Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

School of Computer Science and Statistics

Localised Internet Scanning Infrastructure

Jordan Myers
15323206

B.A. (Mod) Integrated Computer Science
Final Year Project April 2019
Supervisor: Dr. Stephen Farrell

Declaration

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

Signed: ______________________ Date: ______________________

Abstract

A short summary of the problem investigated, the approach taken and the key findings. This should be around 400 words, or less.

This should be on a separate page.

Acknowledgements

You should acknowledge any help that you have received (for example from technical staff), or input provided by, for example, a company.

Acronyms

IANA	Internet Assigned Numbers Authority
MAC	Message Authentication Code
SMTP	Simple Mail Transfer Protocol
SSH	Secure Shell
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol

Contents

List of Figures

List of Tables

1 Introduction

1.1 Goal

The goal of this project is to design and implement a data store and data visualisation tool for a localised internet scanning infrastructure which supports Ireland-sized internet scans. Specifically, this report focuses on outlining how to use the Elastic Stack (1) for this purpose. The project uses Elasticsearch for data storage and Kibana for data visualisation and analysis. These should be able to run on a single machine with limited resources. The report aims to explain the process of installing and configuring the Elastic Stack, inserting the raw scan results (stored in JSON format) into Elasticsearch and visualising the results in Kibana. It outlines a command line tool developed to simplify this process.

1.2 Motivation

While large-scale, internet-wide scanning is a well-explored topic, more local scans could produce more actionable results. This project focuses on Ireland-wide scans, but it could be used for other similarly sized scans. The hope is that by using local knowledge and building relationships with key stakeholders (e.g. network operators, hosting providers, etc.), identified issues can be examined more closely and mitigations can be devised and implemented more effectively.

1.3 Report Structure

Chapter 2 outlines the background of the project and related work. It explains what internet scanning is, some of the reasons for it and some of the technologies used.

Chapter 3 explains the implementation of Elasticsearch as a data store and some of the challenges associated with it.

Chapter 4 describes the use of Kibana as a data visualisation tool and looks at some high-level visualisations and the insights provided by them.

Chapter 5 attempts to determine the success of the project by comparing implementations of the same experiment with and without the infrastructure implemented in this project both in terms of code (quantity and readability) and execution time. It also highlights some of the advantages and disadvantages of the new solution.

Chapter 6 provides some final remarks on the project.

Chapter 7 highlights some work that could be done following on from this project.

2 Background

2.1 Internet Scanning

2.1.1 ZMap/ZGrab

2.1.2 Censys.io

2.2 Transport Layer Internet Protocols

2.2.1 TCP

Transmission Control Protocol (TCP)

2.2.2 UDP

User Datagram Protocol (UDP)

2.3 Application Layer Internet Protocols

2.3.1 SSH

The Secure Shell (SSH) protocol is a network protocol which allows secure remote access over insecure networks (2). It provides an encrypted tunnel between the client and the server, as well as client and server authentication. It runs over TCP and has been assigned port 22 by IANA (3).

2.3.2 SMTP

Simple Mail Transfer Protocol (SMTP) is an electronic mail protocol designed to transfer mail reliably and efficiently. SMTP can run over any transport service which provides a reliable ordered data stream, most commonly TCP (4). IANA has assigned ports 25 and 587 to SMTP (3). Port 25 is most commonly used between mail servers, while port 587 is reserved for email message submission (5).

2.3.3 POP3

Post Office Protocol version 3 (POP3) is an electronic mail retrieval protocol used by email clients to retrieve mail that a mail server is holding for it(6). Once an email has been retrieved by the client, it is deleted from the server. It runs over TCP and has been assigned port 110 by IANA (3).

2.3.4 IMAP & IMAPS

Internet Message Access Protocol (IMAP) is an electron mail protocol which allows a client to read manipulate emails on a server (7). Unlike POP3, emails are not deleted on the server after they are read by a client. This allows synchronisation of a single mailbox across multiple clients, among other advantages. It can run on any reliable data stream, usually TCP. It has been assigned port 143 by IANA (3).

IMAPS is a secure version of IMAP which uses TLS. While IMAPS can be used on port 143 using the STARTTLS mechanism, it is recommended that port 993 is instead used for implicit TLS (3, 8).

2.3.5 HTTPS

Hypertext Transfer Protocol Secure (HTTPS) is a secure version of HTTP which uses TLS. It can be run over any transport service which provides a reliable connection-oriented data stream, usually TCP (9). It has been assigned port 443 by IANA (3).

