University of Dublin

TRINITY COLLEGE

A Survey of Web Servers in Trinity College Dublin

Michael Power
B.A. (Mod.) Integrated Computer Science

Final Year Project May 2018
Supervisor: Dr. Stephen Farrell

School of Computer Science and Statistics

O'Reilly Institute, Trinity College, Dublin 2, Ireland

Declaration

I hereby declare that this thesis is entirely my own work and that it has not been submitted as an exercise for a degree at any other university.

_______________________________________ April 26, 2018

Michael Power

Permission to Lend

I agree that the Library and other agents of the College may lend or copy this thesis upon request.

_________________________________ April 26, 2018

Michael Power

Acknowledgements

To be done

Contents

List of Figures

Abstract

As the number of internet devices grows, so to does the difficultly to monitor these devices effectively. This report details the use of ZMap a port scanner and ZGrab an application banner grabber within Trinity College Dublin To survey Web Servers. System administrators have often hundreds of hosts to consider when monitoring Web Servers. The use of the above tools to audit these Web Servers in order to deal with security issues that if left unattended could potential lead to devastating effects is of the utmost importance for any organisation that aims to mitigate these risks, as well as using these tools to study vulnerabilities in order to better defend from attacks, since the availability of tools such as these leads to the potential of attackers finding vulnerability hosts. Scanning at an Internet wide level has shown great promise for uncovering security problems [3] thus the same should be true at a University Campus level.

As well as deploying and testing the tool within Trinity College Dublin, I also hope to be able to interpret the output, and communicate that to site owners/system admins in order to help make their web a bit better and more secure.

Chapter 1

Introduction

1.1 Goals of Report

As with any device that is connected to the internet, the security of these systems is of concern. Even when following strict policies, miss configurations of servers can lead to possible issues which could significant impact. The aim of this project is to deploy a local instance of ZMap and ZGrab within Trinity College Dublin, running scans on both port 80 and 443 in order to survey web servers, in an effort to build a picture of what the current state of Web Servers within the college campus looks like.

I will also outline the steps that I've taken in order for other organisations and institutions to replicate what I've done here within Trinity College Dublin.

This goal will be analysed with the following questions in mind:

- Which IP addresses are listening on port 80, port 443 and both ports?
- The variation in the number of host on at a certain time?
- How many IP addresses resolve to a Hostname?
- What make of servers are being used?
- What does the current state of Certificates look like within the college?
- Are there any out of date versions of TLS being used?
- For those IP addresses that don't resolve to a Hostname, do their certificates lead to one?
- What Signature and Key Algorithms are being used?
- Is secure renegotiation false for any IP addresses?
- Are there any public keys that are used across multiple IP addresses?
- The variation in public key lengths?
- How many IP addresses are managing redirects correctly?

1.2 Outline

The remainder of this report is organized as follows:

- **Chapter 2** provides the reader with the required background information.

- **Chapter 3** shows other relavent work conducted in this area.
- **Chapter 4** explains the overall design of the investigation as well as the ethical concerns around it.
- **Chapter 5** details the implementation of ZMap and ZGrab within Trinity College Dublin as well as describing the various programs and scripts used to gather the data.
- **Chapter 6** presents the results of the scans conducted.
- **Chapter 7** provides an analysis of the findings in surveying Web Servers within Trinity College Dublin.
- **Chapter 8** describes the future work that could be done to extend this project.

Chapter 2

Background

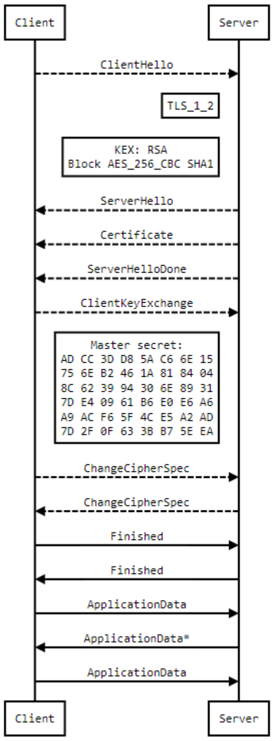
2.1 Web Server

Web servers are one of components that make up the overall web-application architecture. Communication between between a web server and a clients browser is done typical through a protocol called Hyper Text Transfer Protocol or HTTP for short. The general way in which a Client initiates communication between a web server is through a Web Browser such as Chrome or Firefox, The browser sends a request to the web sever of some resource typically a web page or some other file type, if the resource doesn't exist an error is returned to the client, generally in the form of a 404 or some other status code. As well as handling request, the server may sometimes be required to handle processing of forms inputted by the client[1]. Almost any device that is connected to the internet can host a web server, such as a printer or mobile phone.

