Adversarial Analysis of Internet Censorship Systems

Oliver James Farnan

Balliol College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

August 2019

To Elladora, who changed the course of my life.

Acknowledgements

This work has been shaped and written by the past twelve years. During this time I have taken inspiration and encouragement from a host of friends and mentors, and I would like to acknowledge their support.

To most obvious influence to this work is Joss Wright, whose shared values led me towards the path of censorship research. Whether we're talking about weird data science or defending a run on my server, our constant meetings have motivated me to take the next steps towards whatever I've been working on. Productivity is important, but friendship is more important, so thank you.

To Andrew Martin for carrying the CDT on his shoulders and giving strategic direction to my stamp collecting. You've taught so much to so many of us, in much more important things than cyber security.

The biggest boost I ever got was from Liz Sokolowski, without whom I would never have considered academia.

I am grateful to Alex Darer for providing as close as you can to a colleague during a long and lonely Doctorate. It has been been good having someone to talk to about all of this when everyone else is sick of hearing about it.

Our work has been interwoven from early days, and this would look very different if you weren't around.

My time here wouldn't have been the same without Florian Egloff and Rodrigo Carvalho. Oxford is lonelier now you've gone. I know I can turn to you in the future, whether for pirates, diplomacy or socks. And of course Stefanye and Rachel and Kelly. It was good having you guys around.

Danny Rigby has allowed me to keep my wits sharp while pursuing a vanity project. You should really catch up yourself.

Lee Renaut who has spent his evenings teaching me silly things. And to Sam, Joe, Paulo, Andy, Joss, Oscar, Alex, Antony, Dan and Silent Dan, Julian, Liam and Rob, and everyone I'm forgetting. Thanks for helping me unwind in painful and uncomfortable ways.

David Hobbs and Maureen York for your untiring efforts to keep everything running smoothly. For the students here you're part administrators and part parents, so thanks with everything you've helped smooth over throughout the years. It is very much appreciated.

This work was supported by EPSRC through the Centre for Doctoral Training in Cyber Security, University of Oxford.

Abstract

Many countries perform internet censorship in an attempt to control the information and material that its population can access. With no standard way that this is performed, most countries build a bespoke platform to carry out their aims. Information targeted for censorship is set at the state level, and then this is enforced by technical controls implemented either directly by the national government or by Internet Service Providers.

The details of these systems are rarely made public. As they each target information on the internet in different ways it can be difficult to gather details on what exactly they block, or how exactly they do it. This lack of knowledge can lead to situations where we experience unknown behaviour on the internet.

Lack of understanding can lead to unpredictable behaviour on the internet. Network traffic can be interfered with in ways that wasn't intended. In some cases this can happen in situations where neither the sender or receiver are under the direct jurisdiction of the censorship regime affecting them.

Unlike surveillance which can be passive, for censorship an observable action must take place. This gives us an opportunity to study these actions. We can observe when network connections are reset, packets are dropped, or incorrect results are given to queries. These actions allow us to study how such systems are implemented, and the nature of the content that they're blocking.

In this work we focus on the technical implementation of these internet censorship systems. How have they been implemented, and how do they go about blocking unwanted content? We initially focus on the Great Firewall of China, but eventually branch out to look at other internet censorship systems.

We also look at censorship avoidance technologies. Users in censorious regimes turn to technologies such as Tor or VPNs in an attempt to access information they would not otherwise be able to access. We explore how users use these technologies to access censored information, and whether the use of these technologies themselves can be correlated with internet censorship.

Contents

List of Figures

Chapter 1

Introduction

The internet continues to grow in ad-hoc and complex ways, through the actions of different actors driven by different motives. The desires of these actors are diverse and at times in conflict: some actors want free speech, while others hope to censor information; some believe in privacy, while others aspire to surveil upon them; some push for stronger security, while others deliberately weaken it.

Although some of this conflict happens out in the open, with public debates and discussion, laws and regulation, much occurs behind closed doors. The internet is a technical entity, and even when laws are passed following public consensus, the technical details of their enforcement are often unavailable. Whether through deliberate subterfuge, some sense of security through obscurity, or simply because it is considered none of our business, these details often do not have the visibility they deserve.

As academics we want to study the internet; how people interact with it, what it is used for, and how it has grown over time. Ultimately, many of these are social questions

that cannot be answered through purely technical analysis. The internet has embedded itself so heavily into everyday use, that these questions lead us to answers much deeper than simply which packets are where, which protocol is most efficient, and which data is unable to pass a particular boundary.

We are in a difficult position when conflicts of the internet occur in private. The answers to our questions can be hidden, incomplete or obfuscated, preventing us from having ready access to the information necessary for correct or complete conclusions. There are times when to understand the internet and its relationship with people, we need knowledge of its constituent parts. When this knowledge is hidden, we are missing key parts of the whole.

We have no reason to believe that the internet will stop growing in complexity. Different actors pulling in different directions, combining with the increasing scale as the internet grows (regardless of metric: users, hosts, data) are leading to a system of increasing complexity. While this is the case, there is an equal need for growth and innovation in the observations and methods for performing them.

This is the research situation in which we find ourselves, and to which we hope we have contributed. We aim to have shone a light on some of these conflicts, their effects on users, and to have developed methodologies and techniques that others can use in the future. While we hope that our discoveries are useful in themselves, our ultimate goal is that our techniques are useful to others when applied to different environments and contexts. In chapter 2 we discuss many of the previous efforts to solve these problems, and in chapters 3, 4, 5 and 6 we provide our own approaches that we hope can drive this field forward.

1.1 Internet Censorship: Side-Channels and Adversarial Analysis

Our research has focused on analysing internet censorship systems, and the effects that they have on the internet. Many phenomena in networking are well documented and described, written out in RFCs or measured as metrics. We instead focused on those where this is not the case, and where direct observation is difficult.

When the subject under observation is obscured or hidden, it makes it difficult to generate useful and representative data to base research on. When we cannot rely on direct observation, we instead must focus on the consequences. By observing these consequences, we can infer information about what caused them.

Slightly different from inference, is the ability to use side-channels to glean information about a system. Whereas inference focuses on the differences in whether a system is present or not, side-channel analysis gives information on accidental leakage of information of a system, sometimes in response to direct interrogation.

Apart from censorship there are other actions that may interfere with the free flow of the internet. Surveillance aims to passively listen to traffic, but there are times when active interference is required. Network neutrality violations and commercial routing agreements can route traffic in unexpected ways based on its content or owner. Advanced traffic hijacking attacks such as QUANTUM [93] and the Great Cannon [66] deliberately break protocols, invisible to the casual user. Throughout our work we have described these kind of phenomena as internet anomalies.

It should be noted that unobservability is a property of perspective, and given that the internet is a man made system, in many situations someone will have knowledge of it. We focus on those cases where this knowledge is not known to the research community or the wider public. This may be because they are deliberately hidden, but that is not necessarily true for all anomalies. Similarly, while the public does not have direct observation of a given anomaly, with the right access to responsible nodes, analysis could be as simple as reading the configuration file. When direct access is impossible we must instead turn to adversarial analysis. We do not believe this sense of perspective devalues the exploration of such events.

As well as focussing on censorship techniques, we also want to increase the capability of analysis for these other types of internet anomaly. While simply describing our observations is in itself a useful contribution, our primary hope is for our techniques to be adopted by others who are exploring similar internet anomalies.

The beginning of our work focuses on censorship in China. As researchers we are starting to build a good picture of how this filtering is being performed, but there is still a large amount of uncharted territory regarding how it is directed, what effects the censors hope to achieve, and the impact it has on users within its sphere of influence.

Towards the end of this thesis we broaden into other censorship environments. China was a good starting point to become familiar with the state of the art observation and analysis of a particular anomaly, but there are others that deserve similar attention. We identified two key challenges to our work:

1) Firstly, much of the information we are interested in is deliberately hidden. This leads

to us having to make observations often with an incomplete understanding of a system. We do not have the design documentation or system configuration, and may have to make interpretations based on the limited data that we are able to observe.

The raw data for our work comes from both self-generated data, as well as data that has been generated and made available by others. There currently exists a glut of data from relevant projects in this field and little analysis of it. We believe it valid to explore the datasets offered by other projects, and these datasets give differing insights into the systems that we look at. Combining our self-observed data with data from other sources allows us to to provide a deeper analysis than would otherwise be possible.

At times, analysing hidden systems can lead to results that seem to conflict with each other, and it can be difficult to interpret which are correct. There are many observations published within the field of censorship where later research has found differing results. This is not surprising, especially given that the systems we are observing are bound to change over time, normally with no public announcement or press release. It is essential to understand that these observations can only ever represent a single snapshot in time, and that no work can comprehensively and exhaustively describe them.

2) Secondly, we have tried to introduce techniques that can be applied over multiple systems and networks, and are not unique to a single scenario. For example, while some of our work is specific for the GFW, we hope that the techniques we used to perform these analyses are also valuable in other contexts. As many of these systems are bespoke it can be difficult to identify approaches that can be applied to the gamut, but nevertheless the methods of studying them have more general use. Throughout our work we were conscious of the danger of limiting our study to the GFW, and tried to apply our

approach to other environments when we could.

An example of others' work that can be applied over multiple systems, is the work of Weaver *et al.* describing how to detect forged TCP reset packets [109]. TCP resets are ubiquitous throughout the internet. Although this research was initially focused on the GFW, describing how to detect forged TCP resets is useful over a broader field, and is not limited to censorship or the GFW.

1.2 Objectives

Firstly, we have tried to explore and describe some of the major censorship platforms on the internet. Of this, our initial example was the GFW. It was created deliberately, and affects hundreds of millions of users of the internet in both predictable and unpredictable ways.

As a complex system with emergent behaviours, the internet often acts in unpredictable ways. Systems can be installed without the full consequences being adequately explored or predicted. An example of this, and one which fits within our analysis of the GFW, is collateral DNS filtering. It has been observed that DNS requests will at times be poisoned even if neither the DNS resolver, nor the initial DNS server, reside under the poisoner's influence [3]. When a DNS server does not know the answer to a received request, often it will recurse to receive a response from an authority that does[1]. These recursive requests can travel the internet via unexpected paths, and through parts of the internet subject to filtration or poisoning, resulting in the initial request being impacted

[1] This depends on both the specification of the request, as well as the configuration of the server.

by an entity it would not normally be subject to.

Secondly, we aimed to provide knowledge of how these anomalies affect the use and users of the internet. While describing phenomena is useful, we want to open up a second level of analysis beyond simply stating what we have observed. We have explored the data available for patterns, trends and new areas of investigation, and we hope this can lead to exploration of non-technical aspects.

As mentioned we have combined self-generated data with data provide by third parties. Throughout out research we performed a large amount of internet scanning and mapping, testing for responses to certain requests and traffic. This has resulted in a large amount of data to analyse. Once we have this data we cross-referenced it against other data sets offered by third parties. Data gathering is cheap, and an increasing number of organisations are gathering and store it for future analysis. Organisations such as OONI [38] and DNSDB [37] have made some of this data available, and we have used it in conjunction with our own scanning data.

Thirdly, we aimed to provide methods for others to study anomalous behaviour and its consequences. Where we are able to observe and describe anomalies, we hope our published methods can be put to use in other parts of internet analysis. An inspiration for this is Ensafi's work detecting intentional packet drops, which shows that similar properties and identifying attributes are shared between different censorship systems [32].

1.3 Why China?

China is the most populous country in the world. It is controlled politically by the Community Party of China, who maintain a tight grip on the distribution of information to their population. This control extends to public speech, books and journalism, but also of course the internet. When researching internet censorship, it is impossible to ignore China and its Great Firewall. Much of the earliest internet censorship research is focused on China, and the body of peer-reviewed work focusing on China is far larger than on any single other country.

This large research corpus gives us a greater understanding of the GFW than any other censorship platform. It covers both technical implementations, the focus on this thesis, as well as the policy decisions that shape its targeting. Although worth studying China for its own reasons, this understanding has given insights into other censorship platforms. The GFW is not an off-the-shelf system, but techniques adopted in China have also been observed elsewhere.

While this work has not intentionally focused on China, much of the published research it is based upon looks primarily at the GFW. This was not the intention starting the research, but China dominates the work described in Chapters 3 and 4. Chapter 5 and 6 are more open scope, and although they include China, also compare censorship platforms in other countries.

One interesting outcome of the focus on Chinese censorship is that it has given us the ability to watch it change over time. Academics have watched it grow from nascent

ideas to a comprehensive and well developed platform for culling the spread of unwanted information. Our work has contributed to that, and helped document this change.

1.4 Do Censors Care?

Censorship is by its nature an overt action. If content is not blocked we can see it, but if it is blocked we cannot see it. Contrast this to surveillance, which although is often performed by similar actors, can be harder to ascertain whether it has taken place. As long as we have the capability to step outside of a censorship platform's control – for example by accessing it from a different country – then we have the power to observe the difference between the censored and uncensored material.

Perhaps because of this, censorship platforms seem to do little to evade or hamper efforts to research them. Censorship techniques do not appear to go to great lengths to differentiate between traffic created by security researchers versus their target populations. This openness makes censorship systems an ideal system to study when it comes to network manipulation. Censorship platforms have high-performance requirements, requiring the capability to block or alter content quickly and often invisibly and seamlessly to an end user. It is probably this performance requirement that encourages 'good enough' censorship systems, rather than trying to distinguish between regular internet use and the scans of researchers.

1.5 Approach

Our approach has been to study different censorship systems and compare them with each other in an attempt to identify similar properties. We have a deliberate focus on censorship systems, which can be difficult to observe directly, thus leaving us to rely on inference for their analysis as opposed to direct observation. Previously the state of the art for this field often focused on a single censorship system at a time, and there is less work which attempts to look at these systems as a whole. This has changed somewhat during the duration of our research, and our work has contributed in a small way to that shift. We hope that in the future this perspective shift will assist in the detection, observation and analysis of future systems, whether they are internet censorship or related internet anomalies.

We consider this thesis successful if our work assists or influences the future analysis or censorship systems or other internet anomalies. We believe we have been successful in this, as the majority of the work within this thesis is published and receiving citations and discussion by peers in related fields. We hope that discussion is not limited to the findings of our papers, but that our techniques and approaches will be used and developed by others.

Conversely, if our impact is limited to describing the current state of systems we have not filled one of our key goals. Although our results may produce interesting findings, it will be disappointing if the techniques are not used and developed in future research.

1.6 Completed Work Supporting this Thesis

During the course of our work we have made several investigations into censorship and network inference. Firstly, we studied the GFW for signs of Deep Packet Inspection (DPI). We injected content from queries that we knew would be banned (such as HTTP and DNS requests containing banned keywords) into other types of internet traffic. We found that these requests were not filtered, indicating that the GFW's filtering checks the type of packet (at the transport layer) before deciding whether the packet should be filtered. Then we mapped out the Autonomous Systems (AS) and nodes that make up the Chinese internet, and were affected by the GFW. We performed mapping attempts to every /24 network range that exists within China, and identified which AS and ISP (or other owner) they belonged to. We observed the paths and routes into China to help us with future attempts at mapping where filtering was occurring. This was an exploratory project to gain an understanding of the GFW and its capabilities. This research makes up the first part of chapter 3.

Secondly, we performed an analysis of GFW DNS Poisoning, based on the ability to send requests to North Korean internet address ranges. While APNIC [5] has assigned address space to North Korea, we found that this space is only routable via the Chinese internet. We identified some DNS servers in North Korea and sent them requests that the GFW was expected to filter. We observed that although traffic to some North Korean network ranges were filtered (primarily those hosted within China itself), many of the requests were successful (with no poisoning attempt by the GFW). This paper makes up

the second part of chapter 3.

The next piece of work was looking at the DNS cache state of DNS servers within China, ignoring the responses from the GFW itself. This paper describes how Chinese DNS poisoning corrupts the DNS cache of DNS servers in China, and then makes observations about the DNS results returned. We then searched for historic use of the responses given by the GFW, and observed how the amount of censored domains changed over time. This was published in the Workshop on Privacy in the Electronic Society at CCS 2017, and makes up the bulk of chapter 4.

After that we worked on an approach to estimate frequency of access to a particular domain based on how frequently DNS records are checked. This used a technique called DNS cache snooping. We performed DNS cache snooping against the internal DNS servers of privacy providing DNS servers, and observe the frequency with which domains are accessed on them. We then repeated the process for domains that we know are censored in certain countries: China, Indonesia, Iran, and Turkey. This process built upon the lists generated by our collaborative work described below. This work was published in the International Workshop on Traffic Measurements for Cybersecurity at IEEE S&P 2019, is is described in chapter 5.

1.7 Collaborative Work

There are three collaborative papers supporting this thesis. The first paper sets out to address a need for a censorship analyser based on Tor user data identified by other researchers in the field [112]. Our exploration of this problem has led us to the use of

Principal Component Analysis (PCA) [53], a statistical technique to link potential correlated variables. We attempted to use Tor connection figures as a measure for Tor's national connectivity, and use that to detect broader censorship events. Using PCA we provided a measure of global censorship activity, based on Tor usage statistics and divided by country [97]. Our approach takes the time series for Tor users at any given time, and observes for national variance against the global norm. Using this approach we are able to observe any national abnormalities in Tor usage, and ignore global trends that cause spikes or dips in the time series. This work was published in 2018 Proceedings of the 10th ACM Conference on Web Science.

The second paper describes a technique to find censored content on the internet, and generate lists of content that is blocked. During the course of our research we found that while several organisations produced lists of censored web domains, they were curated by hand and made no attempt to be comprehensive. To remedy this we wanted to introduce a systematic approach for creating and maintaining such lists. Our method takes a seed list of filtered websites and analyses the content on them to break out the relevant keywords. We then run these keywords through search engines to find other pages that contain them, and then test if these pages are themselves censored. This is repeated as an iterative process. This work was published in 2017 Network Traffic Measurement and Analysis Conference.

The third paper built upon the second, and expands the technique to cover three additional countries: Indonesia, Iran, and Turkey. We introduced new methods to test if a website is filtered in a particular country, which allowed us to apply the technique from the previous paper to other countries. We used this approach to generate filtered

domain lists an order of magnitude larger than any other publicly available lists. This work was published in 2018 Proceedings of the 10th ACM Conference on Web Science.

Overviews of these papers can be found in chapter 6.

Chapter 2

Background - the story so far

2.1 Literature Review

The internet plays an increasingly important role in our day-to-day lives. In an attempt to streamline and convenience daily tasks, many actions that were previously carried out in the online world are being moved online. Internet shopping and banking; concepts that would be alien to most even 25 years ago, are now every day occurrences, with those that do not use them often being the exception rather than the rule.

As there is no overarching body governing the internet: governments, service providers and technology companies add their own idiosyncratic networks and bespoke services to the internet in an ad-hoc fashion. In order to offer these services a great deal of user information is collected and there are others out there who want to gain access to it.

But while these services provide increased convenience, they also open up new sur-

faces for actors to exploit. Actions that previously only existed in the real world, such as theft or surveillance, have developed online equivalents. To protect users and allow them to receive the potential benefit offered by the internet, some guarantee towards its security is necessary, and for the internet to thrive it requires the existence of strong and enforceable security measures.

Research has tried to keep pace with this changing internet. While the technologies it is built upon are often well understood, new challenges and outcomes occur when deployed on such a vast scale. Approaches that are well understood and documented in a local setting are not always applicable on the societal scope of the internet. Technologies which are developed and tested for localised use, before being deployed and depended upon at a scale that they were not originally envisaged for.

Internet research must also consider a key confounding factor over pure networking and computer science research: the impact of human interaction. The internet is both used and administered by people, with divergent goals and agendas, sometimes striving for different objectives, or at times even in direct competition with each other. This is especially true within the context of security, which involves protecting the users of the internet from each other.

Privacy is an aspect of this, with many users expecting or at least desiring their online activity to remain private, whilst others are dedicating time and resources trying to gather as much information about these same users and their actions as possible. Exploring these conflicts presents an array of difficulties for researchers. Rarely do all sets of actors publish their means, motives and successes, so indirect means of discovering and documenting these conflicts for analysis must be taken.

The security requirements of the internet, combined with its ad-hoc development, have led to a large amount of complexity. With different domains being owned and controlled by different actors, using different techniques, predicting behaviour on the internet can be difficult.

The increasing development and use of the internet has led it to becoming one of the most complicated man-made systems in existence, with many viewing it as a complex system where actions can lead to unpredictable behaviours which are difficult to predict or model [78]. When we are unable to directly observe these actions and their consequent effects, we must often infer their presence from the measurable effects of their interference.

The internet contains many different components interacting with each other in myriad ways. On a micro level most of these components are well understood (e.g. how an individual switch routes packets or what connections a firewall drops), but when placed in the context of their larger environment, some details may be missing (e.g. why the switch is configured the way it is or what bug is causing some unexpected behaviour). Furthermore, while some components are well understood by a few, these details are not revealed and made public (e.g. how an ISP performs packet shaping, or how a country performs filtering of undesirable content). In cases like these we must try to detect, identify and infer the behaviour taking place to gain an understanding of the larger system.

One of the fields that encourages this complexity is internet censorship. Internet censorship is an area where the filtering taking place is deliberate and targeted, yet the details of the filtering system are rarely made public. Users of the internet interact with

the filtering system and are affected by it, at times without even being aware that the content they are looking at or for is being filtered. Additionally, censorship clearly interacts with and is based on human and non-technical constraints, such as what material is being filtered and why (e.g. is it political? Why is it undesirable?), ensuring that research into it cannot be complete without also looking at factors outside the immediate network components enforcing these actions. Traditional analysis of censorship systems often only take one aspect of this into consideration, but a more holistic view must be adopted to fully understand what is taking place. A good overview of these techniques can be found in Access Denied [23] from 2008. In chapter 3, Tools and Technology of Internet Censorship, Murdoch and Anderson lay out the technical options that internet censors have available to them. Although now a decade old, the techniques described still cover the majority of internet censorship programmes.

One of the largest and most multifaceted internet filtering programmes, and one that we look at in depth, is taking place in China. The GFW is not a single type of filtering, but instead an umbrella term that covers a range of censorship techniques used within China, including both technical (e.g. DNS redirects, TCP RSTs) and non-technical (e.g. political pressure) means.

While there is a wide breadth of measures in place, King believes that GFW censorship is primarily aimed at preventing collective action within China [56]. In a serendipitous piece of research that was not initially aimed at censorship, King found that discussion of sensitive topics was often tolerated, but any discussion that could prompt collective action was consistently removed. Previously, exploration by Crandall et al. came to a similar conclusion, and argued that the GFW was not in fact a firewall, but

more akin to a panopticon, where users of China's internet felt they were under constant observation which in turn induced self censorship. Their analysis of the keywords that the GFW attempted to filter found that instead of trying to remove all 'harmful' communication, filtering was instead focused on ensuring that users felt they had to self censor. While there is some time between these papers (2007 for Crandall's piece, to 2013 for King's) they both argue that instead of trying to filter everything, a well thought out approach to censorship is in practice rather than an attempt to block all undesirable discussion.

On the technical side of things, two key approaches have been identified as working together to target web traffic. These are HTTP request filtering, and DNS poisoning. Clayton et al. [17] found in 2006 that when GET requests were made to web servers within China, if the requests contained banned keywords (for example '`falun`' or '`facebook`') then the requesting host and the web server would both be sent TCP RST packets. These typically reached the requesting host before the legitimate response, resulting in each host dropping the connection believing that the other host had sent the RST. This filtering was bidirectional, with HTTP requests both into and out of China checked for filtered content, and potentially receiving a poisoned response. They observed that this filtering could be circumvented by ignoring the TCP RST sent to each host.

A similar technique was discovered by Lowe et al. [63] in 2007, but instead of focusing on HTTP requests it was targeted at DNS requests. Lowe observed that DNS requests containing banned keywords were receiving incorrect DNS responses. In a similar manner to that discovered by Clayton et al. [17] with HTTP requests, an intermediary node

along the path between the requesting host and the server was injecting its own invalid response. This response caused the DNS resolver on the user's host to cache the wrong IP address for their request, and so instead of getting the website they wanted, made the request to a different IP address.

Somewhere around this time there appears to be a switch in methodology: where previously the GFW had attempted to filter each HTTP request containing banned keywords, it instead began to rely more heavily on DNS filtering. There are likely a number of reasons for this. Firstly, there are fewer DNS requests than HTTP requests to monitor. The task of monitoring each HTTP request and attempting to poison those with banned keywords is a much more resource intensive task than focusing on the relative bottleneck of the DNS system. Secondly, HTTP requests are made over TCP connections, while DNS requests (or at least standard DNS requests) are made over UDP. While UDP is stateless, TCP segments are sent within a TCP stream with attempts to guarantee its reliability. Significantly for the GFW's poisoning methodology is the inclusion of TCP sequence numbers. It was observed that when the GFW attempted to poison an HTTP connection (over TCP), it was forced to make a guess at the TCP sequence number that each host was expecting [17]. If this TCP sequence number was wrong, the hosts would assume that the segment was incorrect and drop it. Using this approach, it was observed that the GFW would attempt a scattershot approach covering many different TCP sequence numbers, in an attempt to guess the correct one, an often resource intensive and inefficient approach.

These two papers set the foundations and direction for much contemporary technical research into the GFW, and there was an increasing amount of analysis that followed on

from them. They led to a wide variety of topics, covering everything from the technical details of how the GFW filters unwanted content, to the social and political reasons that some content is filtered.

