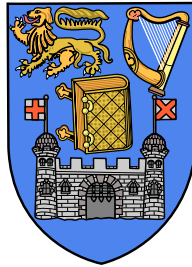University of Dublin

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.

_____ May 2, 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.

_____ May 2, 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 layer scanner 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 further problems is of the utmost importance for any organization that aims to mitigate such 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 as well as showing the state of public facing web servers [8] thus the same should be true at a University Campus level.

As well as deploying and testing the tool within Trinity College Dublin, this report 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

!!!!!!!!!!!!!!!!MIND TENSE USED (PAST OR FUTURE), PICK ONE AND KEEP IT, ALSO MIND THE USE OF I WHEN DESCRIBING THE REPORT!!!!!!!!!!!!!!!!

As with any device that is connected to the Internet, the security capabilities of these systems is of concern. Even when following strict policies, miss configurations of Web Servers can lead to possible issues down the line. 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.

This report will also outline the steps that were taken in order for other organizations

and institutions to replicate what has been done here within Trinity College Dublin.

This goal will be analyzed 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 have a Hostname?

- Trying to categorize IP addresses?

- What does the current state of Certificates look like within the college?

- How many Certificates are self signed certs?

- How many Certificates are browser trusted?

- How many Certificates have expired at the time of writing this report?

- Are there any out of date versions of TLS being used?

- For those IP addresses that don't have 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** aims to familiarise the reader with the required background information.

- **Chapter 3** highlights other relevant work conducted in the area of network scanning.

- **Chapter 4** explains the overall design of the investigation as well as the ethical concerns it entails.

- **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 study conducted.

- **Chapter 7** provides an analysis of the findings in surveying Web Servers within Trinity College Dublin.

- **Chapter 8** suggest future work that could be done following the findings of the work done in this project.

Chapter 2

Background

This chapter will introduce Web Servers along with there underlying technology. It will also discuss the main scanning tools used to conduct the study.

2.1 Web Server

Web servers are one of the many components that make up the overall Web Application Architecture. Communication between between a Web Server and a Clients browser is typical done 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 requests, the server may sometimes be required to handle processing of forms

inputted by the client[5]. 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 scenarios such as being incorrectly configured[29] or lacking proper privilege management.

2.2 Port

Port numbers are the original and most extensively used means for application and service identification on the Internet. Ports are 16-bit numbers, and the combination of source and destination port umbers together with the IP addresses of the communicating end systems uniquely identifies a session of a given transport protocol [6] such as TCP(Transmission control protocol) which provides reliability and safety of packets sent or UDP (User Datagram Protocol) unlike TCP does not guarantee delivery of packets, UDP is widely used in streaming of videos where speed is more important than reliability [23]. Some Port numbers are also known by their associated service names or protocol used for the communication such as "https" for port number 443 and "http" for port number 80. Port numbers range from 0 to 65535 with Port 0 being reserved, leaving 65535 that could be used. There are three distinct classes of Port numbers:

- the System Ports, also known as the Well Known Ports, from 0-1023 (assigned by IANA)

- the User Ports, also known as the Registered Ports, from 1024- 49151 (assigned

by IANA)

- the Dynamic Ports, also known as the Private or Ephemeral Ports, from 49152-65535 (never assigned)

2.3   HTTP and HTTPS

The Hypertext transfer Protocol (HTTP) is application layer protocol most commonly associated with the world wide web ("www"). HTTP connections are made on Port 80, HTTP does not encrypt the data that is exchanged between the client and server [17]. This is solved with HTTPS or HTTP over TLS which provides a secure session between the client and server encrypting the data exchange between the two parties [35].

2.4   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. Over the years there have been many versions of TLS/SSL implementations, the first being in 1994 when Netscape developed SSLv1.O however this was never released publicly as it was vulnerable to replay attacks[39]. This drove the development of SSLv2.0 in 1995, but along with its predecessor it too had problems. In total there were 4 main deficiencies with SSLv2.0, one of these for example was not protecting handshake messages which grants a man-in-the-middle to trick the client in selecting a weaker cipher suite than it would normally choose [41]. Then in 1996 SSLv3.0 was

distributed to combat the issues with SSLv2.0 gaining a large popularity in the process [39]. In mid 1996 development of these protocols moved to the Internet Engineering Task Force (IETF) and so too did the name of the protocol changing from SSL to TLS. To date there have been four releases of TLS with TLSv1.0 being released in 1999, TLSv1.1 in 2006, TLSv1.2 in 2008[39] and in 2014 a draft of TLSv1.3 was developed [7].
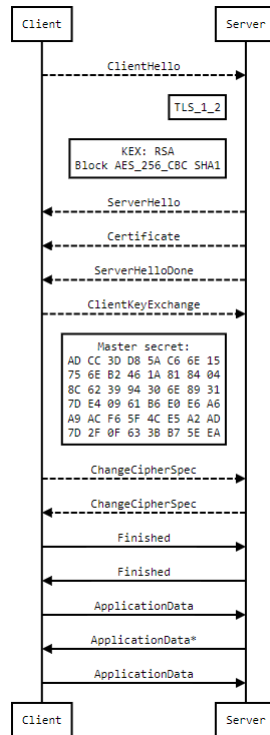
In order for a secure connection to be established the client first sends a *Client Hello* message which includes the version of TLS supported by the client a random byte string that is used for generating the pre-master secret, along with the list of cipher suites and compression methods in order of the clients preference [30].A session identifier is also included to identify the session between the client and

server[39]. 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 pre-master secret.The Server then sends a *ServerHelloDone* 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 [30][39].

2.5 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 that can be conducted:

1. **Vertical Scans:** The vertical port scan is one in which a single host is scanned across multiple 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.[25]

There are a number Port Scanning tools that are widely available such as Nmap [27] and Nessus [3], this project exclusively uses ZMap [14] to conduct the Port Scans.

2.5.1 ZMap

!!!!!!!!!!!!!!! SHOW AND DESCRIBE SAMEPLE OUTPUT !!!!!!!!!!!!!!

ZMap is an open-source network scanner optimized for efficiently performing Internet-scale network surveys [14], 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 entire 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 minutes using a 1 Gbps Ethernet link[14]. The default configuration for
ZMap unfortunately only allows scans for one specified port in the given IP range
(CIDR Block)[18], which in this study 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 [15]

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.[14]

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
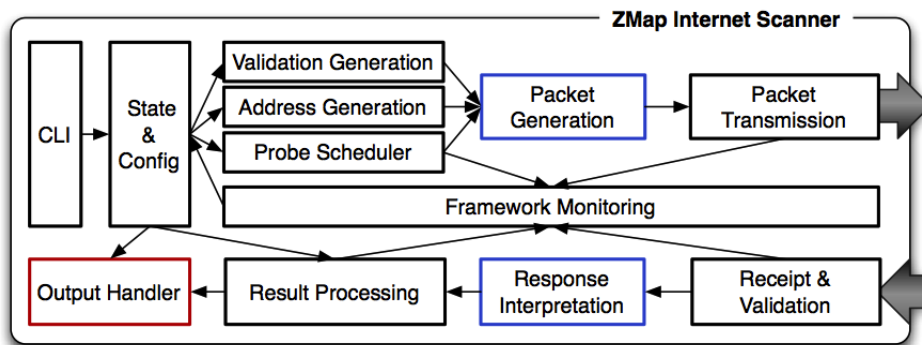best practices when conducting these scans[14] page 12.

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 in this generation are excluded according to a configuration file which omits sites/IP ranges such as any reserved address allocations along with individual host who may wish to opt out of future scans [14] 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. 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 parsed by the *Result Processing* module which processes the results and outputs them either to the console via stdout or to a comma separated file (csv), the results can also be piped to another process directly such as ZGrab [14].

Figure 2.1: a nice plot1

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[14].

ZMap send packets at Ethernet Level in order to cache packet values and reduce the overhead on the Kernel. ZMap implements a probing technique known as SYN scanning or half-open scanning [14]. This was chosen 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)[14].

2.6   Banner Grabbing

Banner grabbing is a technique used to gain information about a device on a network, this is obtained by establishing a connection with the device and observing the output from the connection [24]. 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).

2.6.1 ZGrab

!!!!!!!!!!!!!! SHOW AND DESCRIBE SAMEPLE OUTPUT !!!!!!!!!!!!!!