Web servers play a significant role in terms of the overall workings of web applications and security issues relating to web servers can come in a number of ways such as being incorrectly configured[12] along with lack of privilege management.

2.2 Transport Layer Security

TLS or SSL as it may sometimes be referred to is a protocol that allows secure communication between a client and server over the Internet, which is typically between a Browser and Web Server. Over the years there have been many versions of TLS/SSL implementations, the first being in 1994 when Netscape developed SSL 1.O however this was never released publicly as it was vulnerable to replay attack[14]. This drove the development of SSL 2.0 in 1995, but along with its predecessor it too had problems. Then in 1996 SSL 3.0 was distributed to combat the issues with SSL 2.0 gaining a large popularity in the process [14]. In mid 1996 development of these protocols moved to the Internet Engineering Task Force and so too did the name of the protocol changing from SSL to TLS. To date there have been four releases of TLS with TLSv1 being released in 1999, TLSv1.1 in 2006, TLSv1.2 in 2008[14] and in 2014 a draft of TLSv1.3 was developed [2].



In order for a secure connection to be established the client first sends a *Client Hello* message which includes the version of TLS supported by the client a random byte string that is used for calculations later on in the handshake such as generating the pre-master secret, along with the list of cipher suites and compression methods in order of the clients preference [13]. As well as this the client sends an empty session identifier[14]. The server responds with a *ServerHello* message indicating the version of TLS that will be used along with the appropriate cipher suite, compression method and session Identifier. The server also generates its own random byte string. Next a *Certificate* is sent from the server to the Client whenever the key exchange method uses certificates. This certificate message contains a chain of certificates which the client will use to validate the server says who they say they are. Following this a Server Key Exchange Message is sent if the certificate sent by the server lacks sufficient data for the client to exchange a premaster secret.The Server then sends a *ServerHello Done* message to the Client to signal that the client can now continue with its turn of the key exchange related messages. The Client now sends a *ClientKeyExchange* message to the server which includes the pre-master secret created by the client. Both the Client and the Server send a *ChangeCipherSpec* message to alert each other that all future exchanges within this session will be encrypted with the chosen cipher suite and keys. Finally the Client sends a *Finished* Message to the Server and likewise The Server sends a *Finished* message to the Client, both of theses exchanges are used in order to verify that the key exchange and authentication process were successful as well as indicating that each respective part of the TLS handshake is complete. From this point on all *ApplicationData* exchange between the client and server is protected based on the current connection state [13][14]

2.3 Port Scanning

Port scanners are tools that are used to check for devices on open ports, these tools could be used by system administrator to monitor devices on a network or by a malicious party hoping to find vulnerable devices to infect.
There are three main grades of port scans.

1. **Vertical Scans:** The vertical port scan is one in which a single host is scanned across mulitple ports.

2. **Horizontal Scans:** The horizontal scan is one in which a single port is

scanned against multiple hosts.

3. **Block Scans:** The Block scan is a combination of vertical and horizontal scans.[10]

2.3.1 ZMap

ZMap is an open-source network scanner optimized for efficiently performing Internet-scale network surveys [7], developed in 2013 by Zakir Durumeric, Eric Wustrow and J. Alex Halderman at the University of Michigan they have been able to effectively diminish the time required to scan the IPv4 address space to a matter of minutes. The architecture allows sending and receiving components to run asynchronously and enables a single source machine to comprehensively scan every host in the public IPv4 address space for a particular open TCP port in under 45 mins using a 1 Gbps Ethernet link[7]. The default configuration for ZMap unfortunately only allows scans for one specified port in the given IP range (CIDR Block), which in our case and many others requires users of this tool to write further programmes to investigate hosts on more than one specified port. ZMap currently has fully implemented probe modules for TCP SYN scans, ICMP, DNS queries, UPnP, BACNET, and can send a large number of UDP probes [8]

In order to randomise the scans conducted on the target address space, ZMap remains stateless. To avoid keeping state ZMap sends a fixed number of probs per target with the default sending only one probe.[7]

One of the problems of internet wide scanning is that it uses a large amount of bandwidth, the creators of ZMap have introduced seven points in relation to the best practices when conducting these scans[7] page 12.

