP1x64)
      AS Layer1 : AMD64PagedMemory (Kernel AS)
      AS Layer2 : FileAddressSpace (C:\Users\iqra\Desktop\volatility_2.5.win.standalone\KAKASHI-20180130-071833.raw)
      PAE type : No PAE
      DTB : 0x1aa000L
      KDBG : 0xf8019ed26530L
      Number of Processors : 4
      Image Type (Service Pack) : 0
      KPCR for CPU 0 : 0xfffff8019ed75000L
      KPCR for CPU 1 : 0xffffd00182760000L
      KPCR for CPU 2 : 0xffffd001827e0000L
      KPCR for CPU 3 : 0xffffd0017e5db000L
      KUSER_SHARED_DATA : 0xfffff78000000000L
      Image date and time : 2018-01-30 07:18:35 UTC+0000
      Image local date and time : 2018-01-30 12:18:35 +0500

```

In the Volatility Framework's imageinfo command tells the profile, the person should use the parameter --profile=PROFILE when they use other plugins. There can be more profile suggestion than just one if those profiles seem to be related closely. Moreover, the KDBG structure's kernel address is printed by it that will be employed by the plugins such as pslist and components to look for the process as well as module list heads.

```

C:\Users\iqra\Desktop\volatility_2.5.win.standalone>volatility-2.5.standalone.exe -f KAKASHI-20180130-071833.raw --profile=Win2012R2x64 kdbgscan
Volatility Foundation Volatility Framework 2.5
*****
Instantiating KDBG using: Unnamed AS Win2012R2x64 (6.3.9601 64bit)
Offset (U) : 0xf8019ed26530
Offset (P) : 0x13b926530
KdCopyDataBlock (U) : 0xf8019ec591f4
Block encoded : Yes
Wait never : 0xdfc9d575c00e2e3a
Wait always : 0x1c5c7b706aaf2800
KDBG owner tag check : True
Profile suggestion (KDBGHeader): Win2012R2x64
Version64 : 0xf8019ed26e58 (Major: 15, Minor: 9600)
Service Pack (CnNtCSDVersion) : 0
Build string (NtBuildLab) : 9600.18895.amd64fre.winblue_ltsh
PsActiveProcessHead : 0xfffff8019ed3f460 (65 processes)
PsLoadedModuleList : 0xfffff8019ed596d0 (167 modules)
KernelBase : 0xfffff8019ea8c000 (Matches MZ: True)
Major (OptionalHeader) : 6
Minor (OptionalHeader) : 3
KPCR : 0xfffff8019ed75000 (CPU 0)
KPCR : 0xffffd00182760000 (CPU 1)
KPCR : 0xffffd001827e0000 (CPU 2)
KPCR : 0xffffd0017e5db000 (CPU 3)

C:\Users\iqra\Desktop\volatility_2.5.win.standalone>

```

As opposite to imageinfo that shows profile suggestions, KDGBscan is intended to recognize the right profile and the right KDBG address. This plugin looks for the KDBG header signatures interrelated to Volatility profiles and checks them by

applying sanity checks. The number of times the sanity checks performed and the interminability of the output depends on if Volatility can catch a DTB, so make sure you use the correct profile if you already know it.

The netscan command is used to scan for network artifacts in 32- and 64-bit Windows memory dumps. It manages to find the TCP endpoints and listeners and UDP endpoints and listeners. It can distinguish among IPv4 and IPv6 and depending on the applicability, it can also print the local and the remote IP addresses and the local and remote port, it can also easily print the time when the communication socket was bound or when the machine had an established connection, and its current state, if there is a TCP connection. Here we have found the established connection between our device and a VPN server IP.

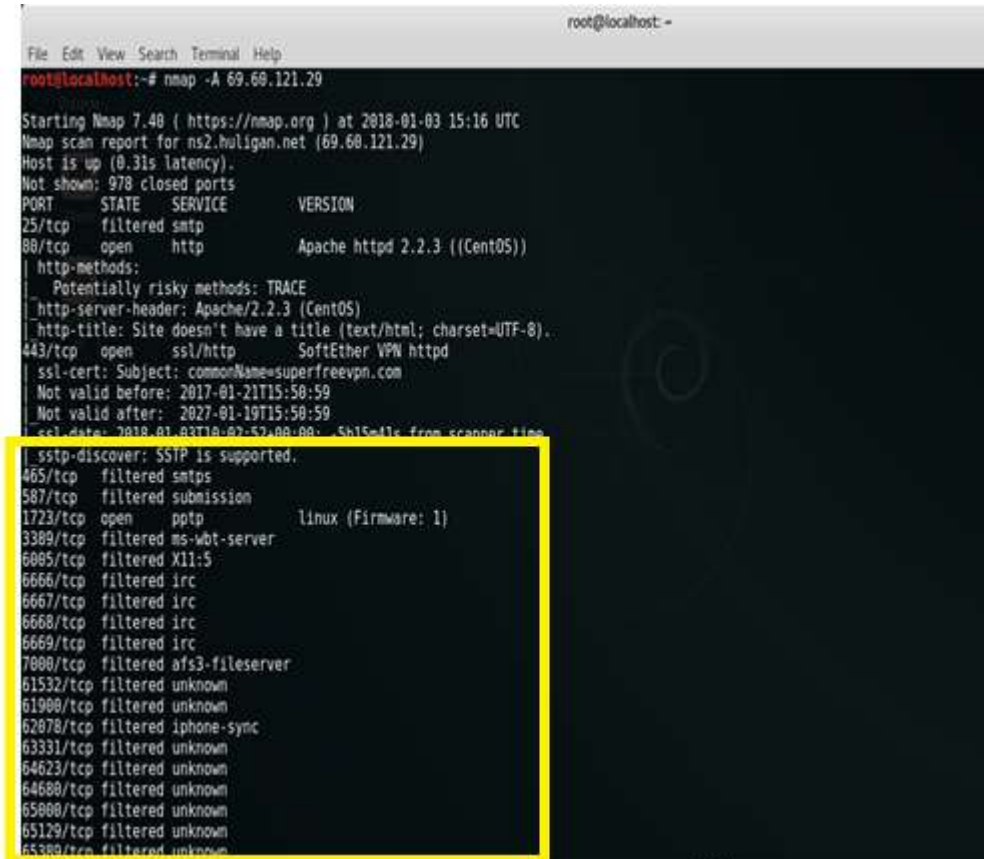


In above screen shot we are able to tell that VPN server has allotted the host IP address of 10.110.32.52 and communicating through the port number 52359. Through Netscan we discover the VPN server private IP 104.20.123.38 and we can tell it is communicating through port 443.

## 4.3 Zero knowledge test:

### 4.3.1 Port Scanning:

After acquiring the VPN server IP address, we performed zero knowledge test on the VPN server, starting with port scanning by Nmap.



```
root@localhost:~# nmap -A 69.68.121.29
Starting Nmap 7.40 ( https://nmap.org ) at 2018-01-03 15:16 UTC
Nmap scan report for ns2.huligan.net (69.68.121.29)
Host is up (0.31s latency).
Not shown: 978 closed ports
PORT      STATE SERVICE        VERSION
25/tcp    filtered smtp
80/tcp    open  http           Apache httpd 2.2.3 ((CentOS))
|_ http-methods:
|_ Potentially risky methods: TRACE
|_ http-server-header: Apache/2.2.3 (CentOS)
|_ http-title: Site doesn't have a title (text/html; charset=UTF-8).
443/tcp   open  ssl/http       SoftEther VPN httpd
|_ ssl-cert: Subject: commonName=superfreevpn.com
|_ Not valid before: 2017-01-21T15:50:59
|_ Not valid after: 2027-01-19T15:50:59
|_ ssl_date: 2018-01-03T18:07:52.00:00 -5h15m11s from scanner time
|_ sstp-discover: SSTP is supported.
465/tcp   filtered smtps
587/tcp   filtered submission
1723/tcp  open  pptp           linux (Firmware: 1)
3389/tcp  filtered ms-wbt-server
6005/tcp  filtered X11:5
6666/tcp  filtered irc
6667/tcp  filtered irc
6668/tcp  filtered irc
6669/tcp  filtered irc
7000/tcp  filtered afs3-fileserver
61532/tcp filtered unknown
61900/tcp filtered unknown
62078/tcp filtered iphone-sync
63331/tcp filtered unknown
64623/tcp filtered unknown
64680/tcp filtered unknown
65000/tcp filtered unknown
65129/tcp filtered unknown
65380/tcp filtered unknown
```

The following are the open ports to VPN type. using default ports:

Type of VPN Implementation	Port
IPsec	UDP 500
PPTP/L2TP	TCP 1723
SSL	TCP 443

Table 4.3.1 Mapping of Open ports to VPN type

### 4.3.2 Packet Tracing:

The Nmap scan report showed the operating ports and their states whether they are open, closed, or filtered. It also has shown potentially risky methods that may compromise a client's anonymity i.e. http TRACE.