Of note is the work done by Anonymous (using the email address zion.vlab@gmail.com). These anonymous researchers have followed on from Clayton and Lowe by attempting to infer additional information about the GFW's technical information filtering with the limited information that can be observed. Their first paper expanded upon Lowe's [63] method of limiting the TTL values of DNS attempts made into China, in order to determine where the filtering was taking place [3]. They increased their analysis to cover all the AS within the internet, and found that there were instances where DNS requests were being collaterally poisoned. They observed situations where although neither DNS resolver nor DNS server were located within China, at some point their requests may have strayed into it. These requests were inspected by the GFW, and if they were found to contain filtered keywords (e.g. 'facebook'), the resolver was sent a response containing an invalid IP address.

There have been other attempts to identify where censorship occurs within the GFW. One of the first (and indeed one that the second Anonymous paper relies on) is by Xu et al., looking at identifying where censorship occurs within the GFW [118]. They break it down by AS, and sort by the owner. They found that the majority of DNS poisoning occurs on China's internet backbone, and that different regional ISPs have different approaches to censorship.

Taking it a step further, Anonymous attempted to identify and measure the devices performing DNS poisoning [4]. They built upon Lowe's work [63] and combined it with

the King method [47] to map out where DNS poisoning was occurring. They found that the majority of locations where DNS poisoning was taking place were within the border ASs of China's network, and primarily targeted at requests going into or out of China. As well as this, they performed a large scale evaluation of the domains that are filtered by the GFW. They attempted to resolve all listed Alexa domains (130 million individual domains) and found that of these; around 35 thousand were censored by the GFW. Through subsequent analyses they were able to identify exact terms that were filtered by the GFW. They then ordered a method for attempting to estimate the amount of requests a node dealt with, and provided this analysis for a single node.

Other researchers such as Wright approached mapping GFW censorship from a different angle. Instead of attempting to base it by where on the network the filtering was occurring, Wright focused on the different DNS responses that were received [116]. He found that responses varied depending on where in China the response was intercepted, and found evidence for a decentralisation of filtering based on a centrally coordinated policy. This evidence indicates that it is wrong to look at the GFW as a single filtering device, but to think of it as different entities' interpretation and implementation of the desires of the central policy.

Of the DNS based censorship observed, there are at least two distinct types used within China. The first approach, as discussed above, is caused by filtering devices sending poisoned DNS responses to requests containing banned keywords. The local host's DNS resolver makes the request and receives two or more DNS responses: one from the legitimate DNS server, and one from the GFW's poisoning devices. This is the technique that has been most heavily studied in the current literature. The second approach is that

the DNS servers in China are themselves poisoned. This has not been studied in depth by other researchers, and is something we explore later in this thesis.

In April 2015 a new action has been observed by the GFW, one that is being referred to as 'The Great Cannon' [66]. The Great Cannon is a Distributed Denial of Service attack, where users of Chinese internet are given incorrect responses to web HTTP requests. Instead of receiving the web content they want, users receive JavaScript code which makes further requests to other sites. The combined traffic of these multiple requests, spread over a great many users, has been enough to prevent the targeted web servers responding to legitimate users

At first this appears to have nothing in common with previous GFW activity, but upon analysis this practice's modus operandi is similar to how it handles both HTTP filtering and DNS poisoning. These three approaches work by observing a request through China's internet, and sending a false or incorrect response to this request. While there are many other ways the GFW could filter content (e.g. IP blacklisting, blackholing, dropping packets, redirecting to a warning page, disconnecting users, etc.) these three approaches are based on injecting a trusted response to a users' request, over application layer protocols.

One observable difference between the GFW and the Great Cannon is that it is believed that the Great Cannon sits inline with the internet connection [66]. That is, whereas the GFW observes network traffic from elsewhere in the AS, and is not directly responsible for forwarding network packets, the Great Cannon exists on a node directly in-line with internet traffic into and out of China. Whilst the GFW poisons traffic with fake TCP RST and DNS responses, the Great Cannon has the ability to entirely prevent

packets reaching their destination.

The Great Cannon has been identified in two separate locations, and in both instances it was co-located with GFW filtering nodes. Additionally, it shares a similar TTL side-channel fingerprinting vulnerability, indicating that they may share some of the same source code. However, the Great Cannot is not optimised for censorship as the GFW is. The Great Cannon only attempts to filter the first packet in any connection, and any subsequent requests will pass through uninhibited

While no similar attacks have been observed previously (at least in complexity or scale), this approach shares characteristics with an NSA technique revealed by leaked internal documents. The NSA technique known as QUANTUM also allows packet identification and response, relying on a privileged position in the network. In an similar attack to the Great Cannon, QUANTUM allows the NSA to respond to HTTP requests with their own (incorrect) response. This has been used to target Tor users and harvest HTTP session IDs [93]. Like the GFW and Great Cannon, this is possible by having access to the backbone that routine internet traffic passes over. Unlike these techniques, it is a targeted approach, only affecting specific targeted users. The technology for doing this is simple, and this technique is trivial for any country (or other organisation) with the correct network vantage.

The GFW makes it difficult for Chinese internet users to access filtered material without expert knowledge, and often those who can circumvent it fear the consequences of being detected. One possible way around this is to use anonymisation services such as Tor. While Tor is a Western funded service, its authors intend for it to be available around the world, especially in countries suffering from repression of information [95].

Unsurprisingly, the GFW has countered user attempts to access Tor within China. In October 2011 it was observed that Tor bridges within China were being blocked and becoming unavailable minutes after their deployment.

Wilde [111] followed this up to discover what was causing this. He found that Tor bridges were being identified by their Tor client cipher list, and were subsequently scanned and blocked. In 2012 Winter and Lindskog followed this up with a much more in-depth analysis [114]. They found that after GFW DPI detected a cipher list that indicated that an IP and Port tuple was acting as a Tor bridge, an IP address would scan this IP and Port and try to initiate a Tor connection. If this connection was successful, this IP and Port tuple would be blacklisted. Once this IP and Port stopped responding as a Tor bridge, the IP and port tuple would be unblacklisted after 12 hours. They then identified the IP addresses responsible for this scanning, and mapped them back to ASs within China. They found that they came from the same three ASs that Xu et al. [118] had previously identified as being responsible for a large amount of GFW filtering. Winter followed on from this with a proposal to use a 'lightweight' censorship analyser for Tor [112]. To address this Winters suggests adding additional metrics onto Tor users' systems, to report any breaks in the connection.

Many of these papers have different outcomes, showing that the GFW has changed over time. One recent paper to explicitly explore this change is by Wang et al. from 2017 [108]. They describe a GFW that has adapted to defeat previous censorship evasion techniques, resulting in an arms race between the GFW and evasion techniques. Taking these changes into consideration they describe revised ways of bypassing the GFW, for both TCP and HTTP based filtering.

And so internet censorship within China continues to evolve. The extensive body of research shows how from the relatively crude method of attempting to filter HTTP requests containing banned keywords has become something more sophisticated and directed; where sensitive topics are tolerated as long as it does not incite collective action. It has allowed us to observe how filtering within China has changed over time. We have seen where new methods have been adopted and deployed, and in some cases, where old methodologies have been discontinued.

Network analysis techniques

Techniques such as those discussed above are often developed behind closed doors, and rarely published for discussion. The knowledge that the academic world has of them is largely based upon network analysis techniques that help us infer how such techniques work. While this list is by no means exhaustive, we wanted to introduce and discuss some of the key ones that have helped with the discovery of such hidden approaches so far.

Making observations about hidden network effects can be difficult, as often the only information available is a limited set provided by the effect that is under observation. This information is often deliberately hidden or obfuscated. In cases such as these, unable to directly observe what is occurring, we must instead infer it using the information that is available. This is possible when network manipulation has side effects which may be unnoticed (or uncared about) from the parties performing the manipulation.

A good example of this is analysis based on the IP Identification `IPID` field. This is a

field contained within the header of IP packets, used to allow IP packets to be fragmented and reassembled as they travel past nodes with different maximum transmission units (MTU). Historically these values have increased incrementally, allowing observers to discover information about the hosts performing network manipulation. This has been used by both Anonymous [4] and Roya Ensafi et al. [33][32] to discover information about the GFW that would otherwise likely be unknown. As this field has no effect for normal users, it otherwise goes unnoticed.

Another technique that has played a large part in the observation of the GFW, is the DNS-traceroute method, first proposed by Lowe et al. [63]. Lowe observed that the GFW attempts to poison DNS requests that pass through its domain, even if they are sent to an IP address that does not contain a DNS service. Lowe used this knowledge to send DNS requests with incremental IP TTL values, to limit how many hops they would travel from the DNS resolving host. Using this technique researchers have been able to map out the nodes and areas that are under this particular influence of the GFW (the DNS poisoning). Anonymous took this a step further, combining the IPID analysis with the King approach assist with censorship mapping [4].

The King method [47] uses recursive DNS queries to estimate the latency between two hosts. It is based on the observation that most hosts are close to their DNS name servers. By making a request for a illegitimate domain name, it relies upon the DNS server being observed performing a recursive lookup to the authoritative name server for the chosen domain. The difference in latency between responses direction from the queried name server, and those that had to make requests to the authoritative name server, is approximately the latency between the two DNS servers. Using this approach,

King can been used to estimate the latency between any two points on the internet. King is a good example of exploiting existing networking protocols to infer more about the layout of the internet than is explicitly given.

A technique common to many online censorship schemes is forgery of TCP RST packets. This is where an filtering agent sends TCP RST packets to one or both hosts, in an attempt to make either close the connection. Weaver et al. [109] have gone into detail on how to detect spoofed RST packets. They found that the most obvious giveaways, IPID and TTL fields, are not reliable as there is already a large amount of variation within regular network traffic. Instead they offer other metrics, as well as a tool, that can be used to reliably identify not just when RST packet injection is taking place, but also often fingerprint the specific type of device carrying out the packet forgery.

2.2 Common Filtering Techniques

There are many potential implementations for censors to choose from, and different countries have chosen to implement their censorship systems in different ways. Each country will have different considerations in what it wants blocked. Is web content enough, or do should it target other protocols (for example email or messengers)? If it is web, will unencrypted traffic be enough, or do they also want to look at the increasing amount of web traffic that is encrypted? Do any of the off-the-shelf platforms suffice, or do they need a bespoke system? Should censorship be surreptitious, or should public warnings be displayed when users try to access filtered material?

There are two parts that have to take place to censorship live internet traffic: firstly

the unwanted content has to be detected, and then something must close the connection or block the traffic. In order for something to be blocked, unwanted content must have previously been classified manually: for example a country choosing that it wants to block connections to `facebook.com`. This manual classification may include domains, topic content, individuals, or organisations. Once manual classification has taken place, there must be an automated step to identify this traffic on the network, as traffic moves too quickly for this to be an interactive process. This identification can be based on either packet headers or packet payload. Checking packet headers is normally a quicker and less intensive process than checking the packet payload. Anything below the packet header is considered 'deep packet inspection', although this term can be ambiguous due to the layered approach of network packets.

Once unwanted traffic has been identified on the network, the next step is to block the traffic. This can be done by tearing down the connection between the user client and the server, by redirecting the user, or simply by preventing traffic flowing between the two hosts.

There are other ways censors can remove content, for example by physically shutting down a web server or persuading its hosts to do so. Legal processes can force website owners or hosts to take down a site or its content, or social pressures may cause actors to self-censor. These methods are not automated real time processes, and are not the primary scope of this work.

Throughout the rest of this chapter we will cover some of the more common types of traffic filtering.

IP address blocking

Internet Protocol (IP) is a networking layer protocol that routes packets between two or more remote hosts. IP is used to connect different local networks together that each share a physical network, and make sure that requests from one host correctly reach the other hosts on the network. While the IP protocol is in use on networks that are not connected to the internet, all traffic over the internet is routed with IP.

One of the simplest way of blocking internet content is by blocking the IP address for the location it is hosted at. This is normally done at the ISP level, and there are a variety of ways it can be implemented[1]. While IP address blocking can be used to target websites it is not usually the ideal tool for this purpose. Websites can choose to change their IP addresses, and one website may be found on multiple IP addresses. The address infrastructure for websites is becoming increasingly complex with additional layers of content delivery networks (CDN), load balancers and virtualisation.

Instead, IP address blocking has been increasingly used to target persistent infrastructure which cannot switch addresses so easily. An example of this is Turkey blocking `8.8.8.8` and `8.8.4.4` in 2014 [123]. These are the IP addresses of Google's public DNS service, and internet users in Turkey were using them to evade state censorship of DNS. With the block in place users were no longer able to use Google's DNS service, and so many went back to using state controlled services.

[1] One of these is BGP hijacking, which we will discuss later in this chapter.

DNS manipulation

The Domain Name System (DNS) is the distributed system for converting human readable domain names (such as 'facebook.com') into machine routable IP addresses (such as 157.240.1.35). Almost every computer on the internet will have a DNS server configured, which is where they send requests for the IP addresses of domains they need to send traffic to. It is accessed by, and sits on top of, the IP layer.

DNS is also of the easiest targets for those wishing to block content. Most internet and web connections first require access to DNS to determine where to send requests, and censors are able to filter traffic by interfering with either the requests or servers that make up the internet DNS network.

DNS has several properties that make it an interesting target for censors. Firstly, it is an essential protocol and is required for normal web use, as well as other methods of information sharing on the internet. While DNS is used for name resolution for many protocols and services, web content (using HTTP and HTTPS) is often the focal target for censors. Other services such as email, chat, or media sharing are of interest to censors, but web content is the primary way which information is openly shared and accessed at scale. Although internet web content can be accessed without using DNS (by entering the IP address directly into a browser) this rarely takes place – the domain name is normally used, and translation from this domain name requires DNS.

Secondly, DNS is not encrypted. It is a cleartext protocol, making it straightforward for network operators to monitor, log, and when necessary, edit DNS requests and responses. As well as this DNS has little integrity protection. This means that censors can

respond to DNS requests with their own responses, controlling the content that users access.

Thirdly, DNS has a structured server architecture. Tampering with DNS requires manipulation of a finite number of servers. These servers can be targeted directly, but also propagate manipulation to other DNS servers. Due to this architecture there is a one to many relationship between DNS servers and web servers, as well as DNS servers and web users.

Many users use their default DNS server settings which are set by their ISP. At times this allows for a soft method of censorship: by forcing – whether legally or coercively – ISPs to block or redirect requests. These servers also provides a convenient central depository for logging and surveillance of historic web domain use.

DNS Redirection

DNS redirection is when the DNS record for a domain has been configured to point to a different IP address, and not the IP address found in the authoritative record for that domain. This non-authoritative IP address can perform a range of actions, including:

- Display a warning page. This page can optionally warn users that the original domain is filtered.
- Log access attempts and record the IP address of hosts attempting to access it.
- Perform a man-in-the-middle attack, for example by pretending to be the website hosted on the domain the user thinks they are connecting to.

- Record user HTTP session information such as username, password and cookies. These could be used for more granular user identification than simple IP addresses, or even compromising user accounts.
- Nothing, and simply drop the traffic. There may not even be a server there.

For example, a host attempts to access `facebook.com`. The host makes a DNS request to get the IP address from `facebook.com`. The DNS server is performing DNS redirection, and instead of returning the correct address of `157.240.1.35`, it returns `37.61.54.158`. The host now sends requests to `facebook.com` to `37.61.54.158` instead of `157.240.1.35`. Connections to `37.61.54.158` could be logged and dropped, both preventing users accessing `facebook.com` and recording the IP address and other details of hosts attempting to access it.

Iran is one country that uses DNS redirection in this way [7]. When users attempt to access a blacklisted site they instead get redirected to a centralised server that presents them with a warning page. As well as giving warnings to users trying to access unwanted content, the server that serves this page is capable of collecting information about the hosts which connect to it.

DNS Poisoning

DNS poisoning is a technique for corrupting DNS records by having them point to the wrong IP address. When hosts – whether they are servers or clients – make a request to a DNS server, a third party eavesdropper can respond with a poisoned response. This poisoned response contains a different IP address to the valid response. Future re-

[7]

Figure 2.1: Iranian DNS Redirection Landing Page

quests to this domain are sent to this wrong IP address. A DNS server that receives a poisoned response will cache this record, and so it will be propagated to other hosts making requests for that domain, including other DNS servers. In this way it is related to DNS redirection, and servers acting this way are a subset of servers performing DNS redirection.

DNS poisoning is done on a large scale in China [36].

HTTP Request or Response analysis

HTTP is the application layer protocol that sits on top of TCP to deliver web content to users. Users make HTTP requests to a web server, and the web server responds with an HTTP response. These responses contain higher-level web languages such as HTML, CSS, JavaScript, or other web content such as media. HTTP requests contain a path for

```
GET / HTTP/1.1
Host: www.wikipedia.org
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Upgrade-Insecure-Requests: 1
Connection: close
Cookie: WMF-Last-Access-Global=12-Aug-2019; GeoIP=GB:ENG:Oxford:51.72:-1.29:v4
```

Figure 2.2: HTTP Header

a specific Originally HTTP requests contained a path for a specific file or resource that existed within the directory structure of the web server, but over time this process has been abstracted and expanded to allow for dynamic individual content.

HTTP request or response analysis are techniques to attempt to identify or block web content based on the content of the HTTP headers. When users attempt to access web content they send an HTTP request header that contains fields such as the type of request, the path, the cookies, and information about the content. Likewise, when a server responds to HTTP requests it replies with an HTTP response header of similar fields. Censors can monitor these fields for content they desire to be filtered, and then choose whether take action to block that content.

The most critical part of the HTTP header for the the purposes of censorship is the path variable after the HTTP method. In typical use, this reveals to censors the specific information that users are attempting to access, such as a specific page, image or video[2].

By using the path header censors can target individual pages on a web domain, in-

[2]It should be noted that using more modern dynamic web design paradigms this may not necessarily be the case. For example the root path of the `facebook.com` domain displays a wide range of dynamic content, which differs depending on which user is making the request.

stead of having to filter the entire domain. This gives them a greater degree of granularity than they would by attempting to categorise by domain alone. For example, a censor could choose to block a specific undesirable story from `www.bbc.com`, and not be forced to block the entire domain.

Although this granular targeting is highly desirable by censors, it comes with drawbacks. Chiefly amongst these is that it only works if censors can see these headers, and by using SSL/TLS web providers hide these headers from censors' eyes[3]. This is a problem with the increasing use of HTTPS over HTTP for routine web connections. Many web pages now offer at least the option of an HTTPS version, and with this censors have no choice but to block the whole page.

As well as this, HTTP analysis is primarily a method just for finding material to filter, and does not offer an easy option for blocking the connection. To maintain the granularity of filtering this can be done with techniques such as TCP `RSTs`, but these methods have their own implementation difficulties. Often it is easier to simply block at the domain name or IP address level, but doing this removes the advantage of granularity that HTTP header analysis can provide.

Any analysis of the HTTP message body below the headers should be considered Deep Packet Inspection (DPI) and is discussed below.

[3]Server Name Identification (SNI) is a special case, and is visible outside of the encrypted tunnel. It is designed to allow multiple HTTPS domains to be hosted on the same web server, but censors can use it to identify the domain users are attempting to connect to, even if they are using HTTPS.

TCP connection interference

TCP is a transport layer protocol that sits between IP at layer 3 and, for web content, HTTP at layer 5. It is a connected protocol, meaning it requires two hosts to establish a connection with each other before data can be transferred between them. Once this connection has been established data can be sent back and forth between the hosts. As this connection must be set up and maintained in order for any data to be transferred, this offers censors some opportunities to interfere with it to filter out unwanted content.

One of the simplest way of interfering with the TCP connection is using the RST flag. By sending a spoofed packet with the IP address of the other host and a TCP RST flag, the receiving host will think the other host wants to close the connection. This method is commonly used by censors once unwanted traffic has been identified by other means [108].

BGP hijacking

Border Gateway Protocol (BGP) is the protocol used to route traffic between different Autonomous Systems (AS). Different AS declare that they have a route to part of the internet, and connected AS can access it through them. BGP is an older protocol and security was not a priority, and so this process is done largely on trust. At times, this trust has been exploited.

In the past there have been both accidental and deliberate internet anomalies caused by incorrect BGP route announcements. In 2008 Pakistan Telecom tried to block YouTube by announcing that they had a short path to its IP address [30]. As a result,

traffic intended for YouTube from all over the world was redirected to Pakistan, making the site unavailable for a large proportion of global users. In 2010 China Telecom announced routes that redirected 15% of the worlds internet traffic into China. This undoubtedly included a large amount of sensitive traffic, which would have been available to anyone listening on the Chinese network.

BGP blackholing can be an effective way of implementing an IP address block (as was tried in the Pakistan case). If an AS declares that they have a route to an IP address, nearby traffic will be sent towards that node, instead of towards the legitimate destination of the address. This technique also has other uses, such as blackholing traffic from distributed denial of service attacks (DDoS).

Deep Packet Inspection

Deep Packet Inspection describes network monitoring and filtering techniques that look below the packet headers that make up network traffic. Where the other techniques described typically focus on one specific protocol (and with censorship systems work by blocking that), DPI allows analysis of the packet payload. When employed for internet censorship this allows censors to analyse the actual content being accessed.

The term DPI does not specify that the network traffic will be blocked, only that an analysis of content is being performed. It is up to the censors and the specifics of the implementation to determine what action can be taken when sensitive information is identified. If the DPI system is acting as an inline network node along the path between the client and server it can simply choose to stop forwarding this traffic, preventing

connections between these two hosts. However DPI is often performed outside the direct path of network traffic (with port mirroring) which means that any filtering must be carried out by other means.

Even if the DPI system sits outside the direct network path and chooses not to take direct action to close the connection, actors have the option of logging and recording traffic details. This can include identifying features of both hosts such as IP address or domain name, as well as details of the content itself. Full packet capture can be used to create complete static offline replicas of the observed traffic.

As with all passive surveillance, DPI monitoring of this kind is difficult to detect. However, it is a mature and well understood approach with off-the-shelf and open source implementations available, and is common amongst organisations with high security requirements for their internal networks. As such, it is likely that many states run DPI censorship or surveillance platforms. There have been several known instances of states deploying off-the-shelf DPI platforms [52][61], as well cases where bespoke systems are believed to have been used [108].

The term DPI simply means that the layers under the packet header are being inspected. As different protocols layer the header for one layer within the payload of the layer above, the direct classification of DPI can be confusing, and dependant on the context. What counts as DPI in one context may not be considered DPI by others. Within this thesis we describe DPI to mean analysis of the payload section of network packets, below all protocol headers. We should point out that we have not always managed to clearly stick to this definition, and in some of our work searching for DPI by the GFW, we do so by injecting faux HTTP headers into packet payloads.

Increasing use of encryption

In 2010 the Firesheep [15] extension for Firefox was released, which acted as a packet sniffer for session cookies on the local network. At the time very little traffic was encrypted as standard, and most user interaction, including authentication, was performed in cleartext. This left it vulnerable for interception on the local network – or indeed anywhere the traffic traversed. While these attacks had always been possible with a little technical know-how, Firesheep trivialised this process, allowing users of the extension with little skill to quickly access others' accounts, simply by sharing the same local network.

While HTTPS had allowed the encryption of web traffic since 1994 it was still not used regularly in 2010. The Firesheep incident was the catalyst that changed that, and started a sudden shift towards encrypting more and more web traffic. Despite resistance at the time – with Facebook claiming HTTPS would slow the site down too much – within a few months an HTTPS option was offered, with a full and default roll-out the following year. Google soon followed suit for their default search engine. Indeed, the company has been one of the biggest proponent for further encryption of web content as standard. In 2014 they offered a ranking boost in their search engine for HTTPS pages, and in 2016 listing non-encrypted authentication pages as 'Not secure' in Chrome. This year the 'Not secure' label has been applied to all webpages not using HTTPS, even those just hosting static content.

While this is the most impactful example of the increase of encryption as standard for internet traffic, similar pushes have occurred for other protocols. Over the past few years

there has been a move for full end-to-end encryption between users of chat applications. WhatsApp has bought this feature to the masses, using the Signal protocol to protect its users [18]. In fact this push has not been limited to data in transit, and Apple have started enabling disk encryption by default on some of its devices [49].

While some of this push can be attributed to incidents such as Firesheep, the biggest influence has been the extent of nation state network monitoring revealed in the Snowdon leaks [45]. These leaks of the US National Security Agency showed the extent of the internet surveillance being performed by the US and other Five Eyes[4] governments. As a response internet organisations increased the speed of rollout for encrypting data in transit.

States have tried to counter this by attacking or exploiting the Public Key Infrastructure (PKI) that is used to provide users with asymmetric encryption keys. There have been examples of nation states either stealing certificates to use, or in some cases using their own certificates to perform man-in-the-middle attacks [60][59]. Encryption provided by HTTPS is only secure as long as the private keys these certificates use are not compromised, and nation states have a lot of ways of potentially accessing these.

One other important example in the context of this work is the move towards encrypting DNS traffic. At present in 2019 most DNS traffic is still unencrypted by default. In fact even when a user is accessing content encrypted through HTTPS, the user's machine is still likely making DNS requests for domains and subdomains in cleartext, giving network operators some idea of the content the user is accessing.

[4] An information sharing agreement between Australian, Canadian, New Zealand, UK and US intelligence organisations.

Over the past few years there have been several movements towards increasing use of cryptography for DNS. The first of these has been the specification and gradual implementation of DNSSEC [8]. DNSSEC does not attempt to encrypt the DNS requests or responses themselves, but signs DNS records to prevent tampering: providing integrity to DNS queries, but not confidentiality. This process could prevent attacks such as DNS poisoning which rely on responding to DNS queries with forged responses. As DNS poisoning is in widespread use for internet censorship this would force censors to find alternative ways to block content.