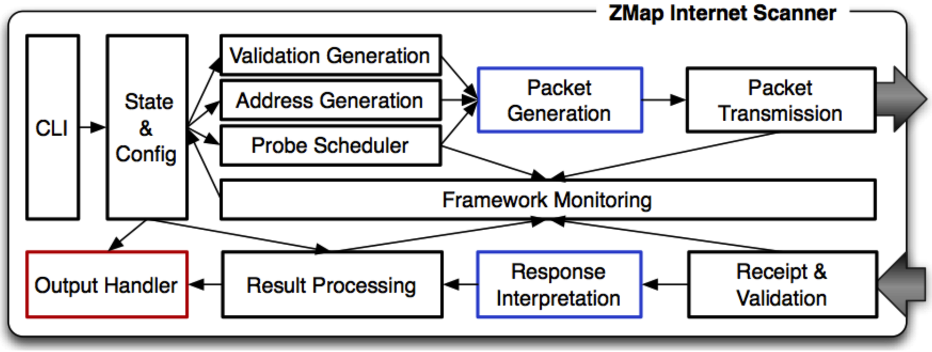


Figure 1. ZMap Architecture.

ZMap uses a modular design to support many types of probes and integration with a variety of research applications. Instructions are entered Via the Command Line (CLI) which are then parsed to the *State & Config* module. The next two modules *Validation Generation* and *Address Generation* are responsible for generating the target space/address range while also validating this space to ensure any host/IP addresses are excluded according to a configuration file which omits sites/IP ranges such as any reserved address allocations along with others who may wish to opt out of future scans [7] while the *Probe Scheduler* sets the timing of the soon to be sent probes. Next we have the *Packet Generation* module which is responsible for generating the required probe packets as per the type of scan we are conducting , the *Packet Transmission* module is responsible for sending these packets to their intended destination. ZMap also provides Extensible probe modules which can be customized for different kinds of probes, and are responsible for generating probe packets and interpreting whether incoming packets are valid responses. The *Framework Monitoring* oversees every packet that is sent and received, while the *Receipt & Validation* module responds to TCP SYN-ACK packets and discard packets clearly not initiated by the scan by cross checking the source and destination ports of the packets. The *Response Interpretation"* interprets the responses from those that have been validated by the *Receipt & Validation* module. Before the results of the probes are outputted they are first brought to the *Result Processing* module which processes the results and outputs these either to the console via stdout or to be piped to another process directly such as ZGrab or outputted to a comma separated file (csv) [7].

The speed at which ZMap sends packets is performed as fast as the source's CPU or NIC allows. This speed however at which ZMap sends probes is a cause for concern, as sending them in numerical order would probably overload and cause a network failure. So in order to counteract this ZMap uses a random permutation of the address space, iterating over a multiplicative group of integers modulo p, with p being slightly larger than 2^{32}. By choosing p to be a prime, ZMap guarantees that the group is cyclic and will reach all addresses in the IPv4 address space once per cycle. To select a new permutation for each scan, a new primitive root of the multiplicative group and a new random starting address are chosen. ZMap efficiently finds random primitive roots of the multiplicative group by utilizing the isomorphism $(Zp-1,+)\ =(Z*p,x)$ and mapping roots of$(Zp-1,+)$ into the multiplicative group via the function f(x) =nx where n is a known primitive root of$(Z/pZ)x$. Once this primitive root is ZMap cycles through the target address space by applying the group operation to the current address. The scan is finished once the initially scanned IP address is reached[7].

ZMap send packets at Ethernet Level in order to cache packet values and reduce to overhead on the Kernel. ZMap implements a probing technique known as SYN scanning or half-open scanning [7]. This was chosen to instead of performing a full TCP handshake based on the reduced number of exchanged packets. In the situation where a host is unreachable or does not respond, only a single packet is used in the exchange (a SYN from the scanner); in the case of a closed port, two packets are exchanged (a SYN answered with a RST); and in the situation where the port is open, three packets are exchanged (a SYN, a SYN-ACK reply, and a RST from the scanner which will close the connection)[7].

