Analysis of the Security Vulnerabilities in Virtual Private Networks

LITERATURE REVIEW

Atiqa Zafar & Zaryab Khan
School of Electrical Engineering and Computer Sciences,
NUST Islamabad, PK

ABSTRACT

With the rapid increase in the usage of VPNs by both individuals and organizations to transmit sensitive information over an insecure public network, VPNs have become attractive target to hackers. VPN traffic is often invisible to IDS monitoring. This means the attacker can intrude without being picked up by the IDS. VPNs are not the impassable, protected systems that we believe them to be and are vulnerable to attacks such as IPv6 leakage, DNS hijacking, Offline Password Cracking, Man-in-the-Middle Attacks and Malware Infections etc. By VPN hijacking, the attacker can not only access the sensitive data being transmitted but can also get an access to the internal network resources. This can lead to massive security concerns.

In our research, we wish to investigate such potential security risks by studying the core VPN technologies. The work shall be extended by discussing a range of best practices and countermeasure techniques to deal with these vulnerabilities when implementing a virtual private network. The purpose of writing this Literature Review paper is to convey to our readers what knowledge and ideas have been established on a topic, and what their strengths and weaknesses are.

CONTENTS

ABSTRACT 1
1. Introduction 3
2. Survey of the core technologies used by VPN service 4
2.1. Tunneling 4
2.2. Point to Point Tunneling Protocol (PPTP) 4
2.2.1 Tunneling in PPTP 6
2.2.3. PPTP Data Channel 6
2.3. Layer Two Tunneling Protocol (L2TP) 7
2.3.1. Double Encapsulation 8
2.3.2. Conclusion 8
2.4. Secure Socket Tunneling Protocol (SSTL) 8
2.4.1. SSTP - an extension of VPN 9
2.4.2. How SSTP based VPN connection works in seven steps 9
2.4.3. Conclusion 10
3. Potential Security Risks 11
3.1. Compromise of User Anonymity 12
3.2. Username Enumeration Vulnerabilities 15
3.3. Offline Password Cracking 16
3.4. Impersonating Client & VPN Hijacking 17
3.5. MitM Attacks and Traffic Leakage using IPv6 18
3.6. Routing Table Attacks 22
3.7. DNS Hijacking Attacks 22
4. Already-Proposed Defenses 25
4.1. Defense against IPv6 Leakage 25
4.2 . Defense against DNS Hijacking 26
4.3 . Authentication Vulnerabilities 27
4.4. Configuration Issues Management 27
5. Conclusion 28
References 29

1. Introduction

Virtual Private Network (VPN) services have become increasingly popular as a cost-effective way of communicating private information securely over an insecure public network. Both organizations and individuals are rapidly joining the VPN usage bandwagon to securely enter an internal network to access resources, data and communications, or simply as a way to work around regional restrictions on content and anonymity. However, VPNs are not the impenetrable, secure systems that we believe them to be and are wide open to attacks such as IPv6 leakage, DNS hijacking, Offline Password Cracking, Man-in-the-Middle Attacks and Malware Infections etc. Over half of the 16 VPN services, looked into by a group of researchers[19] from Sapienza University of Rome and Queen Mary University of London, used the Point-to-Point Tunneling Protocol with MS-CHAPv2 authentications, which makes them vulnerable to brute force hacks.

The purpose of writing this Literature Review paper is to convey to our readers what knowledge and ideas have been established on a topic, and what their strengths and weaknesses are. This paper is organized as follows. We first survey the tunneling technologies most commonly used by VPN service providers. Many VPNS still rely on outdated technologies such as PPTP (with MS-CHAPv2), that can be easily cracked through brute-force attacks. The work is extended by discussing a range of detection and defense practices to deal with these vulnerabilities when implementing a virtual private network.

2. Survey of the core technologies used by VPN service

2.1. Tunneling

A tunnel is a mechanism used to ship a foreign protocol across a network that normally wouldn't support it. Tunneling protocols allow you to use, for example, IP to send another protocol in the "data" portion of the IP datagram. Most tunneling protocols operate at layer 4, which means they are implemented as a protocol that replaces something like TCP or UDP.[1] Tunneling is generally done by encapsulating the private network data and protocol information within the public network transmission units so that the private network protocol information appears to the public network as data. Tunneling allows the use of the Internet, which is a public network, to convey data on behalf of a private network [2]. In tunneling, the data are broken into smaller pieces called packets as they move along the tunnel for transport. As the packets move through the tunnel, they are encrypted and another process called encapsulation occurs. The private network data and the protocol information that goes with it are encapsulated in public network transmission units for sending. The units look like public data, allowing them to be transmitted across the Internet. Encapsulation allows the packets to arrive at their proper destination. At the final destination, de-capsulation and decryption occur. Tunneling is a way for communication to be conducted over a private network but tunneled through a public network. This is particularly useful in a corporate setting and also offers security features such as encryption options. [3] A VPN is established using 3 techniques namely authentication, tunneling and encryption. The authentication and tunneling details are provided below. As far as encryption is concerned, all of the VPN technology listed below use the best encryption techniques such as Blowfish, AES and 3DES [49]. The increased key length ensures that decryption by malicious users is not possible.

2.2. Point to Point Tunneling Protocol (PPTP)

PPTP [5] (Point to Point Tunneling Protocol) is an extension of the PPP protocol. This protocol is defined as per RFC 1171. Organizations use PPTP to securely transmit data across a VPN on the internet. This is done by embedding its own network protocol

within the TCP/IP packets transmitted over the Internet. This process is referred to as tunneling or encapsulation. PPTP in the simplest form works by encapsulating packets inside PPP packets, which are in turn encapsulated in Generic Routing Encapsulation packets. These packets are sent over the networking IP layer to the destination PPTP server and back. The structure of a PPTP packet is shown in Figure 1.1.

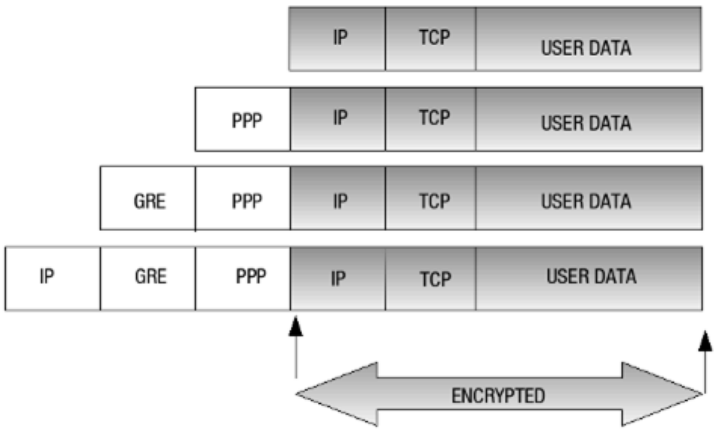

Figure 1.1 structure of a PPTP packet

The PPTP protocol provides the following key security measures:

- Authentication: VPN client's identity is checked by the server and the authentication is restricted to authorized users only using the MS-CHAP and MS-CHAPv2. Audit and accounting capabilities may be used to monitor who accessed which information and when. It must be noted however that MS-CHAPv2 has been proven to be easily cracked [25] and hence making the authentication not so reliable.
- Tunneling
- Encryption: The data is encrypted to make sure malicious users are not able to read the data. Blowfish or AES [49] may be used as encryption techniques
- Compression: Only the necessary amount of information to transmit the data is used

PPTP VPN servers use 2 authentication protocols namely PAP and CHAP:

- PAP: The Password Authentication Protocol is used to authenticate a user with a Network Access Server (NAS). PAP however is insecure as it sends user names and passwords over the network in clear text.
- CHAP: Challenge Handshake Authentication Protocol functions as follows:

I. Server sends a challenge to the requesting client.
II. Client uses this challenge and the password to calculate a response. This response is then sent to the server.
III. Server checks the provided response against the response it expected. If they match, authentication is successful otherwise the connection is terminated.

2.2.1 Tunneling in PPTP

IP tunneling mechanism is used where the packet formats and the addressing used by the VPN might not be the same as which is used to route the tunneled packet across the Internet. Hence the GRE protocol is used. Two hosts are involved in the deployment of PPTP; a PPTP Client who has access to the internet and a PPTP Server

Encryption and network security is implemented by the GRE.