The second movement has been towards fully encrypting the DNS requests and responses themselves, providing confidentiality against network sniffers. There have been two notable implementations of this: DNSCrypt and DNS over HTTPS (DoH). For these approaches to work they have to be implemented on both the client making the request, as well as the server supporting it. While there have been some practical uses over the past few years, they are only recently starting to gain widespread adoption.

At the start of 2018 Cloudflare introduced their new DNS service: `1.1.1.1`. This service supports DNS over HTTPS, and Cloudflare have also provided Windows, Linux and macOS DoH daemons to ensure encrypted queries are the default. As well as this, Mozilla has partnered with Cloudflare to implement default DoH lookups for URLs accessed in Firefox. This feature is the first real step towards providing encrypted DNS queries for regular users[5].

[5]It is worth noting that there have been some disagreements as to whether the Firefox implementation is good for overall security as it is pushing all Firefox users towards the same DNS service.

Others?

While we have covered many of the common censorship techniques we have not tried to be exhaustive. We have chosen to describe those that directly apply to our work here, but there are other techniques and variations that we have not included. A good description of some of these are included in chapter 3 of Access Denied [23].

One trend that is worth mentioning is the increasing use of off-the-shelf censorship technologies. Technologies such as Bluecoat, Netsweeper, and SmartFilter offer increasingly powerful censorship capability to those who do not have the resources to create their own. The proliferation of these services may suggest that over time we may see a trend towards homogenisation of censorship techniques, rather than the bespoke solutions implemented by governments of the past [101].

Now that we have laid out the situation as is, we begin to move into our own original research. The next four chapters present original work that was written throughout the course of this doctorate, studying the censorship engines via their side channels, and identifying the collateral effects that they causes.

Chapter 3

Where Does Internet Censorship Occur?

In this chapter we present our early work looking at censorship by the GFW.

We began our study of censorship in China by looking at ways data is censored in China. This was done from a perspective of technique, and does not focus on the types of content that are blocked or the social implications of the blocking. Most of this chapter is specific to censorship in China and the GFW, and the research described aims to give insight into how one of the biggest global censorship engines in the world limits access to unwanted content. Section 3.1 looks at traffic previously known to have been blocked by the GFW, and whether similar content in different traffic formats also trigger censorship, while section 3.2 begins to look at the possible effects of collateral censorship for other countries whose internet traffic passes through or near censored regions.

3.1 An initial look at Deep Packet Inspection employed by the Great Firewall

In this section we describe a series of tests on the Deep Packet Inspection capabilities of the Great Firewall of China. Previous work has found there exists some DPI capability [99], and these tests focus on discovering how widespread it is. We find that DPI based filtering is not as easy to trigger as expected, and our tests proved negative in some instances where DPI filtering was expected to take place based on previous research. We believe that this is due to filtering optimisation of the GFW in an effort to improve its efficiency given limited processing capability. Our work supports a growing view that the GFW is not able to fully monitor all network traffic in China, and must make sacrifices to focus primarily on key methods of information exchange, such as web traffic [108].

Context

In chapter 2 we discussed DPI and the powers it offers internet censors. Whereas other forms of censorship may be limited to blocking an entire IP address, website or domain, DPI allows a more granular approach, enabling the targeting of a single file such as a webpage or image.

The GFW uses a range of techniques, and DPI just is one of them [99][103]. Multiple layers of fast header and protocol based filtering prevent the majority of unwanted content from reaching users, but this style of filtering may allow some content through. DPI

is an expensive process, and it is not practical to use it to censor the bulk of information traversing the massive and growing Chinese internet.

DPI involves looking at the payload of the packet: the information that sits below the multiple layers of header that are required to get the payload to its destination. Most filtering is aimed at specific protocols, and filtering based on these and their headers is quick and efficient, as the content being checked is in the same place every time. But sometimes the headers do not give enough indication of the content, and the only way to know what is being sent is to evaluate the content itself.

Within China the official name for the GFW is the Golden Shield, and it aims to protect users and supervise them as they use the internet. The content it blocks covers a range material the Chinese Government deems inappropriate, such as historical events, political ideologies and calls for collective action [56].

At the time when we carried out this research most prior work on the GFW had focused on higher level internet protocols (such as DNS [63][116], HTTP [17] and Tor [114]), and lower level filtering had not been examined to the same extent. Specifically, keyword filtering [11], HTTP GET request filtering [17][77] and DNS lookup filtering [121], which are the primary methods of content filtering employed by the GFW.

Within this section we outline methods for evaluating DPI used within a censorship network such as the GFW. DPI based filtering has the potential of blocking a wider range of content than others, including data within a variety of network protocols from ICMP and UDP to FTP and IRC. Although these protocols have been covered by other work in the past, they are generally not the primary focus.

One of the fundamental drawbacks of DPI is that it is slow and costly to implement

[120], and it is likely that its use will increase as the cost of real-time network data analysis decreases. We believe this area requires further research, and as such this study was our attempt at detecting, mapping and analysing DPI based filtering, specifically in the context of the GFW.

As DPI filtering is resource intensive, often there is pre-filtering of data so that only specific types are fully analysed, for example if web traffic content is being inspected it is reasonable for filters to trust requests made to certain known domains. We have seen this kind of optimisation with the GFW, and there is evidence of filtering techniques being adapted to make better use of available filtering resources [77]. One of the key areas of interest for DPI filtering is determining what is filtered and what is not, and how this corresponds to the requirements and desires of the censoring agency.

As we briefly covered in chapter 2, the term DPI filtering can be ambiguous. While it refers to inspecting the content below the headers, the layered way that network protocols encapsulate each other means that it can be difficult to differentiate between what exactly is header and what is content. Within this section we use to refer to payload data that is not used as part of a network, protocol, but this definition is blurred when we insert faux HTTP headers within FTP payload data.

Goals

At this time there was widespread belief that DPI filtering was taking place in China, but not much research on when it occurs, where it occurs, or what it was looking for. The purpose of these experiments was to look for signs of DPI filtering taking place,

and to see how easy it was to trigger. To this end we ran several experiments looking to insert potentially filtered requests into innocuous packets to see if the GFW would identify and interrupt them.

Methodology

Some additional areas had to be considered before analysis of the DPI capabilities of the GFW could be performed.

String Matching DPI filtering attempts to find specific strings within the payload data of network traffic. String matching can either be literal strings, or permeations such as regular expressions. Once this string or pattern has been found the censor can take action to close the connection.

Fast string matching is dependent on quickly eliminating packets which do not meet its filtration criteria. This can be achieved either by specifying the characteristics of the packet such as header data or incoming interface [89], or by fast real-time sorting techniques such as bloom filtering [25]. If there is no match at this stage, the packet can be ignored and forwarded to the next network node (assuming a blacklist approach is being used). If the criteria are met, then the string matching expression is applied to determine whether the packet should be filtered. If the packet fails both the pre-selection and the string matching, it is flagged and some type of filtering action can be taken. This could be simply dropping the packet, or (as is the case with the GFW) taking other actions to close the connection between hosts.

Filtering Techniques Once a packet or stream has been flagged, there are different techniques that can be taken to filter the content that is to be censored. The most common method of filtering observed by the GFW is to send TCP RST packets to both ends of the connection (both the client and the webserver) [32][118]. This causes both hosts to end the TCP connection, making each think that the other has terminated the connection.

An advantage of this technique is that it can be performed out of band from the network traffic itself. The data can be copied to another node for analysis, which can perform the string matching, leaving the primary networking nodes free to continue to process data streams as fast as possible. If the matching criteria are met, the RST packets can be sent from the analysing node. This setup is described by Clayton et al [17].

Test String - Falun For the tests performed in this analysis, the test string '`falun`' was used to test the response to censorship. '`falun`' is a highly censored word within China [39], and there is precedence for its use for testing filtering of the GFW [118][77]. Censorship of '`falun`' is due to the highly persecuted Falun Gong movement. Falun Gong is a Chinese spiritual discipline that has been under heavy scrutiny from the Chinese government since its founding in the mid-1990s.

Directional Filtering When filtering network traffic it can be useful to consider the direction of the traffic. Within a secure organisation it is standard to prevent suspicious connections coming in from the outside internet, while perhaps being more lenient towards those connections originating from within. With censorship it makes sense for

the opposite to occur, with the censors less interested in foreign users accessing content from within the censored realm, and instead focusing on their own users accessing unwanted content from outside. It is important to be aware of this distinction when considering why something does or does not appear to be filtered.

Stateful filtering can take this concept further, and filter traffic based on the state of the connection. This can take into consideration not just the direction of the traffic but also other features, such as such as whether a full connection has already been established, whether the packet is a request or a response, and whether a packet makes up a correct and expected part of a data stream. Want et al. wrote a recent paper focusing on stateful filtering from the GFW and how to evade it [108].

Location of Tests All of the tests described in this section were carried out from hosts outside of China. This made it difficult to comprehensively test DPI filtering due to the possible directional nature of the GFW [77].

Effort was made to perform tests that could either circumvent the directional nature of the filtering, or be agnostic to directional filters. Where possible, tests were performed trying both hosts (those within and outside of China) as both the source and destination for establishing connections. Additionally, protocols were used that made it more difficult for stateful filtering to determine where the request originated.

Experiments

Several methods for analysing DPI filtering were created and tested. They aim to build upon previous work by improving existing techniques, as well as using previous

techniques from different domains to focus specifically on DPI filtering. In the following sections we improve upon previous techniques for mapping out the Chinese internet and AS topology, a method for testing DPI using FTP, and tests over HTTP GET requests to test the stateful nature of the GFW.

Improved Network Mapping To carry out mapping of censorship similar to that as performed by Xueyang, Mao and Halderman [118],a up-to-date network map of China was needed. When we attempted to repeat their process, we implemented a more thorough way of mapping internet infrastructure than the previous technique.

The first change to our technique was to increase the numbers of address ranges scanned within an AS. The previous approach only used a single address to represent an entire AS. Instead of one traceroute to each AS, we tracerouted to all of the /24 address blocks within each AS.

As well as this, our method found more ASs than the previous method, although it is likely that some of this is attributable to an increase in the number of ASs used within the country since 2010. The previous approach attempted to find ASs by referencing the list of Chinese ASN against records within Routeview [102] and RIPE [87] databases. This returned address prefixes for 76 Chinese ASs. Repeating this we discovered that some known ASNs were missing. To account for these we combined the aforementioned list with additional ASNs found on Robtex [88] and which led to IP address ranges for 121 ASs. As a result of these two changes, our mapping connected the routes to 27,526 address ranges, compared to the 76 in the paper it was based on. This yielded a substantially different infrastructure map of China than if the previous approach had been used.

One drawback of our process is that it was only performed from Oxford and not other locations. This resulted in a tree branching out from the source towards the target ranges. As a result, while it discovers many pathways into the Chinese internet, it is missing many joining connections between different AS within China.

The data we obtained was loaded into igraph [20], a network analysis tool for R. This covers 78,249 total hops, and 2241 unique nodes. The following figure shows a map of the infrastructure routes in our data, produced by igraph. The larger the node on the diagram, the more routes passed through it.

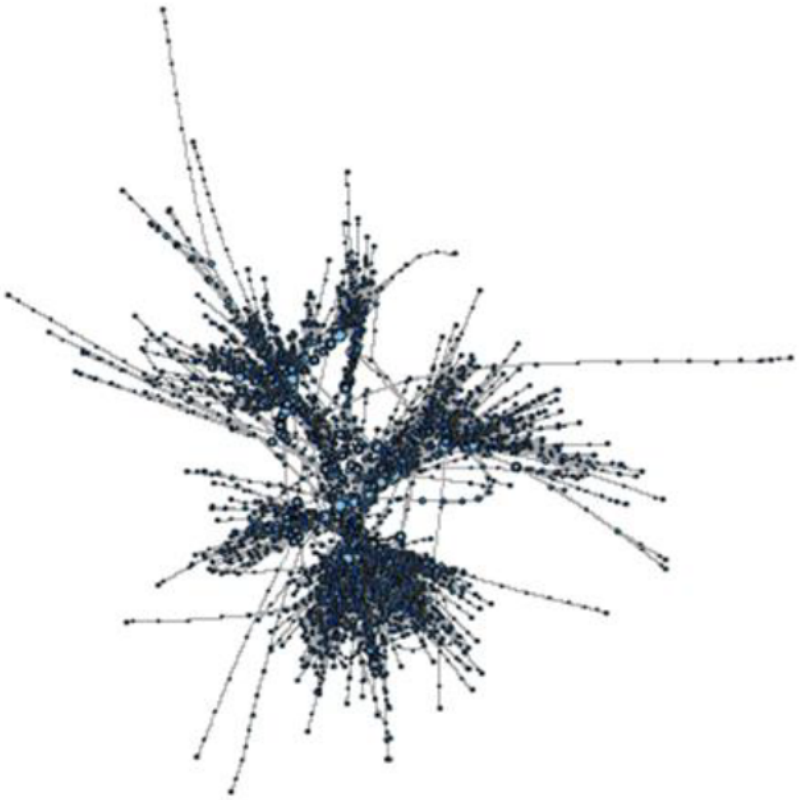


Figure 3.1: Map of Chinese internet (as mapped from Oxford)

The 27,526 unique address ranges we discovered covered 374,726,879 unique IP addresses within to Chinese ASs. A breakdown of the address range of each of these can

be found in the following table.

Breakdown of IP Ranges in China

Address Range	Occurrences	Unique IPs	Total
32	7	1	7
31	0	2	0
30	20	4	80
29	11	8	88
28	15	16	240
27	10	32	320
26	16	64	1,024
25	8	128	1,024
24	10,110	256	2,588,160
23	3,524	512	1,804,288
22	2,608	1,024	2,670,592
21	2,046	2,048	4,190,208
20	2,276	4,096	9,322,496
19	1,347	8,192	11,034,624
18	1,002	16,384	16,416,768
17	584	32,768	19,136,512
16	987	65,536	64,684,032
15	293	131,072	38,404,096
14	160	262,144	41,943,040
13	112	524,288	58,720,256
12	45	1,048,576	47,185,920
11	13	2,097,152	27,262,976
10	7	4,194,304	29,360,128
9	0	8,388,608	0
8	0	16,777,216	0
7	0	33,554,432	0
6	0	67,108,864	0
5	0	1.34E+08	0
4	0	2.68E+08	0
3	0	5.37E+08	0
2	0	1.07E+09	0
1	0	2.15E+09	0
		Total unique IPs	374,726,879

FTP To test DPI within China several tests were performed using FTP. FTP was chosen as it provides a variety of ways of transferring data between two hosts, including options for the remote host to establish a reverse connection with the local host. This is important as it allows testing of bidirectional filtering.

This experiment was carried out on servers within the Chinese address space and within Chinese ASs. To find these addresses, the data from the mapping of the Chinese network was consulted. This method provided a list of 27 thousand address ranges, covering 374 million different unique IP addresses.

These addresses were then scanned using Zmap [29]. Zmap is a port scanning tool developed by Durumeric et al., optimised for performing a survey scan over a wide variety of hosts. Unlike port scanners such as Nmap [64], Zmap is asynchronous and stateless. This gives it a great increase in speed at the cost of some reliability. This unreliability only affects the false negative rate, with around 2% of negatives being false negatives if only a single packet is sent to each address. There are alternative fast asynchronous or stateless port scanners, such as MassScan [42],Unicornscan [62] and scanrand [55] which are arguably even faster [42].

Zmap sent a single SYN on TCP/21 to all 374 million addresses, and gave an output for all of those which responded with a SYN-ACK. This resulted in 164,135 hosts that responded, indicating that they had a service running on this port. This is the default port for the FTP control service, indicating that these servers may be running it.

This list was then checked to verify which of these hosts were actually responding to FTP requests. We were specifically interested in hosts which gave anonymous FTP access. Anonymous FTP typically accepts null credentials, or either '`ftp`'

or ‘anonymous’ as the username. These credentials will give READ, WRITE or READ/WRITE access to the service.

Metasploit was used to test if the hosts responding on TCP/21 supported anonymous FTP. Metasploit tests this by attempting to log on to the service and requesting that a test file be uploaded, downloaded, and then deleted. Based on these responses (and the banner the service gives as you login) we were able to identify 5401 (of 164135 responding on TCP/21) FTP servers providing anonymous functionality within the Chinese address ranges. Of these 539 also allowed anonymous WRITE actions. Anonymous FTP is a default off feature [106], meaning hosts providing this service have gone out of their way to allow it.

Several tests were conducted on these hosts. Each of these tests covered different combinations of data to see if any filtering was in place. Hosts belonging to corporation organisations were selected for these tests, over those hosts belonging to individuals, or those which could not be identified. The first test involved an innocuous file containing the string ‘text1’, which was uploaded, downloaded and deleted from the FTP server. This was our neutral test used to confirm no filtering of data containing innocent content.

The second test involved a file containing the string ‘falun Falun’, which was uploaded, downloaded, and deleted from the FTP server. This test was performed with both active and passive FTP modes. Passive FTP mode transfers the file from the remote host to the local host using the TCP data steam that has already been connected between the two hosts. Active FTP mode creates a new TCP data stream from the remote host to the local host. This data stream was used to test whether stateful filtering was in place, as it allowed testing of connections both going into and coming out of China.

The third test involved a file containing a fake HTTP GET request. HTTP GET filtering has been observed in China in the past [77]. As DPI filtering is based on string matching, depending on the exact regular expression used this could trigger any DPI used for filtering genuine HTTP GET requests. As active FTP mode was used this connection was initiated from a host within China, attempting to access blacklisted material outside of the Chinese network.
The file contained:

```
GET /falun.Falun HTTP/1.1
Hostname:********
```

No filtering or censorship techniques were observed on any of these tests. This indicates that if DPI filtering is taking place, it is either not taking place on the ports FTP is using (TCP/20, TCP/21 and the ephemeral ports), or that other criteria of the string matching are not being met.

HTTP GET Requests Previous work has shown the use of HTTP GET request filtering [77][118]. To test the current functionality and statefulness of this filtering, we performed a series of HTTP GET request tests. These tests were performed on websites within China, from connections outside of China. The 100 sites chosen for this test were the Alexa Top 100 sites in China [2]. This test was based on the HTTP GET request tests performed by Jong Chun Park and Jedidiah R Crandall [77] and those by Clayton et al [17].

We made several HTTP GET requests to each of the chosen sites. These requests

attempted to request a series of websites from each server. Five requests were sent to each websites, requesting the pages / (the top level page), `/index.html`, `/hellohello.html`, `/falun.html`, and `/falunFalun.html`. These requests were made using GNU Wget [72] and HTTP version 1.0. Altogether 500 HTTP GET requests were made to webservers within China from external hosts.

We sent these requests so that we could observe examples of valid requests (/ and `/index.html`), non-censored but failed attempts (such as a `404 - not found`, redirect or other HTTP errors), and potentially censored requests (`/falun.html` or `/falunFalun.html`). The failed requests were then compared to the censored requests to determine if there was any difference in the HTML response.

Both `/falun.html` and `/falunFalun.html` were used in case the word Falun was split between packets. Previous research has indicated that censorship systems (the GFW in particular) have failed to prevent content when the filtered string is split over multiple packet payloads [77].

Censorship of these requests was likely to occur in one of two places: at the HTTP response (e.g. a fake `404`, or an HTML page missing content) or at the TCP connection (e.g. with the connection set via a RST packet). Firstly, we analysed the HTML response content from the webservers. We looked for any differences between the pages of the invalid requests (`/hellohello.html`) and those using potentially filtered requests (`/falun.html`, `/falunFalun.html`). No differences in the HTTP response were noted between the invalid group of innocent requests, and the invalid group of potentially filtered requests.

Secondly, during this period network traffic on the end hosts was recorded and eval-

uated. There were no signs of lower level filtering attempts (such as unexpected TCP RST packets) from either the servers themselves, or nodes between our local host and the webserver. All requests for non-existent content returned the same network traffic as those for potentially filtered requests. No unexpected RST requests were observed.

Results

Our results from these experiments were negative. This included testing FTP with sensitive payloads both heading into and coming out of Chinese ASs, as well as HTTP requests for sensitive URLs. There was no censorship indicating DPI techniques that were present for the tests carried out for this research.

A key restriction of these tests was that they were performed from hosts outside of the GFW. While efforts were made to perform meaningful tests given what was available, this placed limitations on what we were able to carry out.

The surprising result of these tests was that it was not as easy to trigger GFW filtering as was expected. Carrying out tests which should have triggered filtering according to previous work [17] came up negative this time.

A possible explanation for this may be that there is an on-going effort to optimise what the GFW is filtering, and that it does not have the capability to monitor all traffic within its domain. This fits with the description laid out in previous work [77]. DPI censorship capabilities are dependent on the speed and memory of the filtering tools. With this in mind, it makes sense for the operators of the GFW to make efforts to limit the amount of work it has to do by ignoring certain data types, protocols and ports.

This may have manifested itself twice within our tests. Firstly, performing HTTP GET requests for known filtered content should have been filtered according to earlier research [17]. That these tests are now not triggering filtering could indicate that the GFW has been changed to ignore requests from outside of China.

We found that the FTP active mode tests performed data transfers initiated from hosts allowing anonymous WRITE from within China. These tests sent data containing known filtered keywords from within China to hosts without it, over TCP ports 20, 21 as well as ephemeral ports, and there appeared to be no filtering of this content.

This indicates that certain ports may have been 'whitelisted' (those known for the transfer of large amounts of data) or 'blacklisted' (those used for the spread of information). Given that we know that Chinese censorship is aimed at counteracting the flow of certain information [56], it makes sense for them to focus on those protocols (and ports) which are typically used for this. The main suspect of this would of course be TCP/80 and HTTP.

Overall, our work agrees with others that there is a continual effort to optimise the efficiency of the GFW. New research in this field frequently finds inconsistencies with older work, and some of this is likely due to optimisation [108]. Our work builds on that growing list of inconsistencies, and offers new techniques and methods for testing the GFW and its capabilities.

Since carrying out this research a similar approach has been used to map global DPI based censorship. Instead of using FTP like in our experiments, VanderSloot et al. [104] built an approach where they inserted suspected filtered requests inside the Echo protocol [81]. While this protocol is old and rarely used, they were still able to

find almost 60,000 global Echo servers. Echo, like FTP, allowed the authors to test the bidirectionality of censorship. In the same way we tried, they inserted known filtered HTTP requests inside the Echo protocol and waited for TCP RSTs to see if the connection was filtered.

Why was this approach successful, while our approach was negative? We believe the main reason is that their tests were deployed on a far wider scale than ours, both in terms of the number of servers they sent requests to, as well as the number of potentially filtered requests send to each server. Similarly, all of our requests were between Oxford and China, again limiting the potential for observing censorship. These factors gave them many more attempts to be filtered than our experiment, even though fundamentally we were using a similar technique.

While it is frustrating that our results came back negative, it shows the importance of following a good idea with engineering and implementation. The VanderSloot et al. paper did not come out until four years after our initial experiment, which was ample time to expand on our approach and test it on a larger scale. Had we done this it would likely have led to another high impact publication.

Legal and ethical considerations

Censorship research has the potential to introduce difficult legal and ethical concerns, and this project highlights that risk. By knowingly scanning, logging into, and sending data between anonymous FTP servers the question arises are any of these actions illegal or unethical.

There are two potential issues. Firstly, is there a risk to the administrators of anonymous FTP servers within China when we send contentious content to them. Censors may see this content going back and forth to the servers and decide to investigate it, possibly holding those administrators personally responsible. Secondly, is logging into services such as FTP illegal or unethical, as it is unlikely the administrators of those servers originally envisaged them being used for such purposes. Is using anonymous credentials the same thing as hacking into the service? While Illegality and unethicality are not the same thing, one often implies the other. Many unethical actions are illegal, and many illegal actions may also be unethical, especially if they put others at risk of prosecution or other punishment.

Dealing with the first issue, we do not believe there is a significant risk to the administrators of these servers. To start with there is a fairly small chance of our data being noticed by a human. The amount of data being send over the internet is vast, and the content we sent tiny. That alone is not a good enough reason though.

The main reason we do not believe there is a significant risk of administrators being blamed for this content is because that theory would not hold up under investigation of the data. The payloads we sent were designed to trigger automatic filtering, but to a human it would be obvious that these were not real requests.

Dealing with the second issue, we do not believe that using anonymous FTP servers in this way is illegal. A 2019 paper expanded on exactly this issue [106]. Vetterl et al. originally wrote a 2018 paper where they connected to honeypots and similar services in order to fingerprint them [105]. When they first tried to publish this paper it was rejected from a leading conference on ethical grounds. The reviewers suggested that this

research was illegal and unethical, and asserted that the researchers could be prosecuted under the UK Computer Misuse Act 1990 or US Computer Fraud and Abuse Act. The authors published their work but redacted all the analysis dependent on logging into the services on the scanned hosts.

Vetterl et al. countered this by producing a follow on paper discussing the legal and ethical concerns of this type of work. They argued that there is a large amount of uniformity on global cybersecurity laws, and that they often come down to the issues of authorisation and whether a security mechanism was bypassed. They argue that as Honeypots and anonymous FTP servers are deliberately configured to allow this access no illegal action has been performed.

Oxford University has a Research Ethics Committee to help assist researchers with these difficult issues. At the time this experiment was performed there was no process for dealing with internet research of this kind, and it is only since that they have realised the requirement for one. Since liaising with them over this project they have tried to create a process to cover this type of work, and this project has been the prototype for how they deal with internet and network scanning research. Although our project has been the first to go through this process, it is still very much a work in progress.

