
An Investigation Into the Security and Privacy of iOS VPN Applications



**Abertay
University**

JACK WILSON

Division of Cybersecurity
School of Design and Informatics
Abertay University, Dundee

A thesis submitted for the degree of
Bachelor of Science with Honours

in

Ethical Hacking

1st May 2018

Word Count: 11,448

Abstract

Due to the increasing number of recommendations for people to use VPN's for privacy reasons, more app developers are creating VPN apps and publishing them on the Apple App Store and Google Play Store. In this 'gold rush', apps are being developed quickly and, in turn, not being developed with security fully in mind.

This paper investigated a selection of free VPN applications available on the Apple App Store (for iOS devices) and test the apps for security and privacy. This includes testing for any traffic being transmitted over plain HTTP, DNS leakage and transmission of personally-identifiable information (such as phone number, IMEI ¹, email address, MAC address) and evaluating the security of the tunnelling protocol used by the VPN.

The testing methodology involved installing free VPN apps on a test device (an iPhone 6 running iOS 11), simulating network traffic for a pre-defined period of time and capturing the traffic (either through ARP spoofing, or through a proxy program such as Burpsuite). This allows for all traffic to be analysed to check for anything being sent without encryption. Other issues that often cause de-anonymisation with VPN applications such as DNS leakage can be tested using websites such as *dnsleaktest.com*.

The research found several common security issues with the VPN applications that were tested, with a large majority of the applications tested failing to implement HTTPS. Additionally, a large majority of the VPN apps failed to route additional user data (such as DNS queries) through the VPN tunnel. Furthermore, just fifteen of the tested applications were found to have correctly implemented the best-recommended tunnelling protocol for user security.

Outside of the regular testing criteria, some other security anomalies were observed with specific apps that included outdated servers with known vulnerabilities, apps giving themselves the ability to perform HTTPS interception and questionable privacy policies.

Keywords: Mobile, Security, Privacy, Virtual Private Network, VPN, iOS.

¹An IMEI is a unique number used to identify mobile devices.

Contents

Abstract	i
1 Chapter 1: Introduction	1
1.1 Intent	2
1.2 Objectives	2
1.3 Scope	2
1.4 Evaluation	2
1.5 Structure	2
2 Chapter 2: Literature Review	3
2.1 Decline in Privacy	3
2.2 Definition of PII	3
2.3 Prior Research	4
2.4 VPN Security Issues	4
2.4.1 DNS Leakage	4
2.4.2 Tunnelling Protocols	5
2.4.3 Authentication Mechanisms	6
2.4.4 IPv6 Leakage	7
2.5 The Limitations of a VPN	7
2.6 Threat Modelling	7
2.7 Summary	8
3 Chapter 3: Methodology	9
3.1 Testing Criteria	9
3.2 Application Selection	9
3.3 Device Preparation	9
3.4 Accounts within Apps	10
3.5 Recording Results	10
3.6 Baseline Application	10
3.7 Packet Capturing	10
3.7.1 Proxy	10
3.7.2 ARP Spoofing	11
3.7.3 Remote Virtual Interfaces	11
3.8 Analysis of PCAP Files	11
3.9 DNS Leakage	14
3.10 Tunnelling Protocol	14
3.11 Permissions	15
3.12 Ethical Considerations	15
4 Chapter 4: Results	16
4.1 Overall Results	16
4.1.1 VPN Categories	16
4.1.2 Permission	17
4.1.3 HTTP within apps	18
4.1.4 PII Leakage	19
4.1.5 DNS Leak within apps	20
4.1.6 Protocol Usage	21
4.2 Examples of PII Leakage	21
4.2.1 App #5	21
4.2.2 App #8	22

4.2.3	App #21	22
4.2.4	App #52	23
4.3	Baseline Application Testing Results	23
4.4	Interesting Findings	24
4.4.1	Configuration file over HTTP	24
4.4.2	Self-Signed Root Certificate	24
4.4.3	Outdated Web Server	27
5	Chapter 5: Discussion	29
5.1	Discussion of Results	29
5.1.1	VPN Categories	29
5.1.2	Permissions	29
5.1.3	HTTP within apps & PII Leakage	30
5.1.4	DNS Leakage	30
5.1.5	Protocol Usage	30
5.2	The Worst Applications	30
5.3	‘Premium’ VPN app vs Cheaper Alternatives	31
5.4	Research Question Answer	31
5.5	Miscellaneous Recommendations for Security Improvements	32
5.5.1	DNS Over TLS	32
5.5.2	DNSSEC	32
5.6	Trust	32
5.7	Noteworthy Articles on Trust	33
5.7.1	“No Logs” VPN Services	34
5.7.2	TunnelBear completes industry-first third-party public security audit	34
5.7.3	Hola Uses User IP’s as endpoints	34
5.7.4	Facebook Onavo Analytics	34
6	Chapter 6: Guidance	36
6.1	Transport Encryption	36
6.2	Tunnelling Protocol Implementation	36
6.3	DNS Leakage	37
6.4	IPv6 Leakage	37
6.5	Unnecessary Information Gathering	37
7	Chapter 7: Conclusion	38
7.1	Future Work	38
7.1.1	Further Protocol Analysis	38
7.1.2	Authentication Mechanisms	38
7.1.3	HTTPS Interception	39
7.1.4	Analysis of App Rankings vs Security	39
8	References	40
9	Bibliography	44
10	Appendices	45
10.1	Appendix 1: PCAP Analysis Script	45
10.2	Appendix 2: Wordlist for Search Script	46
10.3	Appendix 3: Exclusions for Search Script	46

List of Figures

1	How a VPN Works	1
2	Overall Reduction in Concern About Online Privacy	3
3	A DNS Leak Explained	5
4	A DNS Leak Explained	10
5	Command to Start Remote Virtual Interface	11
6	grep Command to Search PCAP Files	12
7	tshark Command to Find Unique IP Addresses	12
8	tshark Command to Find ESP Packets	12
9	Example Output of Search Script	13
10	Example Result from dnsleaktest.com	14
11	Command to Analyse PCAP Files with Bro	14
12	Breakdown of App Categories	16
13	App Permission Requests	17
14	Chart of Apps Using HTTP	18
15	Chart of Apps Leaking PII	19
16	Chart of Apps Leaking DNS	20
17	Chart of Protocols in use	21
18	App #5 Sending Credentials in URL	21
19	App #8 Sending PII over HTTP	22
20	App #21 Sending PII over HTTP	23
21	App #21 Sending PII over HTTP	23
22	VyprVPN DNS Servers	24
23	Install Profile Screen	25
24	Install Profile Screen	26
25	iOS Certificate Trust Warning	27
26	Outdated Django Web Server	28
27	Speedify GPS Permission Request	29

List of Acronyms

Abbreviation	Expanded
VPN	Virtual Private Network
HTTP	Hyper-Text Transfer Protocol
DNS	Domain Name System
ARP	Address Resolution Protocol
MAC	Media Access Control
TLS	Transport Layer Security
SSL	Secure Socket Layer
PII	Personally-Identifiable Information
IKEv2	Internet Key Exchange version 2
PSK	Pre-Shared Key
GPS	Global Positioning System
ISP	Internet Service Provider

1 Chapter 1: Introduction

Virtual Private Networks (VPN's) were originally designed for business use, with two primary purposes: remote access (allowing an employee working remotely to access an internal company network) and connecting various geographically separate sites together over the internet (site-to-site VPN) (IETF, 2000).

Recently companies have taken to offering "VPN's as a service", where users can route all of their internet traffic through the VPN providers servers. Routing all traffic through an encrypted VPN tunnel to the VPN server prevents interception from internet service providers while offering privacy and/or security (i.e. when using untrusted public networks to prevent an attacker from effectively sniffing traffic). This is illustrated in Figure 1, below.

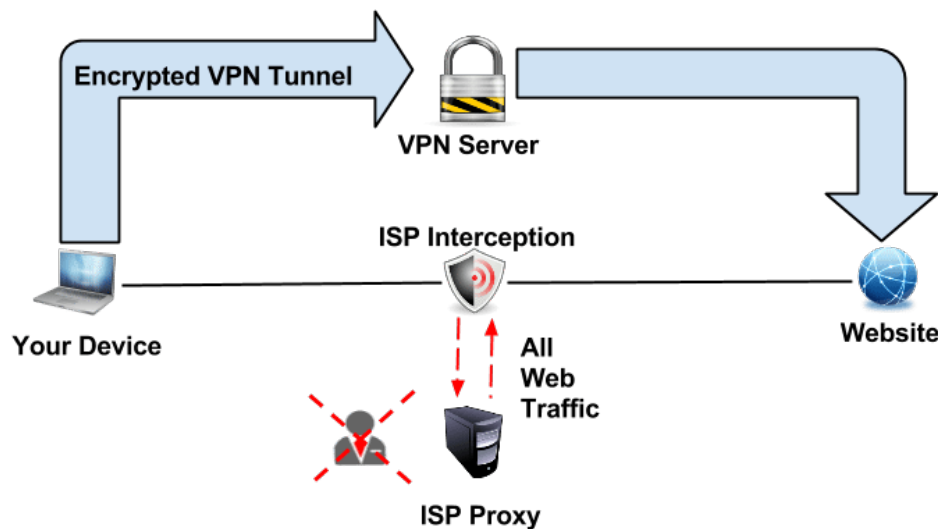


Figure 1: How a VPN Works (The VPN Guru, 2018)

Interest in VPN's has been slowly rising over recent years (Google Trends, no date). This growing interest can be partially attributed to media outlets recommending readers to adopt VPN's to avoid tracking by governments and internet service providers. There are countless examples of similar recommendations and articles both online and in print. (PC Mag, 2018).

While there is some accuracy to the claims in the various articles, some articles will fail to recommend what constitutes a good and secure VPN. Savvy consumers with a low-level of technical knowledge may search their mobile app store of choice (the Apple App Store or the Google Play Store) for "Free VPN", or a similar search term.

While a VPN can be used for privacy on public WiFi networks, to avoid ISP tracking and for bypassing geo-restrictions, a VPN does not solve every privacy issue, and is far from anonymous. Simply logging an IP address for troubleshooting or analytics, or requiring an email address or payment details for an account would already potentially de-anonymise a VPN user.

Picking a VPN goes far beyond the common security issues outlined in this research, there are a host of other factors to consider including: what is a VPN user's threat model? What is a user trying to achieve by using a VPN? A VPN has some limitations and will only protect a VPN user against certain threats. Both VPN limitations and threat modelling are outlined in the respective sections within Chapter 2, below. Additional technical factors (such as those outlined in the trust section of the discussion) may also be considered depending on the user's threat model.

1.1 Intent

Given the increase in popularity and the large availability of “VPN’s as a service”, this research aims to investigate a large selection of free/cheap VPN clients for Apple’s mobile operating system (iOS) to give an overview of the general state of security and privacy within the area. The investigation poses the question:

What is the overall state of security within (free and cheap) iOS VPN clients?

1.2 Objectives

Three main objectives were set to guide the project to a successful completion:

- Evaluate the state of iOS VPN security by following the testing methodology outlined in the methodology section of this report.
- Provide an in-depth analysis of the results based on the testing.
- Write guidance to outline best recommended practices to support iOS developers in creating more secure VPN clients.

1.3 Scope

The applications that were tested had to be available on the iOS platform, in the official app store and compatible with iOS 11. The applications had to be VPN clients that were either free to use or offered a trial period. More details on the selection of apps is discussed in the application selection section of the methodology.

1.4 Evaluation

The tested applications will be evaluated on a variety of criteria, based on the selection of testing undertaken in the methodology section of the paper. Not only can the evaluation be based on the results of the testing methodology, but they can also be broken down based on other criteria (such as if the app is completely free or offers a free trial, and whether an account is mandatory, optional or not required within the applications). Different factors of the results will be more or less important to different users when choosing a VPN app, so portraying the results in a wide variety of ways should help potential purchasers make a well informed decision.

1.5 Structure

Chapter 2 discusses work related to this research, most importantly VPN security research on Android. This is followed in Chapter 3 by an in-depth description of the testing methodology developed and conducted against the various applications. The results from this testing are presented in Chapter 4 and discussed in Chapter 5. Trust associated with choosing a VPN provider and some noteworthy news articles are also discussed in this chapter. Chapter 6 details the guidance offered to developers based on the results of the testing. The projects outcomes are summarised in Chapter 7.

2 Chapter 2: Literature Review

This chapter will give some insights into user privacy and the gradual decline in privacy concerns over time. From there, some detail is given into some of the common issues and misconfiguration errors that can lead to security flaws within VPN clients.

2.1 Decline in Privacy

At present, more products are becoming data driven (i.e. Uber which requires an app and an account compared to traditional taxis). User data being held by more companies has made privacy a more prevalent issue. Studies on user privacy and mobile application privacy have been undertaken which indicate users are becoming less concerned about their privacy while mobile applications are sending more user data than ever.

One such study (Data & Marketing Association, 2015) asked a selection of people their opinions on how their data was used and handled. The key takeaway (that is relevant to VPN security research) is that people care less about their online privacy, and the overall concern has decreased from 84% to 79% between 2012 and 2015. (Shown in *Figure 2*).