ZGrab is one such banner grabber implemented in Go [33] that allows user to perform various application handshakes for a number of different protocols such as HTTP, HTTPS, SSH [43] as well as SMTP[21] to name a few. ZGrab connects to the host by opening up a TCP connections [8]. 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.

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[8] rather than using the chosen cipher suite of the source machine performing the ZGrab. ZGrab can be also be used in conjunction with ZMap to grab service information simultaneous, while a host is being scanned or independently of ZMap by passing the host source IP address or domain name directly into ZGrab. Instructions are passed to ZGrab much in the same way of ZMap via a command line interface (CLI).

In terms of the actual banner grab there are a number of aspects of a web server that we are interested in when we conduct are ZGrab application layer scans in order to see the contrast and similarities underlying the service information of the web Servers in the college.
Here is a list of the fields that this project extracted to examine the web servers in the college:

- Ciphersuite being used to provide encryption of data between the client and server.

- Key Exchange Method used to exchanged cryptographic keys between the client and server.

- TLS/SSL verison being used to provide secure communication between the client and server.

- The Public Key of the Web server embedded in the Certificate.

- The length of the Public Key used.

- Certificate start and end date, to see when the certificate was issued and also when it expires.

- the Signature Algorithm used to sign the Certificate.

- If the Certificate is self signed or not. A self signed certificate is a Certificate that has not been Issued and signed by a Certificate Authority rather the issuer who has created the certificate has signed it themselves with their own private key [20].

- Browser Trusted field to see if the certificate is verified by the browser. This is done by the clients browser who attempts to build a chain of trust from the certificate to a root certificate on the client. The root trust store on the client contains a set of root certificates from trusted CAs to validate against. The browser also checks that the certificate has the correct hostname and the certificate has not yet expired [1]. If at any point there is a failure in the

validation process, this means that the client(browser) is unsure of who the proper identity of the server actually is.

- The Status code and Reason Phrase, The status code itself is a three digit integer that is received by the client as a response from the server, the phrase is a short description for the user [17] [4].

- HTML body of the web server's root page to categorise the servers and their uses.

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 [9], 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 targeting HTTPS were observed, prior to the disclosure of the vulnerability 29 scans from 16 hosts were observed targeting HTTPS.

The University of Michigan performs regular scans for HTTPS hosts[9] 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.[14] With the research still finding some Certificate Authorities still using MD5 to sign their certificates [10]. 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.[8] 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 [11]. 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 [8] 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[42].

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 [26]. 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

4.1 Overview

As we can see from figure 4.1 in section 1 highlighted in blue, the design of the scans consisted of two horizontal set of scans on port 80 and port 443 that would run every hour being executed by a scheduler. After this we would analyse the raw data outputted by ZMap in section 2 in green, once we discovered which IP addresses were listening on which Port along with which IP had an associated host name by doing a reverse DNS lookup in order for us to conduct the ZGrab scans with a domain lookup. Once the data was sufficiently analysed, we conducted 4 sets of ZGrab scans, which were a domain lookup on port 80, a domain lookup on port 443, IP banner grab on port 80 as well as port 443, seen in section 3 of figure 4.1. Lastly the output of the ZGrab scans would be analysed looking for certain fields that would be of interest.
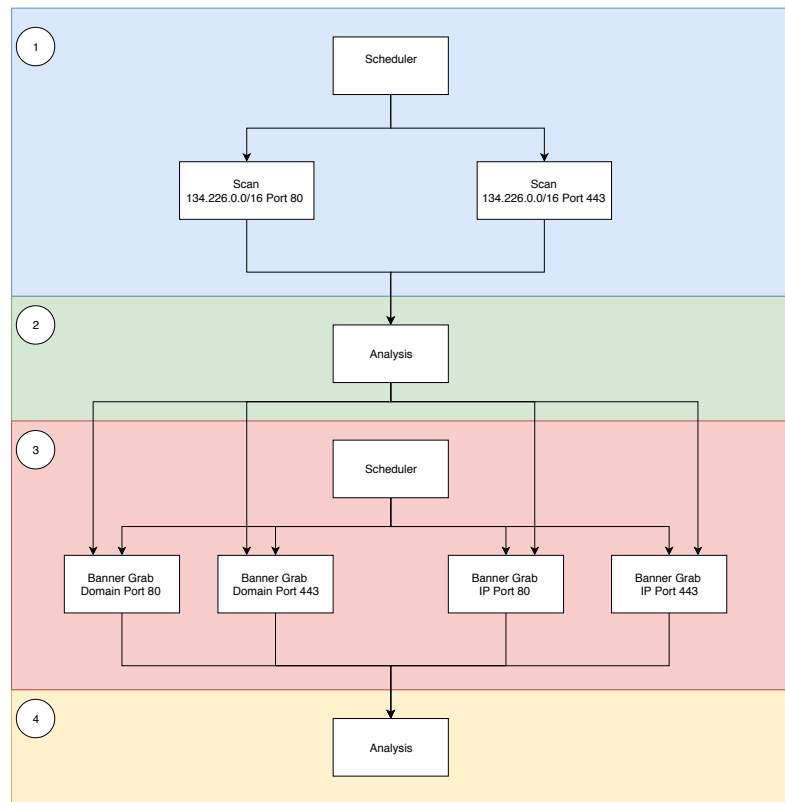
Figure 4.1: a nice plot

4.2 Ethics

As with any type of scanning work that is being conducted, there are a number of ethical considerations to be aware included how the scans should be conducted. We informed the system administrators within the college, stating the intent of the

scans that we were conducting. We also decided to limit the port that would be scanned to port 80 and port 443 to limit the strain that the scans could potentionally have on the network. While there are no codes of conduct as of yet in relation to scanning network, the creators of ZMap offer a few useful best practice guidelines when performing scans[8] [14].

Another consideration to account for is the releasing of data, since the majority of hosts are not publicly visible, for this reason no individual IP addresses are released in the report.

Chapter 5

Implementation

!!!!!!!!!!!!!!!!put in Github link in this section!!!!!!!!!!!!!!!!

This chapter will cover the implementation of the scans and banner grabs conducted in this project

5.1   Technologies Used

The majority of this project was implemented in Python and Shell scripts, here I list the main technologies used:

- ZMap as discussed earlier is a port scanner that finds IP addresses listening on a port by searching an IP/IP range against the desired port.

- ZGrab is used to perform the banner grabs of the IP addresses found.

- Crontab was used to automate/schedule both the ZMap and ZGrab.

- The majority of Python code was used to analyse and sort the data , as well leverage some libraries with python such as the socket library [36] to perform both DNS and reverse DNS lookup.

- Shell Scripts programmes were used to package ZMap and ZGrab in order for it to run with crontab.

- Matplotlib [28] is a python library used to plot the graphs in this report.

5.2 ZMap Implementation

In order to implement the ZMap scans as discussed in the design in Chapter 4, I packaged ZMap within a Shell Script that would be set to run every hour on the hour by crontab which is a time based Scheduler in linux. The ouput of the two ZMap scans would be appended to a csv file lying in the server.

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 in order to do a ZGrab with a domain to determine if there where any different Certificate Alt names between the IP lookup and its corresponding Domain Name. To implement this I used the socket library within python to do a reverse DNS lookup to obtain the hostnames.

5.3   ZGrab Implementation

To Implement the ZGrab scans took a bit more care as different types of machines would have different fields present in some and not in others, this was discovered by testing a few machines which we controlled on both sides in order to get a collection that would be of interest to us. These machines were as follows:

- A site that was signed by A Certificate Authority.

- One that was self Signed.

- A 2006 printer on Port 80.

- A site that wasn't being maintained.

- As well as a two IP addresses on Port 80 and 443.

!!!!!!!!!!!!!!!!PLACE ERROR FILE OF ZGRAB JSON OUTPUT AS AN EXAMPLE!!!!!!!!!!!!!!!!

These banner grabs that were conducted helped us in order to see the difference of the JSON outputs that ZGrab gives. Similarly for the ZGrabs I've implement much in the same for the ZMap scans, instead here we are also doing the domain lookups as well on port 80 and port 443. To extract the fields I was interested in I implemented a pyhton programme that would take the JSON output of the ZGrab and collect those fields we wanted. In order to take account for the fields that could be present in any practicular ZGrab output, exception handling was conducted on fields were attempting to extract the fields, this was also necessary due to the fact that if certain IP wasn't on at the time a ZGrab was being conducted an error would be outputted in the JSON file along with the IP of the ZGrab that was just executed, with the fields we were concerned with being absent.

