Using Blockchain to Create a Decentralised Security Model for Distributed Systems

Adam David Bruce
a.bruce3@newcastle.ac.uk

April 2021

Abstract

//TODO

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 COVID-19 and Cyberattacks

In the summer of 2020 during the midst of the COVID-19 pandemic, universities and research institutions worldwide were working hard to understand the structure of the virus and develop a vaccine in an attempt to return to normality. However, whereas some countries were making fast progress in understanding the virus, others were falling behind, and the virus began to put a strain on healthcare, and increasing critique on governments. In order to keep up with the nations at the forefront of vaccine development, nations turned to state-sponsored cyberattacks in order to both hinder nations, and also obtain research and information about other countries' vaccine efforts. One such example was the threat group 'Cozy Bear', formally known as Advanced Persistent Threat (APT) 29. APT29 used a number of tools to target various organisations involved in COVID-19 vaccine development in Canada, the United States and the United Kingdom. The National Cyber Security Center (NCSC) believe that the intention was highly likely stealing information and intellectual property relating to the vaccine [5].

In addition to the mortality of COVID-19, the virus also caused a number of economic issues across a number of nations. Global stock markets lost $6 trillion in value over size days from 23 to 28 February [6]. This gave private companies no other choice than to make large volumes of staff redundant, which increased job insecurity causing many people to become redundant, and in nations without suitable support or benefits, attackers turned to cybercrime for financial gain. These attacks represented the majority of cyberattacks aimed at both universities and the general public. A study of cyber-crime throughout the COVID-19 pandemic determined that 34% of attacks directly involved financial fraud with a number of attack surfaces used, the majority being phishing, smishing and malware [7].

University attacks became a frequent headline in the UK as universities suffered attacks from different threat actors. A number of threat actors launched attacks against multiple universities in the hope to find a vulnerability in at least one. One such attack was aimed at both Newcastle University and Northumbria University, two universities in extremely close proximity [8, 9]. The attack crippled both Newcastle and Northumbria Universities, however the attackers only managed to exfiltrate data from Newcastle University. Why was the attack successful on both occasions? Why wasn't knowledge of the attack shared?

One reason is that currently, there is no reliable or automated system in place to share this information. Such a system is what this paper will aim to create.

1.2 Distributed Systems

A distributed system is defined by Tanenbaum and van Steen as a "collection of independent computers that appears to its users as a single coherent system" [1]. Such systems are commonplace in peer-to-peer computing and sensor networks where each systems contributes some data via transactions to the system. A distributed system therefore should be autonomous and to the user, should appear as though they are interacting with a single system. Furthermore users and applications should be able to interact with the distributed system in a consistent and uniform way, regardless of where and when system interaction takes place. This requires a common interface provided by a stub which is used to bridge the gap between a programming language or protocol and the distributed system. This stub hides the differences in machine architecture and communication between the computer and the distributed system. The use of stubs creates a new software layer, known as middleware which runs on an Operating System (OS) and exposes distributed functions to higher-level applications and users.

1.3 Decentralised Systems

Reed defines a decentralised computer system as a computer system that "involves separation of the computers in the system by physical distance, by boundaries of administrative responsibility for individual computers and their applications, and by firewalls" [10]. Reed suggests that for a computer system to be decentralised, it must be separated by both physical distance and administrative responsibility, such that no single body administrates the system. One of the most well-known examples of decentralisation is cryptocurrency, a currency which takes no physical form, but instead exists entirely digitally. If cryptocurrency were to be governed by a central body, nefarious transactions could be used to launder money. Using a decentralised system ensures the transaction can only take place if all nodes within the system are in consensus that the transaction is genuine.