"On a scale from 1 to 10 where 1 is 'not at all concerned' and 10 is 'very concerned', how do you rate your levels of concern about the issue of online privacy these days?" | % who answer 7-10

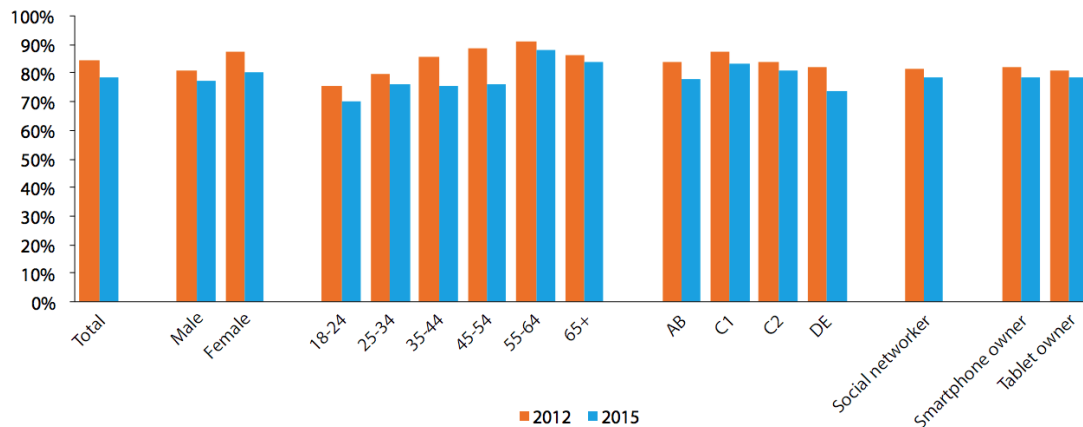


Figure 2: Overall Reduction in Concern About Online Privacy

As well as user attitude towards privacy, developer attitudes towards how users (and their data) are treated by developers must also be considered (this is an influencing factor to the research being undertaken). Developers should ensure they are not taking an excessive amount of unnecessary data and must also ensure that the data is being handled and stored securely.

Some prior research (Ren, Lindorfer, Dubois, Rao, Choffnes and Vallina-Ridriguez, 2017) analysed different versions of Android apps, with releases spanning several years. The research investigated what data about users and their devices was being sent to the developers and third-parties and demonstrated that (generally) apps have increased how much user data is being sent over time.

2.2 Definition of PII

Personally Identifiable Information (PII) is a term that is used to describe any form of information that could identify an individual person or device. Ensuring that PII does not end up in the hands of an adversary is of critical importance. If, for example, a user was using a VPN for security and

privacy on public WiFi but the VPN service was transmitting a username and password over HTTP, an attacker sniffing traffic on the network may be able to steal the credentials and impersonate the user.

The Information Commissioner’s Office (ICO) have created a flowchart-style document that can quickly determine if an item of information could be deemed ‘Personal Data’ (ICO, 2012). This document was partially referenced to create the below list of personally-identifiable information:

- Device Identifier
 - IMEI
 - IP Address
 - Device Serial Number
- User Identifier
 - Name
 - Banking Details
 - Date of Birth
- Contact Information (Phone Number, Email Address etc.)
- Location Data
 - Home/Work Address.
 - GPS Location
- Credentials
 - Username
 - Email Address
 - Password

2.3 Prior Research

Prior research into VPN security has been undertaken, both for mobile and desktop clients at a variety of price points. One such paper is **An Analysis of the Privacy and Security Risks of Android VPN Permission-enabled Apps** (Ikram, Vallina-Rodriguez, Seneviratne, Kaafar and Paxson, 2016) which investigates several common issues (described below) that can reduce the privacy of VPN users on the Android platform. The paper covered a large dataset of 283 apps, but only focused on VPN clients within the Android operating system.

A Glance through the VPN Looking Glass (Perta, Berbera, Tyson, Haddadi and Mei, 2015) is another piece of VPN security research that investigated a significantly smaller dataset (just 14 VPN clients) but clients were analysed on every common operating system (iOS, Android, Windows, OS X and Linux). The research involves testing the selection of VPN clients for known security issues within VPN clients (such as IPv6 leakage and DNS leakage).

Both of the aforementioned papers (Ikram, *et al.* & Perta, *et al.*) show there are issues present within the VPN client ecosystem and outline a variety of the vulnerabilities present that could reduce the privacy of a user, as well as a variety of methods to test for vulnerabilities, but no expansive research has been done on the iOS platform with a focus on free/cheap VPN clients (the majority of non-technical users will likely purchase VPN apps at this price-point). Some of the common security issues found within VPN clients are outlined in the VPN Security Issues section, below.

2.4 VPN Security Issues

There are several common security issues with VPN client that (if present) could reduce the privacy of a user. This does not take into account developing exploits focused specifically at individual applications, but rather on common security issues through misconfiguration errors.

2.4.1 DNS Leakage

A DNS leak is the result of an insecure VPN client making DNS requests to a DNS server not controlled by the VPN provider. Making DNS requests to a third-party allows that third-party (most commonly: an Internet Service Provider’s DNS server or Google DNS) to view the which

websites a user is visiting based on the hostnames that the DNS server is being asked to resolve. A DNS leak is visualised in Figure 3, below.

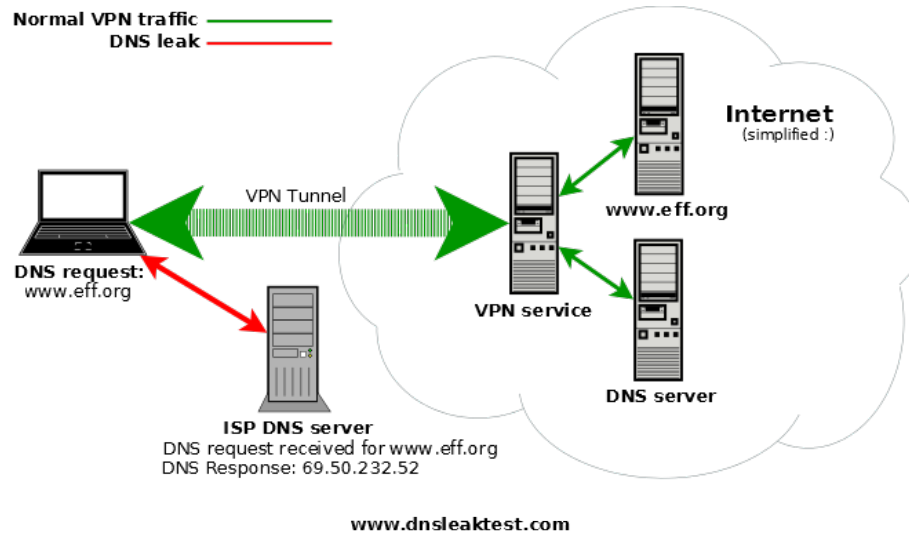


Figure 3: A DNS Leak Explained (dnsleaktest.com, no date)

If a VPN is being used to avoid tracking by ISP's, then a DNS leak to an ISP's DNS server could negate the point of a VPN service in the first place as an ISP can monitor and record the DNS resolve requests being made from a user's source IP address (outside of the VPN tunnel).

If a VPN provider was to run their own DNS server this would mitigate the issue of a DNS leak, as it centralises trust. Rather than trusting a VPN provider with user traffic, and a third-party with DNS requests, both of the above would be handled exclusively by the VPN provider.

2.4.2 Tunnelling Protocols

iOS supports a selection of VPN tunnelling protocols on both iOS and MacOS. Each protocol has a variety of benefits & disadvantages, some are more secure than others and each has support for different authentication mechanisms (Apple, no date). The purpose of investigating this security issue is to determine if the VPN clients are using a secure, modern tunnelling protocol (such as a variant that utilises IPSec) or a less secure, older protocol such as L2TP. The research is unlikely to encounter any PPTP-based VPN's as this protocol was deprecated in iOS 10.

2.4.2.1 IPSec

IPSec is a protocol that is often combined with other protocols (detailed below) that provides enhanced encryption and better authentication mechanisms. IPSec relies on two main protocols: The Authentication Header (AH) which is used to verify the source of the traffic and to ensure that traffic is not tampered with using hashes of the data being sent. The second protocol is the Encapsulated Security Payload (ESP) which uses symmetric encryption to encrypt data being sent between clients. (Hong Kong Government Infosec, 2008).

IPSec has three methods that can be used for authentication: pre-shared keys, encrypted nonces or digital certificates. These authentication mechanisms are detailed in the authentication mechanisms section, below.

2.4.2.2 IKEv2 (with IPSec)

IPSec is responsible for the integrity checking and encryption part of a VPN tunnel, but this is often combined with another protocol - the Internet Key Exchange (IKEv2) - to exchange keys for authentication purposes. This mutual authentication is performed by establishing a security association between two parties (in this case the user and the VPN server) where the shared secret information is exchanged to allow for the communication of the aforementioned AH and ESP protocols. (IETF, 2014).

2.4.2.3 L2TP (with IPSec)

The Layer 2 Tunnelling Protocol (L2TP) is a slightly older tunnelling protocol that is an advancement of the Point-to-Point Tunnelling Protocol (which was deprecated in iOS 10). It was built in a partnership between the PPTP Forum, Cisco and Internet Engineering Task Force: a body who specialises in standardisation of protocols and hosts RFC documents (Cisco, 2008). The L2TP protocol is defined in RFC 2661 (IETF, 1999). By default, L2TP does not encrypt any traffic, it is purely used to establish the VPN tunnel. For this reason L2TP is often combined with IPSec and is known as ‘L2TP over IPSec’. Securing the L2TP protocol using the IPSec suite is defined under RFC 3193. (IETF, 2001).

2.4.2.4 SSL VPN

SSL VPN’s are primarily intended for employees to remotely access internal resources within a company. An SSL-based VPN typically uses port 443 (HTTPS) so there are often no issues with firewalls (which can occur with other VPN protocols which use different ports). SSL VPN’s are browser-based so there is no need for any operating system integration or third-party VPN clients. This also makes them very lightweight in terms of CPU/RAM utilisation (IETF, no date).

2.4.3 Authentication Mechanisms

Authentication is a key part of a secure VPN infrastructure that ensures only valid and legitimate users of a VPN service can access the service. There are three common authentication methods (detailed in the sections below) with a recommendation given for which method to implement.

2.4.3.1 Pre-Shared Keys

A pre-shared key is essentially a password and it authenticates in a similar fashion that a user would connect to a basic home WiFi network, one ‘password’ that every user shares.

If PSK’s are used for authentication and the PSK is not unique for every user, this creates a security weakness. An adversary/attacker could (in theory) impersonate the VPN server and route all traffic through the adversary’s VPN server. (IVPN, no date). It is therefore recommended not to use non-unique pre-shared keys for authentication.

2.4.3.2 Encrypted Nonces

Nonces are randomly generated and unique numbers. To encrypt the nonce, a key exchange between both parties (in this case the VPN provider and the user) must take place to encrypt user’s nonce with the VPN provider’s public RSA key. Given that the nonces are unique this can help to mitigate replay attacks. (eTutorials, no date).

2.4.3.3 Digital Certificates

Digital certificates work in a similar way to encrypted nonces, relying on asymmetric cryptography to encrypt and decrypt data. The difference is that digital certificates are issued by a trusted certificate authority (in the same way SSL certificates are issued). Their main advantage to issuing

certificates in this way is that distribution of certificates is centralised with trusted certificate authorities. This also allows certificate authorities to revoke certificates that have been compromised. (Pitts, 2003).

2.4.4 IPv6 Leakage

Another common issue present within VPN clients that could reduce the privacy of a user and allow them to be tracked is IPv6 leakage. This issue occurs when the operating system makes requests over the IPv6 protocol (rather than IPv4), and due to a lack of support by the VPN client the requests are made outside of the secure VPN tunnel, leaking user information. This issue is also detailed in *A Glance through the VPN Looking Glass* (Perta, Berbera, Tyson, Haddadi and Mei, 2015)

On the iOS platform this issue should (theoretically) be less common as the only VPN tunnelling protocol that supports IPv6 is IKEv2 (Apple, no date). If a tunnelling protocol is in use that does not support IPv6 then IPv6 is disabled system-wide.

2.5 The Limitations of a VPN

Using a VPN can give a user more privacy, assuming the VPN is secure and does not suffer from the issues discussed above. However, a VPN does not make a user untraceable, and is far from anonymous, as some media outlets and VPN advertisements claim.

Most VPN providers require an account to use the service, and doing so could de-anonymise a user (given that a name, email address or payment information could be treated as personally-identifiable information). Even VPN providers that do not require accounts will often log users' source IP addresses (even temporarily) which could also serve to deanonymise a user.

WebUser magazine recently released an issue with the front page boldly claiming VPN's would allow users' to 'Stay 100% Anonymous' while offering guidance on what a VPN is and why consumers should use a VPN service (Irvine, 2018). The general explanation was technically accurate, and a variety of VPN services were recommended at different price points, however, the 'Stay 100% Anonymous' claims are entirely false, due to the fact most VPN services will log personally-identifiable information (e.g. email addresses for accounts or user IP addresses for troubleshooting), thus removing any form of anonymity.

2.6 Threat Modelling

The threat model of a VPN user must be heavily considered before using (and purchasing) a VPN. Security conscious users, political activists and users wishing to circumvent geographic restrictions to access content may use a VPN, but VPN's are not bulletproof, and far from anonymous. The majority of VPN's will log IP addresses of the user (even for a short period of time), and larger, paid VPN providers will often require billing details (such as bank details and a home address). These details can easily de-anonymise a user.

If the objective is to avoid nation state actors or law enforcement, a VPN will not protect the user from this. Many VPN providers have terms in their privacy policies that claim they will cooperate with law enforcement investigations. One such example is VyprVPN (a product owned by Golden Frog), which states in its privacy policy (under the *How Golden Frog Responds To Criminal Investigations* section):

"Golden Frog cooperates fully with law enforcement agencies, yet there must still be a subpoena before Golden Frog provides a member's identifying information - minimal information reasonably calculated to identify and no more". (Golden Frog, no date).

2.7 Summary

The various research, from paid and free VPN clients on various platforms, to the common ‘deanonymisation’ attacks that could compromise a user’s privacy, as well as the research into the gradual decline in user privacy over time indicate a spectrum of issues both from a technological standpoint, but also from a business trust standpoint.

Despite all of the aforementioned research, there is a lack of expansive research into the security of VPN clients on the iOS platform. The research in this paper aims to close the gap and present some insightful research into the subject area.

3 Chapter 3: Methodology

The methodology (detailed below) outlines how applications were selected for testing, how the test device was prepared before each app was tested to ensure consistency and each step of the active testing of applications to determine which common security issues are present in each of the selected applications.

3.1 Testing Criteria

Several areas of the applications' security and privacy were tested including:

- Was data sent over HTTP?
- Did the applications suffer from DNS leakage?
- Which tunnelling protocol was in use?
- Which permissions were applications requesting?

Each of the testing criteria is detailed further into the methodology section.