1. Encryption: PPTP supports PPP-based data encryption mechanisms. These may include popular techniques such as Blowfish, 2fish and 3DES [49].
2. Compression: Only the necessary amount of information to transmit the data is used hence reducing bandwidth and increasing transfer speed. PPTP uses the Compression Control Protocol (CCP) used by the PPP protocol.

Two hosts are involved in the deployment of PPTP:

- A PPTP Client who has access to the internet.
- A PPTP Server

The client and server are connected via the control channel and the data channel

2.2.2. PPTP Control Channel

The control channel kicks off the communication between the client and the server. A TCP connection is established on the port 1723. This control channel negotiates tunnel parameters. These parameters include details about the compression algorithm and encryption mechanisms. The data channel is managed, established and released by the control channel.

2.2.3. PPTP Data Channel

The PPTP data channel is negotiated to secure the data transfer once the control channel is expired. The GRE protocol is used to encapsulate the PPP packets for secure transfer to the server. Once the packets are with the server, it verifies and unwraps these packets before destination host in the LAN receives them.

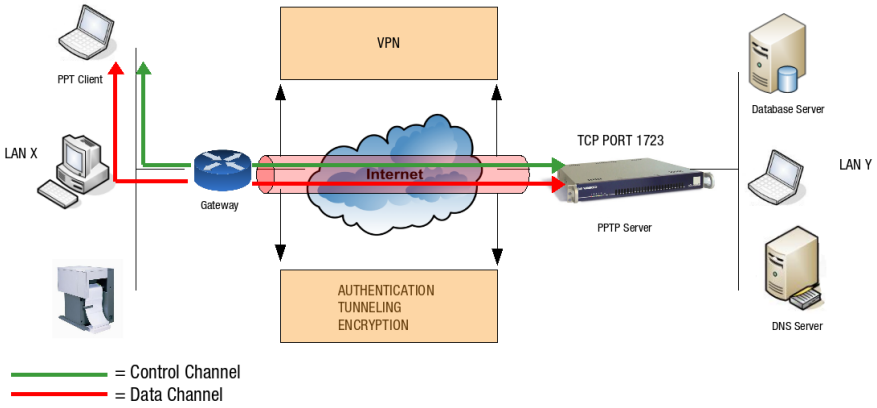

Figure 2.1 A PPP interface with its own IP address is assigned to the server and client once the PPTP VPN is given up.

A strict Firewall Policy should be enforced on the client and server as all network traffic not destined for the local network is routed through the PPP interface, until the PPTP connection is terminated. Only PPTP traffic is routed through the server interface. Depending on the network address assigned to the client's PPP interface, different routing scenarios may apply.

2.3. Layer Two Tunneling Protocol (L2TP)

The L2TP [4] is an extension of the basic PPP. In PPP data packets will be vulnerable during transmission from the client to the server and vice versa. Each data packet being transmitted will be encapsulated by an L2TP header upon implementation of the L2TP. The server will de-multiplex the L2TP packets upon receiving the data. This might seem to be an extra hassle and a waste of precious time but the security improvements made are no small measure.

The L2TP comprises of three parts:

I. Data encapsulation
II. Link Control Protocol for network peer authentication
III. Network Control Protocol, for management of the formation of network layers over the Link Control Protocol after establishment

The L2TP protocol is not very secure on its own which is why L2TP is implemented with the IPSec security measure.

2.3.1. Double Encapsulation

Double encapsulation is one of the main reasons why L2TP is given preference over PPTP. Just like PPTP the first encapsulation occurs and then on top of that a second encapsulation occurs with the IPSec headers to provide security. Encryption keys for the encapsulated data rely heavily on Data Encryption Standards such as AES, DES and blowfish [49] if not on Triple Data Encryption Standards.

2.3.2. Conclusion

The L2TP VPN is an upgraded version of the PPTP protocol. Because of the extensive compatibility it allows and the added security measures, L2TP is easily preferred over PPTP. LT2P is suitable for business use as well as entertainment purposes.

2.4. Secure Socket Tunneling Protocol (SSTL)

Secure Socket Layer uses asymmetric cryptography. Through SSL a secure connection between a server and a client is created. SSL VPN additionally allows users to establish secure remote-access virtually from any internet connection unlike with traditional VPN. Unstable connectivity is no longer an issue. With SSL VPN an entire session is secured, unlike the case with only SSL.

Secure socket tunneling protocol is an application-layer protocol. It employs synchronous communication between 2 programs. It allows many application endpoints over one network connection, between peer nodes, thereby enabling efficient usage of the communication resources that are available to that network.

SSTP protocol is based on SSL and uses TCP Port 443 for transmitting SSTP traffic. Despite being closely related a direct comparison cannot be made between SSL and SSTP as SSTP unlike SSL is only a tunneling protocol. Some of the reasons why SSTP may be preferred over IPSec are the following:

- Authentication is weak in IPSec,
- User clients are required,
- There is no regularity between the coding quality from vendor to vendor
- By default non-IP protocols are not supported,
- IPSec is likely to present problems for remote users attempting to connect from a location that has a limited number of IP addresses.

The default connection protocol is TCP/IP and no static IPs is required in SSL VPN. The client is also unnecessary in most cases.

2.4.1. SSTP - an extension of VPN

The lack of capability of VPN brought the rise of the SSTP. The main drawback of VPN is unstable connectivity due to insufficient coverage areas. SSTP solves the problem by increasing the coverage area ubiquitously. A connection is established over secure HTTPS which allows clients to securely access networks behind NAT routers, firewalls and proxies, without being concerned about port blocking issues. The SSTP VPN tunnel functions over Secure-HTTP hence solving the typical problem faced by L2TP and PPTP. SSTP doesn't require retraining issues due to unchanged end-user VPN controls. The SSTP also has full compatibility with the IPv6. Furthermore it has integration with into MS RRAS client and server, with two factor authentication capabilities. The SSTP can also be controlled and managed using application layer firewalls. Another added advantage of the SSTP is that it is a full network VPN solution and not just an application tunnel for one application. The SSTP is application independent and there is no need to buy expensive and hard to configure firewalls.

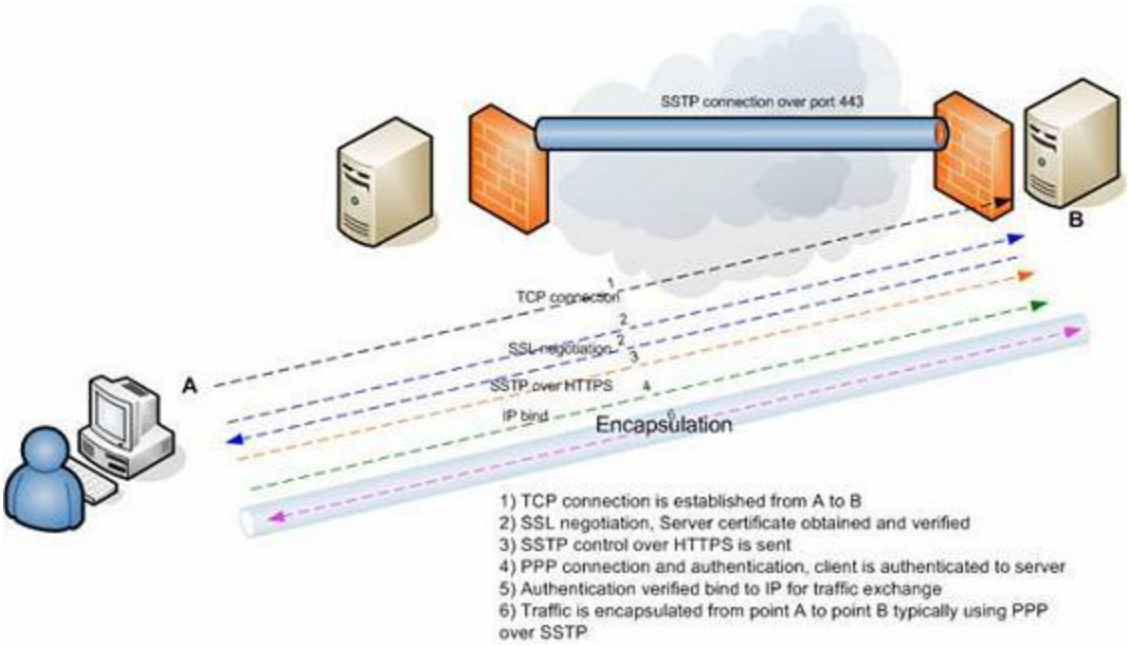

Figure 2.2. The SSTP connection mechanism

2.4.2. How SSTP based VPN connection works in seven steps

1. The SSTP client needs internet connectivity. Once this internet connectivity is verified by the protocol, a TCP connection is established to the server on port 443.