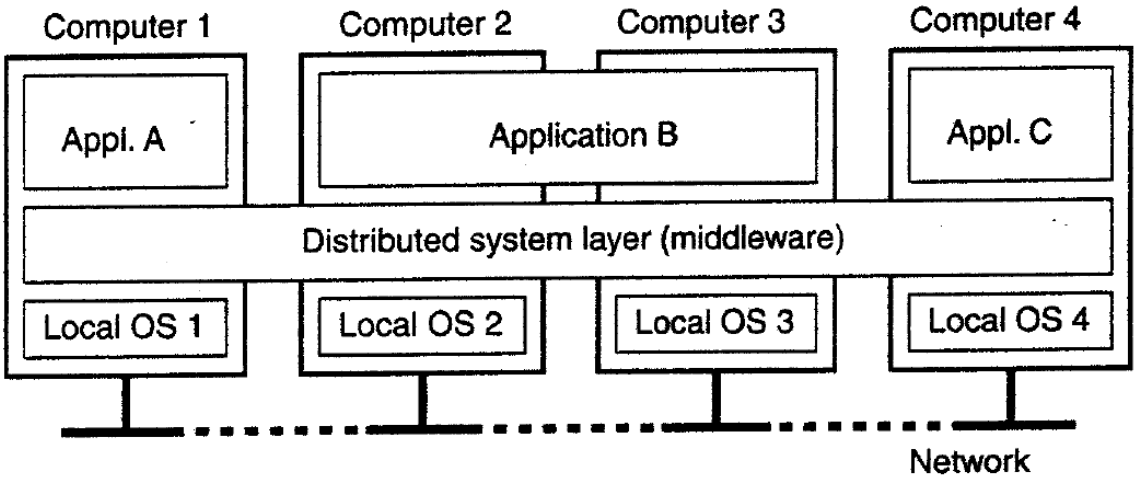

Figure 1.1: A Distributed System visualised as middleware [1]

1.4 Blockchain

Nofer et al. define blockchains as "data sets which are composed of a chain of data packages (blocks) where a block comprises multiple transactions. The blockchain is extended by each additional block and hence represents a complete ledger of the transaction history." [2]. Nofer et al. describe the basic fundamentals of a blockchain, which is that numerous blocks of transactions contribute to a larger chain. This chain is never controlled by a single body, instead a copy of the chain is stored at each node within a system, making blockchain a popular candidate for controlling transactions over a decentralised computer system. Hence, blockchain is the foundation for the vast majority of cryptocurrencies including Bitcoin[11] and Ethereum[12]. One of the key aspects of blockchain is the use of cryptographic hashing algorithms, these algorithms represent a block as a fixed-length string. For a block to be added to the chain, it must contain the hash of the previous block.

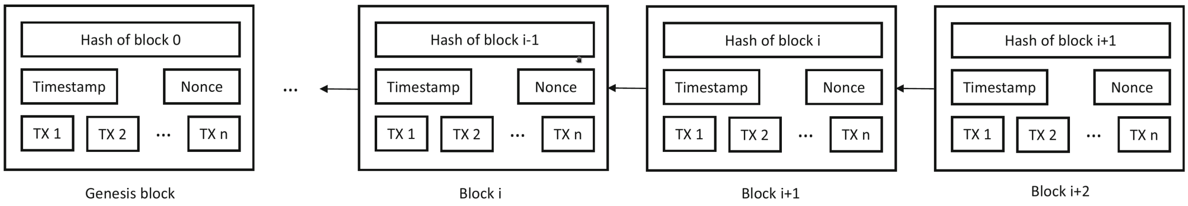

Figure 1.2: An example of a blockchain [2]

Chapter 2

Project Aims and Objectives

2.1 Aim

The original aim for this project was to design and create a decentralised firewall that could communicate knowledge of cyberattacks aimed at universities in real-time, allowing other universities to protect themselves from the same attacks. This system would be distributed, and hence must conform to the previous description of a distributed system in section 1.2. Following an extensive amount of background reading, there appeared to be no existing implementation or design of such a system which inspired me to alter my aim and instead focus entirely on designing a protocol and implementing a stub to demonstrate the protocol's effectiveness. This project will therefore not be implementing a firewall, but instead a system to coordinate firewalls. Further research determined that blockchain was the best choice for the underlying structure for such a protocol, and so this final change shaped the current aim for this project: **Using Blockchain to Create a Decentralised Security Model for Distributed Systems**.