3.2 Application Selection

The applications tested throughout the research were downloaded from the Apple App Store. Due to cost limitations, the applications had to be either completely free to use, or offer a free trial (either time-based or data usage-based). The applications were found by using the search queries (or variations of the queries) listed below:

- "Free VPN"
- "Free VPN for iPhone"
- "Secure VPN"
- "Free VPN Anonymous"
- "Fast Free VPN"

3.3 Device Preparation

To test the applications an iPhone 6 was acquired which was running iOS 11.2.2 to be used as the test device. Automatic updates for both the operating systems and installed applications were disabled to prevent any possible changes to the results throughout the testing period.

During the testing of an application, the only other application open was the Safari browser to test for DNS leakage using www.dnsleaktest.com. No other tabs were open in the browser and no other apps were running in the background to mitigate collecting unnecessary data that may skew the test results.

The iPhone was connected to WiFi, with no SIM card inserted. This ensured all data was sent over WiFi, and nothing was sent using mobile data. Internet traffic was also mirrored over USB to a computer to capture the results. This is detailed in the packet capturing section of the report.

3.4 Accounts within Apps

During the testing, it was determined that there was three options regarding accounts within applications:

- An account was required.
- An account was optional.
- No option to have an account.

Where required or optional, an account was created to determine if any sensitive account credentials were transmitted over HTTP. Naturally, when no option for an account was present, no account was created.

3.5 Recording Results

All testing results (e.g. transmission of PII over HTTP, DNS leakage and tunnelling protocol implementation) were recorded in a Microsoft Excel spreadsheet. This provided a simple, easily searchable method to keep track of results and analyse the results using Excel's build-in graphing features. Each app was also assigned a unique identifier (the **App ID** column in the Excel spreadsheet). An example of the results from one application is shown in the Figure 4, below.

App ID	App Name	Using HTTP?	DNS Leak?	Leaking PII?	Tunneling Protocol Used	Contains ESP Packets?
6	Snap VPN - Unlimited VPN proxy	✓	✓	✗	IKEv2	✓

Figure 4: A DNS Leak Explained (dnsleaktest.com, no date)

3.6 Baseline Application

A baseline application was introduced to the testing that was generally regarded as a 'secure' VPN application. Having personally used VyprVPN for over a year, the app seemed generally secure and the service level (e.g. uptime and server availability) was very good. Given the prior experience with this 'premium' VPN app, it was decided this should be used as a baseline measurement for a 'secure' VPN application.

This app is a slightly more premium option (ranging from £43.50 to £57.50 per year), but it would be interesting to see what benefits and security enhancements are offered at this price-point compared to the selection of cheap and free alternatives being tested.

The testing of the baseline application followed the same testing methodology as the rest of the applications being tested, and this methodology is detailed in the below sections.

3.7 Packet Capturing

There was several options for packet capturing from an iOS device. Each of the options had positive and negative aspects to the method, so they were each evaluated (below) for effectiveness for the purpose of this research.

3.7.1 Proxy

The first option to packet capture would be to set up a proxy server, wherein all traffic passes through the proxy server before being forwarded on as normal traffic. This could be achieved using a proxy such as **Burpsuite** on a laptop on the same network as the test device. There was some issues with this method. The first was only having access to the free version of Burpsuite, which only allowed temporary project and did not allow for saving/loading of the Burp file format (it could however export .xml files).

The second issue was with trying to test using the program on a large enterprise network. The laptop and test device were often assigned to different subnets by the DHCP server meaning that proxying was not possible, and the subnets could not be changed. Finally, some applications were observed to have built-in proxy detection which could have prevented the applications from functioning correctly if they detected a proxy (the effectiveness of this detection was not evaluated).

3.7.2 ARP Spoofing

The second option was to capture traffic using ARP spoofing. This would involve using a program such as **Bettercap** to impersonate a router, forcing all traffic through the device running Bettercap. The issue with this method was that running a spoofing attack on a public WiFi network (or even a home network with other users) can have a network performance impact, and could inadvertently cause for traffic to be captured for all users on the network. This brings some serious ethical implications so it was decided not to use this method.

3.7.3 Remote Virtual Interfaces

The third, and best option is to use a technology called **Remote Virtual Interfaces** (RVI). This technology exists exclusively on iOS 5 or later and requires a device running OS X/macOS to use for packet capturing. The iOS device had to be connected to the MacOS device using a lightning to USB cable.

With the prerequisites installed (Xcode), running the command (shown in Figure 5) in a terminal window created a virtual network interface that mirrored the iOS network traffic to a virtual network interface on the MacOS device.

```
rvictl -s <device UDID>
```

Figure 5: Command to Start Remote Virtual Interface

The Unique Device Identifier (UDID) was determined by connecting the test device to iTunes and retrieving it from the phone information section. The **-s** flag creates the network interface and the **-x** flag with the same command stops and removes the virtual network interface. Once the command was executed, packet capturing software (such as Wireshark) could be directed at the virtual network interface to allow all traffic to be captured. (Apple, 2016).

This was the most effective method to packet capture traffic because it made no difference which subnet each device was on. An additional benefit was (due to the connection between the test device and the laptop being wired) there was no risk of interfering with other users network traffic by packet capturing WiFi traffic.

3.8 Analysis of PCAP Files

The analysis of the PCAP files involved several steps, the first of which involved analysing the HTTP traffic to check if the app was using the HTTP protocol in the first place, and if the HTTP traffic contained any information that could be deemed personally-identifiable. If HTTP was not in use, it was automatically determined that the app was not suffering from a PII leak.

Due to the sheer amount of PCAP files accumulated over the course of the testing, it became infeasible to manually analyse each file for the presence of personally-identifiable information. For this reason a small bash script (shown fully in Appendix 1) was written to parse the PCAP files for a list of keywords that were accumulated through the initial manual analysis.

The main part of the script involved two **grep** commands piped together. This command would take user input for a filename, run **grep** against the file with a wordlist of words commonly associated

with PII leakage (such as username, email, password etc.), then pipe the result of this to a second grep command that would remove words in an exclusions list and display the result to the user. Some exclusions were necessary due to a large quantity of HTTP headers that were of no use to the analysis.

The full wordlist can be viewed in Appendix 2 and the exclusions list can be viewed in Appendix 3.²

```
grep -i -a -f wordlist.txt ${SEARCHTERM} | grep -ivf exclusions.txt
```

Figure 6: grep Command to Search PCAP Files

Breakdown of Command

- -i: Ignore case-sensitivity.
- -a: Process a binary file as if it was a text file.
- -f: Use a wordlist.
- -v: Invert matches.
- \${SEARCHTERM}: The user input for the filename to parse.

The bash script contained a second command that would list all IP addresses within the specified PCAP file and filter for uniqueness:

```
tshark -r ${SEARCHTERM} -T fields -e ip.dst ip.src | sort | uniq
```

Figure 7: tshark Command to Find Unique IP Addresses

A third command was implemented to quickly determine if the VPN applications had a strong implementation of IPsec that utilised the Encapsulating Security Payload (ESP) protocol, shown below:

```
tshark -r ${SEARCHTERM} -O ESP | grep -i -q "Encapsulating Security Payload"
```

Figure 8: tshark Command to Find ESP Packets

The results of all three commands were also outputted to a .txt file for future reference and analysis. An example output from one of the analysed PCAP files is shown in Figure 9, below.

²There are a minimal amount of words in the exclusions list to minimise inadvertently omitting important results while not displaying unnecessary data

```

(Jacks-MBP:PCAPS jack$ ./search.sh

-----
This program will search a PCAP file against a wordlist, show a list of unique IP addresses and check
f the PCAP file contains ESP packets
Enter filename to compare against wordlist
Filename: 8.pcapng
-----

KEYWORD MATCHES:
-----
User-Agent: VPNClient/5.2.0 CFNetwork/893.14.2 Darwin/17.3.0
X-Amz-Cf-Id: G2o0auMzksfR6dMpkMaNAB6QT1c0Bil_oqqiVmhLzL1CwFco-IygaA==
User-Agent: VPNClient/5.2.0 CFNetwork/893.14.2 Darwin/17.3.0
Location: http://www.apple.com/
User-Agent: VPNClient/5.2.0 CFNetwork/893.14.2 Darwin/17.3.0
Location: https://www.apple.com/
User-Agent: VPNClient/5.2.0 CFNetwork/893.14.2 Darwin/17.3.0
X-Amz-Cf-Id: t5J5QHzUp_IMwkDGH7EB8GE021UJA80Bf-C-2vLsgt0zbrfXeVzHYw==
Host: reg.vpnvip.com
User-Agent: VPNClient/5.2.0 CFNetwork/893.14.2 Darwin/17.3.0
User-Agent: wp-iphone/5.2.0
Set-Cookie: PHPSESSID=ptp4as9r2mr8b3qedcv9pp37g1; path=/
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Set-Cookie: PHPSESSID=elick1h8fdh9jnf5hh0k5a28b6; path=/
<member><name>sessionID</name><value><string>elick1h8fdh9jnf5hh0k5a28b6</string></value></member>
Cookie: PHPSESSID=elick1h8fdh9jnf5hh0k5a28b6
User-Agent: wp-iphone/5.2.0
Set-Cookie: PHPSESSID=elick1h8fdh9jnf5hh0k5a28b6; path=/
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
<member><name>credits</name><value><string>0.00 MB</string></value></member>
<member><name>subscription_credits</name><value><string>0.00 MB</string></value></member>
<member><name>subscription_start</name><value><string>2017-10-11 15:31:54</string></value></member>
<member><name>subscription_end</name><value><string>2017-10-21 15:31:54</string></value></member>
<member><name>paid</name><value><boolean>0</boolean></value></member>
<member><name>username</name><value><string>junk@jack.lu</string></value></member>
<member><name>original_credits</name><value><string>0</string></value></member>
<value><string>vpn.vpnvip.com</string></value>
<value><string>uk.vpnvip.com</string></value>
<value><string>de.vpnvip.com</string></value>
<value><string>fr.vpnvip.com</string></value>
<value><string>ca.vpnvip.com</string></value>
<value><string>hk.vpnvip.com</string></value>
<value><string>jp.vpnvip.com</string></value>
USER-AGENT: Google Chrome/63.0.3239.132 Mac OS X
USER-AGENT: Google Chrome/63.0.3239.132 Mac OS X
USER-AGENT: Google Chrome/63.0.3239.132 Mac OS X
USER-AGENT: Google Chrome/63.0.3239.132 Mac OS X
-----

UNIQUE IP ADDRESSES:
-----
103.235.47.15
104.127.28.49
104.68.179.122
104.68.184.133
13.33.50.152
13.33.50.198
17.125.249.11
17.172.224.47
17.173.66.179
17.178.96.59
17.248.149.205
17.252.59.246
17.253.77.203
172.16.5.237
184.86.208.173
193.60.169.52
204.236.132.44
224.0.0.251
239.255.255.250
255.255.255.255
54.241.33.29
74.6.105.9
-----

CHECKING FOR ESP PACKETS...
-----
Does not contain ESP packets
-----
Results printed to output_8.pcapng.txt
Jacks-MBP:PCAPS jack$ █

```

Figure 9: Example Output of Search Script
13

3.9 DNS Leakage

Testing for DNS leakage was a fairly simple process that involved visiting www.dnsleaktest.com while the VPN connection was initiated and running an extended test. This test made a series of DNS requests to determine which DNS queries (by IP address) were being made to and also listed the DNS provider by name (e.g. Google, an ISP or a server operated by the VPN provider). If any DNS requests were made to any servers other than those in control by the VPN provider, the application was determined to be suffering from DNS leakage. An example test result from an app leaking DNS to Google is shown in Figure 10, below.

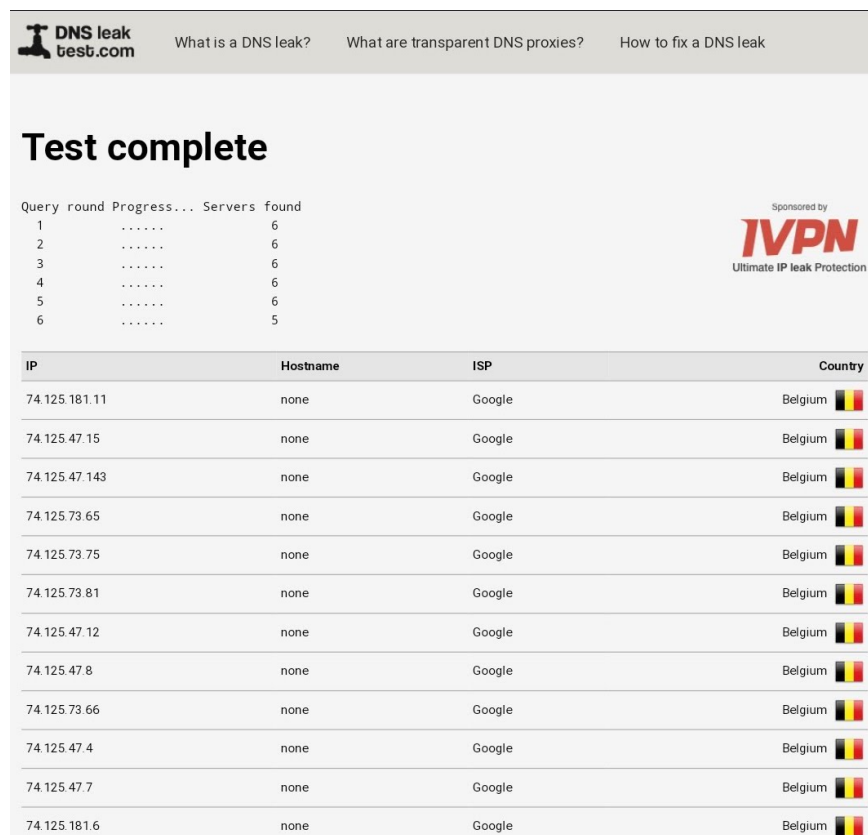


Figure 10: Example Result from dnsleaktest.com

3.10 Tunnelling Protocol

Ideally, apps should be making use of the Encapsulating Security Payload (ESP) protocol within the VPN tunnel, so the aforementioned bash script for automated PCAP analysis was used to check if the PCAP files contained any ESP packets.