2.4 Banner Grabbing

Banner grabbing is a technique used to gain information about a device on a network by establishing a connection and observing the output from the connection [9]. System Administrators can use these tools designed for this application to take inventory of the devices and services on their network. Attackers can also use banner grabbing tools in order to find network devices that are using known applications with well documented vulnerability (MD5 for example)

2.4.1 ZGrab

ZGrab is one such banner grabber implemented in go that allows user to perform various application handshakes for a number of different protocols such as HTTP, HTTPS, SSH as well as SMTP to name a few. ZGrab connects to the host by opening up a TCP connection. ZGab outputs the information in raw JSON format retrieving all the information about the connection handshake such as SSL/TLS information as well as response codes.[3]

For example When performing a TLS handshake with a host, ZGrab offers the cipher suites implemented by the Golang TLS library and logs the chosen cipher suite[3] rather than using the chosen cipher suite of the source machine performing the ZGrab . ZGrab can be also be used inconjuction with ZMap to grab information simultaneous while a host being scanned or independently of ZMap by passing the host directly into ZGrab. Instructions are passed to ZGrab much in the same way of ZMap via a command line interface (CLI)

Chapter 3

Related Work

Other similar scanning works can be seen with Zakir Durumeric, Michael Bailey and J. Alex Halderman the creators of ZMap/ZGrab at the University of Michigan, where in 2014 [4], they published a paper relating to the overall environment of scanners on the Internet by analysing a years worth of data from January 1st 2013 up until May 1st 2014 from a large network telescope revolving around scanning activity with a scan being defined as an ”instance where a source contacted at least 100 unique addresses in our darknet(.0018% of the public IPv4 address space) on the same port and protocol at the minimum estimated Internet-wide scan rate of 10 packets per second (pps)”, they also investigated the way in which organisations were protected themselves against these scans if at all. Motivations behind the many of scans that they discovered were in relation to academic research, while a large proportion of scans were targetting services with know vulnerabilities (e.g. SQL serves). Throughout this period 10.8 million scans from 1.76 million hosts were detected with the distribution of the scans found being 56.4% TCP

SYN packets, 35.0% UDP packets, and 8.6% ICMP echo request packets. The HTTP and HTTPS are among the highest in terms of number of scans found on these protocols which in relation to my own project are the protocols most familiar with port 80 and 443. With malicious users and attackers having the ability to use these tools for various rouge purposes scan detection plays a huge part in defending organisations however according to this paper the vast majority don't regard scanning as a significant threat, that being said within 24 hours of new vulnerabilities being released on devices, they discovered an increase in the number of scans conducted on the ports commonly associated with those devices for example regarding the disclosure of the Heart-bleed Bug which was discovered in March 2014 and publicly disclosed on April 7th 2014. The vulnerability itself allows attackers to remotely dump arbitrary private data from many common and popular servers that support TLS. In the week following the public disclosure 53 scans from 27 host targetting HTTPS were observed, prior to the disclosure of the vulnerability 29 scans from 16 hosts were observed targetting HTTPS.

The University of Michigan performs regular scans for HTTPS hosts[4] in order to track the certificate authority ecosystems in an effort to analyse TLS certificates and the Certificate Authorities that sign them. Between the periods of April 2012 and June 2013 they managed to collect 33.6 million unique X.509 Certificates of which 6.2 million were browser trusted as well as identifying the most common CAs by leaf certificates issued, with GoDaddy.com, Inc accounting for 31% adoption rate of the HTTPS protocol throughout their scans with an increase of 23% in the Alexa Top 1 million websites and 10.9% increase in the number of browser-trusted certificates. Keys and Signatures were also tracked to highlight how ZMap could also be used to mitigate risk and act as defensive tool for researchers but this also has the flip side effect of an attacker using this tool to locate hosts suffering from a new vulnerability within minutes.[7] With the research still finding some Certificate Authorities still using MD5 to sign their certificates [5]. Within the scope of my own project, I will also be investigating the certificates found within the University, looking at certain parameters such as the signature algorithm used to sign the certificate along with the number of self signed and browser trusted certificates within the University.

Censys begun in 2015 and is a platform created by the same team that designed ZMap that helps information security practitioners answer security related questions in an effort to discover new threats and assess there impact they may have. They regularly probe every public IP address and popular domain names through horizontal scans of the IPv4 address space, curating data over time to see changes in protocol adaption for example, and make it accessible through an interactive search engine and API that allows users to ask questions such as ”what percentage of HTTPS servers support SSLv3” by eliminating the labour intensive process of analysing gigabytes of data as well as lowering the barrier to entry for researchers who might not have the performance capabilities to perform these scans.[3] they have also played a central role in the discovery or analysis of some of the most significant Internet-scale vulnerabilities: FREAK, Logjam, DROWN, Heart bleed, and the Mirai botnet. Today however Censys has moved out of the University of Michigan and into it’s own company in order to better serve and expand their capabilities to research offering more enhanced services, technical support, and an even more complete and powerful view of the Internet [6]. Censys could also be used to identify public facing devices that may have been intended to be private within a network but unintentional found it's way to public internet or be used to calculate the risk that known public facing device could have on an organisation [3] by using their website to investigate such questions.