Lessons

There are several key lessons learned from this work, both academic and relating to the specifics of the GFW and censorship research. Regarding the GFW, our initial take-away from this was surprise at the difficulty of triggering DPI based filtering. Based on

the previous literature we believed it would be a straightforward process to trigger, and expected to be above to move into similar experiments once we had found it. We started using keywords and queries we were certain would be filtered in some instances, and this prevented us going further by testing other payloads as we had originally intended.

Regarding internet censorship, we learned that it is constantly changing, and the work that we do is only going to be correct for the duration of the experiment. This is going to be true for most internet based research, as we are exploring a growing and changing system. Because of this it is important that we carefully consider the scientific value of our work, as simply describing the state of a system at a given point in time is not enough.

From an academic perspective, it teaches that it is not enough simply to have a good idea, but that idea must be following through with thoroughness and systematicity. There are mixed feelings associated with the discovery that the same technique was used four years later while our work remains unpublished. We feel both vindicated that the technique works, as well as disappointed that we did not manage to back up the idea with solid experimentation.

3.2 Exploring Collateral Censorship using North Korea

In this section we continue to try to identify where on the internet censorship takes place, and at some of the unexpected effects caused by censorship on the internet. We

explore the collateral effect that censorship of one part of this system can have on neighbouring parts of the system. For this we use the unique example of North Korea, which has limited internet connectivity with the rest of the world. As North Korea has no direct access to the undersea fibre-optic cables that form the internet's backbone, its connectivity to the rest of the world comes through third parties, primarily China and its Great Firewall. Our focus is an analysis of DNS requests passing through the Great Firewall on their way to and from North Korea. We find that the Great Firewall does not filter all such requests, and may be configured to allow some requests to pass through into different autonomous systems.

Context

On February 24th 2008 YouTube became unavailable worldwide for over an hour [50]. Why? After objecting to a cartoon posted on YouTube, Pakistan Telecom attempted to restrict access to the site. They were successful, but also unintentionally blocked access for users all over world. Within this section we look at how the location of filtering can affect other locations, and the unintended effects that actions such as censorship can have. To do this, we continue to examine one of the most comprehensive censorship programmes in the world: China's.

When censoring its citizens, a country has more options than simply removing or filtering unwanted content. Countries can shape the information available to their citizens via other means, such as through political pressure or encouragement of self-censorship. Censorship programmes very rarely rely on a single method, and can combine several

different approaches to restrict unwanted information.

The GFW has been called the 'largest selective suppression of human expression in history' [56]. Rather than referring to a specific method of censorship, the GFW is an umbrella term that covers the range of censorship techniques used by the Chinese government to filter internet content; including both technical and non-technical means. The GFW aims to filter internet content available within China so that it adheres to the ruling party's view of acceptable content.

As we discussed in the previous section, the techniques that the GFW deploys to censor unwanted content have changed over time. Previously identified techniques were found to target information flow over the web, and included HTTP Request and GET filtering [17][77]. These methods however have evolved, and both refinements of existing techniques, as well as the introduction of new techniques, have been observed. Currently, one of the primary methods of censorship used is DNS based censorship.

DNS based censorship works by targeting the Domain Name System that is used to bind IP addresses with the human readable names used to navigate the internet [69][70][14][31]. As mentioned in chapter 2, the DNS system is an ideal target for censors due to its widespread use for finding resources on the internet. All DNS queries follow a specified structure and are sent to a specific port on the server (UDP or TCP 53), making them easy to identify on a network. Additionally, standard DNS queries are not encrypted, allowing a well-placed adversary to see, and in the cases of censorship, to modify DNS queries going through their network.

DNS censorship often works by returning an incorrect IP address for a domain. The content still exists on the internet, and can still be reached directly via its IP address, but

a device trying to reach a website by its name will be directed to the wrong address or page. Alternative methods of DNS based censorship include:

- Returning an NXDOMAIN result (non-existent domain);
- Ignoring the request so that it times out;
- Redirecting the request to a warning page.

The internet is a complex system, where even small changes are capable of leading to emergent behaviour and unpredicted consequences [78]. Given the unpredictability of actions on the internet where changes in such a system often have unintended results, changes to configurations and deployments of technology should be carefully analysed and planned. Unfortunately this rarely happens with censorship, due to the closed nature of these systems. They are designed, deployed and used with little outside consultation or awareness.

The Pakistani blocking of YouTube was not an attack against the DNS system, but BGP poisoning [50]. Instead of spoofing the IP addresses assigned to the domain name as with DNS poisoning, Pakistan Telecom advertised that the IPs belonging to YouTube existed within their network. This was mistakenly broadcast to the internet as a whole, and within several minutes all YouTube traffic was being redirected into Pakistan. Pakistan had no intention of disabling YouTube for the entire internet, but this example highlights that a complex system [78] such as the internet does not always respond in predictable ways. Technical interventions can have unexpected ripples on the rest of the internet.

Goals

After our DPI tests came back negative we wanted to take a step back and focus on the censorship we knew was taking place. DNS poisoning by the GFW was well documented, so we wanted to explore it ourselves in order to gain a better understanding of it.

Reading the literature one thing was unclear: where exactly was this filtering taking place? There had been some debate, and we wanted to look first-hand to gain an informed view.

The situation in North Korea gave us a unique opportunity to do this. With several internet address ranges, one of which was located behind China, and the other which appeared to be operated by China. Would DNS censorship occur to DNS servers within these areas? The idea was simple: send requests to these North Korean assigned ranges, and see which requests were subjected to interference from the GFW.

Methodology

We assessed the effects of the GFW on the North Korean internet in a two-step process. First, we mapped the North Korean internet, before sending DNS requests to various nodes, and their intermediaries. We then collected and analysed their responses.

Background

North Korea has an official address range of 1024 addresses. This is provided by APNIC, and can be found at `175.45.176.0/22`. As well as this range there are

two supplemental address ranges, provided by private organisations. The first of these is provided by China Unicom, and can be found at 210.52.109.0/24. The second of these is provided by SatGate, a Russian Satellite company, and can be found at 77.94.35.0/24. APNIC and RIPE database queries for all three of these address ranges can be seen in the tables below.

The first North Korean address range is a /22 range provided by APNIC, and found at 175.45.176.0/22. This address range started to see use in 2010, and serves official North Korean internet content, including the Korean Central News Agency, Naenara (the country's official web portal), and the Voice of Korea. Traffic routed to and from

this address range travels through domestic internet ASs within China.

APNIC database entry for 175.45.176.0/22

```
inetnum:      175.45.176.0 - 175.45.179.255
netname:      STAR-KP
descr:        Ryugyong-dong
              Potong-gang District
              KP
              SJVC1-AP
country:      SJVC1-AP
admin-c:      ALLOCATED PORTABLE
tech-c:       APNIC-HM
              MAINT-STAR-KP
status:       MAINT-STAR-KP
              -+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
mnt-by:       This object can only be updated by
              APNIC hostmasters.
mnt-lower:    To update this object, please
              contact APNIC hostmasters and
mnt-routes:   include your organisation's account
              name in the subject line.
              -+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
remarks:      IRT-STAR-KP
              hm-changed@apnic.net 20091221
              APNIC
```

The second North Korean address range is a /24 address range provided by China Unicom, and found at 210.52.109.0/24. This address space is signed up to Korea Post and Telecommunications Co., which is the state owned telecommunications organisation within North Korean. Whilst this network formerly hosted official North Korean

services, it is now largely defunct.

```
            APNIC database entry for 210.52.109.0/24
inetnum:      210.52.109.0 - 210.52.109.255
netname:      KPTC
country:      CN
descr:        Customer of CNC
admin-c:      TC254-AP
tech-c:       TC254-AP
status:       ASSIGNED NON-PORTABLE
changed:      cncipaddr@china-netcom.com
              20040803
              MAINT-CN-ZM28
              APNIC
```

When this investigation began, there was a third IP address range assigned to North Korea. This was a `/24` range provided by Russian Satellite Company SatGate, and was available at `77.94.35.0/24`. During the course of our investigation this range was removed from assignment to North Korea, and switched to 'SatGate Lebanon'. Being assigned by a Russian organisation, this address range was not sourced by APNIC as were the others, but by RIPE [87]. It is likely that this address range was never controlled by North Korea itself, as no North Korean hosts or internet activity were observed within this range. It is possible SatGate or a customer is using this range, but as there is little

in the way of observable hosts or activity, it is difficult to say.

RIPE database entry for 77.94.35.0/24

```
inetnum:      77.94.35.0 - 77.94.35.255
netname:      SATGATE-FILESTREAM
country:      KP
descr:        Korean network
admin-c:      AVA205-RIPE
admin-c:      EVE7-RIPE
tech-c:       PPU4-RIPE
tech-c:       ANM47-RIPE
status:       ASSIGNED PA
mnt-by:       SATGATE-MNT
source:       RIPE
```

As well as these internet facing networks, North Korea has an internal intranet, the Kwangmyong [74]. This network contains information and websites published by the Workers Party of Korea, and although official information is limited, is widely believed to be using standard internet and web technologies. North Korea has produced their own operating system (Red Star OS) and browser (Naenara) [75]. Interestingly, the browser contains code attempting to access 10.76.1.11 (an internal IP address), indicating that Kwangmyong is set up as an internal RFC 1918 network [84].

Stage 1: Mapping

We ran a series of experiments against addresses inside North Korean assigned address space. We sent queries to a sample of addresses, including both authentic DNS servers (those servers where a DNS service was observed and responsive) as well as unused addresses (e.g. no host or services were observed or responsive), and left listeners

open on the requesting device, to record all packets sent between the requester and the receiver.

We ran two tests for each of the two address ranges. Firstly, DNS requests were made to any legitimate DNS servers that we located within these ranges. The DNS requests sent included requests for an unfiltered domain (`baidu.cn`), as well as for a filtered domain (`facebook.com`). We then repeated the requests but with incremental IP TTL values (`1..n`). When the value reached 0, an ICMP `type 11` (time exceeded) was sent back to the requester, from the node where it expired. This was to map out the path the DNS requests were taking to the servers. We took this approach in order to preserve, as far as possible, the authentic traffic characteristics of a DNS query. Other approaches, using standard TCP-based traffic traceroute tools, present the possibility of being subjected to differing routing and quality of service policies.

Stage 2: DNS queries

Once the path to the DNS servers was mapped out using this approach, we performed the second stage of our test. We repeated the DNS requests, but changed the destination address to be set to the intermediary nodes that had previously been mapped out. In all cases, other than the final destination node, none of the nodes we were sending requests to were legitimate DNS servers, and therefore we should not have received responses unless interfered with by the GFW. As DNS operates over both TCP and UDP all requests were run using both.

We found that in cases where requests for filtered domains were requested to nodes under the influence of the GFW, the GFW responded by sending a DNS response to

the requester. If a regular DNS service was observed, this response came in addition to that, typically arriving just before the legitimate DNS response was observed (if no regular DNS service responded, the only response was from the GFW). These responses contained an incorrect IP address for the domain name that was being requested. Additionally, they typically contained a different IP TTL parameter than if they had come from the DNS server itself, making them easy to identify.

Results

The first stage allowed us to map the path to the areas we were interested in and the second stage allowed us to check where filtering was performed on nodes along this route.

Mapping

The first North Korean address range is a `/22` address range provided by APNIC, and found at `175.45.176.0/22`. Mapping of the APNIC assigned North Korean address range suggests all routes travel through Chinese internet nodes. This was tested over TCP, UDP and ICMP, and using DNS traceroute attempts (discussed below) [63].

Within this range, two DNS servers were identified. These hosts responded to DNS requests normally, and were not affected by the GFW. These DNS servers had no entries configured for external domains, and gave null responses for entries such as `baidu.cn` and `facebook.com`. They were however configured to respond to reverse DNS queries, and revealed fully qualified domain names for other devices sitting within North

Korean address space. By DNS grinding against the IP addresses found in the range, we were able to discover the fully qualified domain names for these devices. A list of discovered hosts can be found below. Interestingly, one of the hosts is assigned a `.cn` top-level domain.

Services in `175.45.179.0/22`

175.45.176.15	ns1.kptc.kp
175.45.176.16	ns2.kptc.kp
175.45.176.10	smtp.star-co.net.kp
175.45.176.67	naenara.com.kp
175.45.176.70	mail.silibank.net.kp
175.45.179.67	email.kp.col.cn
175.45.179.76	ns1.star.edu.kp

We were surprised that requests into this range were unfiltered, despite those made to nodes en route being filtered. This finding seemed to contradict previous understanding of the GFW [3][17][13]. It has been assumed that all nodes behind an area under the influence of the GFW will be filtered. Given that these requests traversed the network as clear text DNS lookups, they were visible and susceptible to GFW filtration.

While this seemed to indicate that requests made to destinations outside of China are being ignored there is another possibility. Internet routing is notoriously complex, made up of an assortment of routing and peering agreements between different actors. Our traceroute-like technique simply told us where our packets' TTL values are expiring, but the actual routing may take place over different nodes, or the route taken to the destination may be different from the route back. This is a known problem of a normal traceroute technique: that you can never be sure that the intermediary nodes are the same ones that the full path would take. It is possible we were observing the same effect

here.

The second North Korean address range is a /24 address range provided by China Unicom, and found at 210.52.109.0/24. Mapping of this range reveals that it sits in a border AS within Chinese internet. As with the above range; this was tested over TCP, UDP and ICMP traceroute requests, as well as DNS traceroute attempt (discussed below) [63].

DNS Queries

We repeated the queries but with the destination addresses set to the intermediary nodes that we had previously mapped out in stage 1. Using this approach, and observing the DNS responses we received, it was possible to identify which nodes were under the influence of GFW's DNS poisoning.

Testing within the APNIC assigned range The APNIC assigned North Korean range exists on the far side of the Chinese internet. It exists in AS131279, which is owned by STAR-KP. To reach this network space, connections must pass through AS4837, which is owned by CNCGROUP China 169 Backbone.

Firstly, we sent DNS queries for baidu.cn and facebook.com to DNS servers within the APNIC assigned North Korean range. None of these attempts were poisoned or otherwise interfered with by the GFW, even though they were travelling through its area of control. This is in contrast to what other researchers have argued and expected to happen for DNS requests passing through GFW controlled nodes.

We then repeated the queries to the nodes through China, en route to the AP-

NIC assigned range. We observed that when we sent a request for a filtered domain (`facebook.com`) to an intermediary node within China, we received an incorrect DNS response, attempting to poison our query.

Testing within the China Unicom assigned range The China Unicom assigned North Korean range exists within the Chinese internet. It is contained within AS9929, which is owned by CNCNET - China Netcom Corp, and sits on the edge of the Chinese internet, connecting directly with the US. As it exists within the Chinese internet, routes to this network range pass through other nodes within the AS to reach the destination.

All DNS queries sent to the China Unicom range were poisoned and responses from the GFW were received. Regardless of whether active DNS services were present on the destination address, DNS queries for filtered domains (e.g. `facebook.com`) received a DNS response with an incorrect IP address from the GFW.

The tests were repeated for the nodes en route to the China Unicom range. Here too, all UDP requests were subjected to DNS poisoning attempts. Queries sent to each node along the path to the end node were susceptible to DNS poisoning as soon as our queries entered AS9929 and Chinese internet space.

Method of DNS poisoning

Chinese DNS poisoning has been previously discussed in a number of papers [3][4][63][117], and the process is briefly described in chapter 2. It works by monitoring for DNS requests to censored domains, and responding with an incorrect DNS response. The GFW does not attempt to block the packet from either the requester or

the responding DNS server, and both of these are received at their destination as normal.

When requests are made to a DNS server in China, this creates a race condition between the filtering device and the legitimate DNS server: whichever response reaches the requester first 'answers' the query, and the second response is ignored by the host (which typically responds with an ICMP destination unreachable packet). As the 'real' DNS server is the end node in the chain, the filtering device sits closer on the network to the host making the request. Thus ensuring that the result received by the host first is the filtering device's, and is cached as the IP address for this domain.

UDP DNS request showing GFW's additional response

Source	Destination	Protocol	Info
192.168.18.157	58.50.31.11	DNS	Standard query 0xa5ed A facebook.com
58.50.31.11	192.168.18.157	DNS	Standard query response 0xa5ed A 78.16.49.15
58.50.31.11	192.168.18.157	DNS	Standard query response 0xa5ed A 159.106.121.75
192.168.18.157	58.50.31.11	ICMP	Destination unreachable (Port unreachable)

UDP vs. TCP DNS requests

As has been covered previously [3][4][63], DNS responses from the GFW were only observed when DNS requests were made using UDP. The DNS protocol can use either UDP or TCP to make requests and responses; with single requests made over UDP, while TCP is used to support larger requests between different DNS servers such as zone trans-

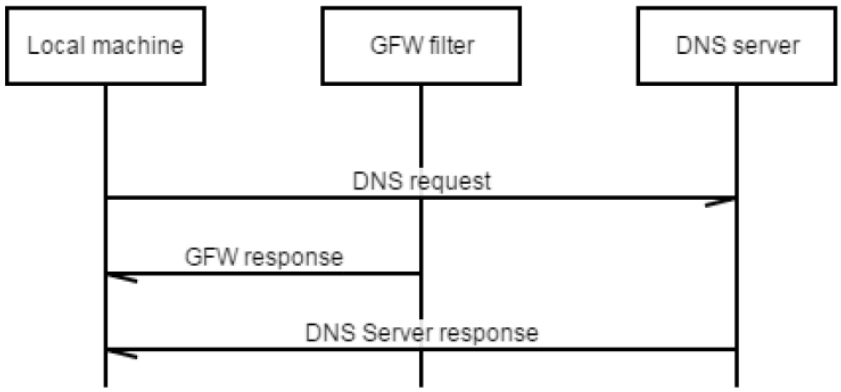


Figure 3.2: Race condition between GFW and DNS server

fers. Typically DNS servers have services listening on both TCP and UDP ports (53), and will respond to requests using whichever protocol the request was made in. Within the GFW, a request made using UDP will be filtered, while an identical request made over TCP will pass through unchallenged.

TCP DNS request unimpeded by GFW.

Source	Destination	Protocol	Info
192.168.18.157	58.50.31.11	DNS	Standard query 0x7125 A facebook.com
58.50.31.11	192.168.18.157	DNS	Standard query response 0x7125 A 78.16.49.15

This demonstrates two possible methods of ignoring GFW filtering. Firstly, incorrect DNS results can be ignored. These are normally the first DNS results to respond to a request. This is due to the advantageous location on the network that the GFW has, sitting between the requesting host and the legitimate DNS server. Secondly, TCP requests can be used. It has been observed in the past that the GFW has deliberately

moved away from using similar methods against TCP traffic, possibly due to the difficulty in 'guessing' the correct TCP sequence number [77]. This resulted in an inefficient attack, where several TCP responses with different sequence numbers would have to be sent to interrupt a single request. Using TCP for DNS queries renders the GFW's current implementation ineffective. This knowledge has been exploited by some users in China to bypass this kind of filtering [3], and we will explore in further in the next chapter.

These tests demonstrates that some DNS requests to North Korean servers are poisoned by Chinese DNS filtering mechanisms, while others are not. In order to determine where this filtering is taking place, we repeated these tests against different nodes along the path to the different North Korean DNS servers. Using this approach identified which nodes en route to North Korea occurred before Great Firewall DNS filtering took place, which nodes are susceptible to this filtering, and which (if any) nodes are being filtered behind the Great Firewall filtering domain.

This finding could indicate 'zones of control' within the GFW. We already know that different organisations are responsible for different areas of the GFW [117], but we believe this is the first documented case of the GFW ignoring data because it falls out of its jurisdiction.

It was unexpected that DNS poisoning existed for nodes en route to the APNIC assigned North Korean address space, but not for the end destination. In these instances the GFW was ignoring DNS requests that it would usually filter, dependent on the destination address of the DNS query. This contradicts the findings or assumptions of previous papers, which have argued that any host behind an area of GFW DNS censorship would be subject to the same interference [3][4][13]. Our work indicates that this is

not always the case, although it is not possible to confirm this due to a limitation of the technique we used and the complexities of internet peering agreements.

Discussing the China Unicom range, it is possible that it is filtered due to oversight or lack of consideration in its configuration. As a `/24` range it only makes up a small part of the AS it sits in. If the filtering devices are configured to filter requests to either a) within China as a whole, or b) within China Unicom's address range, then this configuration would not consider `210.52.109.0/24` as being outside of its area of authority. Given that technical filtering is an intensive process [94], and it is likely that filtering throughput is a bottleneck which places limits on what can be censored [77], it is reasonable for the censors to want to filter by as simple a ruleset as possible. As the North Korean assigned China Unicom range only holds 256 unique IP addresses, a rule to check specifically for requests into this range would be wasteful.

That filtering is limited to UDP DNS requests indicates that the censors want to limit the quantity of data that must be processed. Given what we already know about GFW filtering, it is within their technical capabilities to filter TCP requests (we have seen this with GFW HTTP filtering and with 'The Great Cannon' [66] as discussed in the previous chapter). We believe this focus on UDP DNS requests is a tradeoff aimed at filtering as much content as possible, whilst limiting the amount of data that must be analysed or the amount of requests that must be sent to interrupt a connection. This ties in with what other researchers have discovered, where inefficient methods of censorship (specifically TCP disruption) appear to have been phased out over time [77]. Whilst it is easy to bypass for knowledgeable users [24], this is a relatively low effort method of preventing the majority of users accessing unwanted websites.

Most of the observable methods of censorship by the GFW have been relatively simple, and rely on a specific repeated approach. A packet is observed passing a GFW filter, and in response an 'incorrect' or unexpected answer is sent from the censorship devices to the requesting host. This same approach has been applied to HTTP request filtering (with TCP RSTs) [17], DNS lookups (with fake DNS responses) and the 'Great Cannon' [66] (with URL redirection). This is in contrast to other methods available, such as blocking the connection entirely, or preventing the server's response reaching the requesting host.

Lessons

This project demonstrated that the internet is complex, and that there may be multiple potential causes for an observed effect. It is made up of many regions, each with ad-hoc traffic rules. Observing from a single vantage point may not be enough, and we should increase the size of our data sets in order to increase surety in our conclusions.

From an academic point of view we feel that we should have been more tenacious in polishing both of our early projects to the level required for publication. While they shaped our later work, we were too quick to move on and should have stuck with these research avenues longer. Over time we have come to believe that the best papers are not necessarily those that start with breathtaking and original ideas, but are instead those that methodologically explore and thoroughly analyse an idea.

3.3 Chapter Summary

In section 3.1 we took an initial look at the DPI capabilities of the GFW. The initial step of this process was a mass scanning and mapping the layout of China's internet. Once this was done we sent various payloads which we suspected might trigger DPI based censorship into and out of the country.

Ultimately, nothing we sent in these experiments appeared to be filtered via DPI from the GFW. This was surprising, as the GFW is known to possess some DPI capability [95]. We now believe this is because the scale of our experiment was not large enough to capture DPI censorship taking place. Follow on research by VanderSloot et al. uses an almost identical technique to discover and map global DPI based censorship [104].

In section 3.2, North Korean internet provided us an opportunity to study traffic travelling through the GFW. We used this opportunity to build on the method of DNS traceroute [3][4][63] and performed analysis of intermediary nodes en route to hosts filtered by the GFW. We then made DNS requests through this highly censored area, and analysed which results are poisoned by it. We found that contrary to assumptions made in previous research, DNS requests travelling through the GFW were not always targeted.

These results demonstrated two weaknesses of the GFW's filtering technique. Firstly, a DNS race condition is introduced between the response from the legitimate DNS server and the filtering device, and whichever response is received first will be cached by the requesting host. This first request can be ignored to instead cache the correct IP ad-

dress for the domain name. Secondly, as only UDP traceroute requests are filtered, TCP requests can be used instead. Standard DNS configurations respond to requests made using either protocol, and the GFW does not attempt to filter TCP DNS requests.

This race condition led us to wonder what the results from the actual DNS servers are. There had been studies looking at the responses back from the GFW, but none of these had focused on the responses coming from DNS servers in China. We focus on this in our next chapter: Poisoning the Well.

Chapter 4

Has Internet Censorship Changed?

In this chapter we present our work on how the GFW poisons the DNS servers in China themselves, and does not just interfere with user's DNS queries. This work was peer reviewed and published in Proceedings of the 2016 ACM on Workshop on Privacy in the Electronic Society [36], at ACM CCS.

With the right position on the internet there are seemingly unlimited ways to automate the censorship of information. Given that the internet itself is still developing and changing so rapidly, it should be unsurprising that internet censorship systems are adapting themselves in order to keep up.

As a country of over 1.3 billion people and strict information control requirements the People's Republic of China has a big challenge when it comes to censorship. Their borders stretch thousands of miles and their population consists of a wide variety of disparate peoples, each with their own beliefs. Tension occurs when these beliefs come into conflict with the beliefs of the Communist Party [83].

Given this immense scale it should come as no surprise that a key challenge for the GFW is to operate as efficiently as possible. On the internet this means having an effect on the largest amount of information possible, for the least possible expenditure.

As we saw in the previous chapter DNS infrastructure is a key weapon of choice. While not the only technique used, DNS poisoning is a low effort method of affecting a large amount of users. We have seen that this was not always the primary method used by the GFW, so how has the GFW changed over time? In this chapter we are going to look at the state of DNS itself within China, and how censorship by DNS has changed over time.