2.4 Transport Layer Security (TLS)

Transport Layer Security (TLS) is a cryptographic protocol which provides secure communication over a computer network. Symmetric cryptography is used to encrypt data between two applications using keys unique to that connection, ensuring a private connection. The keys are based on a shared secret, negotiated during the TLS Handshake. Message authentication codes, computed using secure hash functions, are used to ensure a reliable connection (10). The TLS Handshake protocol is used to set up the secure connection.

2.4.1 TLS Handshake

The TLS Handshake is initiated by the client once the TCP connection has been established.

1. **Client Hello:** the client sends a hello message to the server. In it, the client states the version of TLS and the cipher suites it supports, as well as a client nonce (random number).
2. **Server Hello:** the server chooses the highest mutually supported TLS version to use. It chooses a cipher suite from the list of those supported by the client and generates a server nonce (random number). It then sends these three things along with its TLS certificate.
3. **Server Verification:** the client attempts to verify the server's TLS certificate with the issuing Certificate Authority to ensure the server is who it says it is.
4. **Pre-Master Secret:** the client generates another random number called the Pre-Master Secret (PMS). It gets the server's public key from the certificate, encrypts the PMS using the key and sends the encrypted PMS to the server.
5. **Master Secret:** using the client nonce, server nonce and PMS, the client and server independently compute the master secret using the same key derivation function, as specified by the TLS standard (10). They should both get the same result. The master secret is then split up to give a client MAC key, a server MAC key, a client encryption key and a server encryption key.
6. **Client Ready:** the client sends a message, encrypted using the derived client encryption key, to the server to indicate that it is ready to use encrypted messages from now on.
7. **Server Ready:** the server decrypts and verifies the message received from the client and sends a message to the client indicating that all further messages will be encrypted.

2.4.2 Cipher Suite

A cipher suite is a set of algorithms that are used to secure a connection in TLS. A set contains a key exchange algorithm, a symmetric encryption algorithm and a message authentication code (MAC) algorithm. When RSA is used for key exchange, it can also be used to authenticate the server - if the server successfully decrypts the pre-master secret encrypted by the client using the servers public key, it demonstrates that it knows the private key. When Diffie-Hellman is used for key exchange, either fixed parameters can be provided by the server, or it can send temporary parameters using the server key exchange message.

2.4.3 Public Key Fingerprint

A public key fingerprint is a cryptographic hash of a public key. As it is shorter than the key it represents, it can be used in place of the key to simplify some tasks.

2.5 Data Overview

3 Data Store

3.1 Elasticsearch

3.2 Inserting Data

3.3 Data Types and Mappings

3.4 Challenges

Java OOM error on bulk insert

4 Data Analysis

4.1 Kibana

4.1.1 Web Server and TLS Certificate

4.2 Dashboards

4.2.1 Ireland Data

4.2.2 Aviation Data Subset

4.3 Results/Insights

5 Evaluation

5.1 Same Keys

5.1.1 Lines of Code

5.1.2 Run Time

5.2 Scalability

5.3 Usability

6 Conclusion

7 Future Work

Bibliography

[1] Elastic. Elastic Stack. `https://www.elastic.co/`. [Online; accessed 11-April-2019].

[2] Tatu Ylonen and Chris Lonvick. The secure shell (ssh) protocol architecture. Technical report, 2005.

[3] Michelle Cotton, Lars Eggert, Joe Touch, Magnus Westerlund, and Stuart Cheshire. Internet assigned numbers authority (iana) procedures for the management of the service name and transport protocol port number registry. Technical report, 2011.

[4] Jon Postel. Simple mail transfer protocol. Technical report, 1982.

[5] Randall Gellens and J Klensin. Message submission for mail. Technical report, 2011.

[6] John Myers and Marshal Rose. Post office protocol-version 3. Technical report, 1996.

[7] Mark Crispin. Internet message access protocol-version 4rev1. Technical report, 2003.

[8] K Moore and C Newman. Cleartext considered obsolete: Use of transport layer security (tls) for email submission and access. Technical report, 2018.

[9] Eric Rescorla. Http over tls. Technical report, 2000.

[10] Tim Dierks and Eric Rescorla. The transport layer security (tls) protocol version 1.2. Technical report, 2008.

A1 Appendix