5.4 Challenges

!!!!!!!!!!!!!!!!!!!!!!!!!!!! TAKE ABOUT ZGRAB DOCUMENTATION !!!!!!!!!!!!!!!!!!!!!!!!!!!!

One of the Challenges that was encountered is the scheduling of the cron jobs for ZGrab. Throughout ZGrab's conducted not all of the original IP addresses that were found in the ZMap scan presented themselves in the ZGrab data this can be contributed to a number of different reasons, One of these is the fact that the college has a number of IP address that are up less than 90% of the time period for which ZMap scans were conducted as well as ZMap being run every hour on the hour, where as in the original scheduling of ZGrab they were set up to run for every 2 hours on the hour and as a result the hours on either side might have have presented themselves with IP address that are only in those slots. The reason this was done to run every 2 hours was due to ZGrab taking some time to

complete a banner grab especially when an IP or domain wasn't up causing ZGrab to wait for a default of 10 seconds before attempting to abandon the connection. Releasing this I decided to reduce the default timeout to 5 seconds as well as running multiple ZGrabs over smaller list of IPs or domains every hour rather than a single ZGrab being conducted for instance on the either list of IP addresses on port 443 but rather splitting that list of IP addresses on port 443 into 2/3 smaller list with a dedicated ZGrab Script being executed by the crontab. These methods improved the proformance of the ZGrab allowing us to implement more frequent ZGrabs on smaller lists.

Chapter 6

Results

For Comparison of Some results e.g average host per hour look at page 5 and 6 of [14]

Other comparison [8] page 9 percentage of cipher suites being used could have a new column with TCD figure 5

Another comparison can be found with [26] 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 [10] page 4

Another result comparison could be trying to identify CA that sign certs who have had security issues in the past [10] page 6

Another result comparison could be looking at key lengths [10] 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 [10] page 9

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

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

In this section we will present the results of the scans and use them to answer the questions outlined in section 1, in order to better understand the state of Web Servers in the college.

6.1 ZMap Results

!!!!!!!!!!!!!!!!!!FIND PERIODS WHEN SERVER WAS DOWN !!!!!!!!!!!!!!!!!!!!!!!!!!

6.1.1 Variation In IP addresses
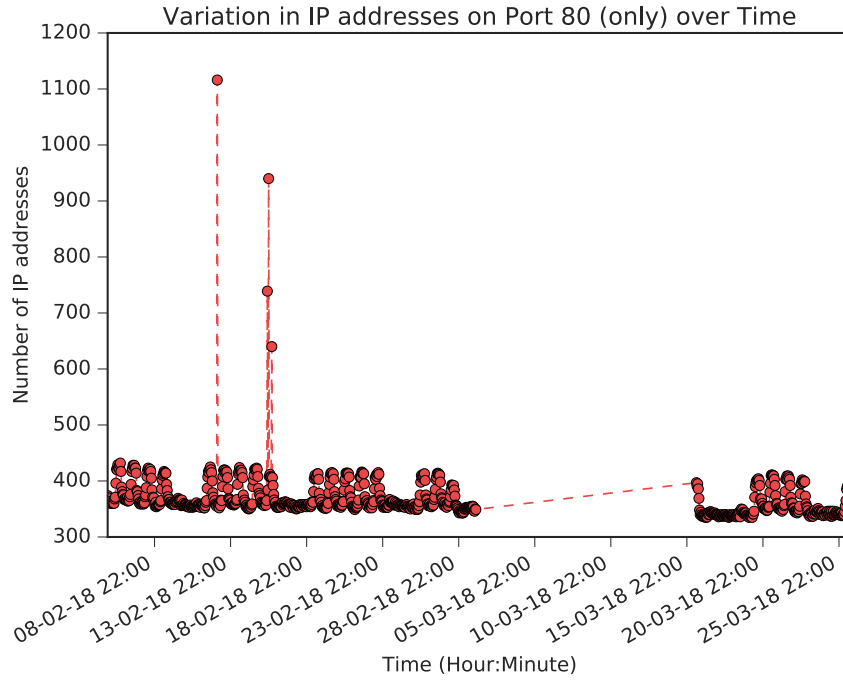
Figure 6.1: a nice plot

The ZMap scans took place between the 5th of February at 8pm up until and including the 26th of March at 6pm. However as we can see in figure 6.1 during the periods 02-03-2018 up until 16-03-2018 no scans were detected as this was during the snow period which caused a power cut in the college, downing Stephen's Server that was conducting the ZMap Scans on both Port 80 and 443. In absence of this there was a total of 827 observational periods in which the ZMap scans took place. We can also see in figure 6.1 an exceptional high count of 1116 IP addresses was recorded on the 13th of February at 1am (2018-02-13T01), this high count was due to an unknown failure of the ZMap scan on port 443, as

	Port 80 (only)	Port 443 (only)	Both Ports
Monday	367	383	377
Tuesday	378	383	377
Wednesday	377	383	377
Thursday	372	383	377
Friday	383	383	377
Saturday	352	383	377
Sunday	349	383	377

Table 6.1: Average Number of IP addresses across week

within this observational slice there were no subsequent IP addresses on port 443 to compare against.

Another intriguing point to note is that the count of IP addresses throughout the week is steady, with the number of IP addresses on port 80 (only) rising throughout the weekdays and falling at the weekend.
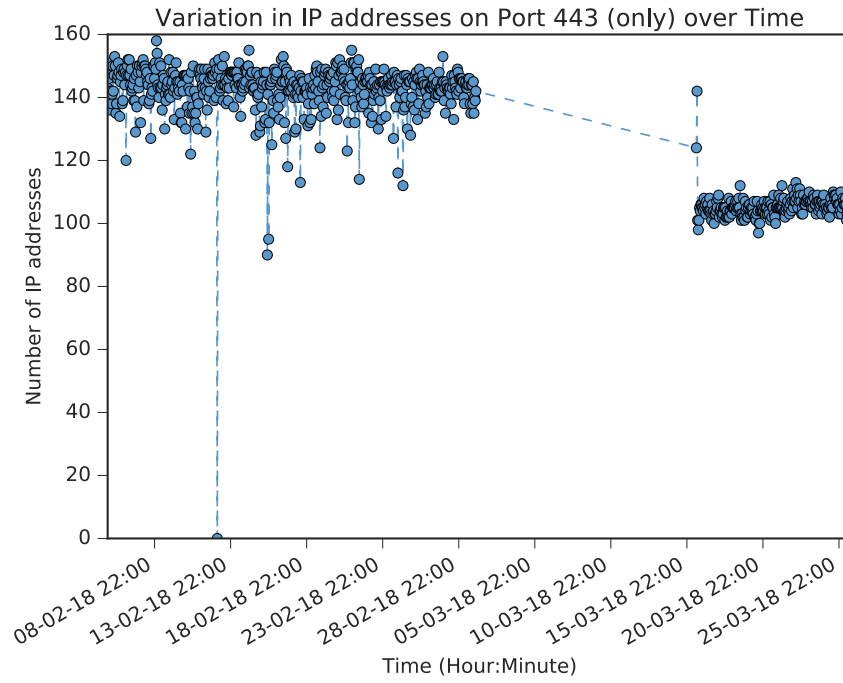
Figure 6.2: a nice plot1

From figure 6.2 we can see the failed scan that appeared on the 13th of Feburary at 1am which lead to the overly high count of IP addresses on port 80 (only) for this observational slice.