The seminal paper in this area is written by Lowe et al. from 2007 [63]. Lowe introduces DNS poisoning in China, and provides an analysis of its implementation. It presents a technique to start mapping this censorship (via editing the TTL field of the DNS query) that has proved useful in subsequent analyses of the GFW. This work built upon earlier work by Clayton et al. [17], which was focused not on DNS based censorship, but on HTTP traffic being filtered by the insertion of TCP RST packets into the TCP stream, causing connections to drop.

More recently, a thorough analysis of the GFW's DNS poisoning was presented in a pair of Anonymous papers produced in 2013 and 2014. In 2013 Anonymous (zion.vlab@gmail.com) began to look into the collateral damage of DNS filtering [3]. They looked at how DNS queries in other countries were affected by the GFW's DNS interception. Their paper deals primarily with DNS servers performing recursive lookups to servers within China, resulting in pollution of their results. They cite examples of lookups that were made where neither the requester nor resolving server resided within

China, yet the results were being poisoned because the resolving server tried to answer by recursively requesting the answer from a DNS server within China. They found that 26% of open recursive resolvers worldwide were vulnerable to result pollution in this manner.

Following on from this in 2014, Anonymous (zion.vlab2@gmail.com) provided a more thorough analysis of GFW DNS poisoning [4]. They built upon the earlier work from Lowe et al. [63] and combined it with the King method [56] to map out where DNS poisoning was occurring. They found that the majority of locations where DNS poisoning was taking place were within the border ASs of China's internet, and was primarily targeted at requests going into or out-of China. As well as this, they performed a large scale evaluation of the domains that are filtered by the GFW. They attempted to resolve all listed Alexa domains (130 million individual domains) and found that of these; around 35 thousand were censored by the GFW. Through subsequent analyses they were able to identify exact terms that were filtered by the GFW. They then offered a method for attempting to estimate the amount of requests a node deals with, and provided this analysis for a single node.

Simultaneously, Wright presented work indicating that it is wrong to view the GFW as a homogeneous filter across the entire country and provided evidence for regional variation in censorship [116]. Again, this evidence focused on DNS poisoning performed by the GFW, and how queries changed depending on where on the network they were intercepted. Wright observed that the responses from the GFW were different depending on where in the country they were intercepted, and found evidence for a decentralisation of filtering based on centrally coordinated policy. This is supported by a preceding

piece by Xu et al. [118], suggesting that different ASs and ISPs within China performed filtering differently.

There has been other analytical work performed on the type of content the GFW filters. In 2007 Crandall et al. presented a technique for determining which [19] characters and keywords the GFW attempted to filter. They found that the GFW did not attempt to block all communication that could be considered harmful, but instead filtered enough to encourage self-censorship. They argued that in this sense, it was closer to a panopticon than a firewall. In a similar discovery, King et al. [56] found that agents of censorship in China did not attempt to filter all communication, but instead focused on that which could have real world consequences. Specifically, they realised the primary focus for censorship was often discussion that encouraged collective expression. Their analysis was not based on direct filtering by the GFW like earlier analyses, but instead on content which had been posted but then later removed.

4.1 Goals

As we saw in the previous chapter, when the GFW observes DNS queries to certain domains it responds by sending a poisoned DNS response to the requesting DNS resolver. Due to its position in the network, this typically reaches the requesting DNS resolver before the response from the DNS server. This results in the requesting DNS resolver caching the poisoned DNS response, and ignoring the response from the DNS server itself. Throughout this paper we have described the DNS response that comes from the DNS server as the 'legitimate' response, and the response that comes from the GFW as

the 'poisoned' response.

Most previous work on Chinese DNS censorship has focused on poisoned responses from GFW infrastructure, and not legitimate responses from DNS servers in China. The purpose of this work was to look at the responses from the DNS servers themselves, as we believe these legitimate responses deserved further investigation. To do so we needed to look at whether the DNS servers themselves responded with the correct IP address, or if instead they are too poisoned. Understanding of the state of the legitimate DNS servers is essential for creating a full picture of censorship within China.

4.2 Methodology

Previous GFW analyses by other researchers have focused on the first DNS responses received to queries. This is typically a poisoned response, coming from systems belonging to the GFW, rather than from the DNS servers [4]. For our analysis we collect not just the first response, but also data from any subsequent DNS responses. This ensures that legitimate responses from DNS servers are captured.

We obtained a list of DNS servers from Public DNS Server list [79]. Of these 1949 were registered in China. We cross referenced this list against the MaxMind [68] GeoIP database, and removed 78 servers which MaxMind listed as outside of China. This left 1871 public DNS servers in China, according to both Public DNS Server List and MaxMind.

We choose 15 domains to act as our test domains. This consisted of 5 control domains, and 10 domains that were previously known to have been poisoned by the GFW. The

choice of these domains was based on the Alexa most popular domains site [2], taken on 18/04/2016.

We used the Alexa top domains list and the ViewDNS.info Chinese Firewall Testing Tool [107] to check each domain listed, sequentially from the most popular domain downwards. This tool makes a DNS request to a variety of servers in China, and flags the domain if any of the requests return an incorrect response. Any domains which showed signs of filtering were then manually checked with DIG [26] to confirm the result.

The most popular domain found to return incorrect IP addresses was `google.com`. This process was then repeated on sequentially less popular domains, until 10 domains returning incorrect IP addresses were found. These popular filtered domains were:

`google.com`	`instagram.com`
`youtube.com`	`blogspot.com`
`facebook.com`	`imgur.com`
`wikipedia.org`	`github.com`
`twitter.com`	`blogger.com`

The control domains were the 5 domains listed as the most popular within China, and consisted of:

`baidu.com`	`sina.com.cn`
`qq.com`	`weibo.com`
`taobao.com`	

We sent a DNS request to each of the 1871 public DNS servers for each of the 15 domains, resulting in 28065 total DNS requests. Each DNS request had a unique and incremental DNS transaction ID value, which was used to determine which response matched to which request. This was needed as many requests had multiple responses due to the GFW. All traffic during experimentation was recorded directly from network traffic, rather than from the software DNS resolver making the request. This was then stored in packet capture format. This process of recording the network traffic itself ensured that all DNS responses were captured, even in cases where multiple DNS answers came for the same query – for example when the GFW had poisoned the request – and that each response could be correctly attributed to the request that prompted it.

4.3 Results

Most requests that were made to DNS servers under the influence of the GFW came back with two DNS responses: the legitimate response from the server, and the poisoned response from the GFW. Our initial expectations were that for DNS requests made to servers under the influence of the GFW, we would receive one correct DNS response from the server, and one incorrect DNS response from the GFW infrastructure.

This was not the case. On most occasions both the legitimate and the poisoned DNS responses were incorrect.

Some DNS requests returned more than two responses. In many cases, none of the responses returned the correct IP address for the domain. For example, sending a DNS request for ‘blogger.com’ to `140.206.217.2` elicited three responses, one from the

legitimate DNS server and two from GFW infrastructure, none of which were correct.

```
localhost,140.206.217.2,blogger.com
140.206.217.2,localhost,37.61.54.158,blogger.com
140.206.217.2,localhost,78.16.49.15,blogger.com
140.206.217.2,localhost,59.24.3.173,blogger.com
```

Many poisoned and incorrect responses from the GFW returned an IP address from a small set of incorrect IP addresses. Not only would the same incorrect IP address be returned for multiple requests to the same domain, but often the same incorrect IP addresses were observed as answers for different filtered domains. For example, requests to both `facebook.com` and `blogger.com` returned results for `37.61.54.158`.

We also observed these same IP addresses as responses within both legitimate and poisoned DNS responses. In many cases we observed both DNS responses coming from this small set of IP addresses.

In total we found 9 IP addresses that were repeated for incorrect results for requests made into the GFW. These IP addresses were not all observed with the same frequency, with the most common address appearing over 11 times more than the least frequent address. These IP addresses are registered with, and geolocate to, different locations and AS around the world, with no obvious pattern. The occurrences figures below came from running the experiment four times over four separate days.

Null 9 IP addresses, and number of domains associated with them

IP Address	Occurrences
37.61.54.158	54076
93.46.8.89	14642
59.24.3.173	4848
78.16.49.15	4841
203.98.7.65	4832
243.185.187.39	4755
159.106.121.75	4754
46.82.174.68	4711
8.7.198.45	4683

Testing the Liveness of the addresses

Once we gathered these IP addresses we wanted to investigate them. The first step we took was to test for their liveness [9]. To do this we scanned each of them using both ICMP echo requests and TCP SYN requests to all possible ports. There was no response to any packet we sent. This either means that there is no host located at these IP addresses, or that if there is a host there the responses are filtered, for example either at the network interface or by an outbound firewall configured to prevent any response.

If there are no hosts are these addresses, the GFW is losing some of its ability to collect data. While the operators of the GFW can log the IP addresses of hosts attempting to connect to filtered sites via the DNS poisoning infrastructure, more information could be gained by observing attempted connections. If full TCP connections were established a large amount of data on users and their machines could be collected, but we can see that this is not happening.

This indicates that the operators of the GFW are not particularly interested in data

on users attempting to access filtered websites. This may seem odd, but given they are not taking simple steps to gather information this is a likely conclusion. Most other DNS redirecting censorship schemes around the world redirect to a website under control of the censors, which can provide additional user information on those trying to access censored material. It seems strange that the GFW does not attempt to do this.

Since we published this research Bano et al. produced a paper specifically dealing with the issue of liveness when scanning [9]. They argue that liveness is not as simple as a yes/no question, and instead describe a taxonomy of liveness.

Investing the past with DNSDB

The next step in our investigation was to look at the historic use of these IP addresses. There is evidence of the use of these addresses going back several years. There has been some limited observation of these addresses in the academic community [116], and informal observation from others [13][122]. Lowe's work from 2007 identified 8 different IP addresses being returned as results from similar requests [63].

For our investigation we used a passive DNS replication database to check for historic use of these IP addresses within DNS. Passive DNS replication is a technique to replicate domain information by passively collecting historic DNS queries and their responses [110]. We searched the passive DNS database DNSDB [37] for instances of these IP addresses.

DNSDB records show that before 2010 these IP addresses were not associated with these filtered domains, and had few, if any, domains associated with them. Some of them

had not historically been observed occurring in any DNS records at all. For example, for `37.61.54.158`, which is the most frequently observed IP address in the set of 9 repeat incorrect IP addresses, up until 2010 there were no domains which returned this IP address. Starting in 2010 there were 1494 domains, then in 2011 there were 6111 domains, in 2012 there were 2008 domains, etc. This process begun on June 30th 2010, starting with the domain `wdxxx.com`. Throughout July 2010 this increased to 55 domains, and by the end of the year DNS servers in China were returning this IP address for at least 1494 domains.

Domains per year associated with 37.61.54.158

Year	Domains
2009	0
2010	1493
2011	6698
2012	2638
2013	1725
2014	1000000+
2015	3728
2016	745566
2017	133204
2018	11*

**as of October 2018*

Note that these values include subdomains. 2014 and 2016's high figures are caused by domains using highly variable strings as subdomains. In 2014 the domain `jptea.cn` had over 1,000,000 listed subdomains, ranging from `a.jptea.cn` to `999999999999.jptea.cn`. In 2016 the domain `jjj.com` has 744492 listed subdomains, mostly made from subdomains such as `zzz96706.jjj.com`,

zzz85965.jjj.com, etc. As the GFW blocks each subdomain individually, these may be attempts to avoid its poisoning.

Although this work was originally carried out in 2016, we repeated the process again in 2018. Whereas before 37.61.54.158 was associated with thousands of different domains, from part way through 2017 new domains were no longer being associated with this address. Over time the number of existing domains associated with this address have decreased, and since September 2018 this IP address doesn't seem to have been used at all in the way it was previously.

Although we observed fewer responses from the GFW for the other 8 IP addresses, they appeared to have a similar number of domains associated with them. For example, the breakdown for 8.7.198.45 is below:

Domains per year associated with 8.7.198.45

Year	Domains
2009	0
2010	1480
2011	6567
2012	2645
2013	1849
2014	1000000+
2015	3098
2016	40678
2017	505427
2018	34584*

**as of October 2018*

One glaring difference is that 37.61.54.158 has become unused, whereas 8.7.198.45 has continued to have traffic redirected towards it. The number of do-

mains associated with each IP address for 2018 (up to October) are as follows:

Null 9 IP addresses, and number of domains associated with them

IP Address	Domains
37.61.54.158	11
93.46.8.89	31705
59.24.3.173	34270
78.16.49.15	31658
203.98.7.65	2677
243.185.187.39	34575
159.106.121.75	1
46.82.174.68	31760
8.7.198.45	34584

The final 11 domains that redirected to 37.61.54.158 over early 2018 were:

```
www.jswpac.com.cn. IN A 37.61.54.158
mostor.cn. IN A 37.61.54.158
www.mostor.cn. IN A 37.61.54.158
www.qmjdw.com. IN A 37.61.54.158
diyanat.com. IN A 37.61.54.158
www.diyanat.com. IN A 37.61.54.158
www.eweiter.com. IN A 37.61.54.158
iioooop.com. IN A 37.61.54.158
www.xylifetree.com. IN A 37.61.54.158
xn-jvrxly8d.xn-rhqv96g. IN A 37.61.54.158
www.xn-jvrxly8d.xn-rhqv96g. IN A 37.61.54.158
```

The domain `www.qmjdw.com` was the single domain associated with `159.106.121.75`.

4.4 Lessons

When we started this work we expected to find that the legitimate response from DNS servers within the GFW contained the correct address for filtered sites, or at least a different incorrect IP address than the one set by the GFW. Instead what we found indicates that the DNS servers themselves have had their results poisoned. Whilst this finding may be obvious in hindsight, we could not find this discovery stated outright in the existing literature. Indeed, some of the papers working in this area appeared to make the assumption that the underlying DNS infrastructure itself could be trusted.

Most of the previous analyses of the GFW's DNS poisoning have discussed the effects of this poisoning on users within China, rather than on the DNS servers themselves. While there has been discussion of collateral DNS poisoning with recursive queries, this has been from the perspective of users outside of China [3][13][34].

We do not believe we were alone in our ignorance of this phenomenon. Indeed, several past studies of the GFW have proposed methods of avoiding poisoning that assumes the servers themselves can be trusted, without specifying the need to configure for the use of alternative DNS servers [63]. Suggested methods include:

- Using TCP for DNS queries
- Using UDP on a non-standard port
- Ignore the first received DNS response

- Identify and ignore poisoned responses

As public DNS servers within China appeared to be poisoned themselves, none of these methods would have worked on their own as the infrastructure itself is poisoned. Instead users must also configure their local DNS resolver to point to an unpoisoned DNS server outside of the influence of the GFW. This includes not just those that are physically within China, but also those that could be affected by collateral censorship [3].

Although there have previously been disagreements about whether the GFW poisons results centrally or along border nodes [19][118][44][40], Anonymous provided strong evidence for the border theory in 2014 [4]. They also found that only a small number of filtered requests within China are actively poisoned: 4% – 16%.

We postulate that the primary use of the GFW's DNS poisoning is not to poison DNS requests of users, but to corrupt the cache of the DNS servers. The majority of Internet users in China make DNS requests to regional servers, and these requests are unlikely to pass border ASs and receive poisoned responses. While much research focuses on direct DNS poisoning and how to avoid it, poisoning of users queries appears to be an occasional and indirect effect.

While we find the repeated use of 9 IP addresses in DNS responses interesting, we were unable to find any connection between them, or hosts located at these addresses. There is evidence that they have been in use since 2010, and that other IP addresses were used for the same purpose previously [63].

4.5 Chapter Summary

In this chapter we presented an analysis of DNS responses from public DNS servers under the influence of the GFW. We focused not just on the poisoned responses, but also looked at the legitimate responses from DNS servers. We found that in many cases the legitimate responses were pointing to the same IP addresses as the poisoned responses, suggesting that the servers themselves have been poisoned.

We also observed 9 incorrect IP addresses that are repeated from both legitimate DNS servers as well as the GFW infrastructure itself. We are not sure why the GFW is responding to DNS requests with these specific IP addresses, and have not found any evidence that there are hosts listening at these addresses. There appears to be no pattern or relationship between these IP addresses, either in terms of logical address space, or geographical registration.

Our findings indicate that even if the GFW does not poison a particular DNS request, if for example a request does not pass any poisoning nodes, the results are still unreliable. They indicate that several proposed methods for avoiding the GFW would not be sufficient alone and that additional steps must be taken, including the use of trusted servers outside of the control of the GFW.

We postulate that this indicates that the GFW's DNS poisoning technique is aimed less at users, and more at the DNS infrastructure itself. This is supported by evidence from Anonymous' 2014 review of the GFW [4]. We believe there is need for further investigation into the propagation of domain information on DNS servers within

and around the GFW. This information is often not trustworthy, even when dealing with some of the most popular domains in the world such as `google.com`, `facebook.com` and `wikipedia.org`.

Now equipped with an understanding of the GFW and how it works, we want to turn our attention to users' circumvention of censorship. One assumption that continued to come up throughout our research is the assumption that users attempt to evade censorship with tools such as Tor and VPNs. While this sounds like a reasonable assumption, it is hard to directly prove. The next chapter looks at VPNs and introduces a technique for determining what domains users of VPNs are actually accessing. We compare these domains against lists of censored domains, to see whether users are using these services to access filtered content.

Chapter 5

Do Users Evade Internet Censorship?

In this chapter we present our work on DNS Cache Snooping of VPN services internal DNS servers. This work was peer reviewed and published in Proceedings of the 2019 IEEE Workshop on Traffic Measurements for Cybersecurity [35], at IEEE Security and Privacy.

Anecdotal evidence suggests an increasing number of people are turning to VPN services for the properties of privacy, anonymity and free communication over the internet. Despite this, there is little research into what these services are actually being used for. In this section we use a technique called DNS cache snooping to determine what domains people were accessing through VPNs. This technique is used to discover whether certain queries have been made against a particular DNS server. Some VPNs operate their own DNS servers, ensuring that any cached queries were made by users of the VPN. We explored three methods of DNS cache snooping and briefly discuss their strengths and limitations. Using the most reliable of the methods, we performed a large scale DNS cache snooping scan against the DNS servers of several major VPN providers. This al-

lowed us to see which domains are actually accessed over VPNs. We ran this technique against popular domains, as well as those known to be censored in certain countries; China, Indonesia, Iran, and Turkey. This work gave a glimpse into what users use VPNs for, and provided a technique for discovering the frequency with which domains records are accessed on a DNS server.

5.1 Context

Increasing global government crackdowns on internet privacy, anonymity and free communication have found users looking for technologies which they believe can help restore these properties. One of these is the Virtual Private Network (VPN). While not originally designed for this purpose, nevertheless, many have turned to them, often over technologies specifically designed with such properties in mind e.g. Tor [27], Psiphon [16], and Signal [67][18]. VPNs allow users to trivially, quickly and cheaply make it appear as if they are connecting from a different IP address and location, giving users a sense of anonymity and allowing access to content that may not be accessible through their regular internet connection. They also encrypt traffic between the client and VPN gateway, preventing or hindering network traffic analysis or surveillance. They are in widespread use in countries with strict internet censorship regimes [71][7], and they are being increasingly targeted to prevent this use. Most VPN providers advertise on the basis of evading censorship and surveillance [41][51][90], and there has been academic work either implicitly or explicitly assuming this connection [76][82][91]. Many censorship regimes, including the GFW attempt to block, remove or regulate VPN use

[48][58].

Despite this, VPNs are not always designed and implemented with strong security, anonymity or privacy guarantees. Due to the ease of setting up a VPN, there are many organisations offering both free and paid VPN services. Many of these are fairly basic, and cannot offer the same level of security as technologies which are designed with these requirements in mind [6][80].

VPNs create an encrypted tunnel between the client and the VPN gateway. Once this tunnel is established, regular IP routing is used to make requests such as accessing web content. This requires access to Domain Name System (DNS) servers. There is no standardised way for VPNs to handle DNS infrastructure, and so most VPNs handle these requests in their own way: some making no provision for DNS requests, others forwarding requests onto public servers such as 8.8.8.8 [28], and others still operating DNS servers themselves. Over time, users of VPN services seeking privacy and anonymity have become aware of the risks of deanonymisation through DNS requests. As a result, an increasing number of VPN services now include a DNS resolution service internal to the VPN. However these DNS servers can reveal information about the users and uses of these VPN services.

5.2 Goals

There are many reasons that users may want to use VPNs, but one of them is for censorship evasion. Using a VPN allows users to access content that may be blocked in their country for a variety of reasons. Bearing this in mind, we wanted to see the

content that users are accessing over VPNs to discover how prevalent the use of VPNs for censorship evasion is.

Seeing what users use VPNs for is difficult, as these are privacy focused services which have no interest in publicising their data. They are built with privacy in mind, albeit often not to high standards. In order to gather data on what users were accessing we had to look at alternate approaches. This led us to DNS cache snooping.

5.3 DNS Cache Snooping

DNS cache snooping is a technique that allows the discovery and analysis of DNS records that a particular server has cached. While it cannot (ordinarily) be used to breach the privacy of individual users, it can be used against services to discover what users are collectively using the service for. While a known technique, there is little academic work focusing on it [43]. Using DNS cache snooping it is possible to perform an analysis on which domains are being queried against that server and how frequently they are being accessed.

Private VPN subscriptions typically come with an application or OpenVPN [119] configuration file which makes changes to the network interface of the client host. There has been some concern in the VPN community about 'DNS leaks', where user anonymity is compromised by DNS requests sent to third party DNS servers [73][98][100]. In response, VPN services are increasingly offering their own internal DNS server to prevent

[1]During our analsysis we found one instance of a possibly misconfigured VPN DNS server which was accessible over the open internet.

requests leaving the network. Normally these servers are only available once connected to the VPN[1].

For this work we focused on those VPN providers which offered their own DNS servers. Once we were connected to the VPN we made repeat DNS queries for domains we were interested in. There were three possible ways to carry out DNS cache snooping scan from this position.

TTL with no recursive queries – The first method is to query a DNS server with the Recursion Desired flag set to 0. This flag is intended to prevent the server from performing a recursive lookup in the case that the entry is not cached locally. If no cached record is returned then we know that no user has made a recent query for this domain or that the DNS server does not return anything when RD is 0. This period is determined by the maximum DNS TTL setting which is configured on the authoritative DNS server for that domain. This value is set in seconds, and states the amount of time a caching DNS server should hold that record before letting it expire. Once it expires, a new request must be sent to get a fresh record. Unfortunately, DNS servers are not required to respect the Recursion Desired flag. Upon receiving these requests some servers perform a standard recursive lookup for that domain. While servers respecting this flag can be found on the open internet, for the purposes of this work we did not find any internal VPN DNS servers that did so.

TTL with recursive queries – The second method is similar but does not rely on the Recursion Desired flag. Instead we wait for the cached record to expire, and then wait for an additional period no longer than the maximum observed TTL for that domain, before making our own query. By subtracting the remaining TTL from the query from

the maximum TTL for that domain, we see how long it took for the record to be refreshed after expiring.

This method relies on knowing the maximum TTL for each record on that server. TTL values are set by the authoritative server for that domain, but we found in practice that some DNS servers do not respect this. In response, a precursor step to this approach is discovering the maximum TTL for a domain on a per server basis.

This approach is less ideal than the first as our queries pollute the DNS server's cache. Because of this there are periods of time where we are unable to observe queries for this domain: as the records are cached and counting down from our request. As a result more data is needed to get the same accuracy in results. However this approach is reliable and works on almost all DNS servers we tested against[2], and is the method that we ultimately used

Time based – The third method of DNS cache snooping is timing based, depending on whether the query is cached on that server, or if the server has to make a recursive lookup for this domain. This technique is ideal for highly frequented domains or active DNS servers, and can identify granularities of frequency of less than one second. Unfortunately this approach is susceptible to network jitter and fluctuating delay variations caused by third party DNS services and CDNs. We found that although there is a large increase in response time when making a request for a domain that is not cached, such response time differences were common. The majority of requests we made that took an increased amount of time to respond to were unpredictable jitter, rather than recursively

[2]The only server did not work on was the aforementioned server that was refreshing its cache before the TTL had expired

retrieving a record from another server. This is likely not an insurmountable problem, and can probably be overcome with enough data and noise filtering techniques.

5.4 Methodology

We ran our experiment against three internal DNS servers belonging to popular subscription VPN services. To make requests to these services a client had to be connected to the VPN. Once connected they were able to make requests and perform DNS cache snooping to see which domains were being accessed. All of these experiments were run during April and May 2018.

Popular Domains – The first stage of our experiment was run against 1000 popular domains. This list was taken from the Majestic Million [65]. We choose this top domains list as it is based on web backlinks, similarly to how we choose our most popular censored domains (described below) [92]. 11 of these domains had expired or were not live [9], so these domains were removed and replaced with new domains from the list to keep the total at 1000. These inactivate domains were primarily service subdomains which had recently been disabled and were no longer in use.

Censored domains – The second stage of the experiment we repeated against popular domains that we knew to be censored. We gathered these domains using the technique described by Darer et al. [21][22]. This list was also based on backlinks from web pages, similar to the Majestic Million, and so domains with more links to it from other sites were chosen over those domains that only had a few backlinks. Domains which were duplicates between the lists were removed, as were domains that had expired and

consistently failed to resolve to an IP address.

Language specific censored domains – For the third and final stage of the experiment we ran the process against language specific domains. One weakness of our method is that we are unable to tell who accessed a record for a domain: only that it was accessed. Many of the domains in these lists contained content that is relevant to a wide variety of nationalities, for example facebook.com or blogger.com. These platforms allow the publication of a wide variety of content, and it is impossible for us to say whether someone is accessing this from say Indonesia (or, to access Indonesian related content), or from another country entirely.