2.2 Objectives

The following objectives provide an outline for what this project hopes to achieve:

1. Evaluate the effectiveness of existing distributed security mechanisms.

2. Investigate methods of establishing connections and synchronising computers within distributed systems.

3. Understand the structure of blockchains and adapt them for firewall transactions.

4. Implement and rest relevant resilience, fault tolerance and security mechanisms.

5. Compare the use of decentralised security mechanisms.

Chapter 3

Background

3.1 Distributed Systems

The primary reference used for distributed systems was Tanenbaum and van Steen's "Distributed Systems: Principles and Paradigms" [1], who's literature provides an in-depth explanation from the fundamental theory of distributed systems to the design and implementation of such systems. Key details that were taken from this publication are detailed below. In general, this book covered the essential components of creating a distributed system, however much of the detail with regards to client-server interactions was not applicable to this project due to it's decentralised nature. Furthermore, a large portion of the book was not of interest to this project as it focuses on distributed processing, which only comprises a small element of this project, hence a large volume of information regarding implementation of processing was not useful.

3.1.1 Architecture

Tanenbaum and van Steen cover many aspects of a distributed system's architecture spanning network, software and physical architecture. This project will implement a decentralised, peer-to-peer network architecture, which will be discussed in detail in section 3.2. The software used will consist primary of stubs, which are used to hide the differences in machine architecture and communication between the computer and the distributed system. The combined use of stubs creates a new software layer, known as middleware which provides a common interface between a client application, and the distributed system. Creating this layer enables applications to communicate via an application-level protocol, which is independent from the protocol spoken by the middleware.

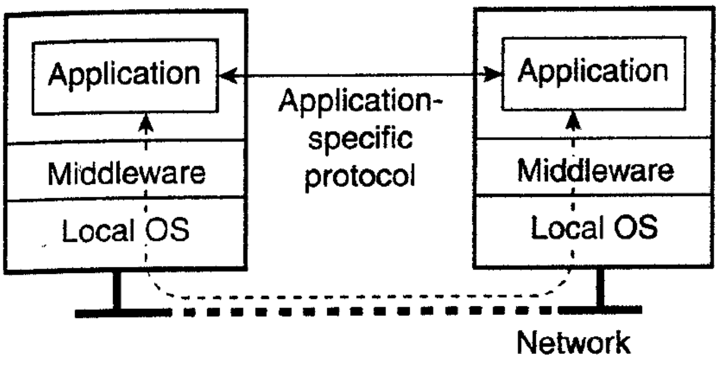

Figure 3.1: Application layer protocol running over middleware [1]

With regards to physical architecture, Tanenbaum and van Steen discuss a number of approaches to client-server architectures, however due to the decentralised nature of this project, non of Tanenbaum and van Steen's classifications apply.

3.1.2 Remote Procedure Calls (RPC)

Tanenbaum and van Steen introduce the concept of a Remote Procedure Call (RPC). RPCs are used to execute some action on a remote node within a distributed system. Tanenbaum and van Steen provide a concise breakdown of the steps required to execute an RPC:

1. The client procedure calls the client stub in the normal way.
2. The client stub builds a message and calls the local Operating System (OS).
3. The client OS sends the message to the remote OS.
4. The remote OS gives the message to the server stub.
5. The server stub unpacks the parameters and calls the server.
6. The server does the work and returns the result to the stub.
7. The server stub packs it in a message and calls it's local OS.
8. The server's OS send the message to the client's acrshortOS.
9. The client's OS sends the message to the client stub.
10. The stub unpacks the result and returns to the client.