```
root@localhost: ~
File Edit View Search Terminal Help
64623/tcp filtered unknown
64688/tcp filtered unknown
65008/tcp filtered unknown
65129/tcp filtered unknown
65309/tcp filtered unknown
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6.18
OS details: Linux 2.6.18
Network Distance: 20 hops
Service Info: Host: local

Host script results:
|_clock-skew: mean: -5h15m41s, deviation: 0s, median: -5h15m41s

TRACEROUTE (using port 1720/tcp)
HOP RTT ADDRESS
1 6.06 ms mobile.router (192.168.1.1)
2 ...
3 35.03 ms 10.10.205.117
4 36.31 ms 10.0.2.149
5 35.04 ms 10.0.1.2
6 36.33 ms UKLI.rwp44.pie.net.pk (202.125.151.106)
7 56.90 ms static-kh1275-P02-pie.net.pk (221.120.254.6)
8 59.05 ms mi77c2-5w494b.pie.net.pk (202.125.128.134)
9 161.90 ms te0-5-0-18.ccr22.mrs01.atlas.cogentco.com (149.14.125.113)
10 171.91 ms be3893.ccr42.par01.atlas.cogentco.com (130.117.50.165)
11 176.88 ms be12489.ccr42.lon13.atlas.cogentco.com (154.54.57.69)
12 246.48 ms be2490.ccr42.jfk02.atlas.cogentco.com (154.54.42.85)
13 362.23 ms be2807.ccr42.dca01.atlas.cogentco.com (154.54.40.110)
14 375.17 ms be2113.ccr42.atl01.atlas.cogentco.com (154.54.24.222)
15 375.19 ms be2123.ccr22.mia01.atlas.cogentco.com (154.54.24.198)
16 391.04 ms te0-0-1-0.agr12.mia01.atlas.cogentco.com (154.54.3.174)
17 341.98 ms te0-0-2-1.nr11.b015452-0.mia01.atlas.cogentco.com (154.24.31.58)
18 294.56 ms 38.104.90.50
19 274.07 ms 69.60.96.103
20 289.40 ms ms2.hulligan.net (69.60.121.29)
```

This particular uses SSL implementation as it uses 443/TCP, and PPTP/L2TP implementation as it uses 1723/TCP.

### 4.3.3 Finger Printing:

After port scanning and packet tracing next we fingerprint using the IKE Scan tool (developed by NTA Monitor). It compares the values of explicit variables in the IPsec packets exchange, against its signature database.

These transforms represent the attributes such as Encryption algorithm, Hash algorithm, Authentication method, and Diffie-Hellman group. And can take up different values such as DES or AES as the encryption algorithm, SHA or MD5 as the hashing algorithm, a PSK or RSA. Diffie-Hellman 1 or 2 as the key distribution algorithm and 28800 seconds as the lifetime. We try these different values to prompt an IKE handshake response from the server if we did not get one from the identification stage.

Once a handshake received, we take note of the acceptable transform set for future scans.

```
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 106.80 seconds
root@localhost:~# ike-scan -H 69.60.121.29
Starting ike-scan 1.9.4 with 1 hosts (http://www.nta-monitor.com/tools/ike-scan/)
69.60.121.29 Main Mode Handshake returned
HDR=(CKY-R=fcf99a1939f66169)
SA=(Enc=3DES Hash=SHA1 Auth=PSK Group=2:modp1024 LifeType=Seconds LifeDuration=28800)
VID=4a131c8107e358455c5728f20e95452f (RFC 3947 NAT-T)
VID=7d9419a65310ca6f2c179d9215529d56 (draft-ietf-ipsec-nat-t-ike-03)
VID=90cb88913ebb696e086381b5ec427b1f (draft-ietf-ipsec-nat-t-ike-02)
VID=cd60464335df21f87cfdb2fcd68b6a448 (draft-ietf-ipsec-nat-t-ike-02)
VID=4485152d1806bbcd80a8a8469579ddcc (draft-ietf-ipsec-nat-t-ike-00)
VID=afcad71368a1f1c96b8696fc77570100 (Dead Peer Detection v1.0)
Ending ike-scan 1.9.4: 1 hosts scanned in 0.720 seconds (1.39 hosts/sec). 1 returned handshake; 0 returned notify
root@localhost:~#
```

In the example, the VPN gateway replies with one returned handshake and the acceptable transform set is

- a) Enc=3DES,
- b) Hash=SHA1
- c) Auth=PSK

#### 4.3.4 Back off Strategy:

Then use retransmission back off strategy where IKE-scan sends the acceptable IKE handshake to the server and stops replying to the server's responses. The server keeps retransmitting packets until it get a response. The -showbackoff option causes ike-scan to record the response time of all of these packets. By carefully analyzing the time difference between each packet being sent from the server, it can successfully fingerprint the VPN gateway vendor.



```

Ending ike-scan 1.9.4: 1 hosts scanned in 70.582 seconds (0.01 hosts/sec) 1 returned handshake; 0 returned notify
root@localhost:~# ike-scan -M --showbackoff 69.60.121.29
Starting ike-scan 1.9.4 with 1 hosts (http://www.nta-monitor.com/tools/ike-scan/)
69.60.121.29 Main Mode Handshake returned
HDR=(CKY-R=016b1b364245260e)
SA=(Enc=3DES Hash=SHA1 Auth=PSK Group=2:modp1024 LifeType=Seconds LifeDuration=28000)
VID=4a131c81078358455c5728f20e95452f (RFC 3947 NAT-T)
VID=7d9419a65310ca6f2c179d9215529d56 (draft-ietf-ipsec-nat-t-ike-03)
VID=99cb80913ebb696e006381b5ec427b1f (draft-ietf-ipsec-nat-t-ike-02\n)
VID=cd60464335df21f87cfd02fc68b6a44b (draft-ietf-ipsec-nat-t-ike-02)
VID=4485152d18b6bbcd0be8a8469579d0cc (draft-ietf-ipsec-nat-t-ike-00)
VID=afcad71368a1fc96b8696fc7757010b (Dead Peer Detection v1.0)

IKE Backoff Patterns:
IP Address      No.    Recv time      Delta Time
69.60.121.29   1      1514993090.657920  0.000000
69.60.121.29   2      1514993092.804102  2.146182
69.60.121.29   3      1514993094.749267  1.945165
69.60.121.29   4      1514993096.812496  2.063229
69.60.121.29   5      1514993098.742406  1.929910
69.60.121.29   6      1514993100.687968  1.945562
69.60.121.29   Implementation guess: UNKNOWN

Some IKE implementations found have unknown backoff fingerprints
If you know the implementation name, and the pattern is reproducible, you
are encouraged to submit the pattern and implementation details for
inclusion in future versions of ike-scan. See:
http://www.nta-monitor.com/tools/ike-scan/submit-patterns.html
Ending ike-scan 1.9.4: 1 hosts scanned in 70.582 seconds (0.01 hosts/sec) 1 returned handshake; 0 returned notify
root@localhost:~# █

```

#### 4.4 Username Enumeration Vulnerabilities:

##### 4.4.1 IKE Aggressive mode:

Many VPNs use IKE Aggressive Mode with pre-shared key (PSK) authentication to authenticate the username and password. In this mode, the client sends an IKE packet containing several ISAKMP payloads to the VPN server that responds with a reply IKE packet. It is to be noted that the client sends the Identity payload and the Server replies with Hash payload, which is an HMAC hash of information including the password (pre-shared key). The Client then sends a third packet containing an HMAC hash of information including the password. IKE aggressive mode do not protect the authentication of data exchange and send authentication of hash in plane text.