Another interesting incident that was observed is the drop in the average number of IP addresses from 145 before the server was cut off to to 105 after the server was back up and running.
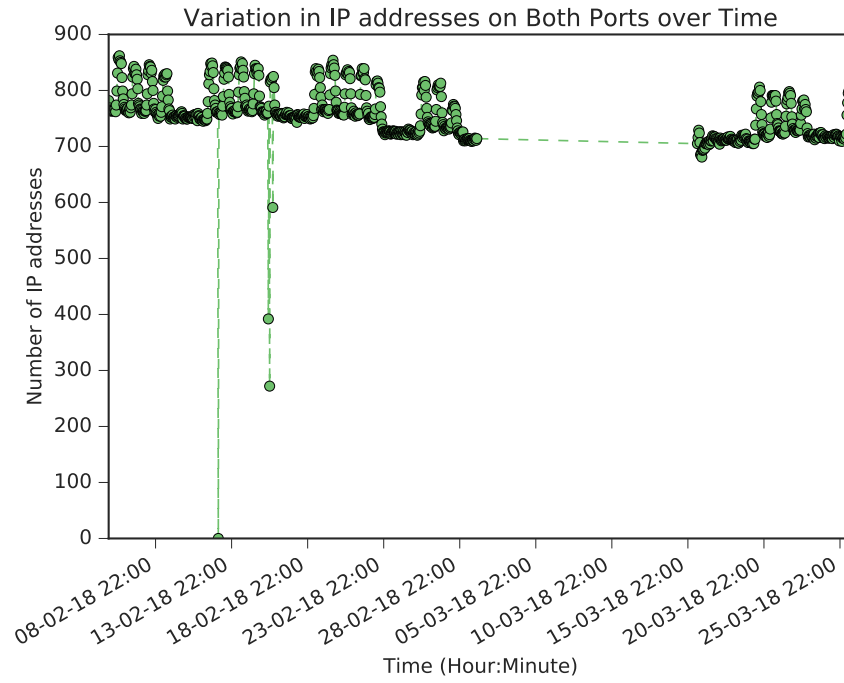
Figure 6.3: a nice plot1

From figure 6.3 we can see a similar trend that was observed in the figure 6.1 with a higher number of IP addresses being present throughout the weekday and falling at the weekend.

There is also no dramatic change in the average number of IP addresses being present before and after the server cut off that was seen in figure 6.2

6.1.2 Port Distribution
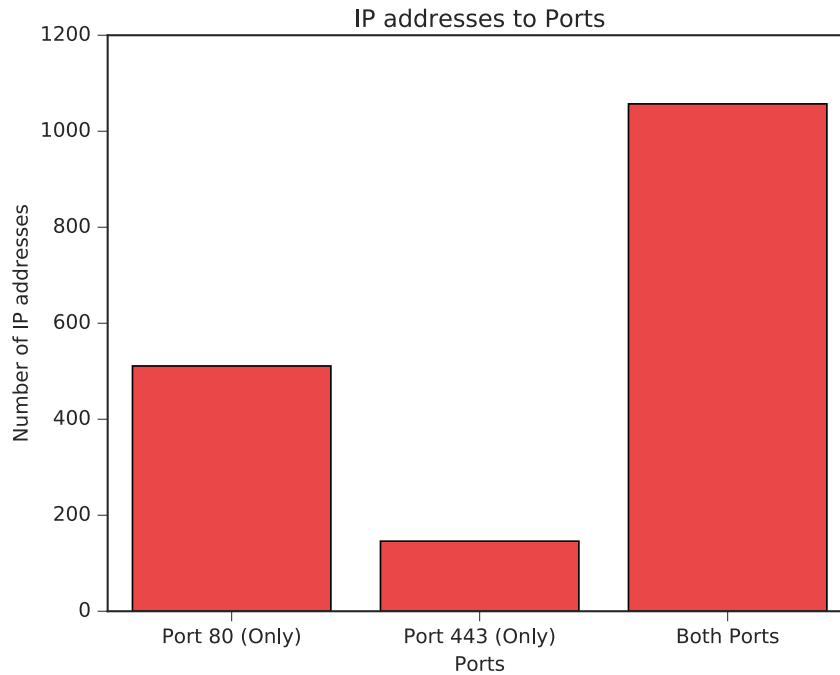
Figure 6.4: a nice plot1

In all there was a total of 1714 unique IP addresses that were seen throughout the ZMap scans, from figure 6.4 we can see a breakdown of the 3 main categories of IP addresses Port 80 (only) which are IP addresses that are were only observed on Port 80 and no other port, Port 443 which were IP addresses only observed on Port 443 and no other port, and finally Both Ports which consisted of IP addresses that were on both Port 80 and 443 at some point throughout the ZMap scans.

We recorded 511 IP addresses on port 80 alone, followed by 146 IP addresses on

port 443 and lastly 1057 IP addresses on Both ports.
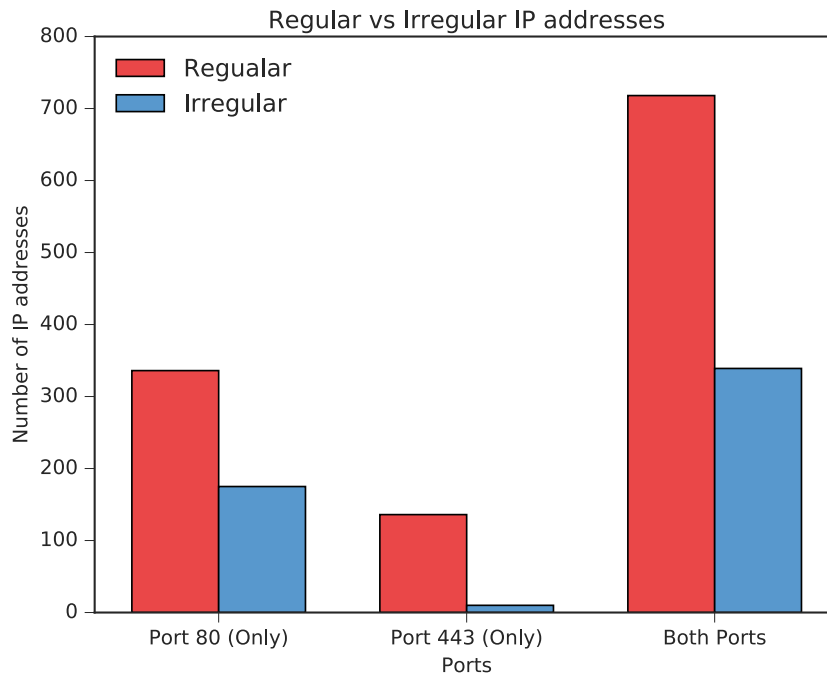
Figure 6.5: a nice plot1

From the scans (figure 6.6) we noticed a number of irregularities in terms of the number of IP addresses that would be up at any given hour, to investigate this aspect of the data we decided to filter the IP addresses in terms of those that are regular which in our case is IP addresses that are seen 90% of the time which is 744 from 827 observational slots. Everything else is consider an Irregular IP address.

The above figure 6.5 shows the comparison between each port category, with a total of 1190 IP addresses being Regular with 336 being on Port 80 (only), 136 on Port 443 (only) and 718 being on Both Ports. Similarly we observed that 524 of

the total IP addresses observed in the scans are Irregular, with 175 of those being on Port 80 (only), 10 being on Port 443 (only) and 339 on Both Ports.
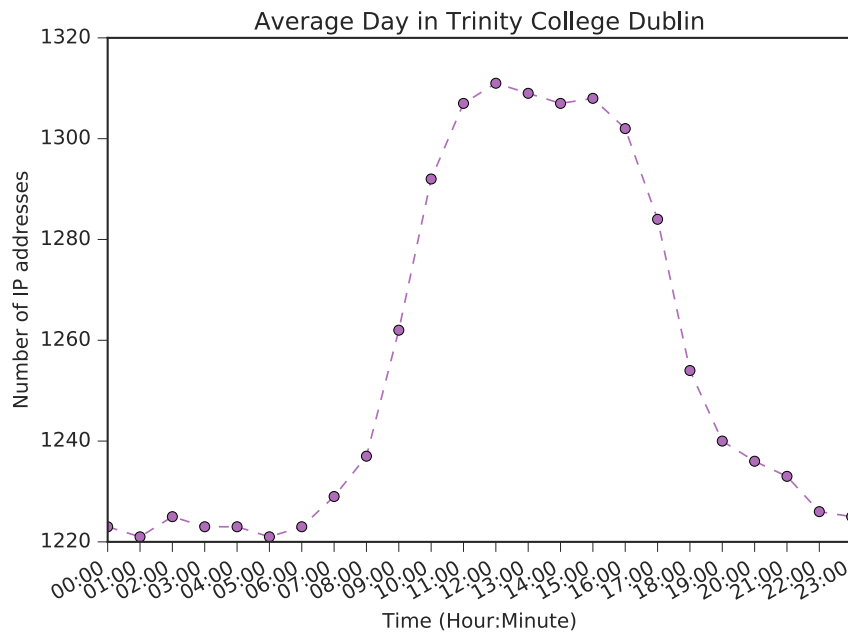
Figure 6.6: a nice plot1