2. SSL negotiation now takes place on top of the already established TCP connection whereby the server certificate is validated. If the certificate is valid, the connection is established, if not the connection is torn down.
3. The client sends an HTTPS request on top of the encrypted SSL session to the server.
4. The client now sends SSTP control packets within the HTTPS session. This in turn establishes the SSTP state machine on both sides for control purposes, both sides now initiate the PPP layer communication.
5. PPP negotiation using SSTP over HTTPS now takes place at both ends. The client is now required to authenticate to the server.
6. The session now binds to the IP interface on both sides and an IP address assigned for routing of traffic.
7. Traffic can now traverse the connection being either IP traffic or otherwise.

2.4.3. Conclusion

SSTP is a great addition to the repertoire of the VPN toolkit to enable users to remotely and securely connect to private networks. The pros of the technology stated above prove that it is a stable and a uniquely useful product among the different variety of technologies that are used to implement VPN's.

3. Potential Security Risks

Many Virtual Private Network (VPN) service providers have become increasingly popular for providing cost-effective way of communicating private information securely over an insecure public network and for boldly claiming about privacy and anonymity without undergoing any standard evaluations. For instance, the AnchorFree's Hotspot Shield website [7] claims to "Hide your IP and ensure anonymous browsing", "Protect yourself from snoopers at Wi-Fi hotspots, hotels, airports, corporate offices" and ""VPN encrypts all traffic". Similarly, Private Tunnel [8] claims "Preventing anyone from viewing or snooping your data exchange across the Internet" and "Preventing anyone from seeing your public IP address". However, none of these VPN service providers are not the impenetrable, secure systems that they claim to be [9-12]. In this section, we shall be investigating the VPN vulnerabilities and surveying a number of researches that show how these popular providers are wide open to potential security risks.

Appelbaum et al. [18] were the first to give a scientific categorization of configuration issues that ought to debilitate the utilization of VPN services achieving privacy and secrecy on the Internet. In addition to other things, they experimentally watched the issue of IPv6 leakage in certain VPNs that we shall be discussing shortly.

R.Hills[20] mentions a portion of the basic VPN security vulnerabilities that NTA Monitor have found amid the most recent three years (2003-5) while performing VPN security tests. The paper focuses on remote access VPN configurations utilizing the IPsec protocol. A portion of the issues that have been seen, for example, the username identification issue, were new revelations, while others are known constraints of the protocols, which are presented because of poor configuration. The paper demonstrates that VPNs are a long way from the impervious frameworks that numerous individuals trust them to be, and that they can really be the feeble connection in a generally secure framework.

A more recent study [19] to come out of the Sapienza University of Rome and Queen Mary University of London has carried out a thorough investigation of the privacy and anonymity features of fourteen popular commercial VPN services available today on the market by subscribing to the services, downloading their recommended clients on both desktop and mobile systems and conducting experiments during the period

Provider	Countries	Servers	Technology	DNS	IPv6-leak	DNS hijacking
Hide My Ass	62	641	OpenVPN, PPTP	OpenDNS	Y	Y
IPVanish	51	135	OpenVPN	Private	Y	Y
Astrill	49	163	OpenVPN, L2TP, PPTP	Private	Y	N
ExpressVPN	45	71	OpenVPN, L2TP, PPTP	Google DNS, Choopa Geo DNS	Y	Y
StrongVPN	19	354	OpenVPN, PPTP	Private	Y	Y
PureVPN	18	131	OpenVPN, L2TP, PPTP	OpenDNS, Google DNS, Others	Y	Y
TorGuard	17	19	OpenVPN	Google DNS	N	Y
AirVPN	15	58	OpenVPN	Private	Y	Y
PrivateInternetAccess	10	18	OpenVPN, L2TP, PPTP	Choopa Geo DNS	N	Y
VyprVPN	8	42	OpenVPN, L2TP, PPTP	Private (VyprDNS)	N	Y
Tunnelbear	8	8	OpenVPN	Google DNS	Y	Y
proXPN	4	20	OpenVPN, PPTP	Google DNS	Y	Y
Mullvad	4	16	OpenVPN	Private	N	Y
Hotspot Shield Elite	3	10	OpenVPN	Google DNS	Y	Y

Table 3.1 VPN Services subjected to the study [19]

September – December, 2014. The VPN services, listed in the table 3.1, were selected on the basis of the first 20 Google results of "Best VPN" or "Anonymous VPN" search queries and on the basis of their advertised distinctive features that were relevant to the study. These include Mullvad - the only provider mentioning IPv6 leakage protection; Hotspot Shield – claiming Wi-Fi security in untrusted hotspots; etc. A detailed discussion of the potential security flaws found in the VPS through the mentioned studies is as follows.

3.1. Compromise of User Anonymity

User information is not hidden from their VPN service provider who may also retain this information. Many VPN services prompt user to enter personal information, or even a valid mobile number at registration time. Some also retain timestamps, the amount of data transmitted, and the client IP address of each VPN connection. The TorrentFreak [13] has been interviewing the leading VPN providers about their logging practices for the four consecutive years (2012-15) [14-17] to figure out how anonymous VPNs truly are. It additionally got some information about other privacy sensitive policies. Hence, VPN service Providers must be blindly trusted to not be malicious, and to not disclose the user traffic to third parties. A number of cases have occurred where the VPN service providers have disclosed the user traffic. For instance, Israel-based Hola - a popular VPN provider used by roughly 46 million

users worldwide to make tracking their internet activity more difficult to track, was caught for selling the bandwidth of individuals using the free version of the software to cover operational costs [9-10].

R. Hills [20] sheds light on the other dimension of user anonymity. Many VPN client programs offer to store the user authentication credentials on the client machine. For some, it is even the default setting. This introduces a security loophole if these credentials are not well protected. Some client programs store the username unencrypted in a file or the registry with weak registry or file permissions for stored credentials and anyone with an access to the machine can obtain it. If the VPN is using IKE Aggressive Mode, then knowing the username can lead to an offline cracking attack against the password. Figure 3.2 shows an example of the username royhills@hotmail.com stored in the registry.

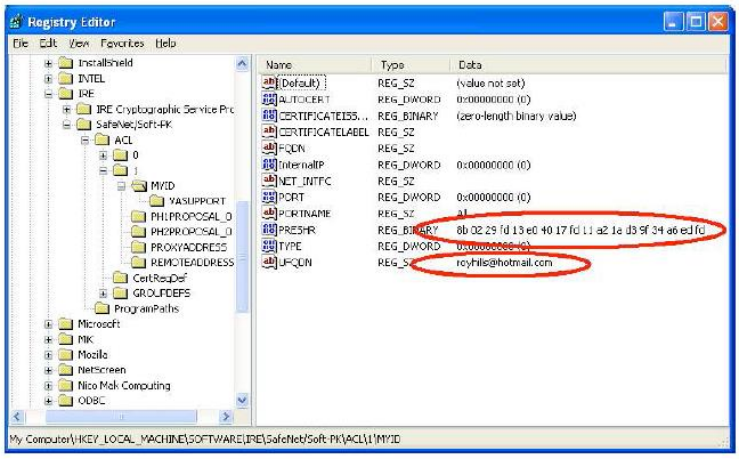

Figure 3.2. Username and obfuscated password stored in the Registry.

Even the password is stored on the client machine in a scrambled form that is often referred to as encryption. However, there is no unique key needed to decrypt it and knowing the encoding algorithm can lead to guessing the actual value. Figure 3.2 shows an example of an obfuscated password stored in the registry with the corresponding clear-text password is W0ntGu355Th15. Many client programs also decrypt this obfuscated version of the password in file/registry when they start up, and store a plaintext version of the password in memory. Thus, starting the VPN client and dumping the process memory with a tool such as pmdump, or crashing the computer to get a dump of physical memory can leak the password. Figure 3.3 shows a memory dump from a VPN client with the clear-text password W0ntGu355Th15 highlighted.