In conjunction with researchers at the University of Michigan, researchers at The International Computer Science Institute and the University of Illinois Urban-Champaign in 2016 have conducted similar HTTPS surveys but by analysing certificates from a large body of sources instead of just one with an aim to obtain a better perspective of the HTTPS ecosystem. In total they combined 8 different data sets observing nearly 17 million unique browser trusted certificates which were valid during August 29 to September 8, 2016 which was the investigation period of their study. Of the 8 datasets analysed Censys(38%) and CT logs(90.5%) accounted for 99.4% coverage of all certificates observed. They are currently working with both parties in order to reduce the discrepancy between either source in order to make each one more or less a near comprehensive view of trusted HTTPS certificates[15].

Researchers at Ajou University, have implemented ZMap within their own University campus in aid of identifying hosts and comparing various scanning techniques such as FIN and Xmas Scans for identifying hosts by conducting scans on a number of ports including port 25 (smtp) , 80(http) and 443 (https). These optional scanning techniques are implemented by creating specific probe modules for ZMap [11]. Some of the hosts they discovered included old web servers of a printers which allowed them to instruct the printer to print test pages as well as gaining access to password protected content of the printer with default passwords of the known devices still in use and common on the internet. While this Report on the other hand uses exclusive SYN scanning the default for ZMap with an aim to identifying regular and irregular Hosts running on port 80 and 443 as well analysing the current configurations of these Web Servers within the university, I was hope to discover similar devices on these ports that were found in Ajou University among others.

This report differs from previous studies in terms of the scope of our dataset as well as the direct line of the questions we intend to answer in order to highlight the state of Web Servers within TCD.

Chapter 4

Design

The design of the scans consisted of two horizontal set of scans on port 80 and port 443 between 05/03/2018 - 31/03/2018 running every hour throughout the set of scans we discovered 1,710 IP addresses in total with x IP addresses on port 80 and y IP addresses on Port 443

With the IP addresses found a further set of Banner grabs were conducted on the results of the horizontal scans, theses banner grabs were done every 2 hours to account for the length of time to adequately complete 1,710 banner grabs

4.1 Technologies Used

4.2 Ethics

see [3] page 2 good internet citizenship as well as ZMap 7 points for conducting scans

4.3 Challenges

Chapter 5

Implementation

To implement the ZMap scans I wrote two scripts that would then be setup on a cron job to run every hour outputting the results to a csv file

To differentiate which IP addresses were listening on to one or more ports I created a python programme to find what IP addresses were listening on Port 80 and which IP addresses were listening on Port 443 doing this I was able to get the intersection of the Two to determine the number that were listening on both Ports. The way in which I found out which IP addresses were actually listening to only port 80, port 443 and which were listening to both ports I used a key value dictionary to determine this with the key being the IP address and value being a list of Ports that particular IP address was one for example an IP address that was listening to only port 80 would have a value of [‘80’] likewise for an IP address that was only listening to port 443 would have a value of [‘443’] and for an IP address that was on Both ports would have a value of [‘80’,’443’] .

In order to determine what IP address resolved to hostname I used the socket library within python to do a reverse DNS lookup in order to convert the IP addresses to hostnames (Talk about purpose of getting a hostname)

– Take about trial runs with Stephens machines in order to figure out what to look out for

Similarly for the ZGrabs I've implement much in the same for the ZMap scans, two scripts One for the IP address on Port 80 and another for the IP address on Port 443

Chapter 6

Results

For Comparison of Some results e.g average host per hour look at page 5 and 6 of [7]

Other comparison [3] page 9 percentage of cipher suites being used could have a new column with TCD figure 5

Another comparison can be found with [11] on page 683 finding out of date printers that haven't been updated in a long time

Another comparison can be found with trying to see how many scans are necessary for discovering all the host [5] page 4

Another result comparison could be trying to identify CA that sign certs who have had security issues in the past [5] page 6