Further to this, Bro (the network security monitor) was used to analyse each PCAP file after the testing was completed to offer further analysis and a breakdown of the protocols found in the PCAP file (Paxson, 1999). This was achieved with the below command:

```
bro -r <filename>.pcap
```

Figure 11: Command to Analyse PCAP Files with Bro

3.11 Permissions

The permissions requests being made by the VPN applications were analysed by interacting with the application and recording (in the Excel spreadsheet) which permissions the application requested. Some permission requests (such as VPN and notification) are expected, however, this section of the testing procedure was to determine if any applications were requesting permission they did not necessarily require (such as access to the media library or to contacts on the device).

3.12 Ethical Considerations

There are several ethical considerations to be made during the undertaking of this project. Firstly; It is against Abertay University's acceptable usage policy to "Attempt to circumvent any of the University's own or linked computing and Information Security measures". This means that using a VPN on the university's network (eduroam) is prohibited. This issue will be mitigated by completing the testing on another network (e.g. at home).

Another ethical issue could be uncovering potentially confidential data while investigating the VPN applications (such as developers passwords or other customer/user data). If any such data was found (e.g. in a Wireshark packet capture), then it would be stored securely for analysis, redacted where necessary in the results section of the thesis and properly disposed of once the research is completed. Depending on the severity of any results, responsible disclosure could also be considered to inform the application developers of the issues uncovered.

There are no other known ethical issues with this project. The project does not involve dealing with anyone's personal data other than the researchers (unless inadvertently uncovered, which is discussed above), there are no human test subjects, no surveys, and no dead or live human tissue involved in the project research.

4 Chapter 4: Results

This chapter details the overall results from the aforementioned testing methodology, detailing the general percentage of apps that met or failed to meet the testing criteria. From here, the results go on to detail some of the more interesting and unique results that were observed outside of the testing methodology.

4.1 Overall Results

A substantial amount of data was recorded during the testing. In total, 57 popular applications were tested, and the wide testing criteria allows for the results to be portrayed in a variety of ways. A decision was made to stop at this number for a couple of reasons: firstly, it was felt that the more popular VPN apps were tested (based on the search terms used to find the apps) and secondly, although dozens more applications existed across the app store, it became harder to find apps that functioned well enough to be fully tested.

4.1.1 VPN Categories

The app store does not have one definitive category for VPN clients, so it was interesting to see what categories app developers chose to categorise their apps under. The productivity and utilities categories on the app store were found to be the most popular choices for categorising VPN apps, however, some were listed under other categories.

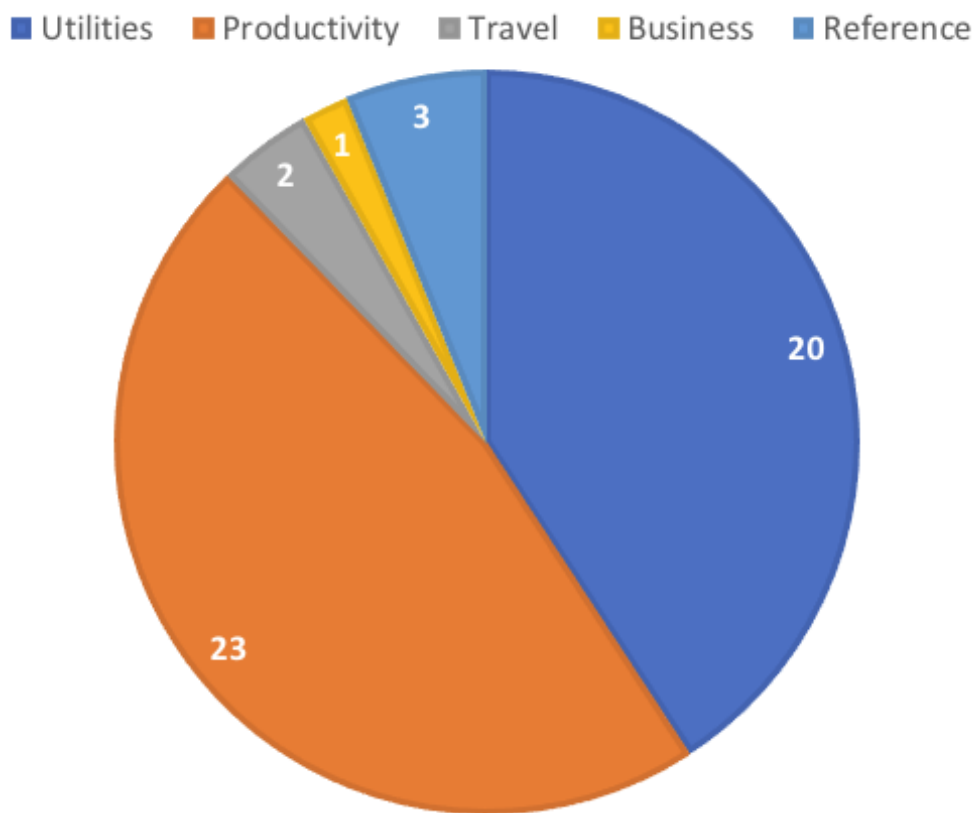


Figure 12: Breakdown of App Categories

4.1.2 Permission

Through interacting with applications during testing, the permissions apps were requesting or requiring was noted in the spreadsheet. Every application naturally required the VPN permission, nineteen of the apps were also requesting permissions to send notifications (e.g. if the app had a set amount of data per month to inform users when their data allowances were running low).

One app required access to GPS to determine user location in order to find the fastest servers to connect to. The only other permission requested by any app was by app #56, which required access to users photo libraries. The permissions are broken down in the below graph.

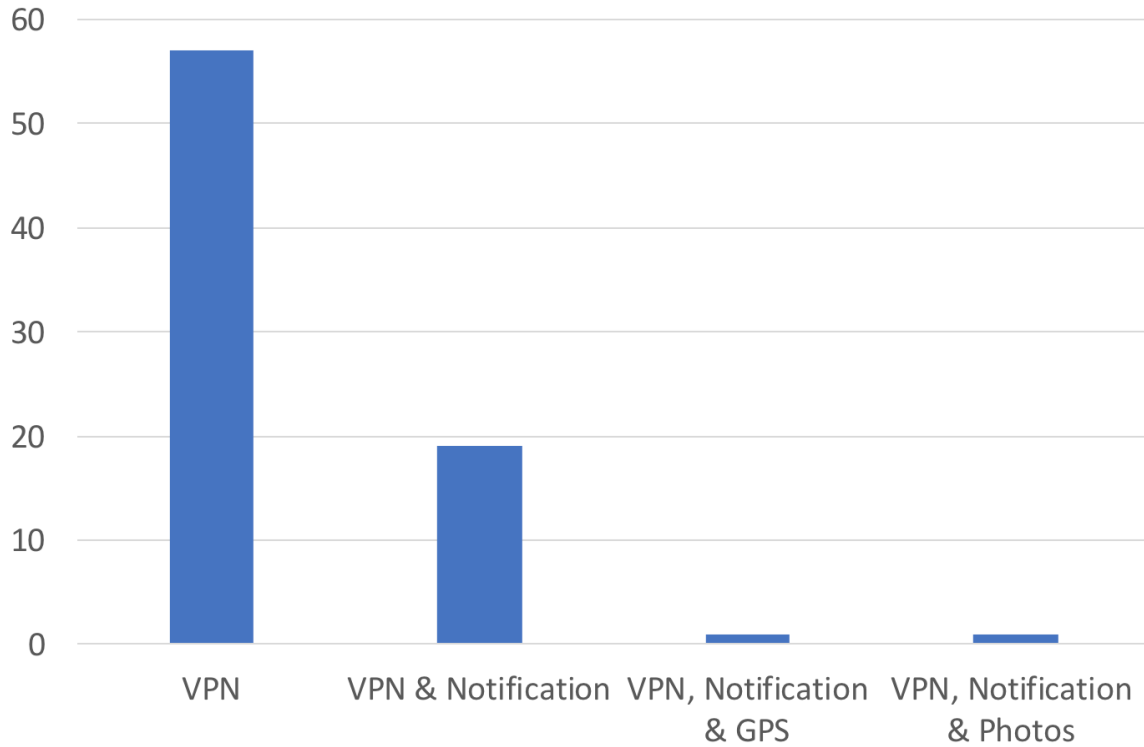


Figure 13: App Permission Requests

4.1.3 HTTP within apps

Out of the 57 apps tested, 39 of these were determined to be using the HTTP protocol (regardless of whether the HTTP traffic contained personally-identifiable information or not). With Apple aiming to enforce mandatory transport encryption (which requires all iOS applications to use HTTPS for all internet traffic), it is important to investigate how many developers are implementing transport encryption across apps. (App Developer Magazine, 2016).

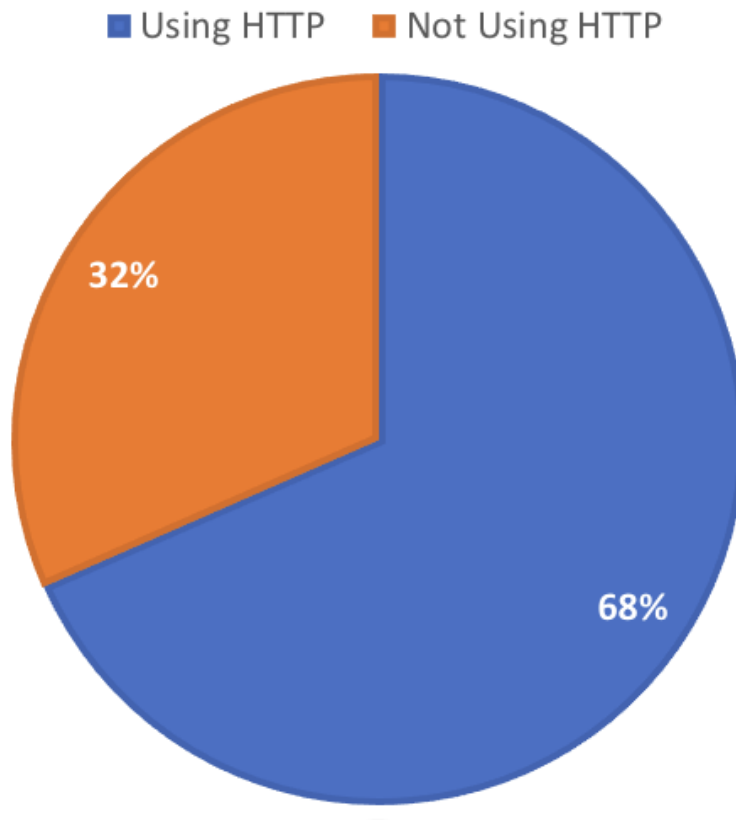


Figure 14: Chart of Apps Using HTTP

4.1.4 PII Leakage

Through the use of keywords within the bash script (and some random manual verification of the script's accuracy), it was determined that 21 of the 57 tested applications were leaking personally identifiable information about users. This included usernames, email addresses, passwords, source IP addresses and GPS coordinates (either determined approximately from the source IP address or by requesting GPS permissions).

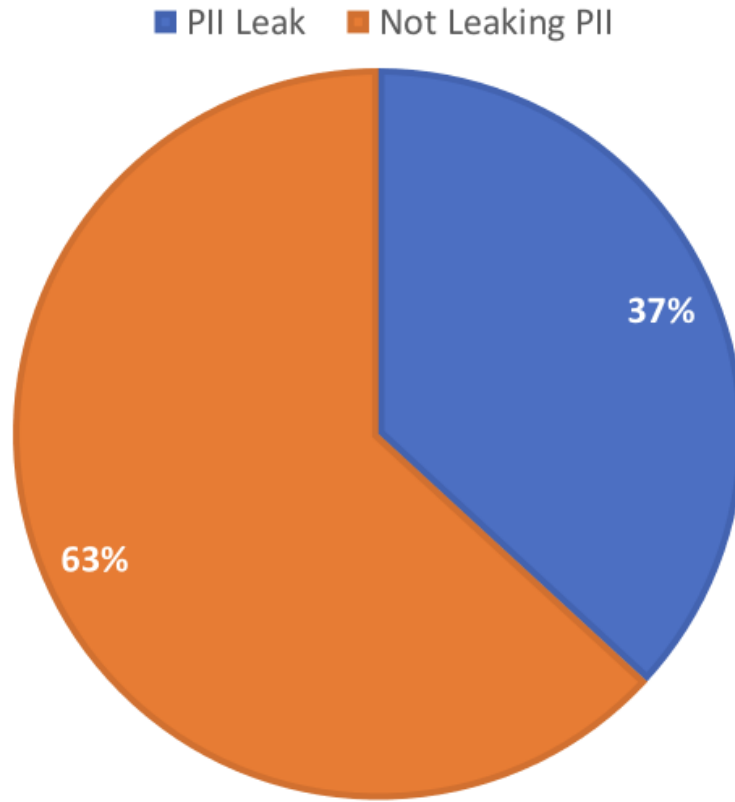


Figure 15: Chart of Apps Leaking PII

4.1.5 DNS Leak within apps

Due to several apps failing to establish a VPN connection, not all of the 57 tested apps could be tested for DNS leakage. Of the 44 apps that were tested, 32 suffered from a DNS leak. Twenty-eight of the apps suffering from a DNS leak were using Google DNS, with some others relying on Level 3 Communications (a large telecommunications provider) for DNS. This meant that the respective DNS provider that DNS was being leaked to could associate DNS requests with a user and therefore track websites that users visit.