```

Applications + Places + Terminal + Wed 15:29
root@localhost: ~
File Edit View Search Terminal Help
VID=4485152d18b6bcb0be8a8469579dccc (draft-ietf-ipsec-nat-t-ike-09)
VID=afcad71368a1f1c96b8696fc77570180 (Dead Peer Detection v1.0)

Worryingly, the aggressive mode does not use a key exchange algorithm like Diffie-
Hellman to protect the authentication data exchange and sends the authentication

IKE Backoff Patterns:
IP Address      No.    Recv time      Delta Time
69.60.121.29   1      1514993180.459109  0.000000
69.60.121.29   2      1514993184.559075  4.100766
69.60.121.29   3      1514993186.501613  1.941738
69.60.121.29   4      1514993188.445904  1.944291
69.60.121.29   5      1514993190.509166  2.063262
69.60.121.29   Implementation guess: UNKNOWN

Some IKE implementations found have unknown backoff fingerprints
If you know the implementation name, and the pattern is reproducible, you
are encouraged to submit the pattern and implementation details for
inclusion in future versions of ike-scan. See:

Ending ike-scan 1.9.4: 1 hosts scanned in 70.473 seconds (0.01 hosts/sec). 1 returned handshake; 0 returned notify
root@localhost:~# ike-scan --pskcrack --aggressive --id=free 69.60.121.29
Starting ike-scan 1.9.4 with 1 hosts (http://www.nta-monitor.com/tools/ike-scan/)
69.60.121.29 Aggressive Mode Handshake returned HDR=(CKY-R=fea93166b77e67d6) SA=(Enc=3DES Hash=SHA1 Auth=PSK Group=2:modp1024 LifeType=Seconds Life
Duration=28800) KeyExchange(128 bytes) Nonce(16 bytes) ID(Type=ID_IPV4_ADDR, Value=69.60.121.29) Hash(20 bytes) VID=4a131c81870358455c5726f20e95452f (
RFC 3947 NAT-T) VID=7d9419e65310ca6f2c179d9215529d56 (draft-ietf-ipsec-nat-t-ike-03) VID=90cb8913ebb696e086381b5ec427b1f (draft-ietf-ipsec-nat-t-ike-
02/n) VID=cd68464335d721f87cfd2fc68b6a440 (draft-ietf-ipsec-nat-t-ike-02) VID=4485152d18b6bcb0be8a8469579dccc (draft-ietf-ipsec-nat-t-ike-09) VID=af
cad71368a1f1c96b8696fc77570180 (Dead Peer Detection v1.0)

IKE PSK parameters (g x:rig xl:cky r:cky l:sal b:ldlr b:ml b:nr b:hash r):
b39bc11ea56fd9616f7b478143fa7a816b385f1592c2775dbff6b66461ebb993cd95ac13b83732f6298bf5eab93261a03cf64442ac7a0856f83266be18e1f9515c84c9fb78908ac1b43e
0951dfff58063202f388f437af8720ff36626a56211f33358f7aa6d87b985efb8c8c363e0ff5fc66eb60c5260d985e11c57fe2a1ab:148ad25ade48d542bf757b20885453fd2c300ca2f65
81356a87e92031349e9363689f995fb01868c534772897f9d2d14cc2a6133168db078c6a2814cc5afcb0404175780b091c79740ab62ac1b5b044760b408bbe03a996a9922d265ff502fcb
f99a853ec3fba4099ca41242b0c4d7477150ab9a1427b68cc503379ffbb015:fea93166b77e67d6:8e040211ec702fcb:000000010000000100000009001010004030002401010000001
0005800200020003000400002000b0001000c000400007000000002404010000000100010002000100040002000b0001000c0004000070000000024030100000001000100
0200020003000400002000b0001000c000400007000000002404010000000100010002000100040002000b0001000c0004000070000000024030100000001000100
38f6478b2fad501ecbc840a194:c8aba02a94abbeec670b160d8776059:f88713580557f919f6a05449a25cb443d6c833be
Ending ike-scan 1.9.4: 1 hosts scanned in 0.353 seconds (2.84 hosts/sec). 1 returned handshake; 0 returned notify
root@localhost:~#

```

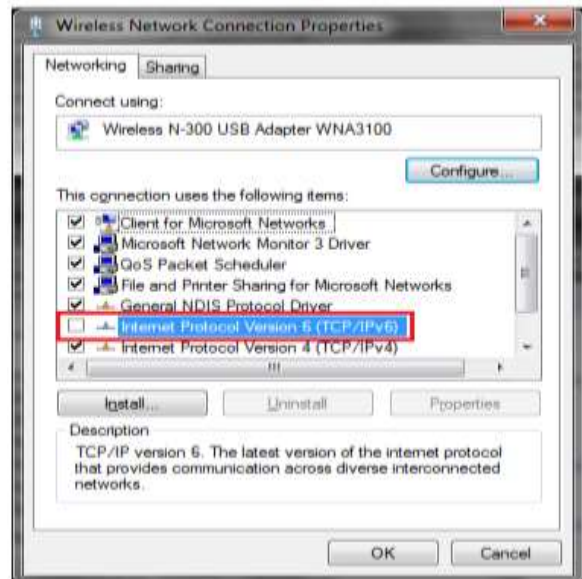
We use the IKE Scan Tool to provoke the VPN server into the aggressive mode of IPsec from the main mode by checking out various combinations. This makes it possible to capture the authentication data and use a brute force or dictionary attack to recover the PSK.

#### 4.5 Countermeasures:

##### 4.5.1 Defense against IPv6 Leakage:

The problem of IPv6 leakage stems from the relationship between VPN and the routing table of the Client Machine managed by the Kernel. We can mitigate this risk by disabling IPv6 traffic on the Client Machine. We can easily disable IPv6 on Windows via the Registry [23], Mac OS, Linux and others.

This defense is feasible but in the face of increasing IPv6 adoption, this shall be a short-term solution. It may also not be an



option for transportable devices that are at times used in a setting where IPv6 connectivity is needed.

#### **4.5.2 Authentication Vulnerabilities:**

Strong authentication by means of certificates, smart cards or tokens can be used when users are connecting to the VPN Server. A smart card stores user profiles, data encryption keys and encryption algorithms.

It is usually required to use a PIN for the smart card to be invoked. We can get an OTP (one time password) by the token card, and whenever a user tries to authenticate on the token and they enter a correct PIN, the card would display the OTP that will lead to grant user access to the network.

#### **4.5.3 Configuration Issues Management:**

Consider the advanced security measures taken by VyprVPN to tackle the configuration issues. The tunnel setup fails if the client routing table is not configured to the DNS Server managed by the VyprVPN. They inspected the traffic with TCPdump and found that on tunnel setup, the VPN client queries three random DNS lookups. If these queries are sent to a third-party DNS Server, the connection is not established and the tunnel shuts down.

The VPN client independently connects the VyprDNS server using the bespoke protocol to check if the queries are correctly received and replied. However, note that the check is only performed directly after the tunnel has been established and can be overcome by delaying the attack for 60 seconds using the DHCP lease time. The study [19] experimentally confirmed the possibility of the route injection attack on VyprVPN by using DHCP Lease time delay.



## 4.6 Top 10 Free VPNs Comparison

<i>Sr.</i>	<i>VPN Name</i>	<i>DNS Leak</i>	<i>IP Leak</i>	<i>WebRTC leak</i>	<i>Speed</i>	<i>Encryption</i>	<i>Tracking</i>
1	Super VPN	✓	✗	✓	Slow	✗	✓
2	Betternet	✓	✗	✓	Slow	✓	✗
3	Cross VPN	✓	✓	✗	Normal	✗	✗
4	Hide.me	✗	✗	✗	High	✗	✗
5	Flash Free VPN	✓	✗	✓	Slow	✓	✓
6	Hotspot Shield VPN	✗	✓	✗	Normal	✓	✓
7	SurfEasy	✗	✗	✓	Normal	✓	✓
8	VPNGate	✓	✗	✗	Slow	✗	✗
9	Windscribe	✗	✗	✗	Normal	✓	✓
10	Tunnelbear	✗	✗	✓	Normal	✓	✗

## CHAPTER 5

### CUSTOMIZED OPEN VPN

#### 5.1 Network Configuration:

##### Step 1: IPv4 Firewall Rules

A firewall blocks unwanted traffic and denies unauthorized network access to increase network safety. A firewall rule is a user-configured access policy that can restrict or permit access to a network. A firewall rule can be used add extra network protection. Here we will allow different ports and protocols like SSH, UDP 1194 to allow open VPN traffic and tunneling at TUN0 interface.

#### /etc/iptables/rules.v4