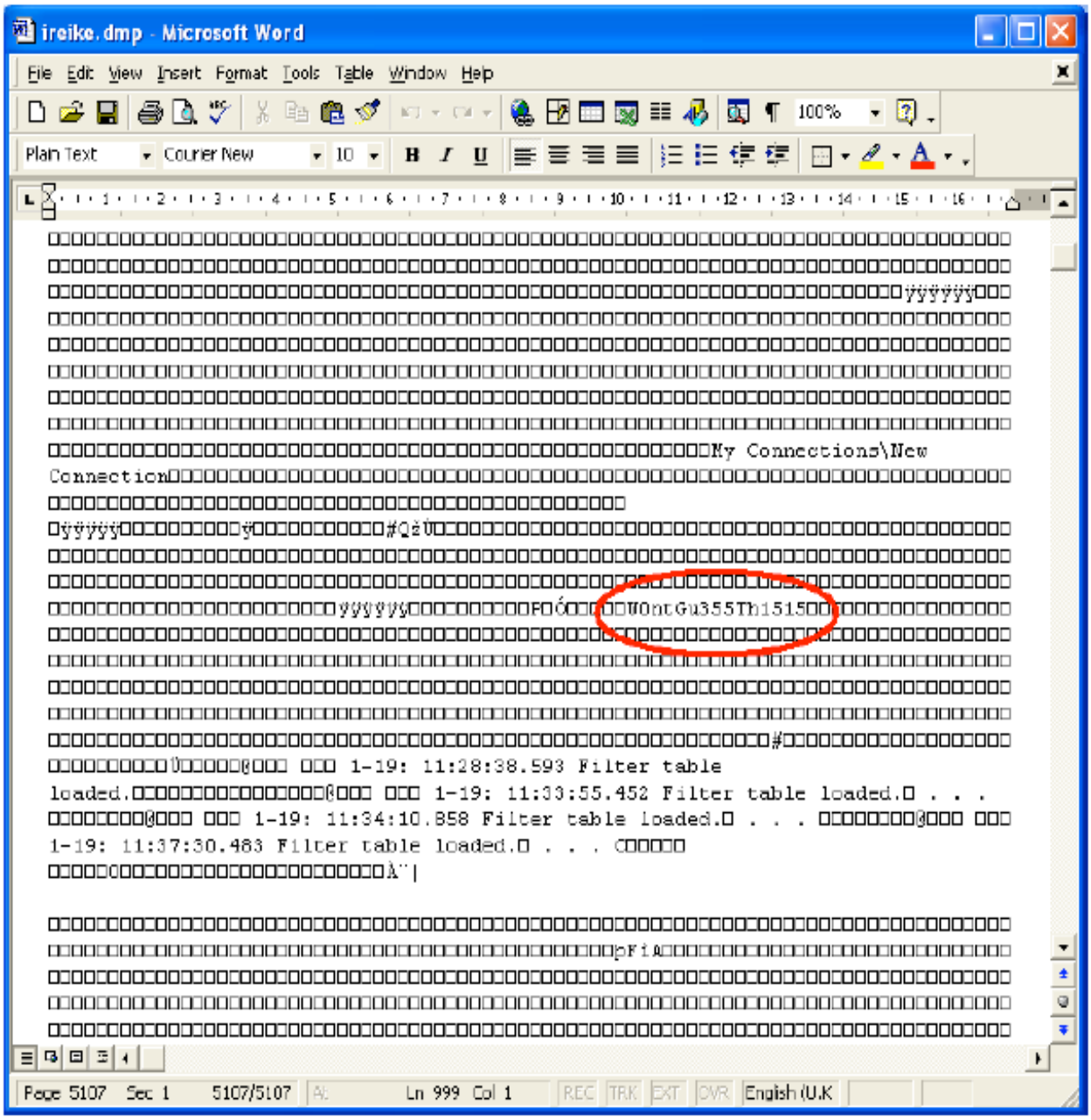

Figure 3.3 shows a memory dump from a VPN client with the clear-text password W0ntGu355Th15 highlighted.

3.2. Username Enumeration Vulnerabilities

The response to an incorrect login attempt should not leak information about which of the authentication credentials (username or password) was incorrect, because this allows an attacker to deduce whether a given username is valid or not, leading to ease in dictionary attacks. Many VPNs use IKE Aggressive Mode with pre-shared key (PSK) authentication to authenticate the username and password. R.Hills [20] points out that many implementations of PSK authentication, failing to abide by this scheme, provide a different response for an invalid username than for a valid one. Figure 3.4 shows the initial packet exchange for aggressive mode PSK authentication. The client sends an IKE packet containing several ISAKMP payloads to the VPN server which responds with a reply IKE packet. Notice 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.

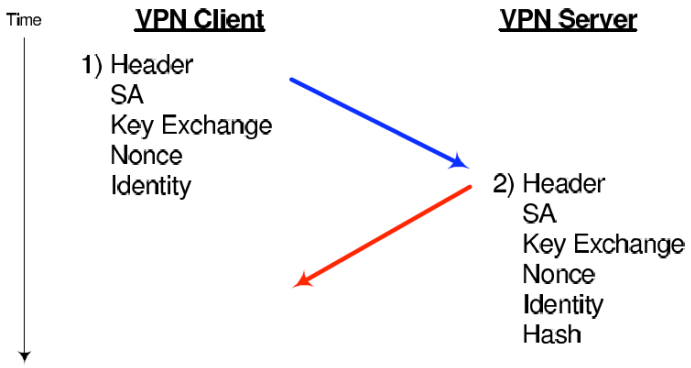

Figure 3.4. Packet Exchange for Aggressive Mode PSK Authentication

Worryingly, some VPN servers only respond to the client if the username is valid, they do not respond at all to invalid usernames; others respond with a notification message if the username is incorrect; and some respond to both valid and invalid usernames, but the hash payload for invalid usernames is calculated using a null password which is easily detectable. Consider the example below:

```
$ ike-scan --aggressive --id=fred 172.16.2.2
Starting ike-scan 1.7 with 1 hosts (http://www.nta-monitor.com/ike-scan/)
172.16.2.2 Aggressive Mode Handshake returned
SA=(Enc=3DES Hash=SHA1 Auth=PSK Group=2:modp1024 LifeType=Seconds
LifeDuration(4)=0x00007080)
KeyExchange(128 bytes)
Nonce(20 bytes)
ID(Type=ID_IPV4_ADDR, Value=172.16.2.2)
Hash(20 bytes)

$ ike-scan --aggressive --id=jim 172.16.2.2
Starting ike-scan 1.7 with 1 hosts (http://www.nta-monitor.com/ike-scan/)
172.16.2.2 Notify message 14 (NO-PROPOSAL-CHOSEN)
```

ike-scan is used to show that the server responds to the valid username "fred" normally but replies with a notification message of NO-PROPOSAL-CHOSEN for an invalid username "jim". This information leakage makes dictionary attacks easy.

3.3. Offline Password Cracking

Given that a valid username has been obtained successfully, an attacker can crack the associated password by using the hash from the VPN Server. RFC 2409 [22] defines the hash payload from the VPN Server as

$$hash_r = prf(skeyid, gx^r | gx^i | cky_r | cky_i | SAi_b | IDir_b)$$

and *skeyid* is defined as:

$$skeyid = prf(psk, Ni_b | Nr_b)$$

where the terms used are:

prf	The pseudo-random HMAC function
gx_r	The responder (VPN Server) public Diffie-Hellman value (in the key exchange payload)
gx_i	The initiator (VPN client) public Diffie-Hellman value (in the key exchange payload)
cky_r	The responder (VPN Server) ISAKMP cookie (in the ISAKMP header)
cky_i	The initiator (VPN client) ISAKMP cookie (in the ISAKMP header)
SAi_b	The body of the initiator (VPN client) SA payload
$IDir_b$	The body of the responder (VPN Server) ID payload
Ni_b	The body of the initiator (VPN client) nonce payload
Nr_b	The body of the responder (VPN Server) nonce payload
psk	The Pre-Shared Key (group password)