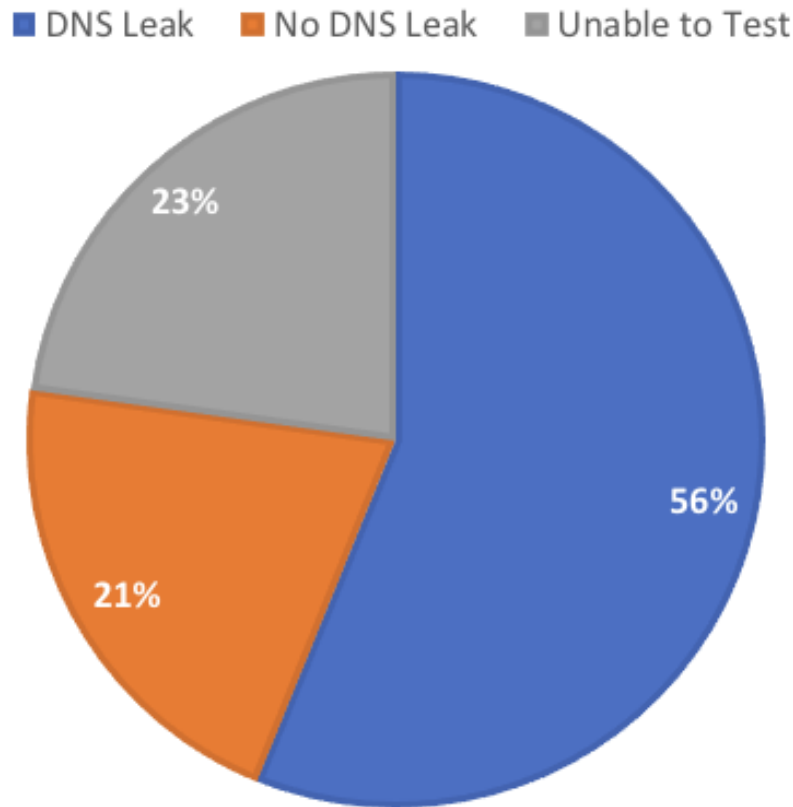


Figure 16: Chart of Apps Leaking DNS

4.1.6 Protocol Usage

Ideally, apps should be using the IPSec protocol (with IKEv2 and the Encapsulating Security Payload). The reasoning behind this is discussed in the guidance section. Just over half of the tested apps were using the IPSec protocol at all, even less were using IKEv2 and ESP alongside IPSec.

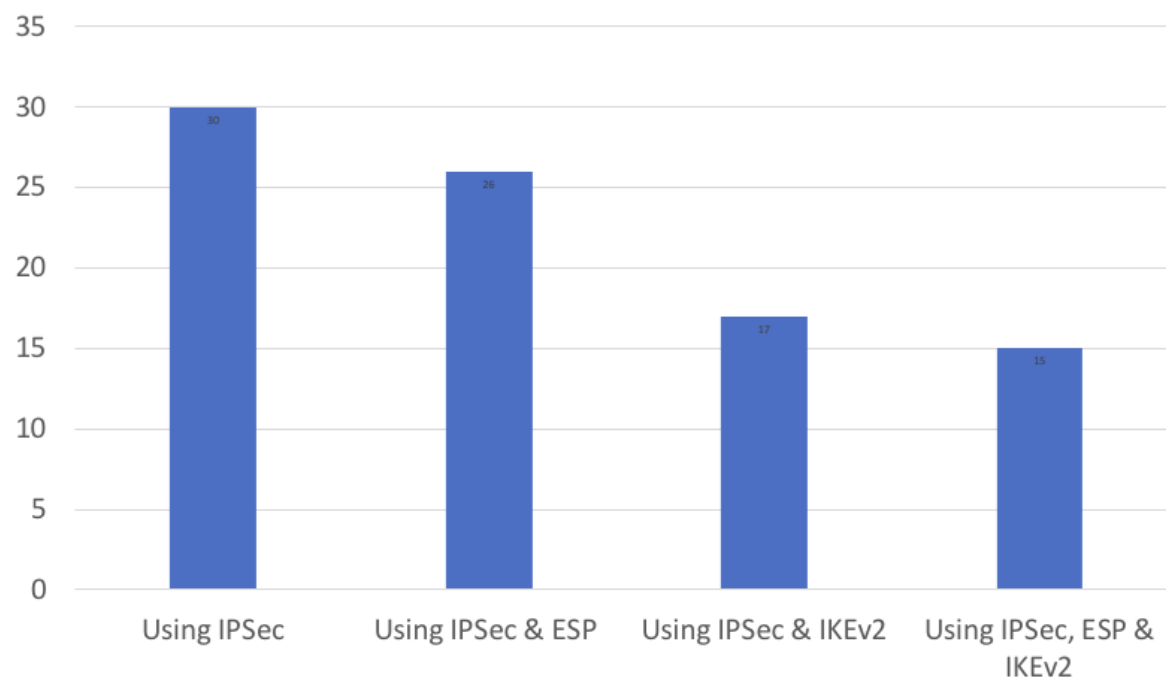


Figure 17: Chart of Protocols in use

4.2 Examples of PII Leakage

This section outlines some of the more critical findings where applications were found to be leaking personally-identifiable information using the insecure HTTP protocol. For obvious reasons, any confidential data (such as device ID's of the researcher) have been either partially or fully redacted.

4.2.1 App #5

During the testing, app #5 (VPN - Fast VPN Master) was observed to be sending personally identifiable information in URL parameters. This included the email and password used to create an account within the app. The 'pass' parameter was determined to be hashing the password using MD5 (this was verified by cracking the MD5 string and verifying it matched the original password). MD5 is a weak hashing algorithm, and despite being a marginal improvement over transmitting password in plain text, is still not an advisable method to securely transmit credentials.

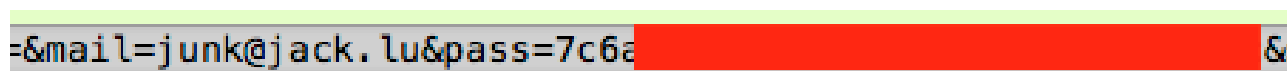


Figure 18: App #5 Sending Credentials in URL

4.2.2 App #8

Application #8 (VPN Master Unlimited vpn proxy) was observed to be sending personally-identifiable information to a server, with the PII enclosed within XML files. This included an email address, a password and a unique device identifier.

```
▼ <value>
  ▼ <string>
    junk@jack.lu
  </string>
</value>
</member>
<member>
  ▼ <name>
    password
  </name>
  ▼ <value>
    ▼ <string>
      [REDACTED]
    </string>
  </value>
</member>
<member>
  ▼ <name>
    lang
  </name>
  ▶ <value>
    </member>
</member>
  ▼ <name>
    deviceid
  </name>
  ▼ <value>
    ▼ <string>
      3024[REDACTED]
    </string>
  </value>
</member>
```

Figure 19: App #8 Sending PII over HTTP

4.2.3 App #21

App #21 (Best VPN Proxy Betternet) was observed to be sending a large quantity of personally-identifiable information over HTTP enclosed within a JSON file. This included the country, city, partial postcode and latitude and longitude coordinates of the user, as well as the user's IP address and internet service provider.

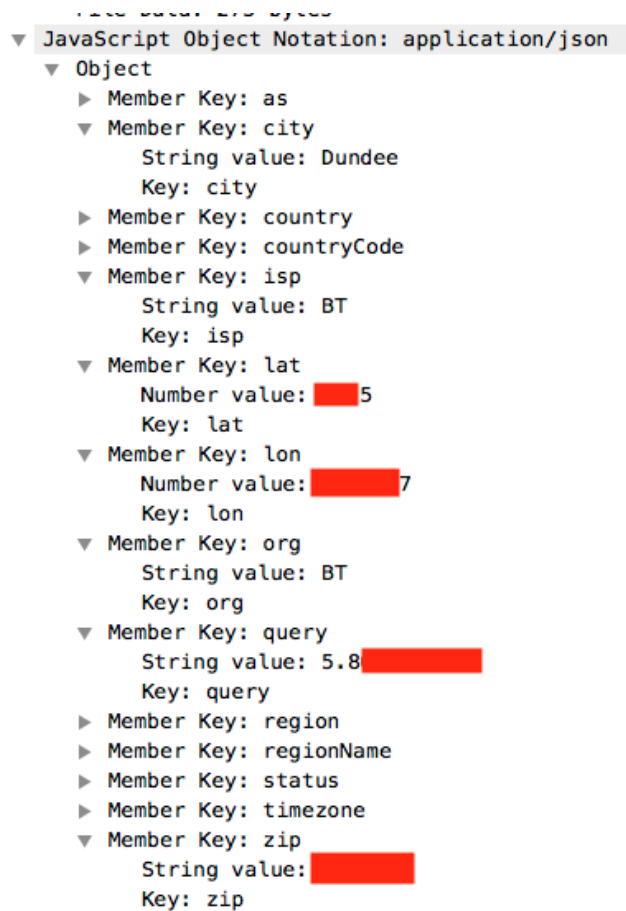


Figure 20: App #21 Sending PII over HTTP

4.2.4 App #52

App #52 (Super VPN - Private browser & wifi) was determined to be transmitting sensitive information over HTTP, although not necessarily that of the user. The photo below shows an IP address (owned by a cloud hosting provider) and credentials used to authenticate the user to the VPN service. Authentication could be improved by using better method such as a digital certificate (as discussed in the authentication mechanisms section).



Figure 21: App #21 Sending PII over HTTP

4.3 Baseline Application Testing Results

VyprVPN was tested as per the same testing methodology as every other application tested, passing against the testing methodology with no problems. The app was not sending any data over HTTP (and therefore not leaking any PII), DNS queries were made to two servers (operated by VyprVPN and hosted in a third-party datacentre). The results of the DNS test are shown in the image below.

Test complete			
Query	round	Progress...	Servers found
1		2
2		2
3		2
4		2
5		2
6		1



IP	Hostname	ISP	Country
149.154.159.219	219.159.154.149.in-addr.arpa	EDIS GmbH	Germany 
46.101.250.113	dns2.de1.goldenfrog.com	DigitalOcean	Germany 

Figure 22: VyprVPN DNS Servers

4.4 Interesting Findings

Some interesting results were discovered that were outside of the regular testing methodology. These results are outlined below and were reported to the developers as slightly more serious security issues.

4.4.1 Configuration file over HTTP

One app was found to be downloading the VPN configuration file over HTTP. Knowing this, an attacker could intercept the configuration file, replace it with a malicious configuration file that included a VPN server controlled by the attacker and route all VPN user traffic through the attacker-controlled VPN server.

4.4.2 Self-Signed Root Certificate

During the setup process of one of the VPN applications, the app requested permission to install a profile on the device. This profile was signed by 'VoiceFive Networks, Inc' with a verified tick, shown in the image below.

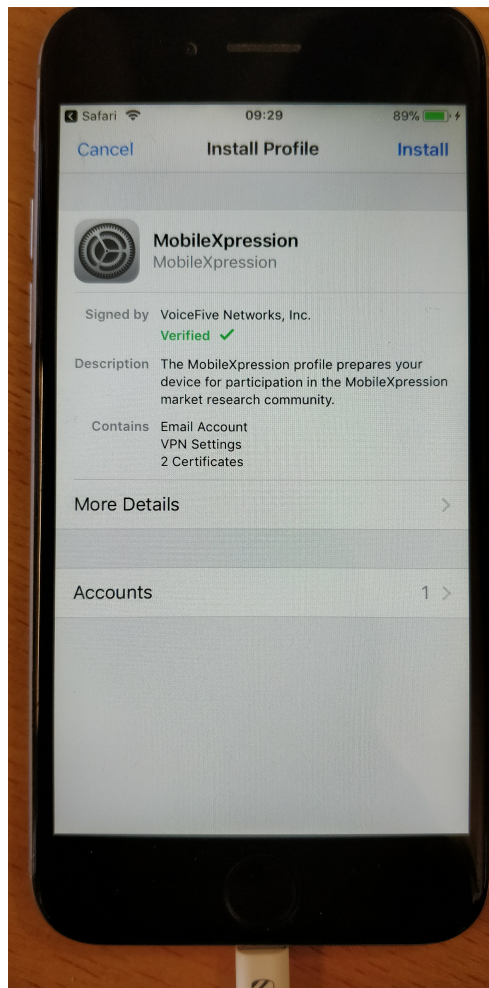


Figure 23: Install Profile Screen

Clicking on 'More Details' revealed that the profile contained VPN settings (as expected), four signing certificates that were issued by Symantec and Verisign (trusted certificate authorities on iOS), and two signing certificates that were issued by 'MobileXpression CA'. The list of trusted root certificates that are preinstalled on iOS does not contain **MobileXpression CA** (Apple, no date), and this certificate was self-signed by the app developer (shown in the image below).

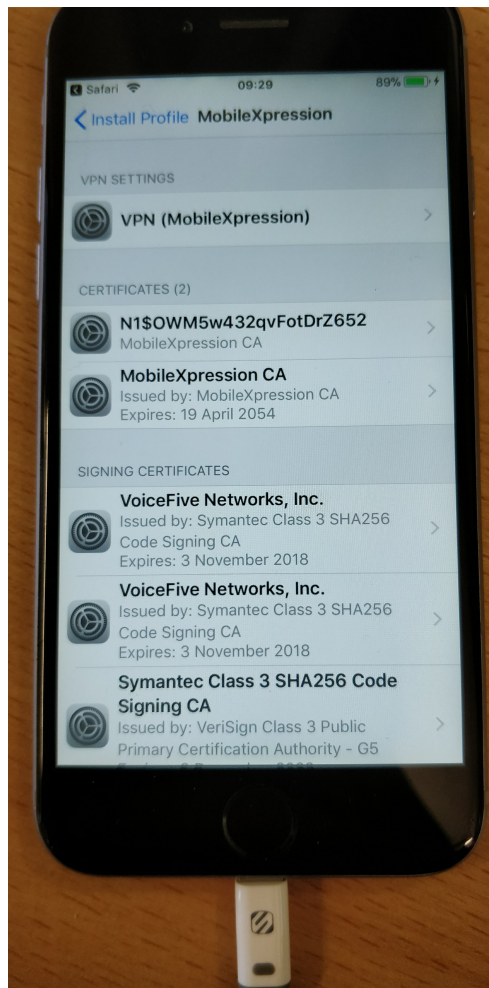


Figure 24: Install Profile Screen

Allowing a developer to install a self-signed certificate on the device is a security concern and would theoretically allow the app developers to intercept encrypted HTTPS traffic to view confidential data such as usernames and passwords for other websites, apps and services, as well as (theoretically) allowing the developers to inject adverts into webpages that users visit.