```
1 *filter
2
3 # Allow all loopback (lo) traffic and reject anything
4 # to localhost that does not originate from lo.
5 -A INPUT -i lo -j ACCEPT
6 -A INPUT ! -i lo -s 127.0.0.0/8 -j REJECT
7 -A OUTPUT -o lo -j ACCEPT
8
9 # Allow ping and ICMP error returns.
10 -A INPUT -p icmp -m state --state NEW --icmp-type 8 -j ACCEPT
11 -A INPUT -p icmp -m state --state ESTABLISHED,RELATED -j ACCEPT
12 -A OUTPUT -p icmp -j ACCEPT
13
14 # Allow SSH.
15 -A INPUT -i eth0 -p tcp -m state --state NEW,ESTABLISHED --dport 22 -j ACCEPT
16 -A OUTPUT -o eth0 -p tcp -m state --state ESTABLISHED --sport 22 -j ACCEPT
17
18 # Allow UDP traffic on port 1194.
19 -A INPUT -i eth0 -p udp -m state --state NEW,ESTABLISHED --dport 1194 -j ACCEPT
20 -A OUTPUT -o eth0 -p udp -m state --state ESTABLISHED --sport 1194 -j ACCEPT
21
22 # Allow DNS resolution and limited HTTP/S on eth0.
23 # Necessary for updating the server and timekeeping.
24 -A INPUT -i eth0 -p udp -m state --state ESTABLISHED --sport 53 -j ACCEPT
25 -A OUTPUT -o eth0 -p udp -m state --state NEW,ESTABLISHED --dport 53 -j ACCEPT
26 -A INPUT -i eth0 -p tcp -m state --state ESTABLISHED --sport 80 -j ACCEPT
27 -A INPUT -i eth0 -p tcp -m state --state ESTABLISHED --sport 443 -j ACCEPT
28 -A OUTPUT -o eth0 -p tcp -m state --state NEW,ESTABLISHED --dport 80 -j ACCEPT
29 -A OUTPUT -o eth0 -p tcp -m state --state NEW,ESTABLISHED --dport 443 -j ACCEPT
30
31 # Allow traffic on the TUN interface so OpenVPN can communicate with eth0.
32 -A INPUT -i tun0 -j ACCEPT
33 -A OUTPUT -o tun0 -j ACCEPT
34
35 # Log any packets which don't fit the rules above.
36 # (optional but useful)
37 -A INPUT -m limit --limit 3/min -j LOG --log-prefix "iptables_INPUT_denied: " --log-level 4
38 -A FORWARD -m limit --limit 3/min -j LOG --log-prefix "iptables_FORWARD_denied: " --log-level 4
39 -A OUTPUT -m limit --limit 3/min -j LOG --log-prefix "iptables_OUTPUT_denied: " --log-level 4
40
41 # then reject them.
42 -A INPUT -j REJECT
43 -A FORWARD -j REJECT
44 -A OUTPUT -j REJECT
45
46 COMMIT
```

## Step 2: Configuring OpenVPN

OpenVPN runs as the root in initial condition. The given user in the server.conf file has as less privileges as any other user (less than root) in any version of Linux. If the same user account gets compromised, the intruder trying to hijack the server will only

have the rights to obtain the user data, but it can also comprise Apache while using mod\_php or different NFS mounts.

To make OpenVPN service run in an entirely different configuration where it has a group and special user is an efficient method to keep it separated from the other processes, particularly if you want to host a web server or a file server on the same machine or server as your VPN server.

### VPN Certificate Authority

A VPN server should not have Client certificates and keys placed in it. It is safer to have them generated locally on a computer and be stored offline, without any internet access. To have the best results it is recommended to have them generated on a computer has a larger RAM size.

There are two ways that you can easily create the certificates and the keys:

- Using the scripts provided by EasyRSA
- By making public key infrastructure (PKI) yourself, for the VPN you want to develop.

### Step 3: Configure EasyRSA

For the configuration of Certificate Authority, we made root directory. The location is arbitrary for your CA. The certificates are created from that directory.

Presets used by EasyRSA contained by the vars file created in /ca folder. For your certificate authority you can provide a name and it will be passed on to client certificates.

```
~/ca/vars
1  # These are the default values for fields
2  # which will be placed in the certificate.
3  # Don't leave any of these fields blank.
4  export KEY_COUNTRY="US"
5  export KEY_PROVINCE="CA"
6  export KEY_CITY="SanFrancisco"
7  export KEY_ORG="Fort-Funston"
8  export KEY_EMAIL="me@myhost.mydomain"
9  export KEY_OU="MyOrganizationalUnit"
```

```

54.37.74.25 - PuTTY
root@vps525133:~/ca/2.0# ./clean-all
root@vps525133:~/ca/2.0# ./build-ca
Generating a 4096 bit RSA private key
.....++
.....++
writing new private key to 'ca.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [DE]:
State or Province Name (full name) [Frankfurt]:
Locality Name (eg, city) [Limburg]:
Organization Name (eg, company) [Tiamat corporation]:
Organizational Unit Name (eg, section) [MyOrganizationalUnit]:
Common Name (eg, your name or your server's hostname) [Tiamat corporation CA]:
Name [Shaukat]:
Email Address [ahmadali_asdzxc@hotmail.com]:
root@vps525133:~/ca/2.0# ./build-key-server server

```

```

<cert>
Certificate:  Data:          Version: 3 (0x2)          Serial Number:          60:b9:d8:d7:76:1
              40:55:1f:6d:b1:e4:e6:48:22:3b:9f:51:ba:a2:54:          b4:1f:96:0
:/CN=ChangeMe          serial:A7:D7:B1:40:B1:12:AD:F9          X509v3 Extended Key Usag
5:71:          74:53:6c:c0:37:cc:7a:05:5c:9d:2d:ae:57:e8:55:fd:6d:11:          1f:bf:d3:e9:33:60:
h2HFPw4Ze4pC4voyed08asCedyAZkwqsxcxjamXthxUPFtwtAxn70rwgB1B1mCRRoUPNRo7utKCLSGE4Qfzp86Ima3nPrTD
</cert>
<key>
-----BEGIN PRIVATE KEY-----MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBCwggSjAgEAAoIBAQC145andmQ3Jtg+8Aah
bo9gDG7pEz/QqKB0w8KiGZaiex485PWnD7g1bIL9x2Zo4IyHx1ULGDKGHZoMxbRUhGEWAayzxW9n2cvuvbsUh5Wn5JDrw0T
6ae1c2e44a2a50b379eee71ed873b6a2b0df7f1f00ca6c84a5965cdf1502dcf3b19d1d698e44f10afb690e943c9322
</tls-auth>

```

#### Step 4: Server Credentials

Sometimes a root certificate can be named as Certificate Authority. That is the certificate will be used to produce key pairs for clients and intermediate authorities. At each step we can add or edit the information used by your certificate. Confirm the signing of the certificate after the completion of question section for the private key.

## Step 5: Client Credentials

The device/machine connecting to the VPN server must have personal key and identifier.

All the other certificate information and other details can be remained the same or shared across all client devices.

## Step 6: Authentication

In order to authenticate with the VPN server, we will first install OpenVPN client on the client computer or machine. After the installation of the OpenVPN client we will import the certificates generated by the certification authority and the client.ovpn file. Using the instructions in the client.ovpn file, the OpenVPN client will use the certificate and key files and start a tunneling request on the tun/tap interface to the remote server.

## 5.2 OpenVPN Configuration Files:

### Step 1: Server Configuration

OpenVPN's server-side configuration file is located in `/etc/openvpn/server.conf`. You can use the contents below to create a new file at that location on your server:

```

/etc/openvpn/server.conf
1 dev tun
2 persist-key
3 persist-tun
4 topology subnet
5 port 1194
6 proto udp
7 keepalive 10 120
8
9 # Location of certificate authority's cert.
10 ca /etc/openvpn/server/ca.crt
11
12 # Location of VPN server's TLS cert.
13 cert /etc/openvpn/server/server.crt
14
15 # Location of server's TLS key
16 key /etc/openvpn/server/server.key
17
18 # Location of DH parameter file.
19 dh /etc/openvpn/server/dh4096.pem
20
21 # The VPN's address block starts here.
22 server 10.89.0.0 255.255.255.0
23
24 explicit-exit-notify 1
25
26 # Drop root privileges and switch to the 'ovpn' user after startup.
27 user ovpn
```