R.Hills[20] points out that the first two IKE Packets contain all of these values above except from the pre-shared key in an unencrypted form. The attacker can perform an offline dictionary attack by running a list of candidate passwords (keys) through the hash function and comparing the resulting hash in each case with the hash that the server sent. If the two match, the correct password is found. This attack is very fast: MDCrack [a MD5 bruteforce tool] can achieve 1.5 million keys per second with pure MD5 and a PIII 700. PSK bruteforcing consists of 4 MD5's, and 4 64 byte XORs....but should still be able to achieve 375,000 IKE keys per second. Preliminary tests in C have shown 26,000 keys per second with un-optimized routines. [23] Consider the example below, showing how ike-scan can be used with a valid username “fred” to fetch the PSK Parameters. It further shows how pskcrack is used to crakc the password against these PSK parameters.

```
$ ike-scan --aggressive --id=fred --pskcrack=fred.psk 172.16.2.2
Starting ike-scan 1.7 with 1 hosts (http://www.nta-monitor.com/ike-scan/)
172.16.2.2 Aggressive Mode Handshake returned
SA=(Enc=3DES Hash=SHA1 Auth=PSK Group=2:modp1024 LifeType=Seconds
LifeDuration(4)=0x00007080)
KeyExchange(128 bytes)
Nonce(20 bytes)
ID(Type=ID_IPV4_ADDR, Value=172.16.2.2)
Hash(20 bytes)

$ psk-crack fred.psk
Starting psk-crack in dictionary cracking mode
key "Liverpool" matches SHA1 hash 1f074be2ce5a3128aea49a4f4fb7752f9fe33670
Ending psk-crack: 10615 iterations in 0.052 seconds (204134.62 iterations
/sec)
```

Pitts [21] further describes this VPN Aggressive mode pre-shared key brute force attack in products of several leading vendors of RFC 2409 compliant VPN devices including Cisco and Checkpoint. The authentication protocol, MS-CHAPv2, of PPTP is also affected by serious security vulnerabilities that have been well-known in the community for years. The papers Schneier[24] and Marlinspike[25] provide key-cracking of MS-CHAPv2.

3.4. Impersonating Client & VPN Hijacking

VPN hijacking consists of impersonating the VPN Client and capturing the established VPN connection. Section 3.2 and 3.3 discuss how a valid username and password can be cracked if the VPN is using IKE Aggressive Mode with pre-shared key (PSK) authentication to authenticate the login credentials. Once cracked, we can use the pre-

shared key (PSK) to complete IKE Phase 1 and establish ISAKMP SA (Internet Security Association and Key Management Protocol) with the Server.

R.Hills[20] discusses how this is all that is needed to hijack the VPN server using XAUTH (Extended Authentication). The attacker installs his machine on the Ethernet link that the VPN Client/Server traffic flows over and use ARP spoofing to redirect the traffic. He then sniffs the username and crack the password using the payloads of IKE Packets. When the VPN client connects, he is tricked to establish an ISAKMP SA to the attacker's machine and not the Server. A second ISAKMP SA is established from the attacker's machine to the VPN Server as the username and password are known. The VPN server will issue an XAUTH challenge to the attacker's machine that is passed on to the client. The client responds with the second username, SecureID PIN and passcode to the attacker that he passes onto the Server to fully authenticate itself. Here, the VPN Security is penetrated. The attacker is free to intercept and modify the traffic or drop the connection with the client and proceed to do IKE Phase 2 with the Server and gain full access to the VPN resources.

3.5. MitM Attacks and Traffic Leakage using IPv6

The VPN client programs only manipulate the IPv4 routing table and not the IPv6 routing table, resulting in all IPv6 traffic bypassing the VPN's virtual interface. Alessandro Mei[19] states that the security concerns stems from the nature of IPv4/6 dual stack implementations on common operating systems that have been introduced to smoothly transition between the two protocols (RFC 4213) to let a network and host to simultaneously operate both IPv4 and IPv6. Hence, most OS have IPv6 enabled and prefer it over IPv4 in line with RFC 6724. Some Operating systems such as Windows 7 and 8 even have DHCPv6, the IPv6-version of DHCP, enabled by default. This means that if they haven't already got an IPv6 connection, they will obtain one from any DHCPv6 server running on the local network [35].

A man in the middle can advertise himself as an IPv6 router, and the VPN client OS will start sending all the traffic to him because IPv6 is preferred. The adversary only needs to be on the same local network as you are, such as a public Wi-Fi. The attack is also known as "SLAAC Attack". Waters [36] shows us what the output of ipconfig looks like on the victim host before the IPv6 interface is connected to the network:

```
Ethernet adapter Local Area Connection:
Connection-specific DNS Suffix  . :
Description . . . . . . . . . . . : Atheros AR8131 PCI-E Gigabit Ethernet
Controller (NDIS 6.20)
Physical Address. . . . . . . . . : 00-26-9E-47-4E-0F
```

DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . :
fe80::119c:ea76:23d4:290d%10(Preferred)
IPv4 Address. . . . . : 192.168.0.2(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Lease Obtained. . . . . : 30 March 2011 23:23:08
Lease Expires . . . . . : 31 March 2011 13:55:33
Default Gateway . . . . . : 192.168.0.251
DHCP Server . . . . . : 192.168.0.251
DHCPv6 IAID . . . . . : 285221771
DHCPv6 Client DUID. . . . . : 00-01-00-01-12-52-C9-D5-00-26-9E-47-4E-0F
DNS Servers . . . . . : 192.168.0.251
NetBIOS over Tcpip. . . . . : Enabled

Note the presence of a link-local IPv6 address that shows the host is IPv6-capable. Once the man in the middle advertise himself as an IPv6 router, the VPN client derives a routable IPv6 address for itself and queries DHCPv6 for further configuration. Shortly, ipconfig will output:

Ethernet adapter Local Area Connection:
Connection-specific DNS Suffix . : pwned.by.v6
Description . . . . . : Atheros AR8131 PCI-E Gigabit Ethernet Controller (NDIS 6.20)
Physical Address. . . . . : 00-26-9E-47-4E-0F
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
IPv6 Address. . . . . .
: 2001:6f8:608:fab:119c:ea76:23d4:290d(Preferred)
Temporary IPv6 Address. . . . . .
: 2001:6f8:608:fab:687a:83f:caa7:8f9c(Preferred)
Link-local IPv6 Address . . . . . :
fe80::119c:ea76:23d4:290d%10(Preferred)
IPv4 Address. . . . . : 192.168.0.2(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Lease Obtained. . . . . : 30 March 2011 23:23:08
Lease Expires . . . . . : 31 March 2011 13:55:33
Default Gateway . . . . . : fe80::225:4bff:fefd:9173%10
192.168.0.251
DHCP Server . . . . . : 192.168.0.251
DHCPv6 IAID . . . . . : 285221771

DHCPv6 Client DUID. . . . . : 00-01-00-01-12-52-C9-D5-00-26-9E-47-4E-0F
DNS Servers . . . . . : 2001:6f8:608:ace::c0a8:5802
192.168.0.251
NetBIOS over Tcpip. . . . . : Enabled
Connection-specific DNS Suffix Search List: pwned.by.v6

Notice that the VPN client OS has begun redirecting all the traffic to the IPv6 default gateway which is actually the link local address of the attackers' interface. Other tools such as SuddenSix [38-39] on Linux presented at DEFCON 21 Hacking Conference, 2012 [37], Evil FOCA [40-41] on Windows also presented at DEFCON 21 and THC-IPv6 with fake_router6[42] on Linux can be used to launch Man-In-The-Middle Attacks using IPv6.

The study [19] and other papers [26-28] further reveal that a small leakage of IPv6 traffic can expose the whole user browsing history even on IPv4 only websites. These mention third party "plug-ins" such as ad brokers, trackers, analytics tools, social media plugins, and the Referer HTTP header that discloses the exact URL of the visited page in the fetching of each of the third party objects embedded in it. Even if one of these fetches are leaked through IPv6 traffic leakage outside the VPN tunnel, then the actual user IP can be compromised along with the page URL that the host is accessing. Figure 3.5 presents the results of studying the number of IPv6 third party objects that the Alexa [29] top 1K IPv4-only websites embed. [19]

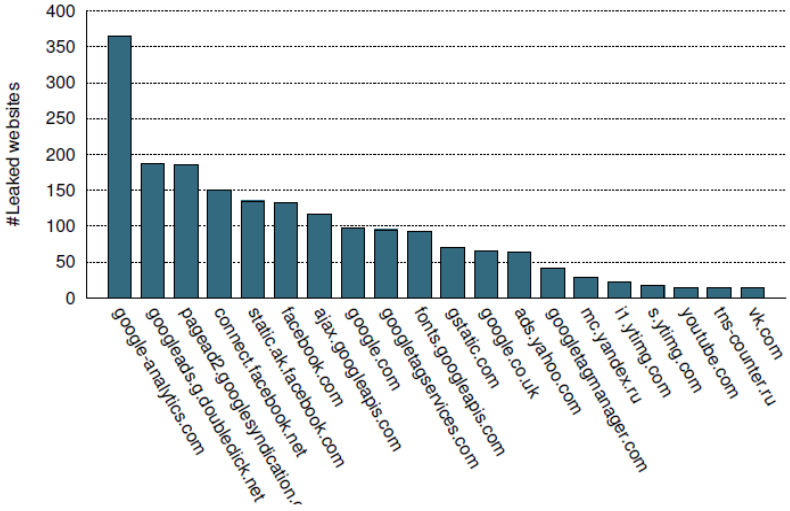

Figure 3.5 92% of the Alexa top 1K IPv4 –only websites embed objects of at least 1 of these third parties.