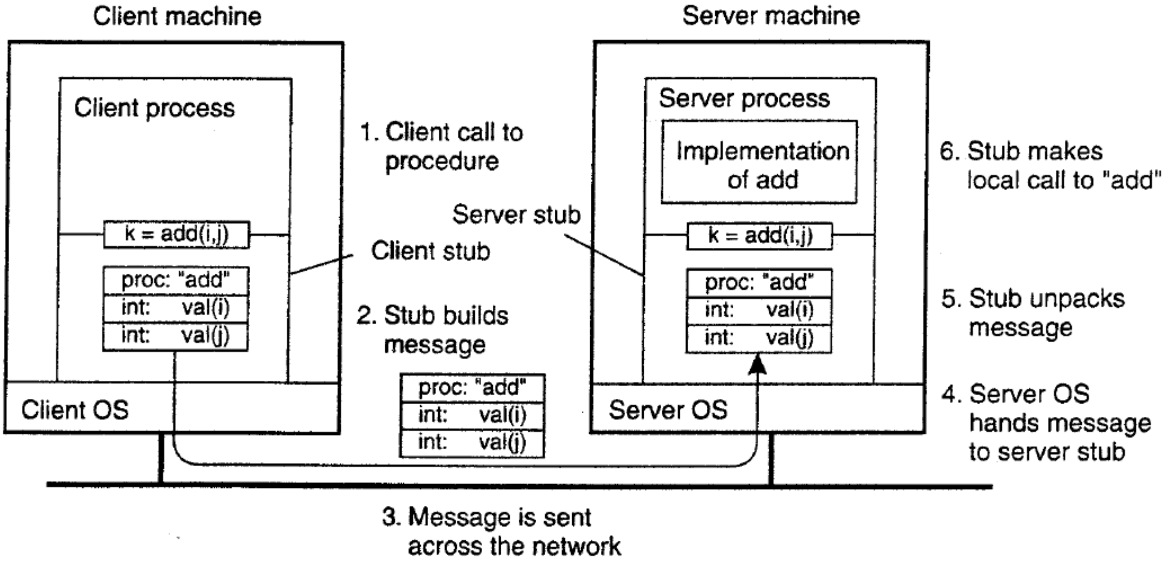

Figure 3.2: A breakdown of an RPC [1]

3.2 Decentralised Systems

Gray's "An Approach to Decentralized Computer Systems" [13] provided the basis for the decentralisation aspect of this project. Gray summarises the advantages of using decentralised systems, a number of which support the argument for using a decentralised topology in this project. The advantages which are relevant to this project are documented below.

- **Capacity**: A decentralised system can support a large number of devices.
- **Response Time**: Having devices in close proximity can reduce response times.
- **Availability**: A failure is likely to be limited to a single site, allowing the rest of the system to continue normal operation.
- **Security**: Removing the central controller in a traditional distributed system removes the risk of an attack compromising the whole system.

Gray's article also looks at how decentralised systems should be designed including data types, network protocols and transactions. There are a number of similarities between Tanenbaum and van Steen's RPCs and the structure Gray proposes for decentralised transactions. For this project however, the finer details proposed by Gray's system are not relevant as the literature uses a large number of examples base heavily on financial transactions, which contain a number of additional complexities over the transactions used within this project.

3.3 Blockchain

The primary reference used for blockchain was Nofer et al.'s "Blockchain"[2]. Nofer et al. provide a high level overview of blockchain, focusing primarily on the structure and implementation, with some consideration of the current applications of blockchain in both financial and non-financial settings. Although concise, this publication provides a valuable summary of the essential components of blocks in order to create a ledger which can accurately trace transactions. Although not essential for this project, Nofer et al. additionally discuss how blockchain can be implemented into smart contracts. In general, this literature was useful in providing a baseline for the structure of blocks within a blockchain, and clearly explained the purpose of each field within the block, which allowed informed decisions to be made in regard to the structure of blocks used in this project.

