University of Birmingham

School of Computer Science
Final year project

VPN over HTTP

Project Report

Author: Daniel Jones (1427970)

BSc Computer Science

supervised by
Dr Ian Batten

Submitted in conformity with the requirements
for the degree of Bsc Computer Science
School of Computer Science
University of Birmingham

VPN over HTTP

Daniel Jones

March 27, 2018

Abstract

Problem: VPN traffic is easy to block, and commonly blocked on free public networks.

Solution: HTTP traffic is rarely blocked, so encoding data into HTTP traffic is one possible way to bypass filtering and blocking on public networks.

Conclusion: It is possible to encode data in such a way that it is difficult to detect that this has been done.

All code that was developed can be found at:
`https://git-teaching.cs.bham.ac.uk/mod-ug-proj-2017/dgj470`

Keywords: VPN, HTTP, Tunnelling, Obfuscation, Steganography

Acknowledgements

I would like to thank my supervisor, Dr Ian Batten for support and guidance throughout the project.

Additionally, I'd like to thank my housemates, friends and family for support throughout both this project and my degree as a whole.

Contents

1 Introduction

The aim of this project is to be able to tunnel data, and by extension operate a VPN over the HTTP protocol. There are multiple reasons why this would want to be done, for example:

- To access services that are blocked by the current network
- To hide the fact that blocked services are being accessed
- To maintain privacy regarding services that are being accessed

This project enables connections to be made and transmit data only over the HTTP protocol, and provide all of the benefits highlighted above.

2 Existing Work

In this section, I explore some related works, and discuss some literature surrounding Steganography, HTTP, DNS tunneling and detecting tunneled traffic.

There has been a lot of work done on the study of steganography, and using different protocols to hide data. The most work has been done on DNS, as it is often below the radar for firewalls and checking if data is being exfiltrated.

2.1 Image Steganography

Steganography is most commonly used for hiding data in unused or unimportant areas of data[1], and the most common form of data for this is images. This is because all of the colour data in an image is not required for a human to see it, and the human eye is very good at filtering out noise[1].

More advanced approaches to steganography can involve identifying redundant data in images[2] which can be better than changing the least significant bit in an image which can be detected by steganalysis[2].

Steganography that is hidden from computers and is hidden from people are quite different things, and can require quite different approaches. The real challenge is to hide data from both.[2].

2.2 DNS steganography

It is possible to hide data very easily in DNS requests, and this is called DNS Tunneling, and it is often used to get around firewalls and hide which websites are being accessed.[3] This paper highlights the point raised in the previous section, that it is very easy for a human to look at the data and see it's not normal, but non-trivial for a computer. The paper describes how the data is detected, and in doing so describes in depth how the data is encoded and tunneled. DNS tunneling as described in the aformentioned paper has a few advantages and disadvantages. The key advantage is that is can be used in locked down networks, as DNS traffic is often let out, but the main disadvantage is that data transfer is very slow with a lot of overhead.

2.3 HTTP protocol

The HTTP 1.1 Protocol[4] is a protocol that describes how data is sent to and from a client, and it describes many areas where data could be included, HTTP traffic can include: Images, HTML, CSS, Javascript, Binary files, and more.
All of which can be used to hide data.

The headers in HTTP are also a potential vector to hide data, as HTTP Headers are whitespace insensitive, and the order in which the headers are sent do not matter[4].

2.4 Detecting tunneled DNS traffic

There are a variety of ways to detect tunneled traffic, from entropy analysis to performing DNS requests[3]. Another way of performing the lookup is to do character frequency analysis[5]. Frequency analysis looks at the difference in frequency of letters in domain names/english words and in random data. It is similar to but not quite the same as entropy analysis, and it is also more effective[5].

3 Background Material

4 Specification

5 Design

6 Implementation

7 Testing

8 Project Management

9 Evaluation

10 Conclusion

11 Bibliography

References

[1] Johnson Neil F and Jajodia Sushil. Exploring steganography: Seeing the unseen. *Computer*, 31(2):26–34, Feb 1998.

[2] Provos N and Honeyman P. Hide and seek: an introduction to steganography. *IEEE Security & Privacy*, 99:32–44, June 2003.

[3] Greg Farnham. Detecting dns tunneling. *SANS Institute Reading Room*, Feb 2013.

[4] Fielding R, Irvine UC, Gettys J, Compaq/W3C, Mogul J, Compaq, Frystyk H, W3C/MIT, Masinter L, Xerox, Leach P, Microsoft, and Berners-Lee T. Hypertext Transfer Protocol – HTTP/1.1. RFC 2616, RFC Editor, June 1999.

[5] Born Kenton and Gustafson Dr. David. Detecting dns tunnels using character frequency analysis. *CoRR*, abs,1004.4358, 2010.

12 Appendices