With this in mind, and to find domains where the content was specific to the four countries that we were looking at, we decided to separate these domains by language. From here we classified domains based on 1) the language of the content of the landing page, and 2) the HTML ISO language code. Domains where these two pieces of information gave conflicting reports had their content checked manually. Domain content classification was performed with the Google Compact Language Detector v3 (CLD3) open library [86]. After language classifying the web content of the pages we removed the domains where the content was primarily in a language that was different to the country it was censored from. For example, we found that shahrvand.com was censored in Iran, and that its primary language was Farsi, so we kept it in the language specific list. Blogger.com contained a large variety of languages and was not specific to any one country, and so was removed. Using this method we ensured we kept only those domains with content in the language that was directly related to the country that had restricted access to it. As these domains were known censored in their home countries,

and as the language of these pages is specific to those countries, using accessing these domains through the VPN were likely to be doing so for the purposes of censorship evasion.

Initially we tried to query the authoritative servers for each domain to discover their maximum TTL, but we found that some servers did not respect this. For these domains we had to manually discover the maximum TTLs being used for each server domain pair. We began this process by sending a single query for each domain. This query returned a TTL value stating when the record was due to expire. We then repeatedly polled the DNS server for that domain as the TTL value counted down. When this value reached 0 it would roll over and return the suspected maximum TTL. This process was rerun until the same maximum TTL had been seen for a given domain 5 times, to ensure we had the correct value. In practice the first observed suspected maximum TTL was the finally accepted value in >95% of cases. Additionally, all observed maximum TTLs were 'round numbers'. Of our entire dataset, >96% were multiples of 60 seconds, and those that were not were multiples of either 15 or 20 seconds. This approach gave us high confidence that we had found the correct maximum TTL for each server domain pair.

One server was found that this approach did not work on. This server reset the cache record for an expiring domain when the record was 3-5 seconds from expiring. This was noticed early when looking for viable servers, and so the experiment was not run against this server.

Another implementation issue we ran into was a VPN service that used multiple DNS servers, each with their own cache. When connecting to this VPN you were assigned into a private network range. Within this range was a DNS server to answer queries.

The range entered was different each time you connected, regardless of the location of the VPN endpoint you choose, and the cache records of the DNS servers was not shared between different network ranges. Our method worked on these DNS servers, but ultimately we did not include them in our data as each one got few hits from other users.

5.5 Results

Over two months we observed just under 6 million DNS queries for the 2000 domains. Some of these were accessed frequently, such as instagram.com with 18378 hits, whereas some were accessed infrequently, such as risheha.com with 1 hit. Our initial analysis of the dataset involved collating the records from the different VPNs into one large dataset, and calculating the frequency which domains were accessed over all observed VPNs. As we know the time when each TTL was due to expire and when it was refreshed (whether by us or someone else) we can calculate the total observed period per domain. We divided this by the total observed refreshes to get an estimate of the average refresh rate per domain.

For each domain we calculate the Poisson arrival rate, λ, of x, where x is the number of events in a fixed time period. The Poisson distribution is chosen in our initial modelling as an appropriate function for modelling count data, under the assumption that the variance and mean of the arrival rate are equal. If we were to require more specific characterisation of features of the traffic we might prefer a more flexible distribution such as the negative binomial, which allows for *overdispersion* in which the variance of

the arrival rate is greater than λ. For our restricted application of ranking mean arrival rate of queries per domain, however, the Poisson is a widely-used and efficient approximation.

$P(x) = \frac{e^{-\lambda}\lambda^{x}}{x!}$

We then ranked each domain sorted by the 'sample lambda' - observed mean arrival rate (distinct from the 'true' arrival rate):

$\hat{\lambda} = \frac{t}{o}$

We calculated each of these to 95% confidence interval:
$b = \pm 1.96\sqrt{\hat{\lambda}/o}$

Discussion around observed domains

Figure 5.1 shows the most frequently accessed domains. Popular social network sites such as twitter.com, instagram.com and weibo.com were highly represented at the top of the list. These sites encourage frequent user interaction, and also often have an app to encourage mobile use.

Some of the domains are on the list because of applications that regularly refresh their domains. High ranking examples of this are steamcommunity.com and soundhound.com, which both operate apps with a clear need for regularly communication with remote servers. Steam is a popular PC gaming application, and also offers a mobile chat application and mobile account authenticator. Soundhound is an audio recognition company, that sends live audio samples to their remote servers in order to identify

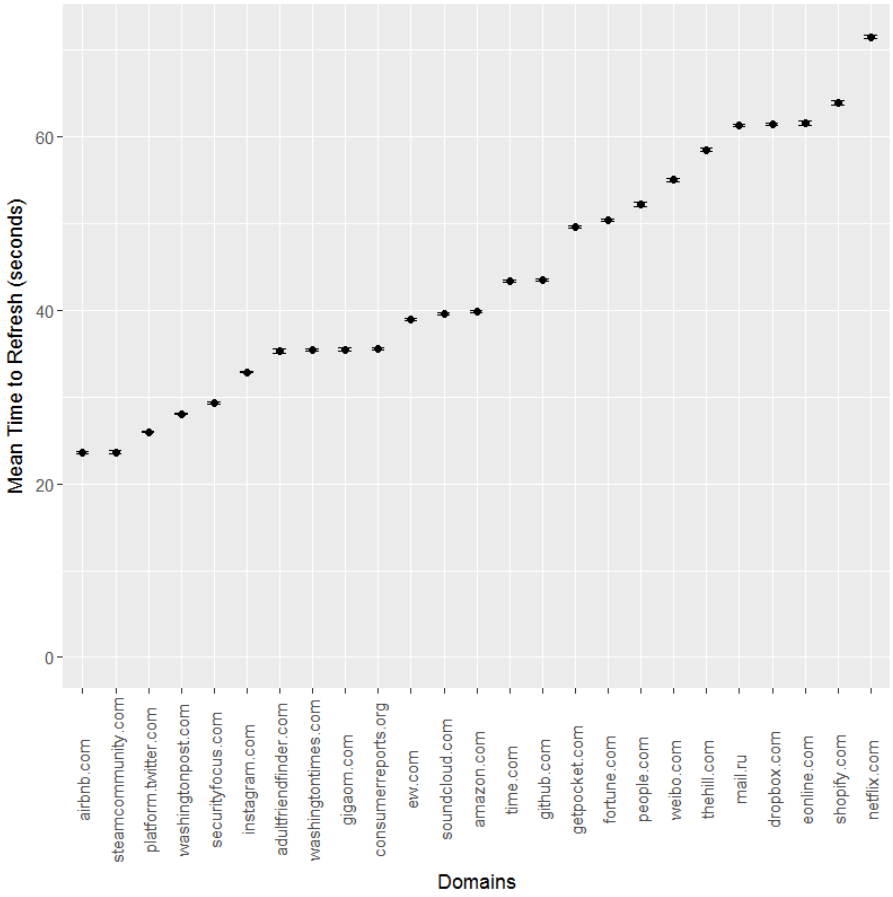


Figure 5.1: Most frequently accessed domains

them. Other domains that fall into this category include ew.com, airbnb.com, and platform.twitter.com. These applications are often in frequent communication with their servers, which may result in a large amount of queries for these domains.

There are some unexpected domains amongst the most frequently accessed which required further investigation. securityfocus.com and gigaom.com are unlikely to be websites that people visit frequently and are not known to be associated with applica-

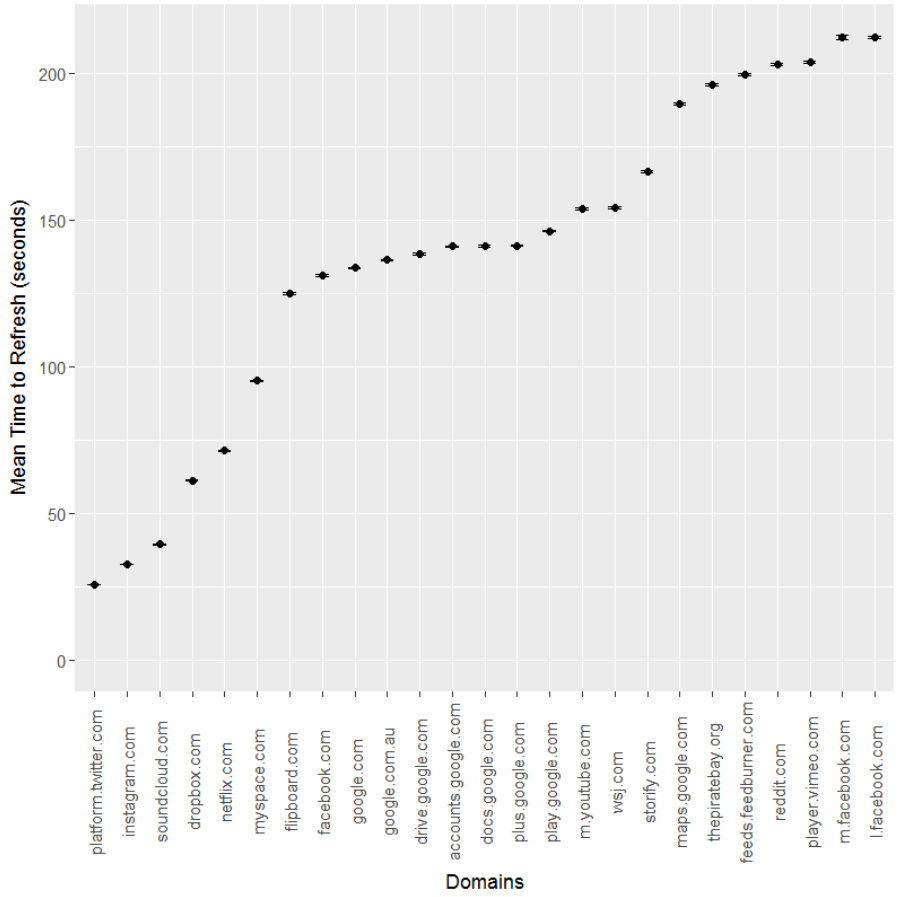


Figure 5.2: Most frequently accessed censored domains

tions. Why are these domains being refreshed so regularly? After further study we now believe that these domains are being refreshed by automated internet research tools. Both organisations do internet research and scanning, and Security Focus also offer a range of downloadable tools. It is likely that some of these scanners or tools are 'phoning home' for information sharing. A single tool making constant requests for a particular domain has the potential to quickly rise towards the top of our most frequently refreshed

domains list.

This uncertainty over what caused the query is one weakness of the approach. From just the DNS cache it is impossible to tell the cause of the query. Other causes of refreshed domains could include web crawlers, hotlinked content, or third party content or scripts embedded into other pages. Each of these would cause a request for the domain name, and from the cache alone it is impossible to know the cause. A consequence of this is that this technique only works if no one else is doing it. If this technique becomes popular amongst researchers, its results become meaningless.

However, as our approach differentiates between parent and subdomain there are some other interesting findings that shine light on the differences between automated and human requests in some circumstances. An example of this is that platform.twitter.com is accessed far more frequently than twitter.com. This subdomain is used by the Twitter API available to third parties. It seems that in our dataset the API subdomain was accessed far more frequently than the human usable site. This exemplifies a pattern amongst the data, where those domains associated with an app or tool feature more frequently than those that are only accessed manually.

We were surprised how infrequently some popular domains are accessed over the VPNs. While there are several domains which have an average refresh period of 20 seconds or less, this drops off rapidly. Of the full dataset only the 20 most frequently accessed domains had an average refresh period of less than 60 seconds. Initially we expected more domains to be refreshed as soon as they expired. Only the 103 highest frequency domains were accessed more regularly than once every 1000 seconds.

Figure 5.2 shows the most frequently access domains that were known to be censored

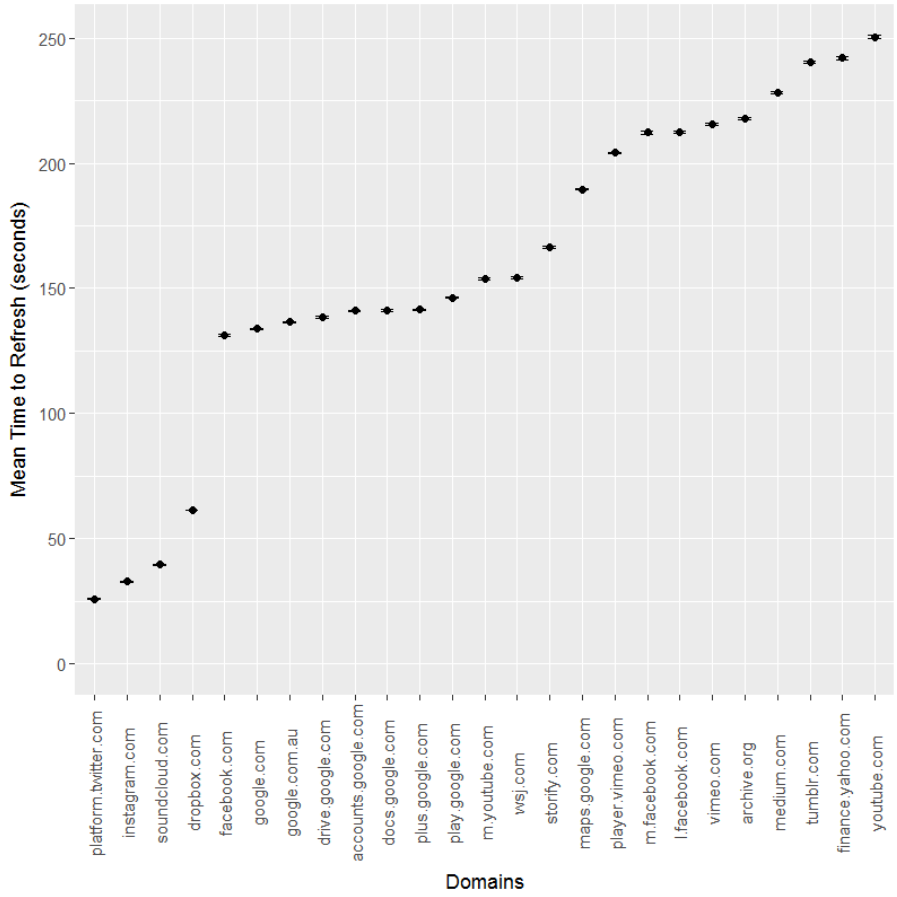


Figure 5.3: Most frequently accessed domains (China)

in one or more countries. There is a large degree of overlap between this and 5.1 as several of the most frequently accessed domains are censored. The majority of these are listed as they are censored in China including social networking sites like twitter.com, instagram.com and facebook.com from China.

Figure 5.3 shows that these sites make up the most frequently accessed domains blocked in China. As well as the traditional social networking sites, there are news

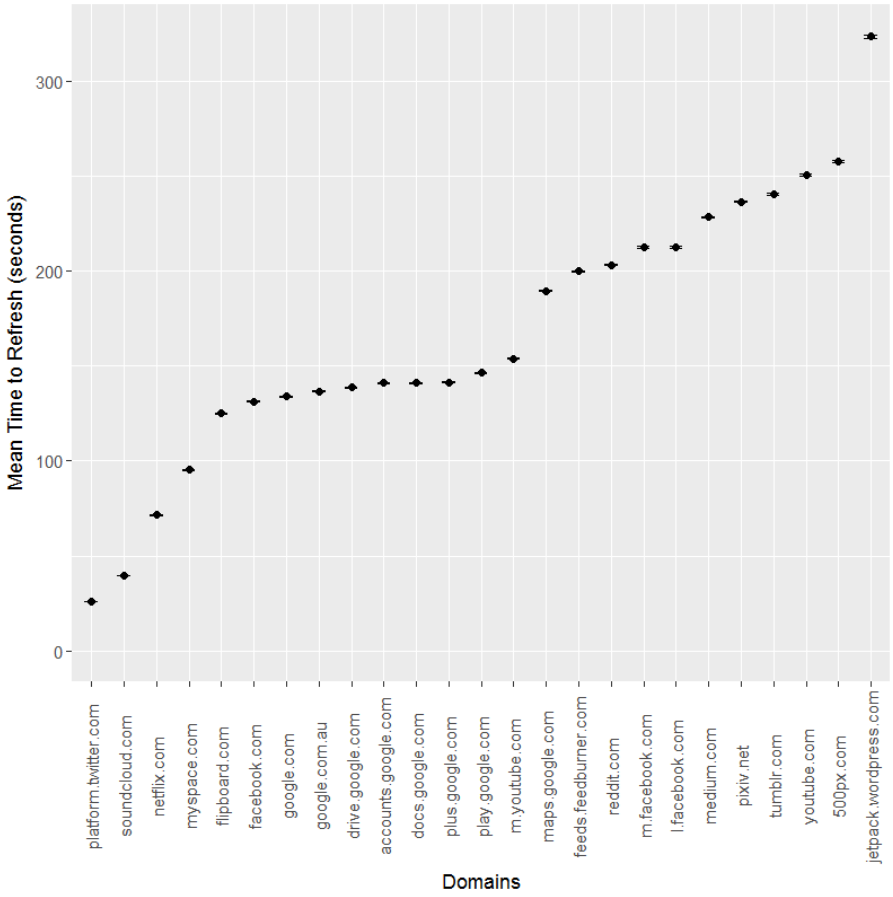


Figure 5.4: Most frequently accessed domains (Iran)

or informational sites such as wsj.com, storyify.com, archive.org, wikipedia.org and blogspot.com. As our method is unable to differentiate between users attempting to access these sites from China versus the rest of the world, it is impossible for us to tell how much of this traffic is from users attempting to circumvent censorship.

Figure 5.4 shows the most frequently accessed domains which are blocked in Iran. We can see Iran blocks many of the same major US domains as China. This includes many

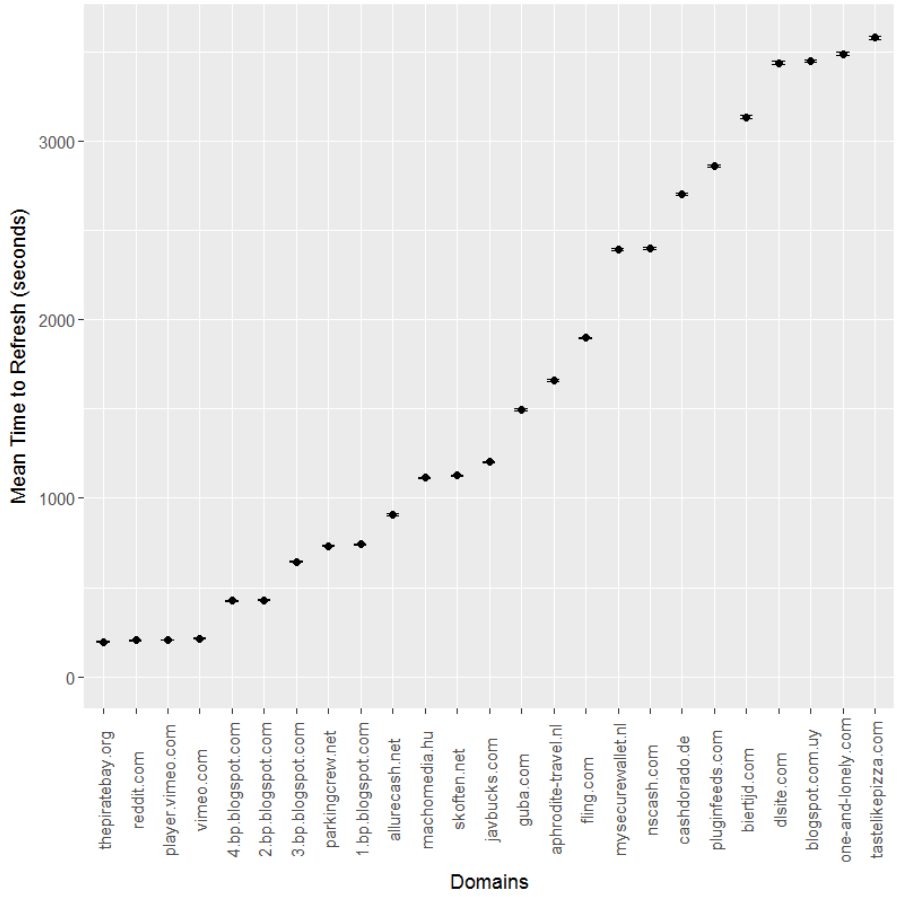


Figure 5.5: Most frequently accessed domains (Indonesia)

social networks, and a large amount of Google associated domains. Indonesia (seen in figure 5.5) and Turkey (seen in figure 5.6) both censor fewer of the highly refreshed domains.

When performing the language specific breakdown, only two of the countries seemed to be blocking a significant amount of local language websites. China and Iran were both found to be blocking local language sites, while Indonesia and Turkey seemed

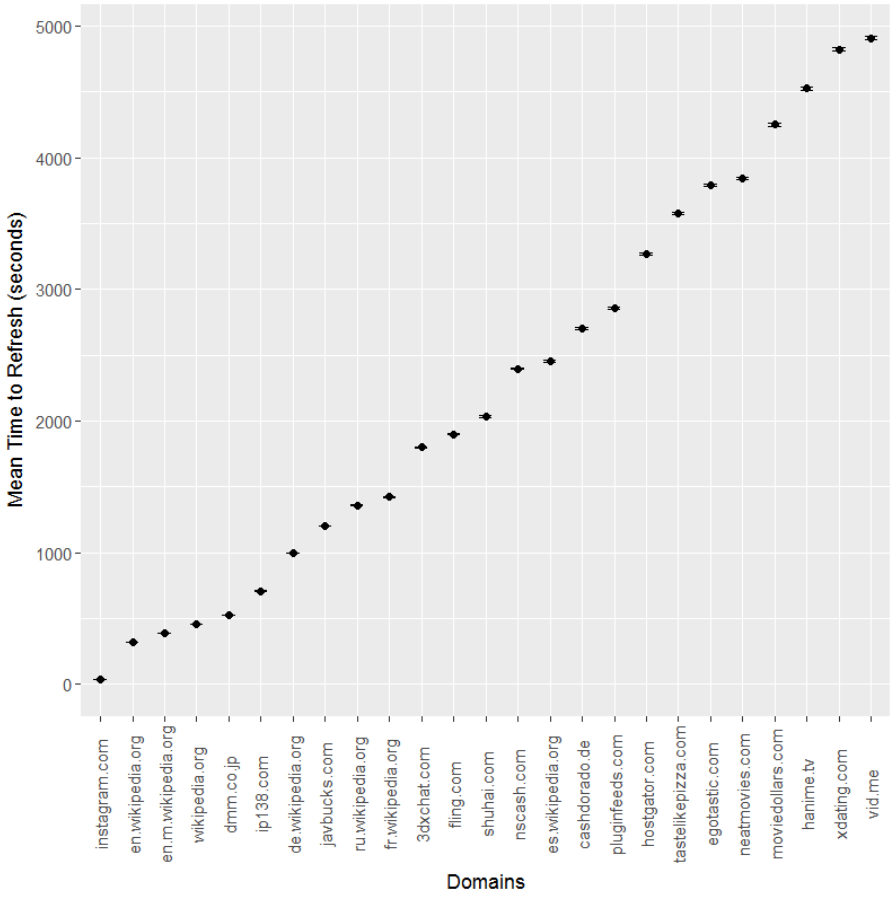


Figure 5.6: Most frequently accessed domains (Turkey)

to block domains primarily written in other languages. Most of the language specific domains were accessed fairly infrequently, but there are some that stand out.

Of the Farsi specific domains censored in Iran in Figure 5.7, radiofarda.com is highly refreshed. Radio Farda is the Farsi language station of Radio Free Europe/Radio Liberty, funded by the U.S government. This was by far the most frequently accessed Iranian blocked Farsi language domain discovered, and was refreshed on average every

150 seconds with 3000 hits over the observed period. It should be noted that there is a Radio Farda mobile app, so it may be affected in a similar way to some of the domains mentioned above. Similarly, in China voachinese.com is the most frequently refreshed blocked language specific domain in Figure 5.8. This is operated by Voice of America, again funded by the US government. Both Radio Liberty and Voice of America are US government 'soft power', and were recently classified as foreign agents by Russia [85]. Because of their funding they are not reliant on being commercially successful in their targeted countries. It is interesting that they are still accessed frequently despite being language specific and blocked in their targeted countries. As these languages are specific to the countries where the domain is blocked in, it is more likely that users of the VPN may be doing so specifically to access blocked content.