```

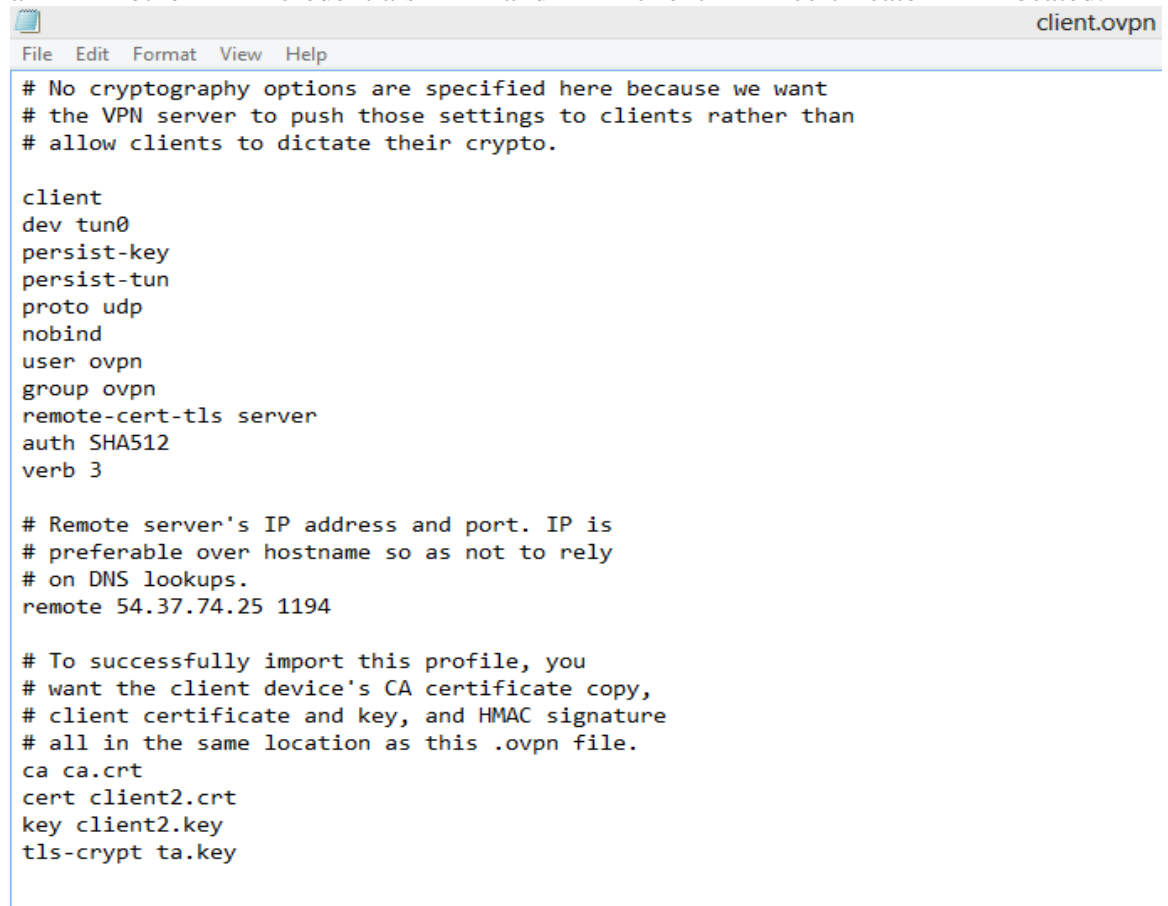
28
29 # OpenVPN process is exclusive member of ovpn group,
30 group ovpn
31
32 # Cryptography options. We force these onto clients by
33 # setting them here and not in client.ovpn. See
34 # 'openvpn --show-tls', 'openvpn --show-ciphers' and
35 # 'openvpn --show-digests' for all supported options.
36 tls-crypt /etc/openvpn/server/ta.key
37 auth SHA512 # This needs to be in client.ovpn too though.
38 tls-version-min 1.2
39 tls-cipher TLS-DHE-RSA-WITH-AES-256-GCM-SHA384:TLS-DHE-RSA-WITH-AES-256-CBC-SHA256
40 ncp-ciphers AES-256-GCM:AES-256-CBC
41
42 # Logging options.
43 ifconfig-pool-persist ip.txt
44 status openvpn-status.log
45 log /var/log/openvpn.log
46 verb 3

```

## Step 2: Client Configuration File:

The client-side OpenVPN's configuration file is Client.ovpn.

When you import an OpenVPN profile ovpn file must be in the same directory where all other credentials and client certificate located.



```

client.ovpn
File Edit Format View Help
# No cryptography options are specified here because we want
# the VPN server to push those settings to clients rather than
# allow clients to dictate their crypto.

client
dev tun0
persist-key
persist-tun
proto udp
nobind
user ovpn
group ovpn
remote-cert-tls server
auth SHA512
verb 3

# Remote server's IP address and port. IP is
# preferable over hostname so as not to rely
# on DNS lookups.
remote 54.37.74.25 1194