Alessandro Mei[19] created a testbed using a IPv4/IPv6 dual stack WiFi LAN to connect a number of hosts running various OS: Linux (Ubuntu 14.04), Windows (8.1 Pro), OSX (Mavericks), iOS 7, and Android (JellyBean, KitKat) to wwww.google.com domain. The test results (table 3.1) show that all desktop VPN client programs leak IPv6 Traffic except for Mullvad, VyprVPN and TorGuard (in Advanced Settings). As for mobile OS, Ios completely disables IPv6 during the VPN tunnel lifetime and thereby, all its VPN Services resist IPv6 Leakage. However, all VPN services on Android are vulnerable to the leakage.

The study[19]also measure the criticality of IPv6 Leakage by investigating how exposed the websites, the mobile traffic and peer-to-peer networks are to IPv6 leakage and public detection, while users retain the belief that all their interactions are securely occurring over the tunnel. It assesses the Alexa rankings for well- known sites that are IPv6-enabled and exhibits the rundown of top IPv6 sites against the number of countries that consider them amongst top 500 in the figure 3.6. All connections with these sites would go around the VPN tunnel and quietly happen over the open native interface.

It further shows that all apps in top 100 most popular applications available on Google Playstore indirectly leak information through third party plug-ins such as advertisements. Viennot et al. [30] found that 75% of all Android apps include Google ad libraries (that supports IPv6) which suggests that all these apps are exposed to the leakage. As for p2p, Vyncke[31] cites the number of IPv6 BitTorrent peers per country, using the methodology detailed in M. Defeche[21], showing that some countries have a significant IPv6 presence and would therefore be extremely vulnerable to IPv6 leakage, such as US.

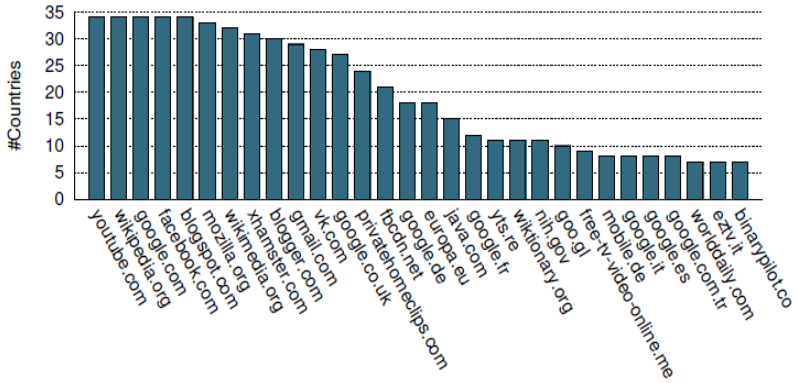

Figure 3.6. Rundown of top IPv6 sites against the number of countries that consider them amongst top 500

3.6. Routing Table Attacks

VPN Client program establishes a tunnel and modifies the host routing table in order to redirect all the traffic towards the virtual network interface created to encrypt and forward the traffic to the VPN remote entry point via the host's active network interface (e.g., Wi-Fi or Ethernet). These steps initiate the tunnel fully to send all user traffic via the VPN in an encrypted form. However, reliance on the correct configuration of the operating system's routing table exposes VPN users to a number of security risks. The crux of the problem is that routing tables are a resource concurrently managed by the operating system, which is unaware of the security needs of the VPN client. After all, to the kernel the active VPN connection is just another virtual network adapter. None of the VPN Clients studied in [19] monitor the routing table to ensure that their initial configuration is not changed. Even small changes to the routing table, either malicious or accidental, results in traffic leakage beyond the VPN tunnel. Consider the Routing Table below provided by Appelbaum et al. [18].

Network Destination	Netmask	Gateway	Interface	Metric
0.0.0.0	0.0.0.0	10.36.13.4	10.36.13.4	1
0.0.0.0	0.0.0.0	192.168.84.1	192.168.84.107	21
10.36.13.4	255.255.255.255	127.0.0.1	127.0.0.1	50
10.255.255.255	255.255.255.255	10.36.13.4	10.36.13.4	50
127.0.0.0	255.0.0.0	127.0.0.1	127.0.0.1	1
192.168.84.0	255.255.255.0	192.168.84.107	192.168.84.107	20
192.168.84.107	255.255.255.255	127.0.0.1	127.0.0.1	20
192.168.84.255	255.255.255.255	192.168.84.107	192.168.84.107	20
208.53.158.59	255.255.255.255	192.168.84.1	192.168.84.107	20
255.255.255.255	255.255.255.255	10.36.13.4	10.36.13.4	1
255.255.255.255	255.255.255.255	192.168.84.107	192.168.84.107	1

It shows how the previous default route (0.0.0.0) is overridden by specifying a lower cost metric for the VPN service. Note that all the existing routing table entries have been retained even though the VPN-installed split tunnel is as narrow as possible, a single IP address 208.53.158.59. The attacker can prompt the client program to connect to the local network, leading to a packet leakage beyond the VPN tunnel.

3.7. DNS Hijacking Attacks

DNS Hijacking or DNS redirection [33][19] is a more concentrated attack to transparently capture all DNS queries, both IPv4 and IPv6, from the VPN Client machine. 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.

Assume that the adversary controls the network's gateway, say the WiFi access point in all these attacks.
In the first case, the attack is simple to carry out, the adversary can simply use DHCP to set the client's DNS server to the one under his control to redirect all DNS queries generated by the host to himself.

In the VPN-Managed configuration, the adversary redirects all DNS queries by modifying the routing table of client [Section 3.4] to make the DNS a local network resource, accessible directly via the LAN rather than through the VPN tunnel. Different Tunneling Protocols modify the client's routing table in different ways.

As cited in the OpenVPN manual [34], the VPN client manipulates the host's routing table by inserting two prefixes: 0/1 and 128/1 rather than deleting the existing default route 0/0 set via DHCP. As these new values are more definite than the default route, all user DNS queries are securely sent through the VPN tunnel interface (usually tun0 or tap0) instead of the host's LAN interface. To hijack the DNS under this configuration, a route injection attack is carried out in which the adversary, with control over the access point (Wi-Fi oruter), reduces the DHCP lease period forcing the victim to periodically re-request new DHCP information. The DHCP renewals allow the adversary to reconfigure the routing table by setting the client's default gateway to a new virtual interface with the IP address of the DNS server used for the VPN. Note that it is easy to note the VPN service provider by passively observing the remote IP of the tunnel. Thence, all DNS queries are redirected to the fake interface on the access point, rather than through the tunnel without the VPN clients detecting the changes.

This attack is ineffectual for PPTP and L2TP tunneling protocols as the client VPN removes or de-prioritizes the existing default route by binding it to the local network interface and sets only one default route 0/0. This allows prevention of any subsequent route to be injected into the routing table by DHCP gateway option as it has a lower priority compared to the default route to the tunnel.
To hijack, the access point (Wi-fi router) assigns the victim an address in a small false subnet that includes the DNS server used by the VPN to bound all the traffic towards the subnet, including that towards the DNS server, to the actual network interface of the victim host (e.g. if the VPN's DNS server were 208.80.112.222, then the victim would be assigned an address in the 208.80.112.0/24 subnet). Thence, this interface gets priority over the default rule imposed by the VPN client.

Table 3.1 lists the experimental results for DNS hijacking against all the VPN clients tested in the study [19], confirming their efficacy. There were, however, some exceptions. First is the Windows 8 resistance to the OpenVPN route injection attack.