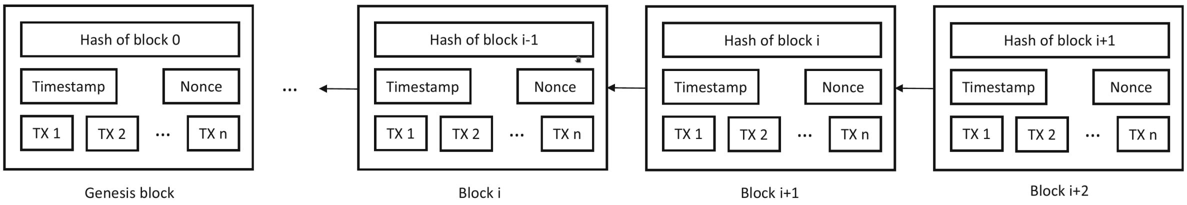

Figure 3.3: An example of a blockchain [2]

3.4 Distributed Security

The primary reference used for distributed security was Rivets and Lampson's "SDSI - A Simple Distributed Security Infrastructure" [3]. This publication provides an in-depth explanation of how a public-key infrastructure can be used in conjunction with access control lists to create a distributed security infrastructure. The majority of the literature within this publication is focussed on creating and issuing certificates, something that is not relevant for this project, however Rivets and Lampson did provide clear requirements over the data structures within such a system. Rivets and Lampson implement a message system similar to that of Tanenbaum and van Steen's in section 3.1. The message system proposed by Rivets and Lampson contains only a type and dictionary of attributes.

```
( type:
  ( Attribute1: value1 )
  ( Attribute2: value2a value2b value2c )
  ... )
```

Figure 3.4: Message format for the SDSI Model [3]

Additionally, Rivets and Lampson detail the concept of objects, which are defined by a type. This type is expressed in the form

```
protocol-name.message-type
```

3.5 Firewalls and Firewall Rules

Al-Shaer and Hazem provide a detailed explanation of how firewall policies should be modelled and managed in “Modeling and Management of Firewall Policies” [14]. This article explores methods of modelling policies and rules and provides a deep analysis of how those rules are interpreted by a firewall. The most relevant discussion within this literature is the structure of a firewall policy which is defined by Al-Shaer and Hazem as a set of rules, which act as records, with the following seven fields:

- **Order**: The priority of a rule.
- **Protocol**: Either Transmission Control Protocol (TCP) or User Datagram Protocol (UDP).
- **Source IP**: The source IP address.
- **Source Port**: The source port.
- **Destination IP**: The destination IP address.
- **Destination Port**: The destination port.
- **Action**: The action the firewall should take (e.g. ACCEPT, DENY).

In regards to this project, there was little other relevant content in the literature. Following the change in this project’s aim detailed in section 2.1, this project was no longer concerned with the implementation of a firewall or the interpretation of firewall rules, which deemed the vast majority of Al-Shaer and Hazem’s publication irrelevant.

Additional background information with regards to how firewalls are integrated into infrastructure came from Dulaney and Eastton’s “CompTIA Security+ Study Guide: Exam SY0-501” [15]. Dulaney and Eastton’s study guide covers a large number of aspects associated with cyber security, including technological, physical and psychological mitigations. With regards to firewalls, this publication details how the placement of firewalls can be used to form a Demilitarised Zone (DMZ), which is a common network layout used by universities, as it permits certain areas of the network to be accessible from outside the local network.