# To successfully import this profile, you
# want the client device's CA certificate copy,
# client certificate and key, and HMAC signature
# all in the same location as this .ovpn file.
ca ca.crt
cert client2.crt
key client2.key
tls-crypt ta.key

```

## CHAPTER 6

### IMPLEMENTATION

#### 6.1 VPN GUI (Graphical User Interface):

The GUI is an interface for a user which allows them to have an interaction with the electronic/computer devices or machines via graphical or photographic icons and visual indicators like visual cues, rather than using text-based UI, text navigation or command labels. In response to apparently steep learning curve due to the unintelligible command-line interfaces, GUIs were introduced.

To design the graphic composition and sequential behavior of any GUI is an essential part of programming any software and application when it comes to making it user friendly. The goal of it is to improve the good organization and user-friendliness for the fundamental logical interface any particular program. To guarantee that the programming language for the visuals in the design is matching to the tasks, many methods of user-oriented design are employed. A GUI uses a mixture of technologies and techniques to deliver a platform that is user friendly, for the jobs of collecting and generating data.

The GUI shown below will be interfaced with the OpenVPN client, the user will be able to select any of the servers from Select Server option, and on clicking 'Connect' the OpenVPN client will read the client.ovpn file and start the authentication process with the intermediate server, which will further connect to the remote server.

Three more buttons (View System Information, IP Leak Test and DNS Leak Test) are added for the user to test whether their anonymity on the internet is intact. The View System Information button will provide information on the client's machine and the buttons IP Leak Test and DNS Leak Test will direct you to online IP leak and DNS leak tests where the client will be shown their public IP and location and the status of DNS leakage.

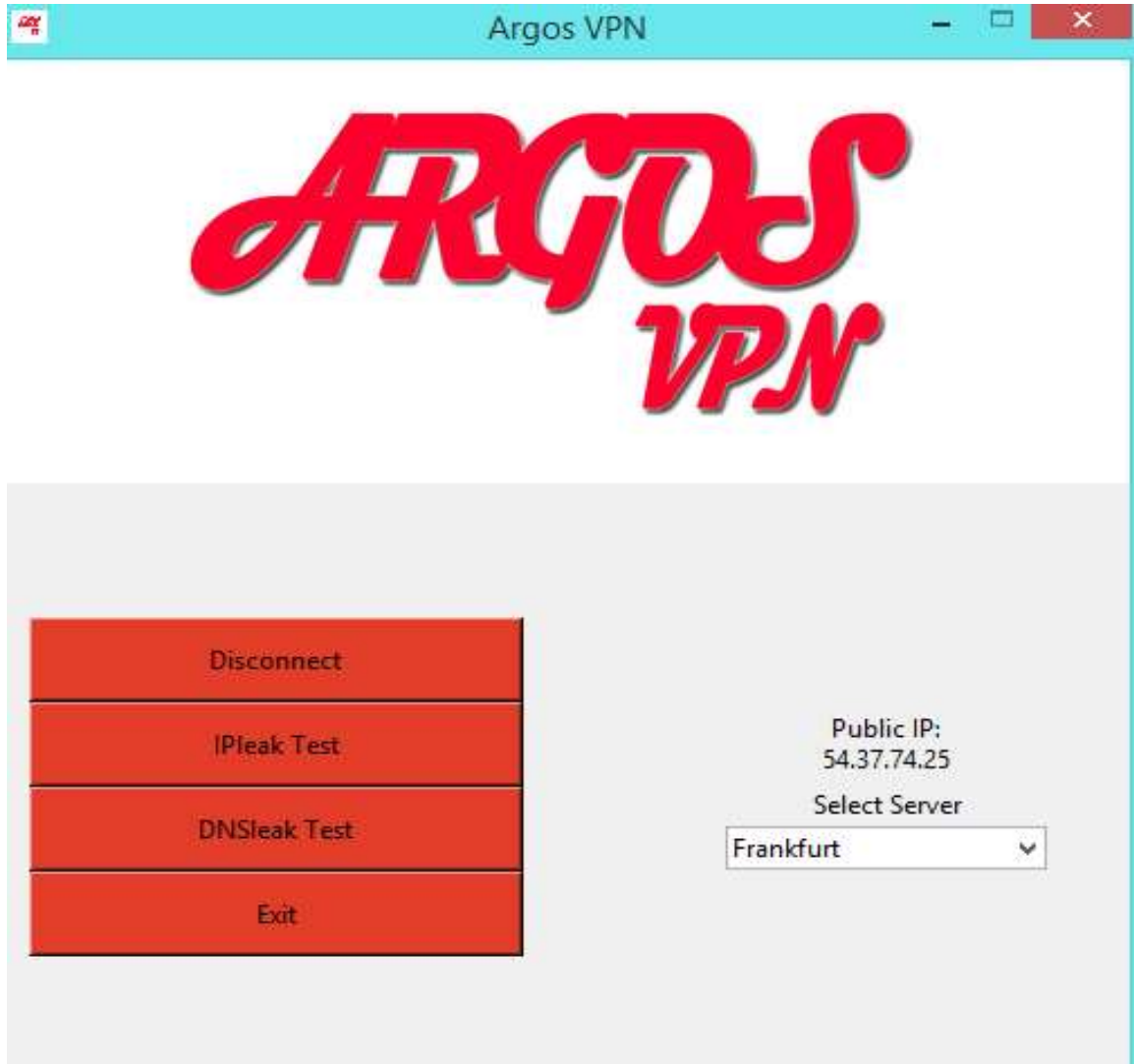


Figure 11: ArgosVPN GUI

## CHAPTER 7

### RESULTS

After integrating GUI with the OpenVPN client, the user can now easily connect to the remote server on a single click, after clicking connect, using the client configuration file to identify the client and to choose a server from a number of servers, the OpenVPN client will connect to any one of the available servers, during the establishment of connection between remote server and the client, the client will first connect to an intermediate server (raspberry pi in our case) that will provide another layer of encryption to client's data and replace its credentials with client's credentials to make the service owners and server owners unable to find out the identity of the client requesting VPN connection.



After the authentication with the intermediate server, the intermediate server will now pose as a client and then establish a connection with the remote server, now OpenVPN will choose the server with the least number of clients connected to it.

On the client's end, while connecting, depending on the traffic on each of the servers, the OpenVPN client will establish tun/tap connection with the server for tunneling traffic, the tun interface at the server creates a network that assigns a private IP for each client connected to it, hence an OpenVPN server is able to connect to multiple clients at a time.

Now that the OpenVPN client and OpenVPN server connection is established, the user of OpenVPN client can browse freely, bypassing the ISP or any other entity eavesdropping on your traffic.

The following tests will now be performed on the OpenVPN server to check if the OpenVPN configuration installed can compromise user anonymity.

IP leak test

Memory Forensics test

### 7.1 Comparison b/w Top 10 Free VPNs with ArgosVPN:

<i>Sr.</i>	<i>VPN Name</i>	<i>DNS Leak</i>	<i>IP Leak</i>	<i>WebRTC leak</i>	<i>Speed</i>	<i>Encryption</i>	<i>Tracking</i>
<b>0</b>	<b>ArgosVPN</b>	✗	✗	✗	<b>High</b>	✓	✗
<b>1</b>	Super VPN	✓	✗	✓	Slow	✗	✓
<b>2</b>	Betternet	✓	✗	✓	Slow	✓	✗
<b>3</b>	Cross VPN	✓	✓	✗	Normal	✗	✗
<b>4</b>	Hide.me	✗	✗	✗	High	✗	✗
<b>5</b>	Flash Free VPN	✓	✗	✓	Slow	✓	✓
<b>6</b>	Hotspot Shield VPN	✗	✓	✗	Normal	✓	✓
<b>7</b>	SurfEasy	✗	✗	✓	Normal	✓	✓
<b>8</b>	VPNGate	✓	✗	✗	Slow	✗	✗
<b>9</b>	Windscribe	✗	✗	✗	Normal	✓	✓
<b>10</b>	Tunnelbear	✗	✗	✓	Normal	✓	✗

## 7.2 IP Leak Test:

**Your IP addresses**

54.37.74.25

Germany

● IPv6 test not reachable. (error)

No forwarded IP detected. If you are using a proxy, it's a transparent proxy.

Browser default: ● IPv4 (228 ms)

Fallback: ● Fail (error)

### Your IP addresses - WebRTC detection

172.27.232.22

Private-Use - [RFC1918]  
IANA Private or Special Address

If you are now connected to a VPN and you see your ISP IP, then your system is [leaking WebRTC requests](#)

### DNS Address - 1 server

54.37.74.25

Germany

No forwarded IP detected. If you are using a proxy, it's a transparent proxy.

## 7.3 WebRTC leak Test:

### WebRTC Leak Test

WebRTC Support Detection :

RTCPeerConnection	✓ True
RTCDataChannel	✓ True
ORTC (Microsoft Edge)	✗ False

IP Address Detection :

Local IP Address	172.27.232.22
Public IP Address	54.37.74.25
IPv6 Address	n/a

WebRTC Media Devices :

Device Enumeration	✓ True
Has Microphone	✓ True
Has Camera	✓ True
Audio-Capture Permissions	?
Video-Capture Permissions	?
Media Devices	kind: audioinput deviceId: default label: n/a

#### 7.4 DNS leak Test:

**My IP:** 54.37.74.25 [Whois](#) **Your anonymity: 90%** Minor remarks regarding your anonymity and security

[Hide IP](#)

Location: Germany (DE), N/A

ISP: OVH Hosting

Hostname: 25.ip-54-37-74.eu

OS: Win8.1

Browser: Chrome 67.0

DNS:

Proxy: No

Anonymizer: No

Blacklist: No

#### 7.5 Internet Speed Test:

# Speed Results

DOWNLOAD SPEED ⓘ 14.0 Mbps

UPLOAD SPEED ⓘ 1.5 Mbps

Latency 295 ms

Protocol IPv4

Host

[Learn about the things that may affect your test results.](#)

## FUTURE WORK

### Fast Multi-Node Networks:

Just like Tor network, using the same technique we can create our own VPNs with more than one layer of encryption or commercialize our own customized multi-layered VPNs. The more the number of servers acting as nodes for a connection, the resilient the privacy.

On the contrary, one might think that by increasing the number of nodes might affect the speed of the network just like in the case of Tor vs VPN, where Tor actually has a relatively slower connection as compared to a VPN because the traffic is routed through three different nodes (servers) globally and then forwarded to the destination server.

This problem can be neutralized by using squid proxy servers integrated with your IPsec or OpenVPN implementation. On the server side squid acts as a proxy server, whenever a client makes a request to visit a web site, squid retrieves the website and then shows it to the client. The client never actually sends a request to the internet as the squid proxy server requests all the sites on the behalf of the client.

As a result, squid speeds up the internet connection on the internal network, make web server pages faster, provides protection to the internal network while you surf the internet, and provides the access only to the authorized users.

### VPNs bypassing port-blocking and DPI firewalls:

In many Networks or LANs where we have port blocking firewalls with DPI (Deep Packet Inspection) systems installed to monitor your personal information and internet traffic, you can use an http tunnel to prevent your traffic to be monitored.

In a port blocking firewall network, the administrators have usually blocked all the ports where a VPN traffic can communicate, and sometimes the administrators only let http/https traffic to go through. To make their firewall even tougher to bypass, they even install a DPI system in it to prevent VPN tunnel requests through port 80 and port 443.

In order to bypass such systems, you can encrypt your VPN traffic in another plain TLS/SSL tunnel which could be http/https or any other protocol allowed in your network which will get you past DPI networks. This can be done by configuring tls-crypt, tunnel or obsfproxy with your VPN implementation.

**Conclusion:**

A VPN is meant to provide you security, anonymity and privacy, and many VPN companies claim to provide you complete safety against the adversaries that can potentially harm your privacy. Here we have completely analyzed those claims by VPN companies, and provided a better solution and alternative to those VPNs guaranteeing you security but are nothing more than just anti-censorship tools.

By running different pentesting and memory forensics tests we were able to see that some VPNs don't encrypt your traffic and make the rest of the world see who does that traffic belong to, some take your credentials without you knowing, and some VPNs run on systems full of backdoors that can be easily exploited by hackers.

The solution we have developed uses an intermediate approach between the way a Tor browser works and a VPN. A user connects to a server and the same server acts as a client to further connect to a remote server selected by the client. In this way, the client credentials are safe as the credentials of our intermediate servers are used instead, by adding an intermediate server we have doubled the encryption layer which means it has twice as strong the encryption as a regular VPN, hence guaranteeing you complete security, anonymity and privacy.

## *Glossary*

**Anonymity:** The act of being anonymous. (See anonymous)

**Anonymous:** Unidentified by any name. In networking terms, it means having no specific reference to such as an IP address or an alias etc.

**Authentication:** A process by which the distinctiveness of a user who wishes to access a system is verified.

**DNS:** The DNS refers to Domain Name System. It is a hierarchical distributed naming system for computers or any other services or resource connected to the internet or a private network.

**DNS Hijacking:** DNS hijacking, also known as DNS redirection is the act of overturning the resolution of DNS queries. A malicious activity subverts a computer's TCP/IP settings so that the computer points at a rogue DNS server. This invalidates the default DNS settings.

**Encapsulation:** Encapsulation refers to the act of wrapping one object within another and so it is used for data hiding. In networking, it is the packaging of one data structure upon another so that the first data structure becomes hidden. For example, in the TCP/IP stack, the application layer encapsulates the transport layer.

**IP:** IP also known as Internet Protocol, is the protocol by which data is transmitted among computers on the internet.

**IPv6:** Internet Protocol version 6 (IPv6) is the most recent version of the Internet Protocol (see Internet Protocol). IPv6 was developed to deal with the exhaustion of the limited IPv4 addresses. IPv6 uses a 128-bit address, theoretically allowing 2128, or approximately  $3.4 \times 10^{38}$  addresses which is more than  $7.9 \times 10^{28}$  times as many as its predecessor, the IPv4.

**IPv6 leakage:** In IPv6 leakage, a MITM (see man in the middle attack) announces itself as an IPv6 router. A VPN client operating system starts sending all the traffic to the man in the middle because IPv6 is preferred by default settings. The VPN client programs do not manipulate the IPv6 routing table. This results in all IPv6 traffic avoiding the VPN's virtual interface.

**L2TP:** The L2TP Layer two tunneling protocol is an extension of the basic Point to Point protocol. The L2TP is used by an Internet service provider to enable VPN's to operate over the internet. The L2TP protocol is not very secure on its own which is why L2TP is implemented with the IPSec security measure.

**Man in the middle attack:** A man in the middle attack is an attack where the attacker secretly relays communication between two parties. The two parties are unaware of the attacker who may also alter the communication data. The malicious actor impersonates himself as a party of communication to the other one and hence gains access to all communication data.

**PPTP:** PPTP refers to Point to Point Tunneling Protocol. PPTP works by encapsulating packets inside PPP packets, which are in turn encapsulated in Generic Routing Encapsulation packets. These packets are sent destination PPTP server and back over the networking IP layer.

**SSL:** SSL stands for secure socket layer. SS is the standard security technology that establishes an encrypted link between a browser and the web server. SSL ensures the privacy and integrity of the data communicated among the protected channel. Millions of websites use SSL in order to secure their online transactions.

**VPN Tunneling:** VPN tunnel is a mechanism used to import a foreign protocol across a network. Tunneling can be done by encapsulating (see encapsulation) the protocol information and private network data.

## Appendix A:

```
from tkinter import *
from tkinter import ttk
from PIL import ImageTk, Image
import os
import urllib.request
import requests
import socket
import platform
from tkinter import messagebox as mBox
from tkinter import scrolledtext
from tkinter import Spinbox
import webbrowser
import time

root = Tk()
root.title("Argos VPN")
root.resizable(width=0 , height=0)
root.geometry('500x500')
frame = Frame(root)
frame.pack()

Hostname=socket.gethostname()
ipv4=socket.gethostbyname(Hostname)
ipv6=socket.gethostbyaddr(Hostname)
sys=platform.system()
intel=platform.processor()

txt="Connect"

topframe = Frame(root)
topframe.pack( side = TOP )

bottomframe = Frame(root)
bottomframe.pack( side= LEFT , padx=10 )

rightframe = Frame(root)
rightframe.pack( side= RIGHT )

url = 'https://whoer.net/'
url1 = 'https://ipleak.net/'

def OpenUrl():
    webbrowser.open_new(url)

def OpenUrl1():
    webbrowser.open_new(url1)

root.iconbitmap(r'D:\GUI\Argos2.ico' )
img = ImageTk.PhotoImage(Image.open("Argoss.jpg"))
panel = Label(topframe, image = img, height=200 )
panel.pack(side = LEFT)

Actualip= None
try:
    Actualip = requests.get(url='http://whatismyip.akamai.com').text
except:
    Actualip = 'No Internet Connection'
```



```

T="Public IP:"
IP=T+'\n'+ Actualip

def _Connect():
    global txt
    if txt=='Connect':
        time.sleep(2)
        txt='Disconnect'
        connectbutton.config(text=txt)
    else:
        time.sleep(2)
        txt='Connect'
        connectbutton.config(text=txt)

def _quit():
    time.sleep(2)
    root.quit()
    root.destroy()
    exit()

connectbutton = Button(bottomframe, text=txt, width=30 , height=2 ,
bg="#E23D28" , fg="Black" , command=_Connect )
connectbutton.grid(row = 0, column = 0, sticky =W+E+N+S )

IPBox= Button(rightframe, text=IP, width=30 , height=2 , borderwidth=0 )
IPBox.grid(row = 0, column = 3, sticky = W+E+N+S )

Label(rightframe, text="Select Server").grid(row=1, column=3, columnspan=20 ,
sticky=W+E+N+S)
User=StringVar()
UserChosen = ttk.Combobox(rightframe , state="readonly" , width=20, height=5 )
UserChosen['values']= ("Frankfurt","USA")
UserChosen.grid(row=2 , column=3, columnspan=20 , sticky=N+S)
UserChosen.current(0)

exitbox= Button(bottomframe, text="Exit" , width=30 , height=2 , bg="#E23D28" ,
fg="Black", command=_quit)
exitbox.grid(row = 5, column = 0, sticky = W+E+N+S )

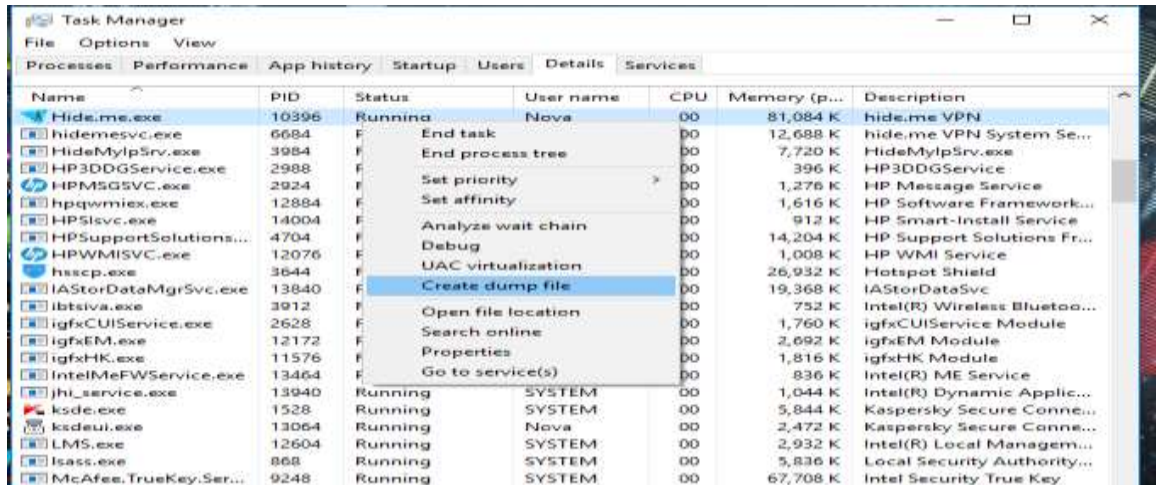
IPleak = Button(bottomframe, text="IPleak Test", width=30 , height=2 ,
bg="#E23D28" , fg="Black", command=OpenUrl1)
IPleak.grid(row = 3, column = 0, sticky = W+E+N+S )

DNSleak = Button(bottomframe, text="DNSleak Test", width=30 , height=2 ,
bg="#E23D28" , fg="Black" , command=OpenUrl)
DNSleak.grid(row = 4, column = 0, sticky = W+E+N+S )
root.mainloop()