When looking at all the IP addresses on average over a 24 hour period as we can see in figure 6.6 there is a large count of IP addresses between 11am to 4pm

Peak time on Average in the college is 12pm midday with 1311 IP addresses being present Off-Peak time are 1 am and 5 am with both counting 1221 IP addresses being present.
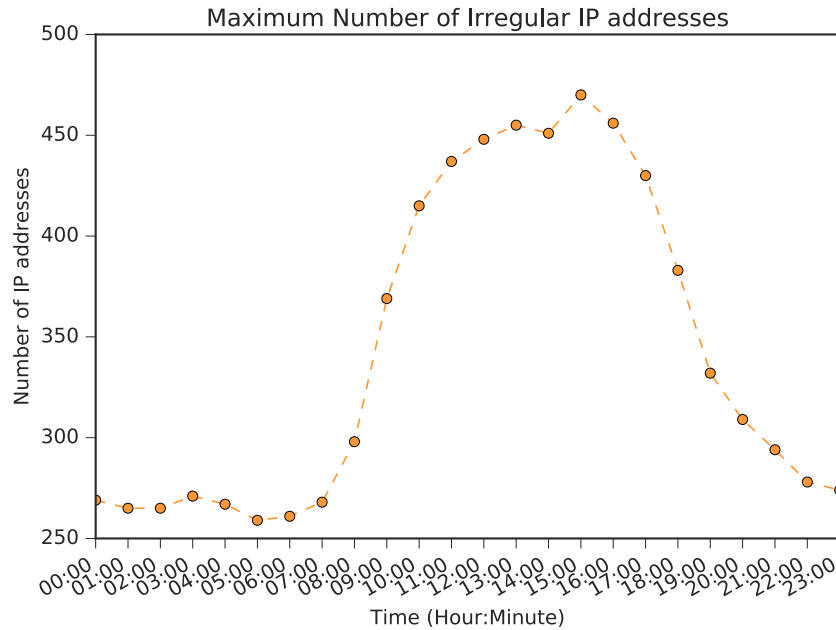
Figure 6.7: a nice plot1

When analysing the hour that an Irregular IP address has ever been on such as in figure 6.7 we see that at some point over the time of the scans particularly at 3pm there has been 470 of the 524 Irregular IP addresses active at this hour throughout the ZMap scans. On the opposite scale 259 IP addresses have be present at 5am at some point in time over the series of scans conducted. This might lead us to believe that the majority of Irregular IP address occurs at 5pm which is not the case when we look at the individual occurrences in figure 6.8.
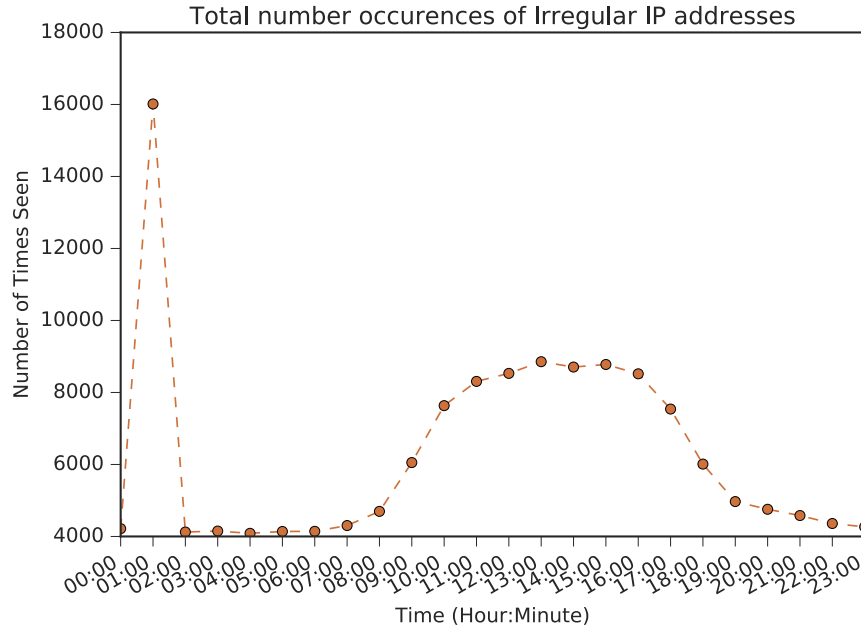
Figure 6.8: a nice plot1

When counting each occurence of an IP address on either port we see in figure 6.8 that there is a large discrepancy between the number of occurrence of an IP and the hour that each IP appears. For example we see in figure 6.8 that the number of occurrences that happen at 1am is 16014 where as at 3pm, 8777 occurrences have taken place throughout the ZMap scanning period which is the second largest number of occurrences. This large difference on further inspection of the data is due to 27 Unique IP address of which only 10 were on port 80 (only), 1 on port 443 (only) and 16 on both ports.

From figure 6.2 we can see that the vast majority of occurrences at 1am account for a major period of the overall number of occurrences. of these 27 that cause this

Total Occurrences	Occurrences at 1am
1520	720
1496	720
1354	716
1294	710
1444	710
1159	702
1018	698
729	688
1633	390
1446	388

Table 6.2: Top 10 Irregular IP addresses seen at 1am

large difference, 17 of them have 50% or greater with the number of occurrences that are seen at 1am compared to every other day in the week combined.

!!!!!!!!!!!!!!!!!!!!!!!!RE DO IPS TO HOSTNAME PIECHART !!!!!!!!!!!!!!!!!!!!

6.1.3 Domain Names
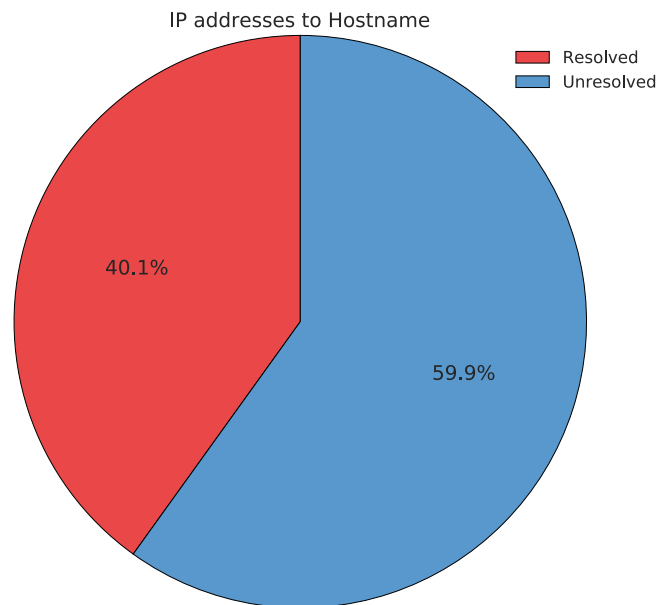
In all there 669 IP addresses that have hostname, this makes up 39% of the overall IP addresses found in the college of those 669 that did have have associating hostnames, 212 were found on port 80 (only), 34 on port 443 (only) and 423 on both ports.

6.2   ZGrab Results
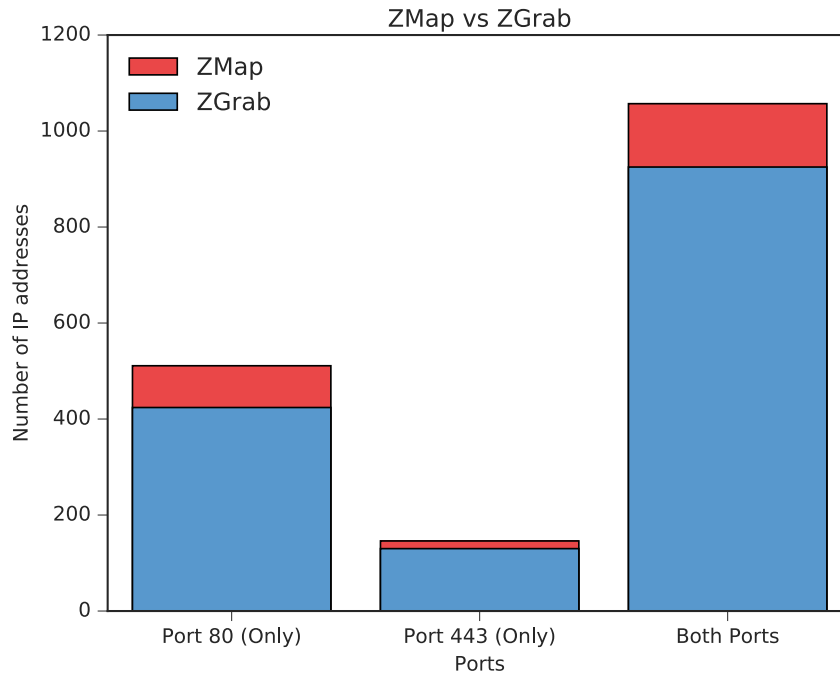
Figure 6.9: a nice plot1

!!!!!!!!!!!!!!!!!!!!!! MIGHT HAVE TO REDO GRAPH ABOVE !!!!!!!!!!!!!!!