When continuing with the installation of the root certificate, iOS will specifically warn users that the unmanaged root certificate will not be trusted by default and that full trust needs to be enabled (shown in the image below). To continue using the VPN app, enabling full trust is a requirement. No other application that was tested required that a self-signed root certificate was installed on the device, this was an anomaly and generally not the best practice for configuring a VPN client on iOS.

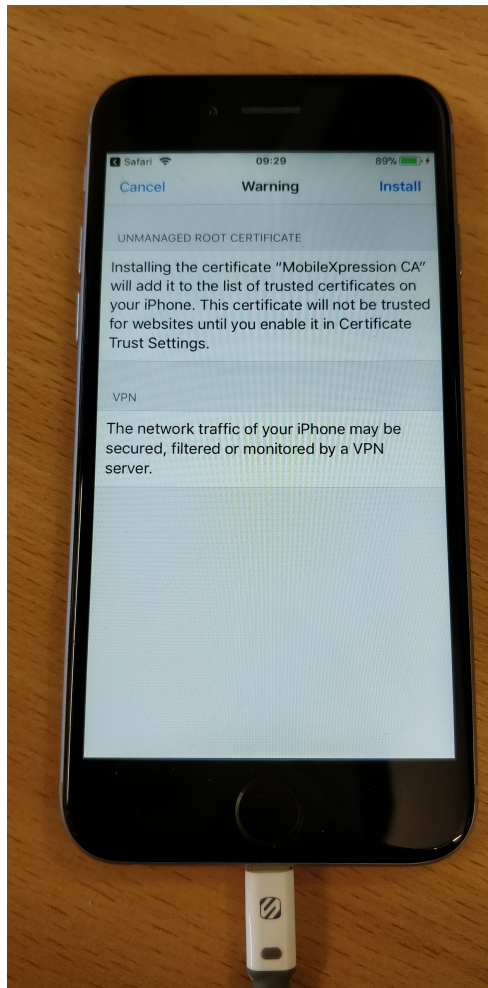


Figure 25: iOS Certificate Trust Warning

4.4.3 Outdated Web Server

One application was determined to be making a strange-looking request to a directory on a web server (this was observed in a Wireshark packet capture). Upon visiting the directory in a browser it was discovered that the server was running *Django*, which is a Python web framework. The developer had left debugging enabled publicly, and the web page displayed debugging information (including the Django version number).

The specific version of Django was checked against the CVE database for known vulnerabilities. It was determined that the installed version of Django was over 3 years old (at the time of writing) and had multiple vulnerabilities that had high severity ratings on the CVE rating scale.

It should be noted that no hacking tools were actively ran against the infrastructure of the VPN provider. A URL was visited in a browser (where requests to the same URL were already being made by the VPN app). The debugging information and version number were publicly visible to anyone on the internet with no authentication required. Responsible disclosure to the VPN provider will be attempted in due course to notify of the issue.

TypeError at /cloudapi/report/serverReachable

unbound method unSupportGetMethod() must be called with Error instance as first argument (got nothing instead)

Request Method: GET
Request URL: http://162.159.138.240/cloudapi/report/serverReachable
Django Version: 1.6.1
Exception Type: TypeError
Exception Value: unbound method unSupportGetMethod() must be called with Error instance as first argument (got nothing instead)
Exception Location: /home/django/162159138240/cloudapi/views/userreport.py in add, line 15
Python Executable: /usr/bin/python
Python Version: 2.7.6
Python Path: ['/home/django/162159138240', '/home/django', '/usr/bin', '/usr/lib/python2.7', '/usr/lib/python2.7/plat-x86_64-linux-gnu', '/usr/lib/python2.7/lib-tk', '/usr/lib/python2.7/lib-old', '/usr/lib/python2.7/lib-dynload', '/usr/local/lib/python2.7/dist-packages', '/usr/lib/python2.7/dist-packages']
Server time: Tue, 30 Jan 2018 08:15:44 +0800

Traceback [Switch to copy-and-paste view](#)

```
/usr/lib/python2.7/dist-packages/django/core/handlers/base.py in get_response
112.         response = wrapped_callback(request, *callback_args, **callback_kwargs)
    ▶ Local vars

/home/django/162159138240/cloudapi/views/userreport.py in add
15.         return Error.unSupportGetMethod()
    ▶ Local vars
```

Request information

GET	No GET data
POST	No POST data

Figure 26: Outdated Django Web Server

5 Chapter 5: Discussion

5.1 Discussion of Results

5.1.1 VPN Categories

There are a large quantity of VPN applications in the App Store. Within the tested results there was inconsistencies with the categorisation of apps: primarily the VPN apps were categorised under **Productivity** or **Utilities**, however, some apps within the selection that were tested were also categorised under **Travel**, **Business** and **Reference**.

This inconsistency in VPN app categorisation could impact how apps are ranked, and what apps are displayed to users (e.g. when browsing specific categories for VPN apps). It would be interesting to compare the results of this research with each apps average rating in the app store and ranking in respective categories. Due to the substantial amount of VPN apps available, it could be beneficial for Apple to introduced a category specifically for VPN's.

5.1.2 Permissions

The results of the permissions testing was surprisingly uninteresting, with almost every application only requesting the fairly standard permissions of VPN and notifications. There were only two applications that differed from this: app #49 (Speedify VPN) which required VPN, notification and GPS permissions. As shown in the image below, the app required access to GPS to 'find the closest Speed Servers'.

Although this does make sense from a technical standpoint, it stood out as the only application doing this through GPS location. Other apps (as evidenced through PII leakage) were able to determine approximate location through the user's IP address rather than through GPS coordinates.

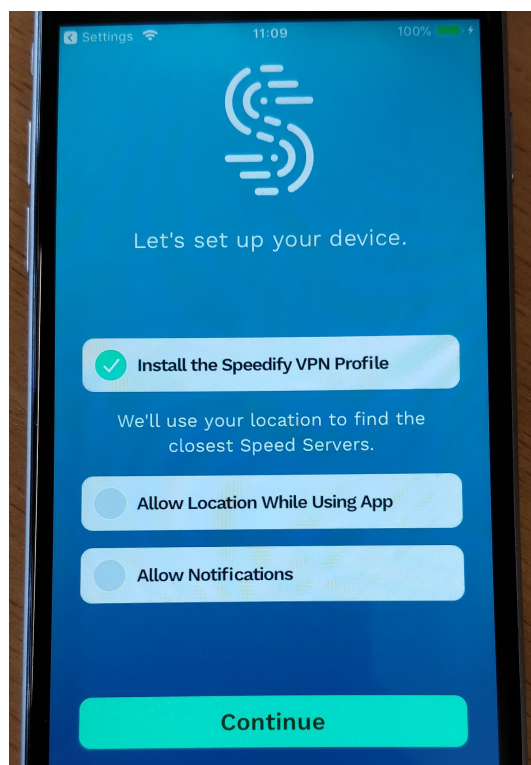


Figure 27: Speedify GPS Permission Request

The second application observed to be requesting unusual permissions for a VPN was app #56 (Cleaner - VPN + Anti Theft). This app required access to VPN and notification permissions (as would be expected), but also required access to the photo library. Upon further investigation this does not seem as strange, or that the applications has any malicious intentions as one of the apps intended functions is to remove duplicate photos.

5.1.3 HTTP within apps & PII Leakage

A large majority of applications were found to still be using HTTP for certain data. Given Apple's plans to enforce mandatory transport encryption, developers should definitely be looking to move away from HTTP sooner rather than later.

The personally-identifiable information leakage was found to be a less-common issue (with just 37% of apps leaking PII under the research's definition of 'PII'), but it was an issue nonetheless. In the age of free and simple services such as **Let's Encrypt** there is absolutely no excuse for developers to not be encrypting usernames, passwords and other personal information that is being sent over the internet.

5.1.4 DNS Leakage

DNS leakage was one of the most prevalent issues found over the course of application testing. For the applications that were tested for this issue, almost 73% were leaking DNS to a third-party organisation. Given that almost all of the DNS leaks were leaking requests to Google also introduces problems for user privacy, allowing Google to track the websites that users visit.

Fixing DNS leakage is a fairly simple issue that involves configuring DNS to route through the VPN tunnel and picking a trusted DNS service that advertises an interest in user privacy. This is further discussed in the DNS leakage section of the guidance in Chapter 6.

5.1.5 Protocol Usage

Out of the 57 apps that were tested, a fairly low amount (15 apps) were found to be using the best-recommended practice of using the IPSec protocol with IKEv2 and the encapsulating security payload. Given the strong security of ESP and the bonus features of IKEv2 it was surprising to see such a low amount of apps supporting both of these protocols. Guidance on implementation of the recommended tunnelling protocol configuration is discussed in more technical detail within the tunnelling protocol section of the guidance chapter.

5.2 The Worst Applications

The three main elements of the testing criteria were the use of HTTP, PII leakage and DNS leakage. Several of the fifty-seven apps that were tested failed all three of these categories, risking users' security and privacy more than usual. The apps below will be some of the first applications where responsible disclosure of the issues is offered to the developers.

App ID	App Name
1	FreeVPN
5	VPN - Fast VPN Master
15	Free VPN for iPhone and iPad
17	MyMobileSecure
18	VPN Dragon - Free VPN, Stable & Fast VPN
19	HexaTech Unlimited VPN Proxy
21	Best VPN Proxy Betternet
25	Whale VPN
28	#VPN - Wi-Fi Hotspot Security
38	VPN Guru - Master of VPN Proxy
39	VPN - Trust VPN Master ???VPN
41	Onion TOR Browser + VPN
45	- VPN
47	VPN 365 - WiFi Security
48	SkyVPN - Best VPN Proxy Shield
56	Cleaner - VPN + Anti Theft

5.3 ‘Premium’ VPN app vs Cheaper Alternatives

VyprVPN was tested as a baseline measurement of what is to be expected of a ‘premium’ VPN application. As would be expected with an app at such a price point (£43.50-£57.50), the application passed the testing criteria outlined in the methodology section.

Given the substantial price difference between the main set of tested apps and VyprVPN, there are some additional features that are offered to enhance user security. This includes all VPN servers being hosted by VyprVPN themselves, rather than with a third-party. One benefit of this is that law enforcement requiring access to servers would contact VyprVPN directly rather than the hosting provider, meaning that the VPN provider would know for definite if a user was being investigated.

VyprVPN also offers support for a variety of protocols: PPTP (on supported devices), L2TP over IPSec, OpenVPN and Chameleon - VyprVPN’s custom-built OpenVPN-based protocol that is intended to prevent VPN blocking (e.g. in countries where VPN’s are restricted) (Goldenfrog, no date).

5.4 Research Question Answer

The research question posed at the beginning of this research was: ‘What is the overall state of security within (free and cheap) iOS VPN clients?’

In short, there are a lot of security issues with a large majority of the apps tested against the criteria outlined in the methodology section. The testing aimed to cover some of the more popular VPN apps, but there are a lot more VPN’s that remain untested. Furthermore, the testing methodology only covered a selection of the common security misconfigurations that can lead to a reduction in privacy for users. There are other factors that could influence a users privacy when using a VPN app.

Despite the countless security issues and strange findings throughout the research, some VPN apps generally offered a good, secure service. This includes apps #22, #29, #31 and #57. All of these applications were ensuring that all traffic was encrypted, DNS was not leaking and IPSec was well implemented with the Encapsulating Security Payload.

5.5 Miscellaneous Recommendations for Security Improvements

The below sections offer some additional recommendations (not specifically relating to the research) that serve to offer extra improvements to the security and privacy of VPN users.

5.5.1 DNS Over TLS

Normally, DNS requests (and in turn the DNS protocol) are sent over plain text, unencrypted. DNS over TLS sends DNS requests using the same level of encryption that HTTPS web traffic uses. This would prevent an adversary sniffing traffic to be able to see DNS requests in transit and (from that) determine which websites a user was visiting. The protocol is further defined under RFC 7858 (IETF, 2016).

Ideally, DNS requests would be made through the secure VPN tunnel, but DNS over TLS adds an extra layer of security, and assists in protecting users even when a VPN connection is not active.

Several recent commits to the Android Open Source Project (similar to GitHub for managing Android versions) indicated that the feature was written into Android (Android Open Source Project, no date) and is expected to be under full release in an upcoming release of the operating system. If Apple were to implement support for DNS over TLS into the iOS (and MacOS) operating systems this would only further strengthen the security of a user's web traffic.

5.5.2 DNSSEC

Another protocol that could be implemented to enhance security is DNSSEC. This protocol provides authentication for the origin of DNS resolutions to verify the integrity of the DNS resolutions, preventing tampering and DNS poisoning attacks. (IETF, 2005).

Despite sounding like a good option for verifying DNS request integrity, its effectiveness has been brought into question for several reasons including having to transition the responsibility of issuing TLS certificates to the controllers of the DNS servers, rather than trusted certificate authorities (such as VeriSign, etc.).

Such actions could politicise the issuing and revocation of TLS certificates. A prime example of this is bit.ly (the url shortener), given that the .ly TLD (and in turn the DNS server) would be under control by the Libyan government.

Additionally, DNSSEC being installed on top-level DNS servers would not mitigate a DNS poisoning attack where DNS queries are being fetched from a cache (e.g. on a home network router).

The Asia-Pacific Network Information Centre (who control IP address allocation and DNSSEC deployment in the Asia-Pacific region) recently undertook a study into DNSSEC adoption. This study found that DNSSEC adoption was already low (sitting at 9% in 2013). There was a gradual increase to a 16% adoption rate in 2014, followed by a decline to 12%, where the adoption rate currently sits. (APNIC, 2018).

Widespread DNSSEC deployment would be complex and expensive, due to almost every piece of internet-speaking software needing to be reprogrammed to understand and handle the errors that DNSSEC could throw if an attack was detected. DNSSEC would also require substantial, widespread adoption to ensure success.

The first known draft of DNSSEC was from 1994, so not only is the technology and cryptography potentially outdated, but if DNSSEC was a valid and useful option it would have probably been adopted long before now given that the original spec is almost 25 years old (Sockpuppet, 2015).