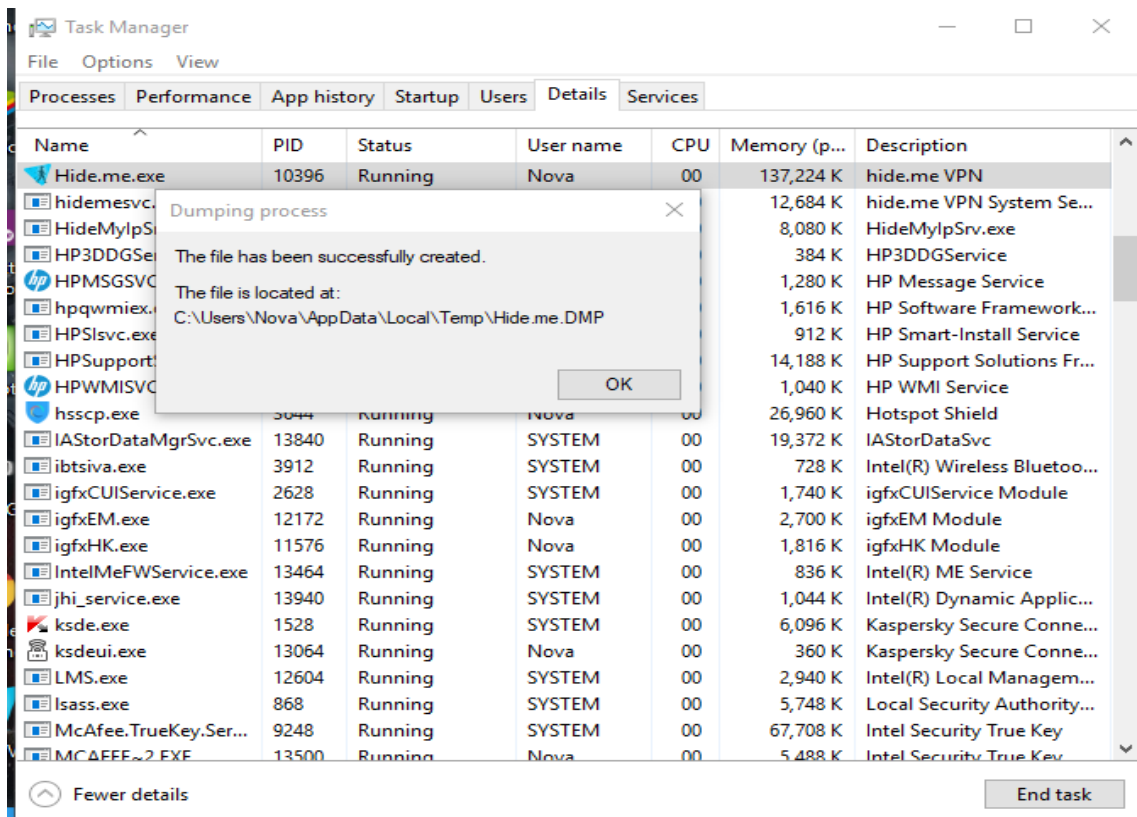
```

## Appendix B:

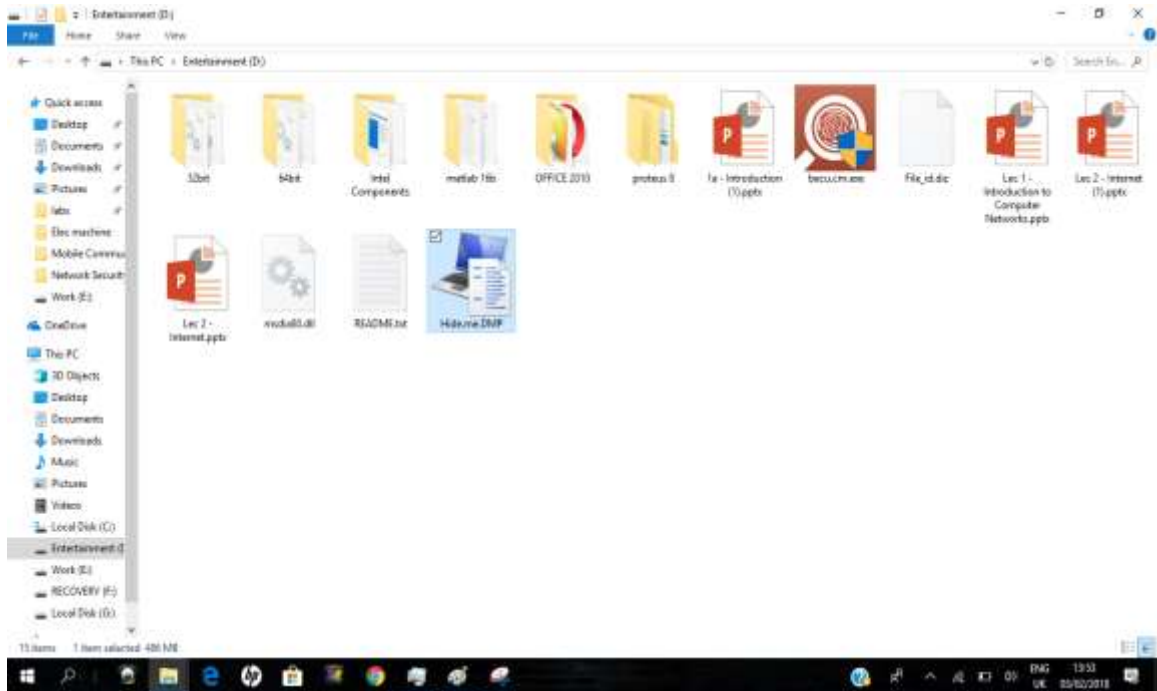
(i)



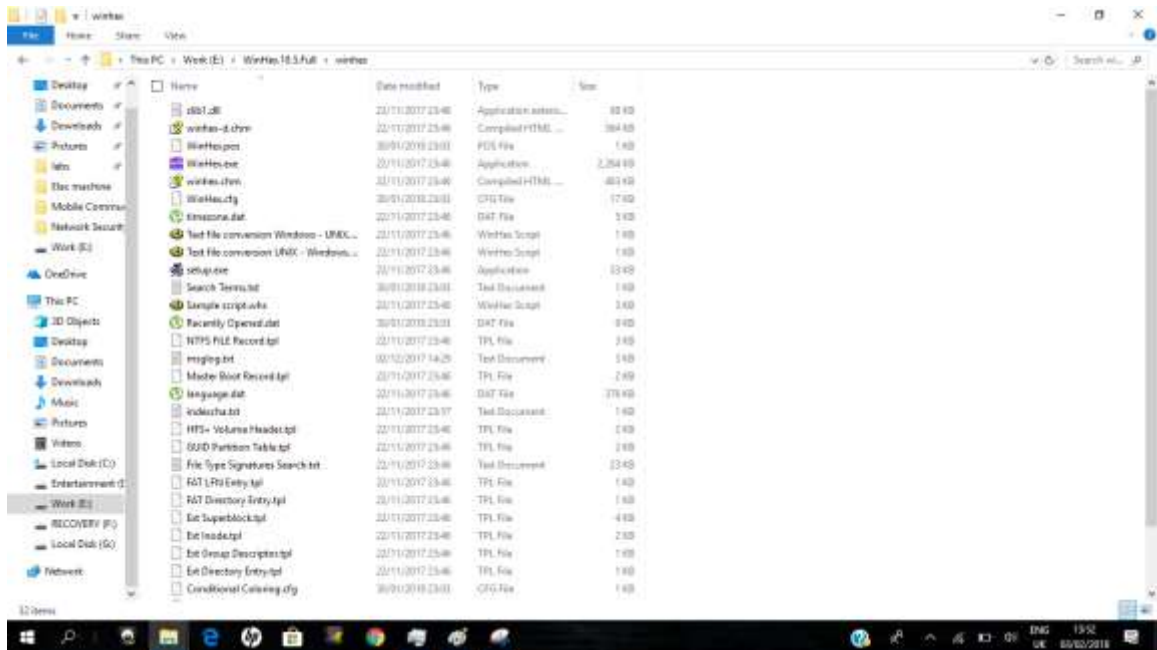
(ii)



(iii)



(iv)





## BIBLIOGRAPHY

1. [https://en.wikipedia.org/wiki/Exploit\\_\(computer\\_security\)](https://en.wikipedia.org/wiki/Exploit_(computer_security))
2. [https://www.top10vpn.com/top10/?v=2&bsid=gen01se1kw001&gclid=EAIaIQobChMI2ufYiPWY1wIV5xXTCh0KbgUGEAAYASAAEgKYP\\_D\\_BwE](https://www.top10vpn.com/top10/?v=2&bsid=gen01se1kw001&gclid=EAIaIQobChMI2ufYiPWY1wIV5xXTCh0KbgUGEAAYASAAEgKYP_D_BwE)
3. <http://searchsecurity.techtarget.com/resources/VPN-security>
4. <https://packetstormsecurity.com/files/36897/pmdump.exe.html>
5. <https://www.elevenpaths.com/labstools/evil-foca/index.html>
6. <http://news.thewindowsclub.com/whitehat-security-releases-free-tool-monitor-dns-hijackings-73716/>
7. <https://vpnpick.com/check-vpn-connection-secure/>
8. [http://www.nta-monitor.com/wiki/index.php/Ike-scan\\_User\\_Guide#IPsec\\_VPN\\_Fingerprinting](http://www.nta-monitor.com/wiki/index.php/Ike-scan_User_Guide#IPsec_VPN_Fingerprinting)
9. <https://www.darknet.org.uk/2008/11/ike-scan-ipsec-vpn-scanner-testing-tool>
10. <https://www.symantec.com/connect/articles/penetration-testing-ipsec-vpns>
11. <http://www.e-fense.com/products.php>
12. <http://www.kitploit.com/2015/09/evil-foca-mitm-dos-dns-hijacking-in.html>
13. <https://tools.kali.org/sniffingspoofing/dnschef>
14. <http://resources.infosecinstitute.com/pen-testing-and-hacking-conferences-a-front-line-experience/#gref>
15. [https://technet.microsoft.com/en-us/library/cc731954\(v=ws.10\).aspx](https://technet.microsoft.com/en-us/library/cc731954(v=ws.10).aspx)
16. Ernesto Van der Sar, "HUGE SECURITY FLAW LEAKS VPN USERS' REAL IPADDRESSES" <https://torrentfreak.com/huge-security-flaw-leaks-vpn-users-real-ip-addresses150130/>
17. Vasile C. Perta\*, Marco V. Barbera, Gareth Tyson, Hamed Haddadi, and Alessandro Mei, "A Glance through the VPN Looking Glass: IPv6 Leakage and DNS Hijacking in Commercial VPN clients", 2015
18. Steve Pitts, "VPN Aggressive Mode Pre-shared Key Brute Force Attack",
19. GIAC practical repository, SANS Institute
20. <https://www.giac.org/paper/gcih/541/vpn-aggressive-mode-pre-shared-keybrute-force-attack/104625>