Second is the Android use of firewall rules [43] instead of routing table changes to force traffic to be routed through the VPN tunnel. The firewall rules completely cut the device off from the local network, allowing traffic to be only routed through the VPN tunnel, thereby preventing the attack. However, Android versions prior to KitKat are vulnerable to the DNS hijacking attack. Third is the Astrill VPN that is not vulnerable to both versions of DNS hijacking as it sets the same IP address for both, the DNS server and the VPN tunnel gateway, thereby making it impossible for the attacker to trick the client into believing that the DNS resides in the LAN.

4. Proposed Defenses

4.1. 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. Grooten[35] proposes to mitigate this risk by disabling IPv6 traffic on the Client Machine. We can easily disable IPv6 on Windows via the Registry [44], Mac OS, Linux and others.

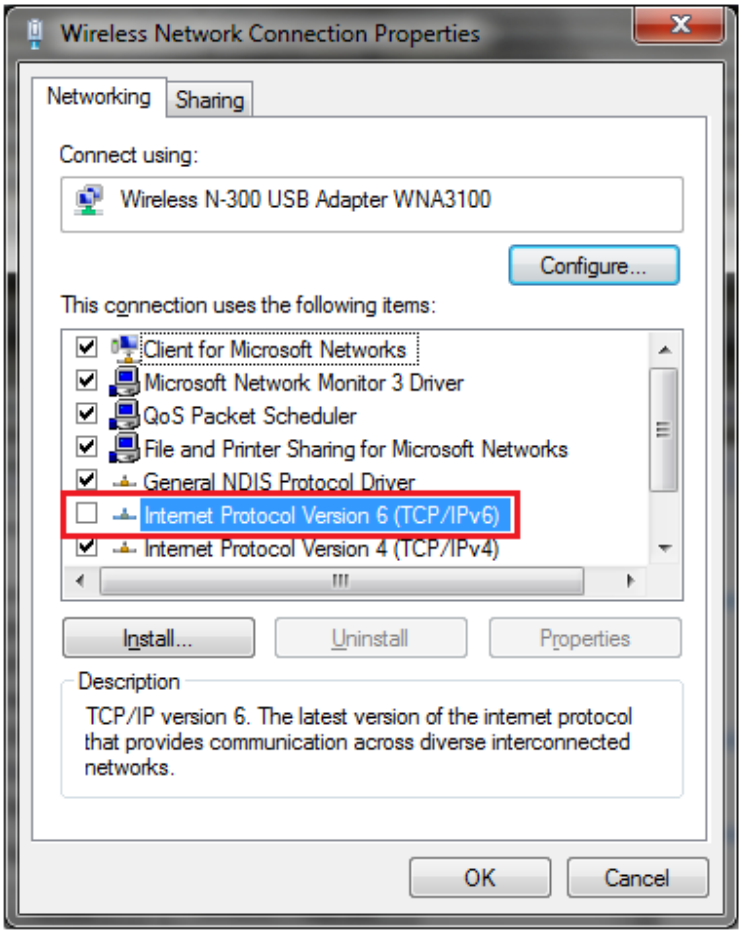

Figure 4.1 Disabling IPv6 on Windows

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. Furthermore, not all Operating Systems (e.g. Android) allow disabling the IPv6 traffic. A better solution will be to make VPN Client program reconfigure the IPv6 routing table as well so that both IPv4 and IPv6 traffic is securely sent through the VPN tunnel. Grooten[35] also mentions RFC6105, an informational document published by the IETF on how

to deal with rogue router advertisements that exploit IPv6 traffic. It proposes Secure Neighbor Discovery (SEND), a solution that is non-trivial to deploy. The RFC also proposes a complement to SEND based on filtering in the layer-2 network fabric, using a variety of filtering criteria, including, for example, SEND status. Unfortunately, most of these mitigation techniques are difficult to employ either due to the lack of a suitable implementation (e.g., SEND), or a lack of capable hardware (e.g., RA Guard or switch ACLs). Cisco also have some tips on first hop security. [45]

Man-in-the-Middle attacks can be detected using Neighbor Discovery Protocol Monitor NDPMon[46]- a diagnostic software application used by IPv6 network administrators for monitoring ICMPv6 packets
It is an IPv6 alike to ArpWatch and has similar basic features to detect suspicious neighbour/router discovery traffic. However, neither RFC6105 nor NDPMon will help to defend against the Attack. The ultimately best way to defend against Man-in-the-Middle Attacks exploiting IPv6 is to ensure that the Client machine always has an IPv6 connection so that no attacker can misuse our default gateway. Waters [36] points out the MitM attack is possible because we are *not* trying to subvert an existing IPv6 network but injecting RAs onto a IPv6-capable IPv4 networks, not native IPv6 or dual stack ones.

4.2. Defense against DNS Hijacking

DNS hijacking can be detected if the VPN Client periodically monitors the DNS Connection rather than just at the tunnel initiation. The Client's Routing Table should also be monitored for changes in the configuration. However, this is risky. The user information may have already been leaked in both of these solution.

DNS hijacking attacks can be defended if we configure the VPN tunnel gateway to have the same IP Address as the DNS resolver. This prevents adversary from producing a split tunnel and fooling the victim host into believing that the DNS is a local resource in the LAN. This solution has been implemented in Astrill VPN [47] and successfully prevent DNS queries to be redirected to the adversary.
Similarly, VyprVPN[48] uses an IP address for the server which is very close to the VPN entry point IP address. Another solution is to use a private DNS. Selecting a fake subnet is easy when a third party DNS service is being used. It becomes difficult when the VPN service provider uses its own private DNS as it ensures that the subnet contains at least three IPs (i.e., the DNS server, the gateway, and the VPN Client). However, the VPN is still at risk of route injection attack when OpenVPN is used with these configurations.

Another solution is to use Firewalls instead of routing table to send packets through the tunnel. This has been implemented in Android KitKat to isolate the mobile device from the LAN. However, this solution is not feasible on desktop computers that need to access resources on LAN. The computers will also not be able to handle DHCP renewals and will be disconnected from the Internet.

4.3. 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 a user profile, encryption keys and algorithms. A PIN number is usually required to invoke the smart card. A token card provides a one-time password. When the user authenticates correctly on the token by entering the correct PIN number, the card will display a one-time passcode that will allow access to the network.

Add-on authentication system, like TACACS+, RADIUS can be also used to create a profile of all VPN users, controlling the access to the private network.

4.4. Configuration Issues Management

Alessandro Mei[19] discuss the advanced security measures taken by VyprVPN to tackle the configuration issues. The researchers noticed that 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, each of which returns an error NXDOMAIN. 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 contacts 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.

Appelbaum [18] also proposes attempts to diminish some of the configuration issues in a platform specific manner. Using OpenVPN or some other TUN/TAP device-based VPN on Linux, we can use Netfilter and iptables to ensure that Operating System only lets the VPN Client program send packets to the network interface and stop any unprotected packets from leaving the physical device unless the VPN is sending them.

5. Conclusion

In this review, we first surveyed the tunneling technologies most commonly used by VPN service providers and discovered that many VPNS still rely on outdated technologies such as PPTP (with MS-CHAPv2), that can be easily cracked through brute-force attacks. The main focus of the literature review was to thoroughly study the research publications discussing the security flaws of the VPN technology. A number of papers were studied and discussed in the review. However, we want to conclude our paper with highlighting the three most important publications dealing with VPN security vulnerabilities to date.

Appelbaum et al. [18] were the first to give a scientific categorization of configuration issues that ought to debilitate the utilization of VPN services achieving privacy and secrecy on the Internet. In addition to other things, they experimentally watched the issue of IPv6 leakage in certain VPNs.

R.Hills[20] mentioned a portion of the basic VPN security vulnerabilities that NTA Monitor have found amid the most recent three years (2003-5) while performing VPN security tests. The paper focused on remote access VPN configurations utilizing the IPsec protocol. A portion of the issues that have been seen, for example, the username identification issue, were new revelations, while others are known constraints of the protocols, which are presented because of poor configuration.

The study [19] conducted at the Sapienza University of Rome and Queen Mary University of London, presented the first ever experimental evaluation of sixteen commercial VPN services, measuring the VPN client's IPv6 traffic being leaked over the native interface and carrying out two DNS hijacking attacks.

Throughout this study we realized the range of detection and defense practices to deal with the security vulnerabilities of VPNs have not been evaluated for their performance and security impact. We aim to complement our analysis of the security vulnerabilities with an evaluation of the countermeasures for VPN vulnerabilities.