Another result comparison could be looking at key lengths [5] page 7 and 8, also look for a paper with recommendations of cryptographic algorithms and key lengths

Another result comparison could be looking at signature algorithms, algorithms used to sign the certificates [5] page 9

Another result comparison looking at ips/hosts sharing multiple public keys, certs e.g Stephens Research and [5] page 10

Another result comparison could be seeing sites vulnerable to FREAK Attack [**?**] page 54
vandersloot2016toward7

Chapter 7

Conclusion

[14] page 62,63

[12] page 316

Chapter 8

Future Work

Would be beneficial if we could find out the total number of host on port 80 and port 443 from system admins to determine whether or not the scans detecting all host on port 80 and port 443 in order to judge the successfulness of ZMap

Looking at the adoption rate of protocols within the college similarly done by [7] page 9 and [5] page 11.

Talk about what I have done could be used as a infrastructure in order for other universities to survey their own web servers as well as maybe having the ability to automate it with alerts being sent out if a certain type of cipher suite is found or old version of TLS is discovered[3] page 5 talks about something similar

Scanning IPv6 address space within the college [7] page 14

Investigate the detection capabilities of ZMap [10] page 2

Along with the point regarding infrastructure deployment I could also make some sort of application around what I have done but giving a score based off of security flaws present, with each Ip/Host getting a score based on how well configured they are [12] page 315

Bibliography

[1] CONALLEN, J. Modeling web application architectures with uml. *Communications of the ACM 42*, 10 (1999), 63–70.

[2] DIERKS, T. The transport layer security (tls) protocol version 1.3.

[3] DURUMERIC, Z., ADRIAN, D., MIRIAN, A., BAILEY, M., AND HALDERMAN, J. A. A search engine backed by internet-wide scanning. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security* (2015), ACM, pp. 542–553.

[4] DURUMERIC, Z., BAILEY, M., AND HALDERMAN, J. A. An internet-wide view of internet-wide scanning. In *USENIX Security Symposium* (2014), pp. 65–78.

[5] DURUMERIC, Z., KASTEN, J., BAILEY, M., AND HALDERMAN, J. A. Analysis of the https certificate ecosystem. In *Proceedings of the 2013 conference on Internet measurement conference* (2013), ACM, pp. 291–304.

[6] DURUMERIC, Z., WUSTROW, E., AND HALDERMAN, J. A. censys, https://censys.io/.

[7] DURUMERIC, Z., WUSTROW, E., AND HALDERMAN, J. A. Zmap: Fast internet-wide scanning and its security applications.

[8] DURUMERIC, Z., WUSTROW, E., AND HALDERMAN, J. A. Zmap, https://github.com/zmap/zmap/blob/master/readme.md.

[9] KONDO, T. S., AND MSELLE, L. J. Penetration testing with banner grabbers and packet sniffers. *Journal of Emerging Trends in Computing and Information Sciences 5*, 4 (2014).

[10] LEE, C. B., ROEDEL, C., AND SILENOK, E. Detection and characterization of port scan attacks. *Univeristy of California, Department of Computer Science and Engineering* (2003).

[11] LEE, S., IM, S.-Y., SHIN, S.-H., ROH, B.-H., AND LEE, C. Implementation and vulnerability test of stealth port scanning attacks using zmap of censys engine. In *Information and Communication Technology Convergence (ICTC), 2016 International Conference on* (2016), IEEE, pp. 681–683.

[12] MENDES, N., NETO, A. A., DURÃES, J., VIEIRA, M., AND MADEIRA, H. Assessing and comparing security of web servers. In *Dependable Computing, 2008. PRDC'08. 14th IEEE Pacific Rim International Symposium on* (2008), IEEE, pp. 313–322.

[13] MIRLEFT. Ocaml-tls demo server, https://tls.nqsb.io/.

[14] TURNER, S. Transport layer security. *IEEE Internet Computing 18*, 6 (2014), 60–63.

[15] VANDERSLOOT, B., AMANN, J., BERNHARD, M., DURUMERIC, Z., BAILEY, M., AND HALDERMAN, J. A. Towards a complete view of the certificate ecosystem. In *Proceedings of the 2016 Internet Measurement Conference* (2016), ACM, pp. 543–549.

Appendix A

A first appendix

the program parameters were . . .

the following table lists all the people who contributed questionnaire responses . . .

Appendix B

A sample of the questionnaire form used

blah blah