!!!!!!!!!!!!!!!!!!!!!! Also state how many Irregular IP addresses we gathered !!!!!!!!!!!!!!!!!!!!!!!!!

The ZGrab scans were first conducted on the 2018-03-27 at 18:00 and finishing on 2018-04-23 at 15:00. In total we managed to collect service information of 1497 unique IP addresses. With 442 being only on port 80, 130 being on Port 443 only and 925 being open on both ports. The IP addresses that were open on both ports in the ZMap scans were not always found in our ZGrab scans. Of the 925 IP

addresses that are open on both ports 669 of those we managed to collect service information being on port 443, 924 of the 925 IP addresses we managed to collect service information regarding port 80. These results seen in figure 6.9 show that not all the IP addresses original scanned in the ZMap scans were present in the ZGrab scans, the main reasons for this were outlined in the Challenges subsection within the Implementation Chapter

6.2.1 Response Codes

!!!!!!!!!!!!!!!!!!!!!!!!!!!DO RESPONSE CODE TABLE !!!!!!!!!!!!!!!!!!!!!!!!!!!

6.2.2   TLS/SSL
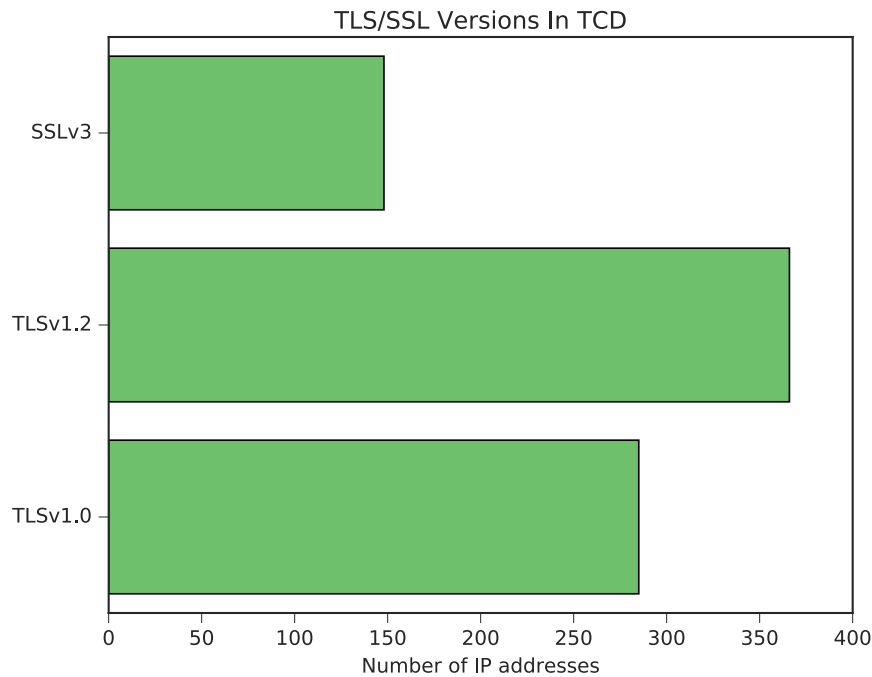
Figure 6.10: a nice plot1

Over the course of the banner grabs we have found 3 different versions of TLS/SSL being implemented in the college to provide encryption of data. It turns out that SSL 3.0 is still being used despite the fact that it is inherently broken and suffers from various attacks such as POODLE[19] [31] and many organisations [38] advise not to use it but use TLSv1.2 instead. One of the reasons as to why this old implementation of SSL/TLS is still being used, might be the fact that many legacy web browser such as IE6/XP only support SSL 3.0 and thus switching off support for SSL 3.0 would prevent browsers from working with the site [32]. The only IP addresses that are using SSL 3.0 also happen to be ones with self signed

certificates as most certificates that are self signed are mainly used in the interal network and under less constrictions in terms on the policy that is enforced. Next we have TLSv1.0 which is the next version of TLS after SSLv3.0, similar advise is given with regards to the use of TLSv1.0 as it too is also a legacy protocol that suffers from problems such as the BEAST attack[19] which is an vulnerability in the implementation of the Cipher Block Chaining (CBC) in TLSv1.0. The most secure of the TLS versions found in the college has the greatest number of IP addresses (366) that are using it, to secure their communication. No version of TLS1.1 was ever discovered throughout the ZGrabs.

6.2.3   Signature Algorithm
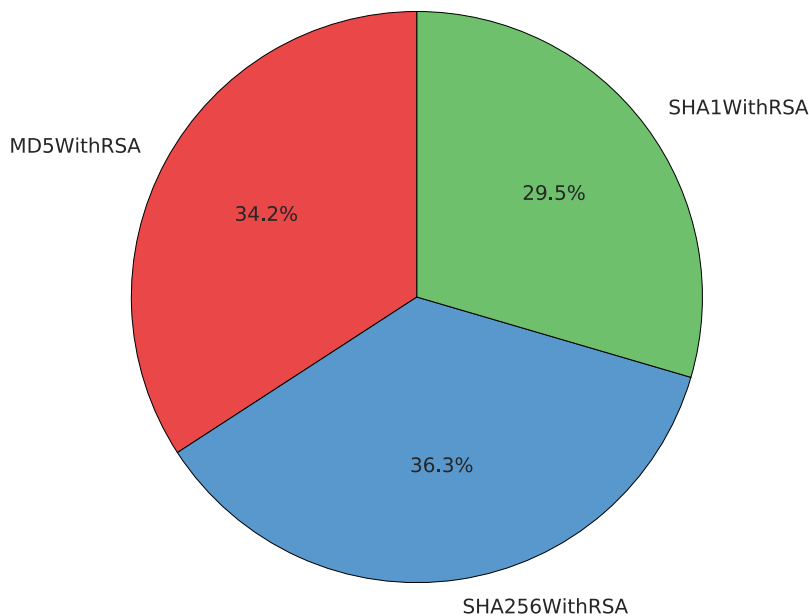
Figure 6.11: a nice plot1

One of the more surprising discoveries is the prevalence of MD5 hash function. In total we found 273 unique IP addresses that are using MD5withRSA to sign their certificates, of these 264 are self signed certificates. According to the Internet Engineering Task Force MD5 is no longer acceptable where collision resistance is required such as digital signatures [40] this is due to the fact that MD5 hash functions allows the construction of different messages with the same MD5 hash resulting in a collision [37]. Along with MD5 being fundamental broken, SHA1 is also now insecure. Both of these signature algorithms in figure 6.11 used to sign

certificates account for 63.7% of 799 IP addresses in the college using unsecured hashing functions.[38]. With the remaining 290 IP addresses possessing certificates signed with SHA256 which is the recommend signature algorithm to use[38].