References

[1] Charlie Schluting, “Networking 101: Understanding Tunneling”, http://www.enterprisenetworkingplanet.com/netsp/article.php/3624566/Networking-101-Understanding-Tunneling.htm, Aug 3, 2006

[2] Margaret Rouse, “tunneling or port forwarding”, http://searchenterprisewan.techtarget.com/definition/tunneling, May 2007

[3] “Tunneling”, techopedia, http://www.techopedia.com/definition/5402/tunneling

[4] Danish Pervez, “L2TP VPN”, http://vpnranks.com/l2tp-vpn/

[5] VASCO DATA SECURITY INTERNATIONAL Gmbh, “Basic PPTP Concepts”, Chapter 2, http://documentation.axsguard.net/manuals/Gatekeeper/8.0.0/output/html_chunked/howto_guides/pptp/generated_output.chunked/ch02.html, 2014

[6] Ricky M Magalhaes, “Secure Socket Tunneling Protocol”, http://www.windowsecurity.com/articles-tutorials/firewalls_and_VPN/Secure-Socket-Tunneling-Protocol.html, 17 April 2007

[7] AnchorFree. Hotspot Shield. http://anchorfree.com/hotspot-shield-VPN-download-windows.php. [Online; July-2012]

[8] Inc. Private Tunnel. Private Tunnel. https://www.privatetunnel.com/index.php/why-private-tunnel.html. [Online; July-2012].

[9] Charlie Osborne, “Hola VPN still riddled with security holes, researchers claim” http://www.zdnet.com/article/hola-vpn-still-riddled-with-security-flaws-researchers-claim/ [Online; June 2015]

[10] Charlie Osborne, “Hola: A free VPN with a side of botnet”

http://www. zdnet.com/article/hola-a-free-vpn-with-a-side-of-botnet/
[Online; May 2015]

[11] Ernesto Van der Sar, "HUGE SECURITY FLAW LEAKS VPN USERS' REAL IP-ADDRESSES"
https://torrentfreak.com/huge-security-flaw-leaks-vpn-users-real-ip-addresses-150130/
[Online; January 2015]

[12] Teresa Hummel, "Security Concerns and the Use of Microsoft Virtual Private Network for Small Businesses", Version 1.4b, SANS Institute 2003

[13] TorrentFreak
https://torrentfreak.com/
[Online; accessed Nov 2015]

[14] Ernesto Van der Sar, "Which VPN Services Take Your Anonymity Seriously? 2012 Edition"
https://torrentfreak.com/vpn-services-that-take-your-anonymity-seriously-2012-edition/
[Online; accessed Nov 2015]

[15] Ernesto Van der Sar, "Which VPN Services Take Your Anonymity Seriously? 2013 Edition"
https://torrentfreak.com/vpn-services-that-take-your-anonymity-seriously-2013-edition/
[Online; accessed Nov 2015]

[16] Ernesto Van der Sar, "Which VPN Services Take Your Anonymity Seriously? 2014 Edition"
https://torrentfreak.com/which-vpn-services-take-your-anonymity-seriously-2014-edition-140315/
[Online; accessed Nov 2015]

[17] Ernesto Van der Sar, "Which VPN Services Take Your Anonymity Seriously? 2015 Edition"
https://torrentfreak.com/anonymous-vpn-service-provider-review-2015-150228/
[Online; accessed Nov 2015]

[18]J. Appelbaum, M. Ray, K. Koscher, and I. Finder, "vpwns: Virtual pwned networks," in 2nd USENIX Workshop on Free and Open Communications on the Internet. USENIX Association, 2012.

[19] 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

[20] R. Hills, “Common VPN Security Flaws Whitepaper”, http://www.nta-monitor.com, January 2005.

[21] Steve Pitts,“VPN Aggressive Mode Pre-shared Key Brute Force Attack”, https://www.giac.org/paper/gcih/541/vpn-aggressive-mode-pre-shared-key-brute-force-attack/104625
GIAC practical repository, SANS Institute 2004

[22] D. Harkins and D. Carrel, RFC 2409 “The Internet Key Exchange (IKE)”, November 1998

[23] IKECrack, Performance
http://ikecrack.sourceforge.net/

[24]Bruce Schneier1, Mudge2, and David Wagner3
Cryptanalysis of Microsoft's PPTP Authentication Extensions (MS-CHAPv2)
https://www.cs.berkeley.edu/~daw/papers/pptpv2.pdf
CQRE '99, Springer-Verlag, 1999, pp. 192-203.

[25]M. Marlinspike, “Divide and Conquer: Cracking MS-CHAPv2 with a 100% success rate,” https://www.cloudcracker.com/blog/2012/07/29/cracking-ms-chap-v2, 2012.

[26] B. Krishnamurthy and C. E. Wills, “Generating a privacy footprint on the Internet,” in Proceedings of the 6th Conference on Internet Measurement. ACM, 2006, pp. 65–70.

[27] B. Krishnamurthy, D. Malandrino, and C. E. Wills, “Measuring Privacy Loss and the Impact of Privacy Protection in Web Browsing,” in Proceedings of the 3rd Symposium on Usable Privacy and Security. ACM, 2007, pp. 52–63.

[28] B. Krishnamurthy and C. Wills, “Privacy diffusion on the Web: a longitudinal perspective,” in Proceedings of the 18th International Conference on World Wide Web. ACM, 2009, pp. 541–550.

[29] “Alexa Top Sites,” http://www.alexa.com/.

[30] N. Viennot, E. Garcia, and J. Nieh, “A Measurement Study of Google Play,” in Proceedings of the 2014 ACM International Conference on Measurement and Modeling of Computer Systems. ACM, 2014, pp. 221–233.

[31] “IPv6-enabled BitTorrent Peers,” https://www.vyncke.org/ipv6status/p2p.php.

[32] M. Defeche, “Measuring IPv6 Traffic in BitTorrent Networks,” 2012, IETF Internet Draft.

[33]DNS Hijacking on Wikipedia, The Free Encyclopedia
https://en.wikipedia.org/wiki/DNS_hijacking
22 November 2015

[34] “OpenVPN,” https://openvpn.net/index.php/open-source.html.

[35] Martijn Grooten, “Researchers demonstrate how IPv6 can easily be used to perform MitM attacks”
https://www.virusbtn.com/blog/2013/08_12.xml
Posted 12 August 2013

[36] Alex Waters, “SLAAC Attack – Olay Windows Network Interception Configuration Vulnerability”
http://resources.infosecinstitute.com/slaac-attack/
Posted April 4, 2011

[37] DEFCON 21 Hacking Conference, 2012
https://www.defcon.org/html/defcon-21/dc-21-index.html

[38] SuddenSix
https://github.com/Neohapsis/suddensix

[39] Scott Behrens & Brent Bandelgar, “MITM ALL THE IPv6 THINGS!”
https://www.defcon.org/images/defcon-21/dc-21-presentations/Behrens-Bandelgar/DEFCON-21-Behrens-Bandelgar-MITM-All-The-IPv6-Things.pdf
DEF CON 21 August 2, 2013

[40] Evil FOCA
https://www.elevenpaths.com/labstools/evil-foca/index.html

[41] Chema Alonso, " Fear the Evil FOCA: mitm attacks using IPv6"
http://www.slideshare.net/chemai64/defcon-21-fear-the-evil-foca-mitm-attacks-using-ipv6
Posted August 2013

[42] THC-IPv6 with fake_router6
https://github.com/vanhauser-thc/thc-ipv6
Last update v3.0 2015-10-16

[43] "Security Enhancements in Android 4.4,"
http://forum.xdadevelopers.com/showpost.php?p=48703545.

[44] Steve Sinchak, "How To Properly Disable IPv6"
http://tweaks.com/windows/40099/how-to-properly-disable-ipv6/

[45] IPv6 First Hop Security—Protecting Your IPv6 Access Network, CISCO
http://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/enterprise-ipv6-solution/whitepaper_c11-602135.html

[46] NDPMON http://ndpmon.sourceforge.net

[47] Astrill VPN https://www.astrill.com/
[48] VyprVPN https://www.goldenfrog.com/vyprvpn

[49] Douglas Crawford, "PPTP vs L2TP vs OpenVPN vs SSTP vs IKEv2",
https://www.bestvpn.com/blog/4147/pptp-vs-l2tp-vs-openvpn-vs-sstp-vs-ikev2/,
December 2014