5.6 Trust

There is one key point to remember when selecting a VPN provider:

A VPN simply moves trust from an ISP to the VPN provider

For this reason, there are several considerations that must be made (White, 2015):

- Will the VPN provider (at least try) to keep user data safe/secure and prevent a data breach?
- Will the VPN provider stick to the claims in their privacy statement? There have been examples in the past where VPN providers have went against the claims they were making in their privacy statement (that are discussed in the ‘No Logs’ VPN Services section, below).
- Does the VPN provider have a policy for logging user data, if so, how long are logs kept for? If a user wants to avoid tracking, having the VPN provider logging user activity could have a detrimental affect. It is therefore important to find if a VPN provider keeps logs, and if so, are the logs kept for a set period of time.
- Will the VPN service sell user data to third-parties? Given that VPN providers have the capability to monitor websites that users visit, it would be possible to profile users browsing habits and interests, and sell this to third-party advertising agencies. This is another potentially bad factor if a VPN user was attempting to avoid tracking.
- Will the VPN provider interfere with user traffic? One VPN app (Hotspot Shield) was found to be actively injecting JavaScript into users’ browsers for advertising and tracking purposes (The Register, 2017).
- Is the VPN provider using a modern encryption protocol for traffic outside of the VPN tunnel (such as TLS) or an older, vulnerable protocol such as SSL? Ensuring that modern protocols with no known vulnerabilities are in use will increase the security of users and their data.
- Does the VPN provider offer an uptime guarantee for the service? This doesn’t directly relate to security, but knowing the service is reliable at any given time is beneficial.
- Does the VPN provider undertake any server hardening measures (e.g. gr-sec and patching for security updates?). These additional steps can further increase the security of users and their data.
- Does the VPN provider store customer passwords in a secure manner? (e.g. bcrypt). If a VPN provider was unfortunate enough to be involved in a data breach, this would make users passwords considerably harder to decrypt.

Some of the factors listed above will mean more or less to different people (depending on each user’s threat model), and some of the factors will also be easier to research than others (depending on the transparency and honesty of VPN providers). In the end, it really comes down to each VPN user to consider the above factors along with the results from the practical work undertaken to make an informed decision on which VPN service to use/purchase.

5.7 Noteworthy Articles on Trust

Recently in the news, several stories have broken that detail potential wrongdoings of VPN providers that could violate trust with users. This section details some of the more noteworthy articles.

5.7.1 “No Logs” VPN Services

In late 2017, an alleged stalker was arrested by the FBI. Part of the evidence associating the suspect to the crime was logs provided to the FBI by PureVPN (The Register, 2017). An excerpt from the privacy policy of PureVPN (PureVPN, 2016) states that:

“we will only share information with authorities having valid subpoenas, warrants, other legal documents”

The statement is completely reasonable, it would be expected that any company would comply with various laws and regulations, however, the same privacy policy also states:

“We Do Not monitor user activity nor do we keep any logs.”

The fact that PureVPN stated explicitly in their privacy policy that they do not keep logs, yet handed over logs to help in the prosecution of a suspect introduces some serious trust implications when picking a VPN provider.

5.7.2 TunnelBear completes industry-first third-party public security audit

TunnelBear, a VPN provider, recently completed an industry-first independent security audit. (TunnelBear, 2017). This involved a third-party security consulting company (Cure53) conducting a ‘penetration test’ on the TunnelBear servers, applications and infrastructure. The results of the initial testing and the follow-up testing (completed after the remediations for the issues were implemented) was released to the general public. (Cure53, 2017).

This was an interesting move from TunnelBear as it shows a good level of transparency and openness as a company. Not only are the company actively trying to find issues within their service so they can be remediated, but the results are being released publicly to demonstrate that the company is taking an interest in keeping customers and their data safe. This action by TunnelBear should also hopefully encourage other VPN providers to follow in their footsteps.

5.7.3 Hola Uses User IP’s as endpoints

A few years ago, **Hola Better Internet** (a VPN provider) was found to be using its users computers as endpoints/VPN servers for other users of the service, rather than running dedicated VPN servers. (Business Insider, 2015). There are several potential issues with this: Firstly, users with a slow internet connection or data caps could experience throttling or additional bills from their ISP due to the potentially large amount of VPN traffic being routed through their computer/router.

The second problem with this practice introduces some legal issues: if **Hola User X** had an active VPN connection and was using the VPN for illegal means (e.g. downloading pirated content) and the endpoint was the computer of **Hola User Y** authorities would be under the impression that **User Y** was browsing illegal content, as the illegal content enters and exits the VPN tunnel at **User Y’s** IP address.

5.7.4 Facebook Onavo Analytics

In 2013, Facebook acquired an analytics company called Onavo. Onavo operate a VPN application on iOS and Android which was one of the applications tested against the methodology outlined earlier in this research. By the definition of the testing criteria, the application was deemed to be secure, ensuring no data was sent over HTTP, no DNS requests were leaked outside of the VPN tunnel and the VPN used the IPSec protocol with ESP.

Despite technically being secure, the app does not offer much user privacy. It recently came to light in various news articles (Wall Street Journal, 2017) that data going through the VPN was redirected to Facebook's servers where it is logged for analytical purposes. The below quote is directly from the Onavo privacy policy (Onavo, 2013).

“Onavo may use your information to: [...] Analyze how you use applications and data. For example, we may combine the information, including personally identifying information, that you provide through your use of the Services with information about you we receive from our Affiliates or third parties for business, analytic, advertising, and other purposes”

6 Chapter 6: Guidance

This section is intended to be a continuation of the previous chapter that discussed some of the actions developers can take to improve the security of VPN applications (based on the findings of the research). The implementation of the guidance will vary from developer-to-developer and application-to-application depending on the current state of security within each application. Implementation of the below steps would most definitely put a VPN client considerably further ahead in terms of security than most VPN clients in the free/cheap market (based on the findings in the dissertation research).

The main security issues and misconfiguration errors (that are explained in detail below) are DNS leakage, IPv6 leakage, insecure protocol implementation, poor authentication mechanisms and a lack of traffic encryption outside of the VPN tunnel.

6.1 Transport Encryption

When a user starts a VPN application, it is common for the app to connect to multiple web servers for several reasons (i.e. for authentication, selecting a VPN server to connect to & to connect to advertising networks to serve ads to the user). What servers apps connect to will vary on a per-application basis but the research found a large majority of the tested VPN applications were using HTTP for connections, which sends user information across the internet in plain text.

Sending potentially confidential user information (such as usernames, passwords, phone IMEI numbers and advertising ID's) across the internet serves to substantially reduce the privacy of a user. Apple intend to enforce mandatory transport encryption for all iOS applications (App Developer Magazine, 2016). This means that when this mandatory encryption is enforced, applications not encrypting all traffic (without an exception approved by Apple) will no longer be supported on later versions of iOS.

HTTPS adoption is growing rapidly - from August 2017 to February 2018 HTTPS adoption within the top 1 million most popular websites has grown from 276,852 websites to 366,005, a change of 32.2% (Helme, 2018). There has never been an easier (or cheaper) time to implement HTTPS. Services such as **Let's Encrypt** offer free DV certificates with a validity period of 3 months and solid documentation on implementing these certificates with an automated renewal process.

Let's Encrypt also recently announced support for wildcard certificates. (Let's Encrypt, 2018). This means (in an environment where a developer runs centralised advertising and authentication) rather than having to attain and manage separate certificates for `example.com`, `auth.example.com` and `ads.example.com`, one wildcard certificate (`*.example.com`) could be issued which would cover all subdomains.

Implementation of HTTPS when an application is reliant on third-party services (e.g. for advertising) could prove to be slightly more difficult. Advertising networks are known to take additional user data that could be personally-identifiable to deliver targeted advertisements.

The recommendation from a technical standpoint would be to move to an advertising network that offers support for HTTPS, but this is a harder specific recommendation to make as changing to a different advertising network could have a direct impact on the profitability of the business/app. Each business/app developer must evaluate customer privacy versus profitability to make an informed, individual decision on a matter such as this.

6.2 Tunnelling Protocol Implementation

The four tunnelling protocols currently supported on iOS (SSL VPN, IPSec, L2TP over IPSec and IKEv2) were outlined in the Literature Review section. Due to being the strongest cryptographically, while having the bonus features of IPv6 support and stability across network changes, it is recommended to implement support for the IKEv2 protocol, with the use of Encapsulating Security Payload (ESP).

Deployment of this protocol relies on the use of the `NEVPNProtocolIKEv2` and `NEVPNProtocolIPSec` classes within the `NetworkExtension` framework. (Apple Developer, no date).

6.3 DNS Leakage

An explanation of a DNS leak was covered in the Literature Review section. The recommended best-practice to prevent a DNS leak is to ensure that all DNS traffic is routed through the VPN tunnel, and preferably to a secure DNS service that advertises an interest in user privacy and support for DNS over TLS³. Some examples include the recently launched Quad9 DNS (built in part by IBM) (Quad9, no date) and the Quad1 DNS resolver, built in collaboration with Cloudflare. (Cloudflare, 2018).

In terms of deployment from a developer perspective, setting the DNS resolver for a VPN tunnel involves implementing the `NEDNSSettings` class within the `NetworkExtension` framework. (Apple Developer, no date).

6.4 IPv6 Leakage

Following the recommendation to implement IKEv2 will result in the VPN having support for IPv6. Similarly to DNS leakage, IPv6 leakage happens when IPv6 traffic is not properly routed through the VPN tunnel. It is recommended to add support for IPv6 through the VPN tunnel using the `NEIPv6Route` and `NEIPv6Settings` classes within the `NetworkExtension` framework (Apple Developer, no date).

6.5 Unnecessary Information Gathering

Throughout the dissertation research, numerous VPN applications were found to be leaking personally-identifiable information that could identify a VPN user. Combining this with the ever increasing amount of data breaches (Digital Guardian, 2015) affecting companies worldwide, there is a substantial risk with handling and storing user information.

On the 25th of May 2018, the General Data Protection Regulation (GDPR) is due to come into effect, which aims to ensure companies handle and store data securely to prevent data breaches. This is backed up by the threat of large financial penalties if companies were to suffer from a data breach.

To minimise the risk of data breaches it is important to ensure best practices are followed in regards to server hardening and patching, but the best option for a developer is to not collect unnecessary user data and stick to the bare minimum amount of data required to keep the app operational.

‘Data that doesn’t exist can’t be stolen or misused’.

³DNS over TLS encrypts DNS requests to prevent tampering/modification of the requests between the VPN server and the DNS server, after traffic exits the VPN tunnel.

7 Chapter 7: Conclusion

In conclusion, this research has proven that some security flaws exist within a large selection of VPN applications on the iOS platform. One of the largest problems found was the lack of HTTPS in use for both confidential and non-confidential data. This issue could be solved by Apple enforcing mandatory transport encryption for all iOS apps (which was announced and then delayed indefinitely) which would require all traffic to be sent over HTTPS to prevent sensitive information leakage (App Developer Magazine, 2016).

The most prevalent issue found was DNS leakage, with 32 of 44 apps tested for this issue suffering from a DNS leak. This issue leads to a substantial reduction in privacy for users, enabling the DNS providers to track users browsing activity based on DNS resolution requests. It is of critical importance that developers route all traffic (including DNS and IPv6) through the VPN tunnel, while also using a secure DNS provider, such as those discussed in the DNS leakage guidance section of the paper.

This research (at present) has not changed anything. Although the findings indicate security issues with a large amount of the VPN applications tested, consumers will (generally) not be aware of this and most will probably not have the time, interest or technical ability to test their VPN application of choice.

This leaves the problems in the hands of the VPN developers. It is crucial that developers understand the various risks and common security misconfigurations that can cause a reduction to user security and implement the necessary fixes. The most common security misconfigurations have been outlined and investigated throughout this paper, with the recommended fixes discussed in the guidance section. Outside of the regular misconfiguration errors, it is of crucial importance that developers are taking the necessary server hardening measures (such as good patching practice).

The guidance section of this report is intended to support developers in creating secure VPN applications. It is hoped that combining responsible disclosure of the results to developers alongside the guidance will offer security improvements to future updates of the VPN applications.

7.1 Future Work

Overall, the research has uncovered a variety of common security issues within VPN applications. Despite this, there are more known issues that can commonly lead to a reduction of privacy for users that were not tested for during the investigation. This section outlines some ideas for future work that is relevant to the research undertaken.

7.1.1 Further Protocol Analysis

Despite being able to determine the protocols in use by some of the VPN applications (through the methods described in the tunnelling protocol section of the methodology), this was fairly inconsistent, only being able to determine the protocols in use by 30 of the tested applications.

Attempting further protocol analysis with Bro (network monitoring software with a built-in protocol analyser) yielded no further success in determining protocols in use. Ideally, further work could include additional research into protocol analysis to better determine how many applications are properly implementing secure tunnelling protocol configurations.

7.1.2 Authentication Mechanisms

Apple's iOS 11 supports four VPN protocols, and each protocol has various methods for authenticating users. Based on the discussion about the most common authentication mechanisms in the authentication mechanisms section of the literature review, it would be recommended to use digital certificates for authentication purposes. Further research could be undertaken to determine which authentication mechanism each application is using which could lead to further insights into the security of each VPN application.

7.1.3 HTTPS Interception

The research was primarily focused on common security issues that can lead to a reduction in privacy for users through leaks and poor protocol implementation, however, it could be interesting to perform HTTPS interception to further determine what personally-identifiable information is being sent to both the VPN developers and third-parties (such as advertisers).

This work would align well with research into the information disclosure to analytics and advertising platforms in mobile applications. This research performed HTTPS interception on popular apps in a variety of categories and found that (for the top 20 most common hosts), 80% of these belonged to advertising or analytics networks (Smart, 2016). Given that VPN's can increase user privacy, what data could potentially be disclosed to third-parties, and what are the privacy and security implications of this?

7.1.4 Analysis of App Rankings vs Security

Given the variety of categories on the app store that VPN apps fall under, it could be interesting to compare the results from this testing against the apps average user reviews and rankings in the applications' respective categories. Are users who are reviewing apps well aware that the apps potentially have security issues?