6.2.4   Cipher Suite
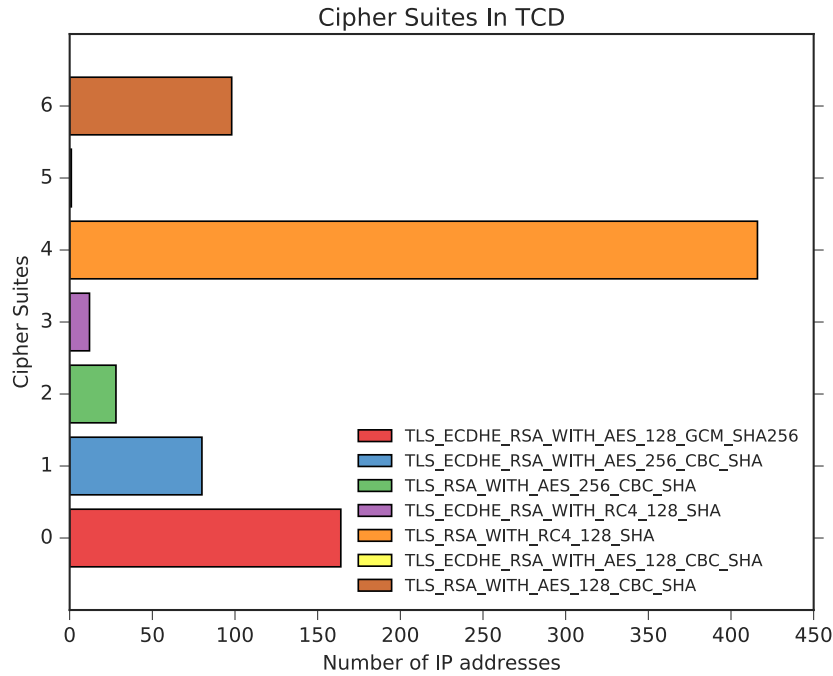
Figure 6.12: a nice plot1

As we can see in figure 6.12 $TLS_R SA_W ITH_R C4_1 28_S HA$ is by far the most widely used cipher suite in the college, even though community at large advise against RC4 cipher suites as the bytes used to encrypt plaintext is not as random as one would hope resulting in attackers being able to exploit the bias in the

keystream that encrypyt the plaintext, uncovering previous encrypted plaintext messages [34]. The most least wide cipher suite found is $TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA$ with only a single IP address offering up this in cipher suite to be used.

6.2.5 Browser Trusted
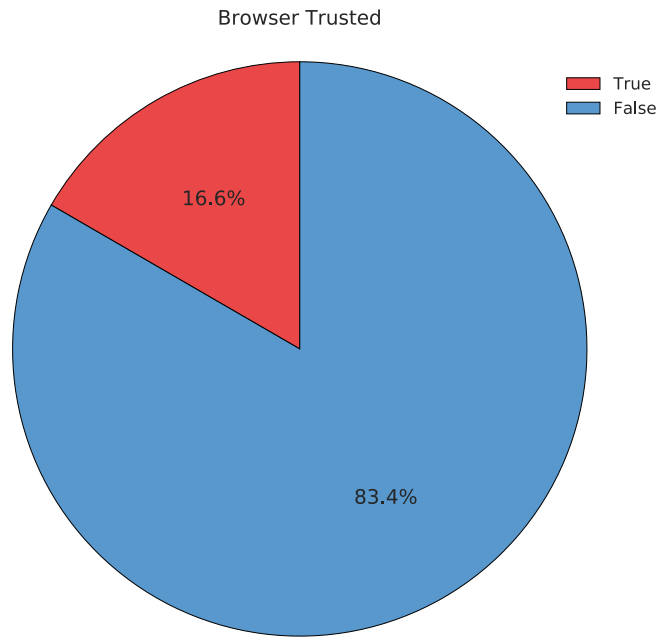
Figure 6.13: a nice plot1

Of the certificates we managed to collect only 133 were browser trusted. Further more the cipher suites that browser trusted certificates use differs from that of the entire certificates collected with 45 IP address of using $TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256$ cipher suites and only 17 using $TLS_RSA_WITH_RC4_128_SHA$. All the browser

trusted certificates have been found to use SHA256withRSA as any certificate signed with MD5 or SHA1 is considered insecure [38].

Issuer	Number of Certificates Issued
TERENA SSL CA 2	10
COMODO RSA Domain Validation Secure Server CA	2
Let's Encrypt Authority X3	7
DigiCert SHA2 High Assurance Server CA	1
TERENA SSL CA 3	112
GeoTrust DV SSL SHA256 CA	1

Table 6.3: Browser Trusted Certificate Issuers

There is a breakdown of the top issuers of broswer trusted certificates within TCD presented in table 6.3 with Lets Encrypt who issue certificates with an expiry of 90 days [16] being the third most popular Issuer of browser trusted certificates in the college. The vast majority of public keys length used stays within the recommend value [38] with 130 using public keys of length 2048, and remanining 3 using public keys of length 4096.

6.2.6 Expired Certificates

Cert Expired
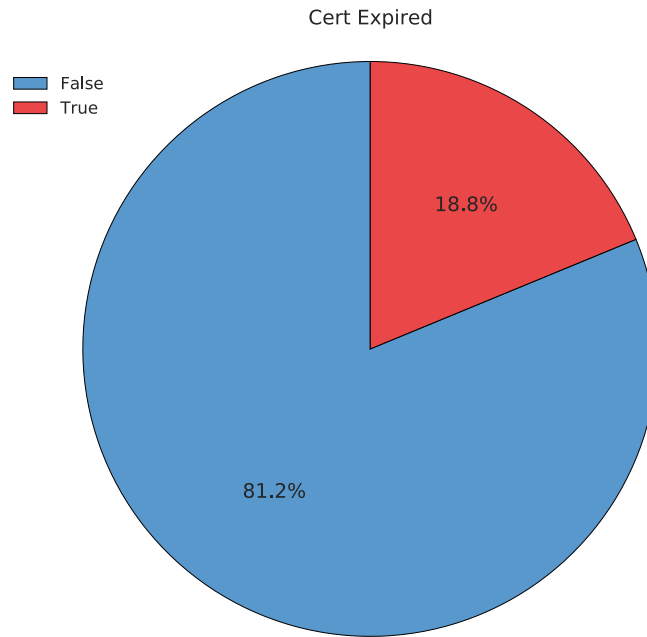
Figure 6.14: a nice plot1

From figure 6.14 we can see that 18.8% which is 150 IP addresses have certificates that are out of date with the most recent certificate expiring in 2018-03-16T15:49:14Z this certificate turned out to be a self signed cert, in terms of certificates that are signed by a CA the most expired on 2018-02-14T15:25:09Z. On the other scale the oldest expired certificates is a self signed cert with it's expire set to 1972-12-31T00:05:01Z, once in terms certs signed by CA the oldest being a cert that expired on 2014-08-01T23:59:59Z. One of the interesting findings in terms of the expired certificates is that there are 17 IP addresses with self signed certificates all

of whom have certificate expired on 2007-01-01T00:00:00Z while also having their certificated beginning on the 2002-01-01T00:00:00Z, they also share the same public key length of 1024 bits as well as being signed with MD5withRSA along with the same cipher suite of $TLS_R SA_W ITH_R C4_1 28_S HA$ all of these IP addresses turned out to be web servers of printer devices (print servers) specifically HP printers. What's interesting about this is that these devices could be an example of an employee being in charge of maintaining these devices and on leaving forgetting to mention that they exist. It also shows us that there is a good chance since these certificates haven't been updated that devices themselves could be using out of date software.

6.2.7 Self Signed Certificates

!!!!!!!!!!!!!!!!!!!!!!!!!!!!INSERT GRAPH OF SELF SIGNED CERTS!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

In total there were 555 IP addresses which had certificates that were self signed. The longest validity period found for a certificate was 27 years which was found on a single certificate with the shortest being valid for 12 weeks. Another finding was that of the 555 IP addresses that had self signed certificates 56 of them shared the same public key, which was the largest number of IP addresses that shared the same public key. they also shared the same issuer who had signed the certificates with MD5withRSA and had the same public key length of 1024 bits.

Public Key Length	Number of Certificates
1024	380
768	2
4096	5
2048	407
3072	1
512	4

Table 6.4: Public key lengths of all IP addresses found in ZGrab scans on port 443

Keeping with public keys, the self signed certificates found in the college contained the only 512 and 768 bit length certificates with regards to the rest of the certificates shown in table 6.4. The National Institute of Standards and Technology (NIST) recommends key sizes of at least 2056 bits, [2] this is due to successful attempts at breaking RSA by factoring for example 512 bit RSA keys were successful factored in 1999 [22] at the moment large numbers are quite difficult to factor but the recommendation by nist is a way to stay ahead of the curve.