5.6 Limitations

While we have mentioned several limitations of this approach during our discussion it is worth laying them out formally. Some of the limitations are specific to our study, such as the inability to tell which country requests come from, while others are fundamental to DNS cache snooping

Cannot tell which country the domains are accessed from – The most significant limitation of our approach is that we have no way of knowing who made a request for a domain, only that someone did. Because of this we cannot be sure whether a domain was accessed to evade censorship or simply because users had their VPN active

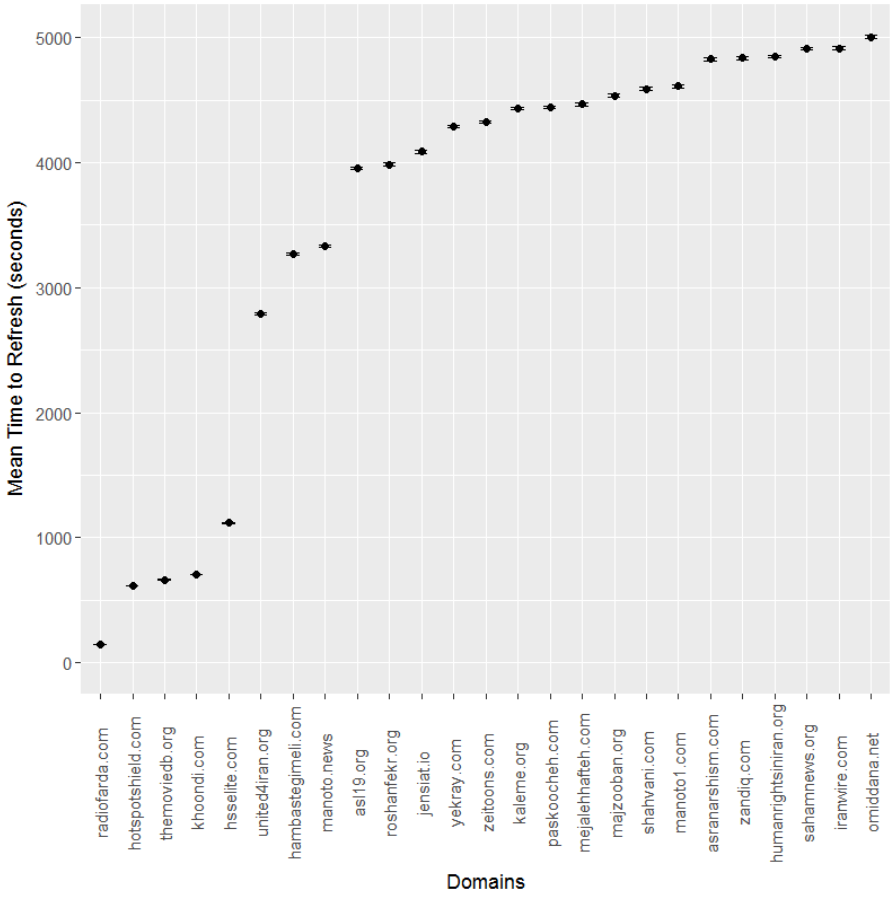


Figure 5.7: Most frequently accessed domains (Farsi language)

for other reasons. This is particularly true for highly popular censored domains such as facebook.com, instagram.com and google.com. Even if these domains were not censored in any country, it is likely they would still be highly trafficked through VPNs simply because of their popularity. For less frequented domains however this is less of an issue. This is especially true for those domains where the content language is only the first language in the region where the domain is blocked, such as those in Chinese and Farsi.

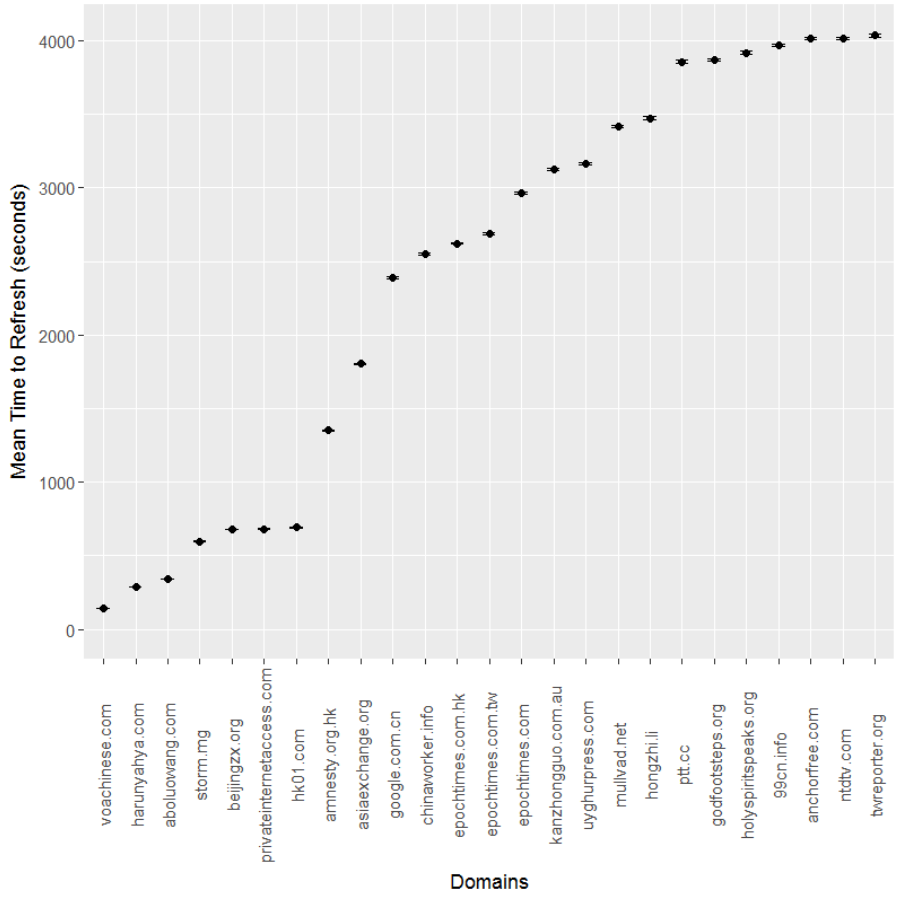


Figure 5.8: Most frequently accessed domains (Chinese language)

The issue of not being able to tell what was the cause of the refresh can be extrapolated to other issues, such as our inability to tell if a request has come from a human user or from an automated app. For some uses of DNS cache snooping this will not be an issue, but for others it renders the technique unusable.

Can only estimate refresh frequency – As our approach relies on new cached records being created, our approach will not observe a request if it was made while

there the record of that domain is already cached. Because of this we are not able to determine exactly how many requests occur over a period of time. Instead we have to estimate it, based on the time between the record expiring and it being refreshed.

If other researchers use the same technique it will corrupt the results – This technique only works as long as only one test is being run against a particular DNS server at a time. If multiple researchers are trying to perform the same technique on the same server it will corrupt the results.

Automated domain refreshing skews results – Apps and tools that periodically make requests for particular domains ensure that the cache record for that domain is requested far more frequently than they otherwise would be. As we are interested in users attempting to bypass censorship, we are less interested in apps that repeatedly poll for specific domains.

Dependent on the specific set up of the VPN service – The reason our study was limited to three VPN services was because most we looked at did not have their DNS set up in this way. Firstly, the majority of VPN services did not have internal DNS servers, and it was typically only the more mature paid services that provided this. Of those that did, several had strange set ups such as those described earlier, which were not susceptible or practical to be cache snooped.

5.7 Lessons

In this chapter we presented DNS cache snooping as a practical method that we used to determine how frequently domains were accessed over three large VPN services. We explored three different forms of DNS cache snooping, and discovered that cache snooping based on the TTL of DNS records is able to stably and consistently reveal information about the state of DNS records cache on a given server and how frequently these records are accessed. Using this approach, we analyse the internal DNS servers of several different VPN providers. These VPN providers advertise their service as a method of evading censorship and surveillance [41][51][90], and evidence suggests they are used for these purposes [6][80]. We ran this experiment for during April and May 2018, across three VPNs and 2000 domains.

During this period we discovered which of these domains were most regularly accessed through the VPN. By looking at the time it took for the records of these domains to be refreshed, we were able to calculate the overall frequency of access for each of these domains. This gave us: 1) an overall ranking of how frequently each domain was accessed through the VPN; 2) an estimate at the total number of requests for each domain through the VPN; 3) a breakdown of which sites and which type of sites are popular for users of VPNs. We did this for popular domains and again for domains we knew to be censored in specific countries: China, Indonesia, Iran, and Turkey. Two of the countries, China and Iran, were found to be blocking a large amount of websites written in their primary languages: Chinese and Farsi respectively. For these countries we provided

breakdowns of the language specific blocked domains that were accessed over the VPN.

5.8 Similar Research

There has not been much work on DNS cache snooping itself. A 2004 paper by Grangeia outlines the possibility of DNS cache snooping, but does not implement it or use it to measure relative popularity of domains [43]. In 2010 Krishnan et al. point out that a similar cache snooping technique may cause loss of privacy, specifically in the context of pre-caching web content [57]. The risk of DNS cache snooping is mentioned in briefly in RFC7626 [12].

Winter et al. [113] has worked on the privacy risk DNS queries pose to users of the Tor service. Their work looks at the risk of identifying users by the DNS requests going into and out of the Tor network. There has been some discussion over the role of VPNs as providers of privacy, anonymity and censorship resistance. One of the earliest evaluations was from Appelbaum et al. in 2012, who performed an assessment of VPNs for these purposes and found them lacking overall [6]. In 2015 Perta et al. performed a similar evaluation, with some time spent on DNS directly [80].

Other uses of DNS cache snooping

While we used DNS cache snooping to see which domains are being accessed over VPNs, there are other potential uses. One which is of interest to us is seeing if it can be used to discover how frequently attempts are made to access censored domains within DNS censored countries. The success of this technique depends on the specific censor-

ship apparatus within a country. The GFW's DNS poisoning is a good example of this. Users make a query for a domain, and the GFW responds with a poisoned DNS response. Users' machines cache this result, stopping them accessing censored content.

However, the original DNS response from the authentic DNS server is still sent [36]. This response can be captured to view the state of this record on the server, including the record's TTL value. By looking at this with our DNS cache snooping implementation it is possible to see how frequently users attempt to access censored domains. Preliminary results indicate this approach works, and it is possible to determine how frequently attempts are made to access censored domains within China for a chosen DNS server. This technique is not limited to China, and is likely applicable to other countries.

We have also looked at applying this technique to the Tor anonymity network. Tor allows clients to make DNS requests through the network, that are then resolved by an exit node. Requests through Tor have a static TTL so it is not as simple as running it against VPN DNS servers [3]. However, there is some degree of flexibility in how exit node operators handle DNS queries. Official guidance from The Tor Project suggests running caching resolvers, and if these can be accessed either from within the Tor network or the public internet this approach is feasible. This is arguably a security misconfiguration issue, although given the lack of specific guidance for DNS handling it is possible to be susceptible while following configuration guidance [96]. Indeed, we were able to proof of concept this against our own chosen-configuration exit node, but with a cursory search we were not able to find other exit nodes set up in a similar fashion. As a result

[3] Another way of doing this against the Tor network may be the timing based scan against an exit node, but Tor adds even more jitter noise which would need to be removed.

we were able to generate equivalent data on Tor domain frequency usage for our exit node, but as we relied upon the configuration of our own exit node to get it we do not believe it would be ethical to publish [54]. Greschbach et al. [46] wrote on the the effect of DNS on Tor's anonymity, while Phillip Winter et al. [113] has looked at the traffic flow identification via DNS requests sent through the Tor network.

5.9 Chapter Summary

In this chapter we presented DNS cache snooping as a practical method that we used to determine how frequently domains are accessed over three large VPN services. We explored three different forms of DNS cache snooping, and discovered that cache snooping based on the TTL of DNS records is able to stably and consistently reveal information about the state of DNS records cache on a given server and how frequently these records were accessed. Using this approach, we analysed the internal DNS servers of several different VPN providers. These VPN providers advertise their service as a method of evading censorship and surveillance [41][51][90], and evidence suggests they are used for these purposes [6][80]. We ran this experiment for during April and May 2018, across three VPNs and 2000 domains.

During this period we discovered which of these domains were most regularly accessed through the VPN. By looking at the time it took for the records of these domains to be refreshed, we were able to calculate the overall frequency of access for each of these domains. This gave us: 1) an overall ranking of how frequently each domain was accessed through the VPN; 2) an estimate at the total number of requests for each domain

through the VPN; 3) a breakdown of which sites and which type of sites are popular for users of VPNs. We did this for popular domains and again for domains we knew to be censored in specific countries: China, Indonesia, Iran, and Turkey.

Two of the countries, China and Iran, were found to be blocking a large amount of websites written in their primary languages: Chinese and Farsi respectively. For these countries we provided breakdowns of the language specific blocked domains that were accessed over the VPN.

The final chapter in this work focuses on the collaborative work carried out during this research period. It describes three published pieces of work, all focusing on censorship. The first piece builds upon the assumption queried in this chapter: that users use tools such as Tor and VPNs to evade censorship. It was resistance to this argument from other academics that encouraged us to study the DNS cache records of VPNs. The second two pieces focus on methods of discovering lists of censored domains, and came about from a need to have lists such as these. The lists of domains we produced with this collaborative work were the lists we referenced earlier in this chapter, and a method of identifying whether content is blocked in a country was used to verify whether domains were actually censored in a given country.

Chapter 6

Collaborative Work - Systematic Censorship Monitoring

6.1 Spotting censorship events by anomalies in Tor user numbers

In this section we discuss our work on using the Tor user figures to spot potentially censorship related network anomalies [115]. The primary author for this work was Joss Wright, who wrote most but not all of the paper. One significant section was written by the author of this thesis, and this author is listed as the third author. This work was published in the 10th ACM Conference on Web Science.

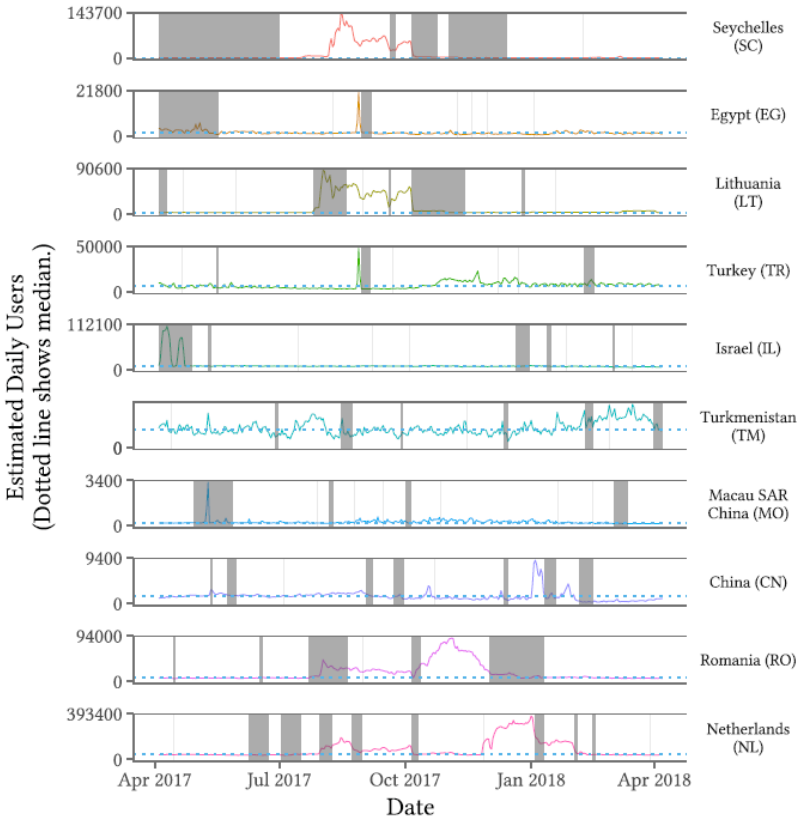
Tor publish their daily user count on the Tor Metrics website [97], separated by country. This data shows the total amount and daily fluctuations of Tor users in any given

country. When plotting this data it seems easy to eyeball anomalous trend-breaking behaviour. Why is Tor suddenly more popular in one country for two weeks, before dropping back to normal levels? We decided to plot these numbers and then algorithmically search for anomalous behaviour, to see how closely these anomalies matched with known censorship events.

While Tor is not designed primarily for censorship resistance, it is reasonable to believe that people are using it in that way. Over time it has become easier to use, and now provides an install bundle that lets users connect to the internet with an anonymous IP in minutes. While this functionality seems more inline with the primary goals of VPNs, VPN services aim to monetise users, either by charging for use or injecting ads into web pages, while Tor is free. Tor also provides a greater deal of anonymity than VPNs, which is an added advantage if users are accessing content they know to be banned by their government and they suspect government reprisals.

The algorithm we use to detect these anomalies is Principle Component Analysis (PCA), which is a statistical technique for identifying key patterns in complex datasets. With the Tor data set it allows us identify anomalous trends in one country, independently of global trends which affect multiple countries. Some of these anomalies may not be visible to the naked eye, such as a slow increase and decrease over a long period of time. But what caused that change? Other events may look anomalous, but actually be part of a regional or global trend that has nothing to do with the specifics of that country.

Using this approach we generate a daily list of alerts, which is in use by various academics and NGOs who are signed up to the list. These alerts show the most anomalous



[115]

Figure 6.1: Ten most anomalous countries

countries, and identify when any anomalies began and when they end.

While our approach uses data from Tor Metrics, it could easily absorb data from elsewhere. Our technique accepts datasets from other time series and then considers this when identifying anomalies. Using this approach it would be possible to collate data from a variety of sources, for example Tor user data, VPN user data, and data showing fluctuating numbers of users to certain websites.

One of the problems with researching internet censorship is that there is often no

accepted and comprehensive ground truth of when censorship events have happened. This made it difficult to calculate true false positive and false negative rates for observed anomalies.

To deal with this we tested our algorithm with generated synthetic time series data. This data contained overall global trends, and also our own injected and controlled anomalies. This allowed us to see the true false positive and false negative rates of our approach, and discover how sensitive it is to minor anomalies.

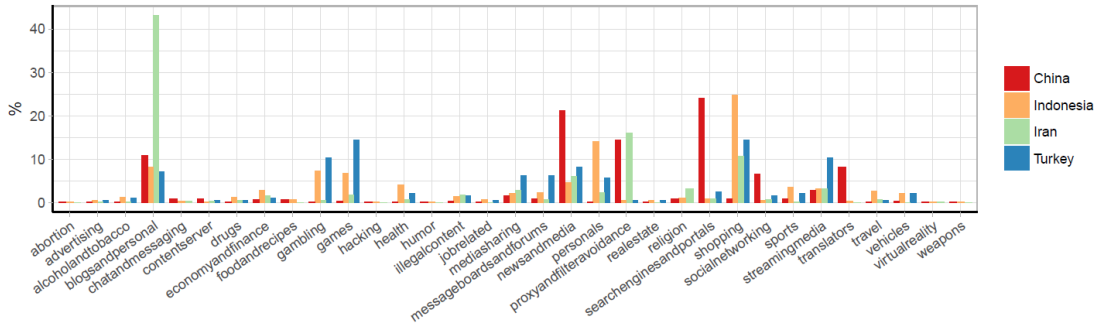
Full details and a copy of this paper can be found in the appendices.

6.2 Discovering blocked websites

In this section we look at two techniques to automate the discovery of blocked content [22][21]. This work was primarily carried out by Alex Darer, who also wrote the peer reviewed and published versions. The author of this thesis is listed as the second author in the published papers that came out of this research. The first paper was published in TMA 2017 Network Traffic Measurement and Analysis Conference, while the second paper was published in 2018 10th ACM Conference on Web Science.

There were times during our research that we needed large lists of censored domains to test against. While some lists existed, they were often missing domains that showed up in our own testing. To address this, we proposed an automated approach to discovering and listing censored content.

The approach outlined in our first paper takes an input list of known censored pages and then retrieves their content. We then break out the descriptive keywords that best



[21]

Figure 6.2: Category breakdown of filtered domains for each target country

define that website, and use a search engine to look for other pages that contain those words. These pages are then checked to see if they are censored, and if they are we repeat the process. This process occurs iteratively and indefinitely, discovering more censored web pages the longer the process runs.

This process was performed against the GFW, and discovered a list of 1355 blocked domains. This list is over 30 times the longer than the most widely-used published list available at the time.

The second paper on this work describes how we modified our process to take other countries into account. We increased our scope from just China, to also include Indonesia, Iran, and Turkey.

This approach discovered 1576 filtered domains in China, 47143 filtered domains in Indonesia, 651 filtererd domains in Iran, and 39725 filtered domains in Turkey. Many of the domains discovered were outside the Alexa Top 1000 [1] domains, and were not previously identified on other filtered domains lists.

Using this data we performed a breakdown of the content of the blocked websites

to analyse which countries are blocking which types of content. We found that the much higher numbers of blocked domains in Indonesia and Turkey were largely due to widespread bans on pornographic websites, while China blocks many news and media sites, and Iran blocks many blogs and personal sites.

Chapter 7

Conclusions

This work presents our research into internet censorship over the past four years. We have explored this by focusing on the side channels that these systems have, and inferring details of their operation where they cannot directly be observed. Censorship systems are rarely transparent about their functionality and operation, and so it is up to researchers to discover this.

We started off by looking for Deep Packet Inspection by the Great Firewall of China. When we could not find strong evidence for this, we began looking at the processes that we definitely knew existed. This led us down the path of focusing on DNS based censorship, initially by the GFW, and then on to other national censorship systems.

During this process we also became interested in censorship evasion techniques, primarily Tor and Virtual Private Networks. We used our knowledge of DNS to look at how these services handle name queries. Nothing concrete came from our evaluations of Tor's name queries, but using an old and esoteric technique we were able to infer how

frequently domains were accessed through several popular VPN services.

We also briefly covered the collaborative work that we produced. This work took place in close proximity to the main body of work presented, and compliments it. Specifically, the interest in censorship evasion came about from attempts to measure censorship events, and our collaborative work for discovering lists of censored domains came about out of necessity for such lists ourselves.

While our early work was aimed primarily at the GFW it was never our intentional focus, and our hope was that the techniques we have introduced can be turned to other censorship systems. We believe this is the case, and our later research has turned to multi-state censorship analyses.

7.1 Future

The battle for censorship is a red queen race [10] that has so far not slowed. Censors are in an arms race with censorship evasion technologies, and we are trying to understanding the effects of each of these.

We have seen them evolve just during the short period with which this work took place, and there is no reason to believe that this is going to slow. The internet opened access to information to people around the world, and governments want to control access to this information. The ways they do this shapes the internet for all of us, and as academics we believe it is important to understand what is taking place.

Apart from the generalities of future internet research, there are some specific lines to follow that have come out of our work.

While updating research for chapter 4 of this work, we discovered that the IP addresses that the GFW is redirecting to have changed. What was the primary redirected IP is now only associated with a handful of domains, while some of the other IPs still seem to be in widespread use. Why is this, and what has prompted this change? It would give insight into the changes behind the GFW if we broke this down by network topology mapping to see where and when these changes have taken place over time.

Additionally, the list of domains pulled from DNSDB that are associated with the null 9 IP addresses has not properly been explored. In our original published work details of these addresses made only a small section towards the end of the paper. There are undoubtedly interesting insights that can come from this list by those who are exploring the social and political issues around censorship.

In a similar vein, the domains pulled from the VPN DNS cache snooping work give some idea what users are actually using these services for. Our work was focused on the inference technique itself, and the analysis of the output was shallow.

Our approach of using DNS cache snooping to calculate domain frequency can be applied to other contexts, not just VPNs. This technique can be applied to almost any organisation that operates its own DNS servers. While we use it to attempt to gauge popularity of domains it could also be turned to other scenarios, such as discovering what software is in use (e.g. windowsupdate.microsoft.com or sadownload.mcafee.com) or estimating the number of hosts on the network (e.g. pool.ntp.org).

Perhaps this is the nature of research, but our work seems to have opened up as many questions as it has answered. Our emphasis on technique and data generation was deliberate, but it is frustrating to leave these threads unexplored. This emphasises

to us the importance of interdisciplinary work when looking at the internet.

With the increasing centrality of the internet to our everyday lives, it is important that we attempt to understand it. Sometimes this research will be assisted by those organisations hosting the internet, but sometimes it must be adversarial. Without this kind of study we would be missing a vital piece of understanding of the internet.

Chapter 8

Glossary

ACK - Acknowledge. The second part of the TCP connection handshake. The ack is the response of acknowledgement of the original message.

A Record - The most basic type of DNS record, used to bind an IP address to a domain or subdomain name. When a requester makes a request for a domain, their DNS resolver tries to find the A record for that domain to discover the IP address. This IP address is used for internetwork routing.

AS - Autonomous System . A region of the internet under control of a single entity. Traffic between ASs is routed using BGP.

ASN - Autonomous System Number. A unique number that references a specific Autonomous System.

Authoritative server - The DNS server that contains the authoritative record for a specific domain.

BGP - Border Gateway Protocol. The protocol that is used to route internet packets

between different AS.

Blackholing - A BGP technique to cause traffic to be routed to a null address, in order to prevent traffic reaching its destination. Can be used to filter traffic for either censorship purposes or to stop DDoS requests reaching their target.

CDN - Content Delivery Network. A set of servers over the internet that distribute web content. Normally geolocated to limit the distance resources have to travel to reach client hosts.

Ciphertext - Data that has been encrypted.

Cleartext - Data sent 'in the clear', e.g. not encrypted.

Data in transit - Data that exists on the network, and is in the state of being sent between hosts or nodes. The antithesis of 'data at rest'.

DSoS - Distributed Denial of Service. An attack that uses distributed hosts to generate a large amount of traffic to a server with the intention of preventing it dealing with other requests.

DoS - Denial of Service. Any attack that prevents a service functioning.

DNS - Domain Name System. The distributed system for converting human readable domain names to machine routable IP addresses.

DNS Poisoning - A technique for corrupting DNS entries with an incorrect answer.

DNS record - A record that can be retrieved using the DNS system. The most common type of record is an A type record, which returns an IP address for a given domain name.

DNS Redirection - A technique for redirecting user attempts to access a web page to a different page.

DoH - DNS over HTTPS. Using HTTPS to provide encrypted DNS queries.

Domain - A human name for a host or group of hosts. Can be accessed over the internet via and internet domain name, or can be used to organise hosts on smaller networks. The concept of a domain has become more abstract over time, with the internet expanding using technologies such as Content Delivery Networks and load balancers.