8 References

- Ikram, M., Vallina-Rodriguez, N., Seneviratne, S., Kaafar, M.A., Paxso, V. (2016) ‘An Analysis of the Privacy and Security Risks of Android VPN Permission-enabled Apps’. doi: 10.1145/2987443.2987471.
- Perta, V.C., Berbera, M.V., Tyson, G., Haddadi, H., Mei, A. (2015) ‘A Glance through the VPN Looking Glass: IPv6 Leakage and DNS Hijacking in Commercial VPN clients’. doi: 10.1515/popets-2015-0006.
- Ren, J., Lindorfer, M., Dubois, D.J., Rao, A., Choffnes, D., Vallina-Rodriguez, N. (2017) ‘Bug Fixes, Improvements, ... and Privacy Leaks’.
- Paxson, V. (1999) ‘Bro: A System for Detecting Network intruders in Real-Time’.
- Smart, I. (2016) ‘An Investigation into the Information Disclosed by Analytics and Advertising in Mobile Applications’.
- IETF (2000) RFC2764. A Framework for IP Based Virtual Private Networks. Available at: <https://tools.ietf.org/pdf/rfc2764.pdf> (Accessed: 28th February 2018).
- IETF (1999) RFC2661. Layer Two Tunnelling Protocol “L2TP”. Available at: <https://tools.ietf.org/pdf/rfc2661.pdf> (Accessed: 28th February 2018).
- IETF (2001) RFC3193. Securing L2TP using IPsec. Available at: <https://tools.ietf.org/pdf/rfc3193.pdf> (Accessed: 28th February 2018).
- IETF (1999) RFC2637. Point-to-Point Tunneling Protocol (PPTP). Available at: <https://tools.ietf.org/pdf/rfc2637.pdf> (Accessed: 28th February 2018).
- IETF (2014) RFC7296. Internet Key Exchange Protocol Version 2 (IKEv2). Available at: <https://tools.ietf.org/pdf/rfc7296.pdf> (Accessed: 3rd March 2018).
- IETF (2016) RFC7858. Specification for DNS over Transport Layer Security (TLS). Available at: <https://tools.ietf.org/pdf/rfc7858.pdf> (Accessed 12th March 2018).
- IETF (2005) RFC4033. DNS Security Introduction and Requirementss. Available at: <https://tools.ietf.org/pdf/rfc4033.pdf> (Accessed 4th April 2018).
- Golden Frog (no date) Privacy Policy. Available at: <https://www.goldenfrog.com/privacy> (Accessed 28th February 2018).
- dnsleaktest.com (no date) What is a DNS leak and why should I care? Available at: <https://www.dnsleaktest.com/what-is-a-dns-leak.html> (Accessed 3rd March 2018).
- Data & Marketing Association (2015) Data Privacy: what the consumer really thinks. Available at: https://dma.org.uk/uploads/ckeditor/Data-privacy-2015-what-consumers-really-thinks_final.pdf (Accessed 4th March 2018).
- Apple (no date) iOS Deployment Reference - VPN overview. Available at: <https://help.apple.com/deployment/ios/#/ior9f7b5ff26> (Accessed 28th February 2018).

Hong Kong Government Infosec (2008) VPN Security. Available at: <https://www.infosec.gov.hk/english/technical/files/vpn.pdf> (Accessed 3rd March 2018).

Information Commissioner's Office (ICO) (2012) What is personal data? - A quick reference guide. Available at: <https://ico.org.uk/media/for-organisations/documents/1549/determining-what-is-personal-data-quick-reference-guide.pdf> (Accessed 5th March 2018).

The VPN Guru (2018) VPN Protocols Explained and Compared. Available at: <https://thevpn.guru/vpn-protocols-explained-info-compare/> (Accessed 6th March 2018).

Cisco (2008) How Virtual Private Networks Work. Available at: <https://www.cisco.com/c/en/us/support/docs/security-vpn/ipsec-negotiation-ike-protocols/14106-how-vpn-works.pdf> (Accessed 6th March 2018).

IETF (no date) SSL VPNs: An IETF Perspective. Available at: <https://www.ietf.org/proceedings/72/slides/saag-4.pdf> (Accessed 8th March 2018).

IVPN (no date) Is using L2TP/IPSec with a public pre-shared key secure?. Available at: <https://www.ivpn.net/knowledgebase/160/Is-using-L2TPorIPSec-with-a-public-pre-shared-key-secure.html> (Accessed 8th March 2018).

Apple (2016) Getting a Packet Trace. Available at: https://developer.apple.com/library/content/qa/qa1176/_index.html#//apple_ref/doc/uid/DTS10001707-CH1-SECIOSPACKETTRACING (Accessed 12th March 2018).

The Hacker News (2017) Google to Add "DNS over TLS" security feature to Android OS. Available at: <https://thehackernews.com/2017/10/android-dns-over-tls.html> (Accessed 12th March 2018).

Android Open Source Project (no date) DNS over TLS commitments. Available at: [https://android-review.googlesource.com/q/topic:dns-dev-opt+\(status:open+OR+status:merged\)](https://android-review.googlesource.com/q/topic:dns-dev-opt+(status:open+OR+status:merged)) (Accessed 12th March 2018).

White, K. (2015) [Twitter] **insert date here**. Available at: <https://twitter.com/kennwhite/status/570062025641951232> (Accessed 12th March 2018).

TunnelBear (2017) TunnelBear Completes Industry-First Consumer VPN Public Security Audit. Available at: https://www.tunnelbear.com/blog/tunnelbear_public_security_audit/ (Accessed 12th March 2018).

Cure53 (2017). TunnelBear Security Assessment Summary 07.2017. Available at: https://cure53.de/summary-report_tunnelbear.pdf (Accessed 12th March 2018).

APNIC (2018). Peak DNSSEC? Available at: <https://blog.apnic.net/2018/02/26/peak-dnssec/> (Accessed 5th April 2018).

Business Insider (2015). A wildly popular Google Chrome extension was being used as a giant bot-net. Available at: <http://uk.businessinsider.com/hola-used-for-botnet-on-chrome-2015-5> (Accessed 5th April 2018).

Wall Street Journal (2017). Facebook's Onavo Gives Social-Media Firm Inside Peek at Rivals' Users. Available at: <https://www.wsj.com/articles/facebooks-onavo-gives-social-media-firm-inside-peek-a>

(Accessed 5th April 2018).

Onavo (2013). Privacy Policy. Available at: http://www.onavo.com/privacy_policy/#UseOfInformation (Accessed 5th April 2018).

The Register (2017). VPN logs helped unmask alleged 'net stalker, say feds. Available at: https://www.theregister.co.uk/2017/10/08/vpn_logs_helped_unmask_alleged_net_stalker_say_feds/ (Accessed 6th April 2018).

PureVPN (2016). PureVPN's Privacy Policy. Available at: <https://www.purevpn.com/privacy-policy.php> (Accessed 6th April 2018).

App Developer Magazine (2016). How Apple's mandatory iOS App Transport Security (ATS) change will affect you. Available at: [https://appdeveloper magazine.com/4664/2016/11/30/how-apple's-mandatory-ios-app-transport-security-\(ats\)-change-will-affect-you/](https://appdeveloper magazine.com/4664/2016/11/30/how-apple's-mandatory-ios-app-transport-security-(ats)-change-will-affect-you/) (Accessed 6th April 2018).

Scott Helme (2018). Alexa Top 1 Million Analysis - February 2018. Available at: <https://scotthelme.co.uk/alexa-top-1-million-analysis-february-2018/> (Accessed 6th April 2018).

Let's Encrypt (2018). ACME v2 and Wildcard Certificate Support is Live. Available at: <https://community.letsencrypt.org/t/acme-v2-and-wildcard-certificate-support-is-live/55579> (Accessed 6th April 2018).

Apple Developer (no date). NEVPNProtocolIPSec. Available at: <https://developer.apple.com/documentation/networkextension/nevpnprotocolipsec> (Accessed 6th April 2018).

Apple Developer (no date). NEDNSSettings. Available at: <https://developer.apple.com/documentation/networkextension/nednssettings> (Accessed 6th April 2018).

Apple Developer (no date). NEIPv6Route. Available at: <https://developer.apple.com/documentation/networkextension/neipv6route> (Accessed 11th April 2018).

Apple Developer (no date). NEIPv6Settings. Available at: <https://developer.apple.com/documentation/networkextension/neipv6settings> (Accessed 11th April 2018).

Quad9 (no date). About Quad9. Available at: <https://www.quad9.net/about/> (Accessed 6th April 2018).

Cloudflare (2018). Introducing DNS Resolver, 1.1.1.1 (not a joke). Available at: <https://blog.cloudflare.com/dns-resolver-1-1-1-1/> (Accessed 6th April 2018).

Digital Guardian (2015). The History of Data Breaches. Available at: <https://digitalguardian.com/blog/history-data-breaches> (Accessed 6th April 2018).

Apple (no date). List of available trusted root certificates in iOS 11. Available at: <https://support.apple.com/en-gb/HT208125> (Accessed 10th April 2018).

Google Trends (no date). VPN. Available at: <https://trends.google.com/trends/explore?date=2010-11-04%202018-04-11&q=VPN> (Accessed 11th April 2018).

PC Mag (2018). You need a VPN, and Here's Why. Available at: <http://uk.pcmag.com/privacy/88655/feature/you-need-a-vpn-and-heres-why> (Accessed 11th April 2018).

Goldenfrog (no date). VPN Protocols. Available at: <https://www.goldenfrog.company/vyprvpn/features/vpn-protocols> (Accessed 12th April 2018).

eTutorials (no date). RSA Encrypted Nonces Overview. Available at: <http://etutorials.org/Networking/Cisco+Certified+Security+Professional+Certification/Part+III+Virtual+Private+Networks+VPNs/Chapter+11+Cisco+IOS+IPSec+Certificate+Authority+Support/RSA+Encrypted+Nonces+Overview/> (Accessed 18th April 2018).

Pitts, S (2003). An Overview of Digital Certificates and How They Are Used in VPN Authentication. Available at: <https://www.giac.org/paper/gsec/2711/overview-digital-certificates-vpn-authentication/104625> (Accessed 18th April 2018).

The Register (2017). Hotspot Shield VPN throws your privacy in the fire, injects ads, JS into browsers ? claim. Available at: https://www.theregister.co.uk/2017/08/07/hotspot_shield_deceives_with_false_privacy_promises_complaint_claims/ (Accessed 19th April 2018).

Sockpuppet (2015). Against DNSSEC. Available at: <https://sockpuppet.org/blog/2015/01/15/against-dnssec/> (Accessed 5th April 2018).

Irvine, R. (2018) 'Stay 100% Anonymous VPNs The Ultimate Guide', WebUser, 21 February - 6 March (Issue 443), p. 40-46.

9 Bibliography

Ren, J., Rao, A., Lindorfer, M., Legout, A., Choffnes, D. (2016) ‘ReCon: Revealing and Controlling PII Leaks in Mobile Network Traffic’. doi 10.1145/2906388.2906392.

Appelbaum, J., Ray, M., Koscher, K., Finder, I. (no date) ‘vpwns: Virtual Pwned Networks’.

Pitts, S. (2004) ‘VPN Aggressive Mode Pre-shared Key Brute Force Attack’.

Cornell (no date). Towards cross device digital tracking via a VPN architecture. Available at: <http://www.cs.cornell.edu/~fnoeke/papers/vpn.pdf> (Accessed 11th April 2018).

Security Stack Exchange (2012). If DNSSEC is so useful, why is its deployment non-existent for top domains?. Available at: <https://security.stackexchange.com/questions/21121/if-dnssec-is-so-useful-why-i> (Accessed 5th April 2018).

10 Appendices

10.1 Appendix 1: PCAP Analysis Script

```
#!/bin/bash

#set -x
#This is for debugging

echo "-----"
echo "This program will search a PCAP file against a wordlist, show a list of unique IP addresses
and check if the PCAP file contains ESP packets"
echo "Enter filename to compare against wordlist"
read -p "Filename: " -e SEARCHTERM
echo "-----"

echo "-----"
echo "KEYWORD MATCHES:"
echo "-----"

if grep -i -a -f wordlist.txt ${SEARCHTERM} | grep -ivf exclusions.txt; then
echo "-----" > output_${SEARCHTERM}.txt
echo "KEYWORD MATCHES:" >> output_${SEARCHTERM}.txt
echo "-----" >> output_${SEARCHTERM}.txt

grep -i -a -f wordlist.txt ${SEARCHTERM} | grep -ivf exclusions.txt >> output_${SEARCHTERM}.txt
else
echo "No keyword matches" > output_${SEARCHTERM}.txt
fi

echo "-----"
echo "UNIQUE IP ADDRESSES:"
echo "-----"

echo "-----" >> output_${SEARCHTERM}.txt
echo "UNIQUE IP ADDRESSES:" >> output_${SEARCHTERM}.txt
echo "-----" >> output_${SEARCHTERM}.txt

tshark -r ${SEARCHTERM} -T fields -e ip.dst ip.src | sort | uniq
tshark -r ${SEARCHTERM} -T fields -e ip.dst ip.src | sort | uniq >> output_${SEARCHTERM}.txt

echo "-----"
echo "CHECKING FOR ESP PACKETS..."
echo "-----"

echo "-----" >> output_${SEARCHTERM}.txt
echo "Contains ESP Packets?" >> output_${SEARCHTERM}.txt
echo "-----" >> output_${SEARCHTERM}.txt

if tshark -r ${SEARCHTERM} -O ESP | grep -i -q "Encapsulating Security Payload"; then
echo "Contains ESP packets: Using IPSec"
echo "True" >> output_${SEARCHTERM}.txt
else
echo "Does not contain ESP packets"
echo "False" >> output_${SEARCHTERM}.txt
fi

echo "-----"
echo "Results printed to output_${SEARCHTERM}.txt"
```

10.2 Appendix 2: Wordlist for Search Script

password	iPhone	card
pass	iOS	credit
pw	11.2.2	debit
user	GPS	source
username	lat	IP
email	latitude	dns
hi@jack.lu	lon	eap
junk@jack.lu	longitude	id
vpn@jack.lu	geo	uid
mac	location	udid
imei	address	
ismi	payment	

10.3 Appendix 3: Exclusions for Search Script

user-agent	accept-encoding	content-encoding
------------	-----------------	------------------