6.2.8 Categorizing the Servers

Port 80 (only)	Port 443 (only)
Web Image Monitor	16
Web Server	21
Printer	15
DHCP Server	1
404	5
Other	3
Hello World	13
Redirect	1
NAS Server	1
Conference Device	8
Projector	1

Table 6.5: Categorizes of Irregular IP addresses

Throughout this product there were 95 successful banner grabs of Irregular IP addresses along with their root page. 63 of these were only on port 80, 2 on port 443 and 30 having known open connections both ports. Of the 30 on both ports 27 were found on port 80 while 25 being found on port 443, 22 of these IP addresses were common between those IP addresses having known connections on both ports in each of the ZGrab scans on port 80 and port 443. Web Image monitors as seen in table 6.5 were one of the most seen types of systems with all of them being exclusively on port 80 only, unfortunately this project hasn't been able to give an in depth information regarding these at the time of writing. Printers make up some

of both the common and irregular IP addresses among the college campus with a variety of models and devices such as HP, Lexmar and Cannon. Conferencing devices were also found with 7 of these being Polycom devices and the remaining being a Tanberg. There were 13 Hello World/test pages which could be either Student projects within the college or pages ensuring a server was functionally running. The actual Web Server pages found consisted of Microsoft and Appache server pages. There were 3 Other that had no distinguishable way to categories them as seen in Table 6.5

Chapter 7

Conclusion

[29] page 316

This project investigates the state of Web Servers within Trinity College Dublin. Within this project I've introduced both ZMap a high speed port scanner and ZGrab an application banner grabber. I've also showed how these applications can be used to effectively answers questions regarding the current state of web servers within TCD such as to what was presented here in Chapter 1.

This project has allowed me to gain a deep understanding of the underlying technologies that aim to make the web more secure. Throughout this project I've been able to investigate interesting security topics such POODLE, BEAST and MD5 collisions.

The methodologies and design of the scans adopted in this project should allow other institutions and organisations who intend to replicate similar results presented

here within their own private network as well as avoiding the challenges incurred in this journey.

With regards to IP addresses on port 443, the version of TLS used and the way it is configured is ultimatley up to the site owners and administrators to ensure that their servers are configured properly the approach taken in this project could be used to identify miss configured servers. For example over time cryptographic algorithms weaken, the methods used within this project could be implemented to locate servers offering weak cipher suites so that the correct configurations can be made.

Chapter 8

Future Work

While this report has demonstrated the use of ZMap and ZGrab on port 80 and port 443, there are many worthwhile areas of future work that could investigated.

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

Similarly to research discussed in Related Work [14] [10] it would be fascinating to see the adoption rate or change over time of IP addresses listening port 80 (HTTP) to ones listening on port 443 (HTTPS). This could carried out by running the scans implemented within this project over a longer time period to see these changes if at all any as well as investigating the possible reasons behind this, for example one reason could be the fact that Google now ranks sites implementing TLS/SSL as higher.

This report only detailed scans conducted on port 80 and port 443, in would be interesting to implement scanning on further ports in order to expand the view of activity that is currently on the college network.

Further expansion on the code produced within this project could be expanded in other to provide institutions/ organisations an easy to install application that would package the entire design, filtering and graphing of the scans conducted as well as producing daily summaries of the network in terms of number IP addresses listening and also notifying the user of any vulnerabilites or weak cipher suites being used such as servers using null cipher suite for example. This could also be extended to provide a ranking of the most secure web servers and more importantly the most vulnerable by using a weighting system similar to other researches [29] who have developed metrics for assessing web server which consist of a list of security recommendations that were taken from a set of interviews from security experts. The question of what database to use is a difficult question as discovered through the ZGrab scans that certain JSON objects produced could have a different number of fields, so if a certain field were not accounted for the schema of the database would not be compatible. Within the github of ZGrab [12] there is the schema of all the fields that could be given out in the output of JSON file which could then be parsed with python code similar to what was produced in this project accounting for all the possible fields and then stored in a sqlLite database for example. One problem with this is that it those not take for the account of the possible updates to ZGrab which could involve the changing of the schema produced in the output of the JSON file along with this the creators

are currently developing ZGrab 2.0 [13] which will replace the current ZGrab, at the time of writing this report it's unsure at the moment what changes there will be between the two versions. While there doesn't seem to be clear cut solution to implement an application that is scalable, this would be an evaluable en devour.

Bibliography

[1] ACER, M. E., STARK, E., FELT, A. P., FAHL, S., BHARGAVA, R., DEV, B., BRAITHWAITE, M., SLEEVI, R., AND TABRIZ, P. Where the wild warnings are: Root causes of chrome https certificate errors. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (2017), ACM, pp. 1407–1420.

[2] BARKER, E. B., AND DANG, Q. H. Recommendation for key management part 3: Application-specific key management guidance. Tech. rep., 2015.

[3] BEALE, J., DERAISON, R., MEER, H., TEMMINGH, R., AND WALT, C. V. D. *Nessus network auditing*. Syngress Publishing, 2004.

[4] BERNERS-LEE, T., FIELDING, R., AND FRYSTYK, H. Hypertext transfer protocol–http/1.0. Tech. rep., 1996.

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

[6] COTTON, M., EGGERT, L., TOUCH, J., WESTERLUND, M., AND CHESHIRE, S. Internet assigned numbers authority (iana) procedures for

the management of the service name and transport protocol port number registry. Tech. rep., 2011.

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

[8] 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.

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

[10] 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.

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

[12] DURUMERIC, Z., WUSTROW, E., AND HALDERMAN, J. A. Zgrab, https://github.com/zmap/zgrab.

[13] DURUMERIC, Z., WUSTROW, E., AND HALDERMAN, J. A. Zgrab, https://github.com/zmap/zgrab2.

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

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

[16] ENCRYPT, L. scoket,https://letsencrypt.org.

[17] FIELDING, R., GETTYS, J., MOGUL, J., FRYSTYK, H., MASINTER, L., LEACH, P., AND BERNERS-LEE, T. Hypertext transfer protocol–http/1.1. Tech. rep., 1999.

[18] FULLER, V., LI, T., YU, J., AND VARADHAN, K. Classless inter-domain routing (cidr): an address assignment and aggregation strategy. Tech. rep., 1993.

[19] HOLZ, R., SHEFFER, Y., AND SAINT-ANDRE, P. Summarizing known attacks on transport layer security (tls) and datagram tls (dtls).

[20] HOUSLEY, R., FORD, W., POLK, W., AND SOLO, D. Internet x. 509 public key infrastructure certificate and crl profile. Tech. rep., 1998.

[21] J. YAO, W. M. Smtp extension for internationalized email. Tech. rep., 2012.

[22] KALISKI JR, B. Rsa factoring challenge. In *Encyclopedia of Cryptography and Security*. Springer, 2011, pp. 1064–1065.

[23] KHAN, I. U., AND HASSAN, M. A. Transport layer protocols and services. *International Journal of Research in Computer and Communication Technology, IJRCCT 5* (2016).

[24] 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).

[25] 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).

[26] 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.

[27] LYON, G. F. *Nmap network scanning: The official Nmap project guide to network discovery and security scanning*. Insecure, 2009.

[28] MATPLOTLIB. matplotlib,https://matplotlib.org/.

[29] 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.

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

[31] MÖLLER, B., DUONG, T., AND KOTOWICZ, K. This poodle bites: exploiting the ssl 3.0 fallback. *Security Advisory* (2014).

[32] OWASP. Transport layer protection cheat sheet, https://www.owasp.org/index.php/.

[33] PIKE, R. The go programming language. *Talk given at Google's Tech Talks* (2009).

[34] POPOV, A. Prohibiting rc4 cipher suites. *Computer Science 2355* (2015), 152–164.

[35] RESCORLA, E. Http over tls.

[36] SOCKET. scoket,https://docs.python.org/2/library/socket.html.

[37] SSLLABS. md5,http://www.win.tue.nl/hashclash/rogue-ca/.

[38] SSLLABS. Transport layer protection cheat sheet,https://github.com/ssllabs/research/wiki/ssl-and-tls-deployment-best-practices.

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

[40] TURNER, S., AND CHEN, L. Updated security considerations for the md5 message-digest and the hmac-md5 algorithms.

[41] TURNER, S., AND POLK, T. Prohibiting secure sockets layer (ssl) version 2.0.

[42] 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.

[43] YLONEN, T., AND LONVICK, C. The secure shell (ssh) authentication protocol.

Appendix A

A first appendix

blah blah . . .

Appendix B

A sample of the questionnaire form used

blah blah