Domain Name - The name of a domain. This could be a network name, but within the context of the internet typically refers to the first part of the address path, where a user wants to request information from. The Domain Name System converts domain names into IP addresses.

Encryption - The process of scrambling data so that it can only be read with the knowledge of a specific decryption key.

FTP - File Transfer Protocol. A protocol for transferring files from one host to another.

Hostname - The name of a specific computer.

HTTP - Hypertext Transfer Protocol. Stateless application layer protocol for requesting content. Typically used for web content, sitting above TCP/IP, and below HTML.

HTTPS - Secure Hypertext Transfer Protocol. Uses SSL or TLS encryption to encrypt the web content.

IP - Internet Protocol. The networking protocol that is used to route traffic between different networks, and which is responsible for carrying traffic over the internet.

IP Address - Internet Protocol Address. The routing address that network nodes use to route traffic between different networks.

IPID - Internet Protocol Identifier. IP Packet field that is used to split and then merge IP packets if they run over network infrastructure with a limited packet size. Can be used in censorship research to infer information about the source of the packet.

ISP - Internet Service Provider. An organisation offering access to the internet. Often, although not always, runs its own network(s) as ASs. Some ISPs do not run their own ASs, and exist via sharing agreements with different organisations.

MTU - Maximum Transmission Units. The maximum packet size that network infrastructure can transfer.

Network layer - The layer that joints different computer networks together and allows data to be routed between them. IP is the biggest network layer protocol in use today.

Node - A piece of network infrastructure that receives packets and passes them to the next node in the path.

Obfuscation - The process of hiding data, making it difficult to read. Obfuscation is normally a reversible process, making it possible to retrieve the original data.

PKI - Public Key Infrastructure. Internet wide infrastructure for distributing public key certificates between users in a trusted way. Its primary practical use is the verify the identity of websites to a user's browser when a secure (HTTPS) site is accessed.

Plaintext - Data intended for encryption, before it is encrypted.

Protocol - A standardised way of communicating over the network.

Root server - A DNS server sitting at the highest tier of the DNS architecture. It is responsible for answering requests for records in the DNS root zone.

RST - Reset. A TCP flag telling the receiver to close the connection.

SSH - Secure Shell. An encrypted protocol for administrating a host remotely.

SSL - Secure Socket Layer. An early method for encrypting internet traffic.

Subdomain - A organisational domain that is a part of a larger domain. In a URL it is represented by the string before the domain name - www is a common subdomain.

SYN - Synchronise. The first part of the TCP connection handshake.

TCP - Transmission Control Protocol. A connected, reliable and ordered transport layer protocol for transferring data between two hosts. It is the commonly used for most internet services, including web content. DNS has the capability to transfer data over TCP, although it defaults to using UDP instead for small queries.

TLD - Top Level Domain. A TLD is a domain that has no other domains sitting above it. Common TLDs include .com, .net, and .uk.

TLS - Transport Layer Security. A replacement for SSL, for encrypting internet traffic.

Tor - The Onion Router. An onion routing protocol designed to provide anonymity to users and web services.

TTL - Time To Live. Header fields in both the IP and DNS packet headers. In the IP header this field counts down the number of hops that the packet has travelled. In the DNS header this field indicates how long in seconds before the cache expires.

UDP - User Datagram Protocol. A best effort transport layer protocol. Unlike TCP, UDP makes no guarantee that messages sent will be received by the target host. While it is used primarily for local network services, it is also the primary transport layer protocol for internet wide user DNS queries.

Web - This is information content typically transferred over HTTP, written in HTML, and accessed in a web browser. 'Web' should be differentiated from 'Internet', as the Internet includes many features and services that are not part of the web.

Bibliography

[1] Alexa, *Million Domains*, `https://www.alexa.com/` (visited on 10/03/2018).

[2] *Alexa - Actionable Analytics for the Web*, (2016) `http://www.alexa.com/` (visited on 08/25/2018).

[3] Anonymous, "The collateral damage of internet censorship by DNS injection", ACM SIGCOMM CCR **42** (2012).

[4] Anonymous, "Towards a Comprehensive Picture of the Great Firewall's DNS Censorship", in USENIX Workshop on Free and Open Communications on the Internet (USENIX Association, 2014).

[5] *APNIC - Apia-Pacific Network Information Centre*, `https://www.apnic.net/` (visited on 08/31/2018).

[6] J. Appelbaum, M. Ray, K. Koscher, and I. Finder, "vpwns: Virtual Pwned Networks", in 2nd USENIX Workshop on Free and Open Communications on the Internet (USENIX Association, 2012).

[7] S. Aryan, H. Aryan, and J. A. Halderman, “Internet Censorship in Iran: A First Look.”, in USENIX Workshop on Free and Open Communications on the Internet (USENIX Association, 2013).

[8] G. Ateniese and S. Mangard, “A new approach to DNS security (DNSSEC)”, in Proceedings of the 8th ACM conference on Computer and Communications Security (ACM, 2001), pp. 86–95.

[9] S. Bano, P. Richter, M. Javed, S. Sundaresan, Z. Durumeric, S. J. Murdoch, R. Mortier, and V. Paxson, “Scanning the Internet for Liveness”, ACM SIGCOMM Computer Communication Review **48**, 2–9 (2018).

[10] W. P. Barnett and M. T. Hansen, “The Red Queen in Organizational Evolution”, Strategic Management Journal **17**, 139–157 (1996).

[11] R. W. Borders et al., “China: Journey to the Heart of Internet Censorship”, (2007).

[12] S. Bortzmeyer, *RFC 7626 - DNS Privacy Considerations*, (2015) `https://tools.ietf.org/html/rfc7626` (visited on 05/25/2018).

[13] M. A. Brown, D. Madory, A. Popescu, and E. Zmijewski, “DNS Tampering and Root Servers”, Renesys Corporation (2010).

[14] R. Bush, M. A. Patton, R. Elz, and S. Bradner, “RFC: 2182 - Selection and Operation of Secondary DNS Servers”, Consultant (1997).

[15] E. Butler, *Firesheep*, (2010) `https://codebutler.github.io/firesheep/` (visited on 10/03/2018).

[16] Citizen Lab, *Psiphon*, (2018) `https://psiphon.ca/` (visited on 05/15/2018).

[17] R. Clayton, S. J. Murdoch, and R. N. Watson, "Ignoring the great firewall of China", in Privacy Enhancing Technologies (Springer, 2006), pp. 20–35.

[18] K. Cohn-Gordon, C. Cremers, B. Dowling, L. Garratt, and D. Stebila, "A Formal Security Analysis of the Signal Messaging Protocol", in Security and Privacy (EuroS&P), 2017 IEEE European Symposium on (IEEE, 2017), pp. 451–466.

[19] J. R. Crandall, D. Zinn, M. Byrd, E. T. Barr, and R. East, "ConceptDoppler: a weather tracker for internet censorship.", in ACM Conference on Computer and Communications Security (2007), pp. 352–365.

[20] G. Csardi and T. Nepusz, "The igraph software package for complex network research", InterJournal, Complex Systems **1695**, 1–9 (2006).

[21] A. Darer, O. Farnan, and J. Wright, "Automated Discovery of Internet Censorship by Web Crawling", Web Science Conference (2018).

[22] A. Darer, O. Farnan, and J. Wright, "FilteredWeb: A framework for the automated search-based discovery of blocked URLs", in Network Traffic Measurement and Analysis Conference (TMA), 2017, pages=1–9 (IEEE, 2017).

[23] R. J. Deibert, J. G. Palfrey, R. Rohozinski, and J. Zittrain, *Access Denied: The Practice and Policy of Global Internet Filtering (Information Revolution and Global Politics)* (2008).

[24] F. Denis and Y. Fu, *DNSCrypt*, `https://dnscrypt.info/`.

[25] S. Dharmapurikar, P. Krishnamurthy, T. Sproull, and J. Lockwood, "Deep Packet Inspection using Parallel Bloom Filters", in 11th symposium on High performance interconnects (IEEE, 2003), pp. 44–51.

[26] *DIG DNS Lookup Utility - Linux man page*, (2018) `https://linux.die.net/man/1/dig` (visited on 10/03/2018).

[27] R. Dingledine, N. Mathewson, and P. Syverson, *Tor: The Second-Generation Onion Router*, tech. rep. (Naval Research Lab Washington DC, 2004).

[28] DNS, *Google Public DNS*, (2018) `https://developers.google.com/speed/public-dns/` (visited on 05/24/2018).

[29] Z. Durumeric, E. Wustrow, and J. A. Halderman, "ZMap: Fast Internet-wide Scanning and Its Security Applications.", in USENIX Security Symposium, Vol. 8 (USENIX Association, 2013), pp. 47–53.

[30] Dyn, *Pakistan Hijacks YouTube*, (2008) `https://dyn.com/blog/pakistan-hijacks-youtube-1/` (visited on 10/03/2018).

[31] R. Elz and R. Bush, *RFC: 2181 - Clarifications to the DNS Specification*, tech. rep. (1997).

[32] R. Ensafi, J. Knockel, G. Alexander, and J. R. Crandall, "Detecting intentional packet drops on the Internet via TCP/IP side channels", in Passive and Active Measurement (Springer, 2014), pp. 109–118.

[33] R. Ensafi, J. C. Park, D. Kapur, and J. R. Crandall, "Idle Port Scanning and Non-interference Analysis of Network Protocol Stacks Using Model Checking", in USENIX Security Symposium (USENIX Association, 2010), pp. 257–272.

[34] M. V. Ereche., *Odd Behaviour on One Node in I root-server*, (2010) https://lists.dns-oarc.net/pipermail/dnsoperations/2010-March/005260.html (visited on 08/31/2018).

[35] O. Farnan, A. Darer, and J. Wright, "Analysing Censorship Circumvention with VPNs via DNS Cache Snooping", in Proceedings of the 2019 IEEE Workshop on Traffic Measurements for Cybersecurity (IEEE, 2019).

[36] O. Farnan, A. Darer, and J. Wright, "Poisoning the Well: Exploring the Great Firewall's Poisoned DNS fResponses", in Proceedings of the 2016 ACM on Workshop on Privacy in the Electronic Society (ACM, 2016), pp. 95–98.

[37] *Farsight Security - DNS Database. https://www.dnsdb.info/*, https://www.dnsdb.info/ (visited on 08/28/2015).

[38] A. Filastò and J. Appelbaum, "OONI: Open Observatory of Network Interference", USENIX Workshop on Free and Open Communications on the Internet (2012).

[39] Freedom House, "Freedom on the Net", (2009).

[40] *GFW Technology Review*, (2010) http://gfwrev.blogspot.com/ (visited on 08/31/2018).

[41] Golden Frog, *How To Bypass Censorship in Russia*, (2018) https://www.goldenfrog.com/vyprvpn/guides/how-to-bypass-censorship-russia (visited on 05/15/2018).

[42] R. Graham, *Mass Scan*, (2013) https://github.com/robertdavidgraham/masscan (visited on 10/03/2018).

[43] L. Grangeia, *DNS Cache Snooping or Snooping the Cache for Fun and Profit*, (2004) http://cs.unc.edu/fabian/course%5C_papers/cache%5C_snooping.pdf (visited on 05/01/2018).

[44] *Online Censorship In China — GreatFire.org*, (2016) https://en.greatfire.org/ (visited on 08/31/2018).

[45] G. Greenwald and E. MacAskill, "NSA Prism program taps into user data of Apple, Google and others", The Guardian **7**, 1–43 (2013).

[46] B. Greschbach, T. Pulls, L. M. Roberts, P. Winter, and N. Feamster, "The Effect of DNS on Tor's Anonymity", arXiv preprint arXiv:1609.08187 (2016).

[47] K. P. Gummadi, S. Saroiu, and S. D. Gribble, "King: Estimating Latency Between Arbitrary Internet End Hosts", in Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment (ACM, 2002), pp. 5–18.

[48] B. Haas, *China moves to block internet VPNs from 2018*, (2017) https://www.theguardian.com/world/2017/jul/11/china-moves-to-block-internet-vpns-from-2018 (visited on 05/15/2018).

[49] A. Hern, *Apple defies FBI and offers encryption by default on new operating system*, (2014) https://www.theguardian.com/technology/2014/oct/17/apple-defies-fbi-encryption-mac-osx (visited on 10/03/2018).

[50] P. Hunter, "Pakistan YouTube block exposes fundamental internet security weakness: Concern that pakistani action affected youtube access elsewhere in world", Computer Fraud & Security **2008**, 10–11 (2008).

[51] IPVanish, *Bypass Censorship*, (2018) https://www.ipvanish.com/bypass-censorship.php (visited on 05/15/2018).

[52] N. M. Jennifer Valentino-DeVries Paul Sonne, *U.S. Firm Acknowledges Syria Uses Its Gear to Block Web*, https://www.wsj.com/articles/SB10001424052970203687504577001911398596328 (visited on 07/03/2019).

[53] I. Jolliffe, "Principal Component Analysis", in *International Encyclopedia of Statistical Science* (Springer, 2011), pp. 1094–1096.

[54] B. Jones, R. Ensafi, N. Feamster, V. Paxson, and N. Weaver, "Ethical Concerns for Censorship Measurement", in Proceedings of the 2015 ACM SIGCOMM Workshop on Ethics in Networked Systems Research (ACM, 2015), pp. 17–19.

[55] D. Kaminsky, *Scanrand*, (2013) http://www.vulnerabilityassessment.co.uk/scanrand.htm (visited on 10/03/2018).

[56] G. King, J. Pan, and M. E. Roberts, "How censorship in China allows government criticism but silences collective expression", American Political Science Review **107**, 326–343 (2013).

[57] S. Krishnan and F. Monrose, "DNS Prefetching and Its Privacy Implications: When Good Things Go Bad", in Proceedings of the 3rd USENIX conference on Large-scale exploits and emergent threats: botnets, spyware, worms, and more (USENIX Association, 2010), pp. 10–10.

[58] H. Kuchler, *Apple removes apps that bypass China's censors*, (2017) https://www.ft.com/content/e83e8034-7543-11e7-90c0-90a9d1bc9691 (visited on 05/15/2018).

[59] M. Kumar, *Kazakhstan Begins Intercepting HTTPS Internet Traffic Of All Citizens Forcefully*, https://thehackernews.com/2019/07/kazakhstan-https-security-certificate.html (visited on 07/19/2019).

[60] A. Langley, "Maintaining digital certificate security", Google Online Security Blog (2015).

[61] R. Lemos, *Blue Coat Appliances Used by Governments to Monitor, Censor Web Traffic*, https://www.eweek.com/security/blue-coat-appliances-used-by-governments-to-monitor-censor-web-traffic (visited on 07/03/2019).

[62] J. Louis, *Unicornscan*, (2013) https://tools.kali.org/information-gathering/unicornscan (visited on 10/03/2018).

[63] G. Lowe, P. Winters, and M. L. Marcus, "The Great DNS Wall of China", MS, New York University **21** (2007).

[64] G. F. Lyon, *Nmap network scanning: The official Nmap project guide to network discovery and security scanning* (Insecure, 2009).

[65] Majestic Million, *Welcome to Majestic Million*, (2018) `https://blog.majestic.com/welcome-to-majestic-million/` (visited on 05/01/2018).

[66] B. Marczak, N. Weaver, J. Dalek, R. Ensafi, D. Fifield, S. McKune, A. Rey, J. Scott-Railton, R. Deibert, and V. Paxson, "China's Great Cannon", Citizen Lab **10** (2015).

[67] M. Marlinspike, *Advanced cryptographic ratcheting*, (2013) `https://signal.org/blog/advanced-ratcheting/` (visited on 05/15/2018).

[68] *MaxMind: IP Geolocation and Online Fraud Prevention*, (2016) `https://www.maxmind.com/` (visited on 08/29/2018).

[69] P. V. Mockapetris, *RFC: 1034 - Domain names - concepts and facilities*, tech. rep. (1987).

[70] P. V. Mockapetris, *RFC: 883 - Domain names: Implementation specification*, tech. rep. (1983).

[71] Z. Nabi, "The Anatomy of Web Censorship in Pakistan", in USENIX Workshop on Free and Open Communications on the Internet (USENIX Association, 2013).

[72] H. Nikšic, *GNU Wget*, `https://www.gnu.org/software/wget/` (visited on 08/17/2018).

[73] NordVPN, *NordVPN - DNS Leak Test*, (2018) https://nordvpn.com/features/dns-leak-test/ (visited on 05/25/2018).

[74] *North Korea Tech*, (2018) http://www.northkoreatech.org/ (visited on 10/03/2018).

[75] *Opening up North Korea*, (2018) http://www.openingupnorthkorea.com/downloads-2 (visited on 10/03/2018).

[76] P. Papadopoulos, N. Kourtellis, and E. P. Markatos, "Exclusive: How the (synced) Cookie Monster breached my encrypted VPN session", in Proceedings of the 11th European Workshop on Systems Security (ACM, 2018), p. 6.

[77] J. C. Park and J. R. Crandall, "Empirical Study of a National-Scale Distributed Intrusion Detection System: Backbone-Level Filtering of HTML Responses in China", in Distributed Computing Systems (ICDCS), 2010 IEEE 30th International Conference on (IEEE, 2010), pp. 315–326.

[78] K. Park and W. Willinger, "The Internet as a Large-Scale Complex System", Oxford University Press **1** (2005).

[79] *Public DNS Server List*, (2016) http://public-dns.info/ (visited on 08/31/2018).

[80] V. C. Perta, M. V. Barbera, G. Tyson, H. Haddadi, and A. Mei, "A glance through the VPN looking glass: IPv6 leakage and DNS hijacking in commercial VPN clients", Proceedings on Privacy Enhancing Technologies **2015**, 77–91 (2015).

[81] J. Postel, *RFC 862 - Echo Protocol*, (1983) https://tools.ietf.org/html/rfc862 (visited on 05/21/2019).

[82] L. Rainie, S. Kiesler, R. Kang, M. Madden, M. Duggan, S. Brown, and L. Dabbish, "Anonymity, Privacy, and Security Online", Pew Research Center **5** (2013).

[83] E. Rauhala, *New evidence emerges of China forcing Muslims into reeducation camps*, (2018) https://www.washingtonpost.com/world/asia%5C_pacific/new-evidence-emerges-that-china-is-forcing-muslims-into-reeducation-camps/2018/08/10/1d6d2f64-8dce-11e8-9b0d-749fb254bc3d%5C_story.html (visited on 09/17/2018).

[84] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, and E. Lear, *Address Allocation for Private Internets*, tech. rep. (1996).

[85] Reuters, *Russia designates Radio Free Europe and Voice of America as foreign agents*, (2017) https://www.reuters.com/article/us-russia-usa-media-restrictions/russia-designates-radio-free-europe-and-voice-of-america-as-foreign-agents-idUSKBN1DZ0MP (visited on 09/15/2018).

[86] J. Riesa, *ropensci/cld3*, (2018) https://github.com/ropensci/cld3) (visited on 05/16/2018).

[87] *RIPE atlas*, https://atlas.ripe.net/ (visited on 10/03/2018).

[88] Robtex, *Welcome to Robtex!*, (2018) https://www.robtex.com (visited on 10/03/2018).

[89] M. Roesch et al., "Snort: Lightweight Intrusion Detection for Networks", in Lisa, Vol. 99, 1 (1999), pp. 229–238.

[90] C. Roswell, *BYPASS INTERNET CENSORSHIP WITH VPN/PROXIES*, (2018) https://thevpn.guru/bypass-internet-censorship-surveillance-vpn-tor-proxies (visited on 05/15/2018).

[91] I. Savchenko and O. Y. Gatsenko, "Analytical Review of Methods of Providing Internet Anonymity", Automatic Control and Computer Sciences **49**, 696–700 (2015).

[92] Q. Scheitle, O. Hohlfeld, J. Gamba, J. Jelten, T. Zimmermann, S. D. Strowes, and N. Vallina-Rodriguez, "A Long Way to the Top: Significance, Structure, and Stability of Internet Top Lists", in Proceedings of the Internet Measurement Conference 2018 (ACM, 2018), pp. 478–493.

[93] B. Schneier, *How the NSA Attacks Tor/Firefox Users With QUANTUM and FOXACID.* https://www.schneier.com/blog/archives/2013/10/how_the_nsa_att.html (visited on 08/31/2018).

[94] R. Sekar, Y. Guang, S. Verma, and T. Shanbhag, "A High-Performance Network Intrusion Detection System", in Proceedings of the 6th ACM Conference on Computer and Communications Security (ACM, 1999), pp. 8–17.

[95] *Tor:Overview. https://www.torproject.org/about/overview.html.en*, https://www.torproject.org/about/overview.html.en (visited on 08/31/2018).

[96] Tor Project, *Tor Relay Guide*, (2018) https://trac.torproject.org/projects/tor/wiki/TorRelayGuide (visited on 05/16/2018).

[97] Tor Project, *Welcome to Tor Metrics*, (2018) https://metrics.torproject.org/ (visited on 10/01/2018).

[98] TorGuard, *TorGuard DNS Leak Test*, (2018) https://torguard.net/vpn-dns-leak-test.php (visited on 05/25/2018).

[99] TorProject.org, *Knock Knock Knockin' on Bridges' Doors*, (2012) https://blog.torproject.org/knock-knock-knockin-bridges-doors/ (visited on 09/24/2018).

[100] K. Townsend, *DNS Leaks*, (2018) https://thebestvpn.com/dns-leaks-causes-fixes/ (visited on 05/25/2018).

[101] M. C. Tschantz, S. Afroz, V. Paxson, et al., "SoK: Towards Grounding Censorship Circumvention in Empiricism", in Security and Privacy, 2016 IEEE Symposium on (IEEE, 2016), pp. 914–933.

[102] University of Oregon, *RouteViews Project*, http://www.routeviews.org (visited on 10/03/2018).

[103] B. VanderSloot, A. McDonald, W. Scott, J. A. Halderman, and R. Ensafi, "Quack: Scalable Remote Measurement of Application-Layer Censorship", in Proceedings of the 27th USENIX Security Symposium (2018).

[104] B. VanderSloot, A. McDonald, W. Scott, J. A. Halderman, and R. Ensafi, "Quack: Scalable Remote Measurement of Application-Layer Censorship", in Proceedings of the 27th USENIX Security Symposium (2018).

[105] A. Vetterl and R. Clayton, "Bitter harvest: systematically fingerprinting low-and medium-interaction honeypots at internet scale", in 12th usenix workshop on offensive technologies - woot 18) (2018).

[106] A. Vetterl, R. Clayton, and I. Walden, "Counting Outdated Honeypots: Legal and Useful", in Proceedings of the 2019 IEEE Workshop on Traffic Measurements for Cybersecurity (IEEE, 2019).

[107] *View DNS Info*, (2016) `http://http://viewdns.info/` (visited on 07/30/2018).

[108] Z. Wang, Y. Cao, Z. Qian, C. Song, and S. V. Krishnamurthy, "Your State is not Mine: A Closer Look at Evading Stateful Internet Censorship", in Proceedings of the 2017 Internet Measurement Conference (ACM, 2017), pp. 114–127.

[109] N. Weaver, R. Sommer, and V. Paxson, "Detecting Forged TCP Reset Packets.", in NDSS (2009).

[110] F. Weimer, "Passive DNS Replication", in FIRST Conference on Computer Security Incident (2005), p. 98.

[111] T. Wilde, "Great Firewall Tor Probing Circa 09 DEC 2011", URl: https://gist. github. com/da3c7a9af01d74cd7de7 (2011).

[112] P. Winter, "Towards a censorship analyser for Tor", in 3rd USENIX Workshop on Free and Open Communications on the Internet (USENIX Association, 2013).

[113] P. Winter, R. Köwer, M. Mulazzani, M. Huber, S. Schrittwieser, S. Lindskog, and E. Weippl, "Spoiled Onions: Exposing Malicious Tor Exit Relays", in International Symposium on Privacy Enhancing Technologies Symposium (Springer, 2014), pp. 304–331.

[114] P. Winter and S. Lindskog, "How the great firewall of China is blocking Tor", Free and Open Communications on the Internet (2012).

[115] J. Wright, A. Darer, and O. Farnan, "On Identifying Anomalies in Tor Usage with Applications in Detecting Internet Censorship", (2018).

[116] J. Wright, "Regional variation in Chinese internet filtering", Information, Communication & Society **17**, 121–141 (2014).

[117] J. Wright, A. Darer, and O. Farnan, "Detecting Internet Filtering from Geographic Time Series", arXiv preprint arXiv:1507.05819 (2015).

[118] X. Xu, Z. M. Mao, and J. A. Halderman, "Internet censorship in China: Where does the filtering occur?", in Passive and Active Measurement (Springer, 2011), pp. 133–142.

[119] J. Yonan, *OpenVPN*, (2001) `https://openvpn.net/` (visited on 05/25/2018).

[120] F. Yu, Z. Chen, Y. Diao, T. Lakshman, and R. H. Katz, "Fast and Memory-Efficient Regular Expression Matching for Deep Packet Inspection", in Proceedings of the 2006 ACM/IEEE symposium on Architecture for networking and communications systems (ACM, 2006), pp. 93–102.

[121] J. Zittrain and B. Edelman, "Internet Filtering in China", IEEE Internet Computing **7**, 70–77 (2003).

[122] E. Zmijewski, *Accidentally Importing Censorship*, (2010) `https://dyn.com/blog/fouling-the-global-nest/` (visited on 09/18/2018).

[123] E. Zmijewski, *Turkish Internet Censorship Takes a New Turn*, (2014) `https://dyn.com/blog/turkish-internet-censorship/` (visited on 09/20/2018).