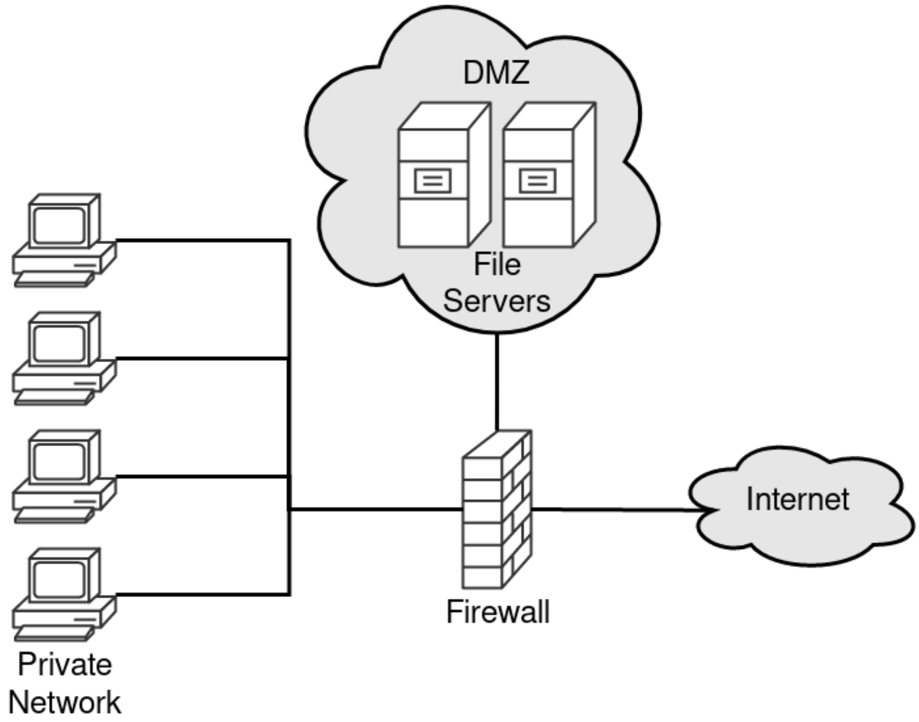

Figure 3.5: Using a firewall to create a DMZ

This information provided a strong understanding into where this project would fit in a standard network model. This publication also covered firewalls, however the information provided was not as detailed as that from Al-Shaer and Hazem, and hence no other aspects of this book were used.

3.6 Fault Tolerance

The primary reference for fault tolerance was Guerraoui and Schiper's "Fault-Tolerance by Replication in Distributed Systems"[4]. This literature details how replication can be used to provide fault tolerance in a distributed system in addition to ensuring consistency is maintained. A number of backup techniques are discussed however the technique that is best suited for application is primary backup replication. Primary backup replication consists of a client invoking an operation which is the applied to the primary data, and then cascaded to a number of additional backups.

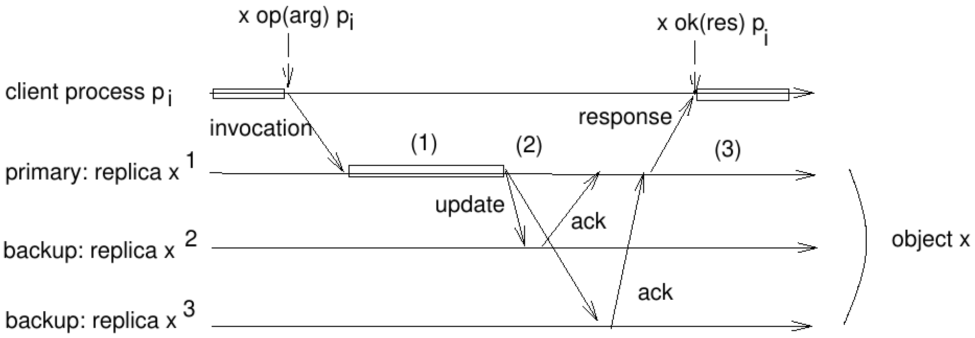

Figure 3.6: Primary backup technique [4]

Guerraoui and Schiper continue to discuss methods of detecting faults and appropriate ways to deal with them. The methods covered however require a much greater level of control than that achievable by a single application running on a standard OS, and hence are not applicable to this project.

3.7 Computer Networks

The primary reference for computer networks was Lammle's book "CompTIA Network+ Study Guide: Exam N10-007" [16]. Lammle's study guide covers a wide range of aspects associated with computer networks including physical implementations, subnets, security and protocols. Two sections which are relevant to this project are network layer protocols (TCP and UDP), and network topologies. Lammle provides a comprehensive explanation of the differences between Transmission Control Protocol (TCP) and User Datagram Protocol (UDP), however the difference that is most relevant to this project is TCP's requirement for a connection to be established prior to transmission. The book explains how establishing a TCP connection takes additional time and blocks the port whilst attempting to establish a connection, which prevents any other connection from being made from that port. This is not ideal for a decentralised system as messages will be sent on an ad-hoc basis, with strict time constraints, and therefore UDP will be more suitable for this project.

With regards to network topologies, Lammle details seven approaches: bus, star, ring, mesh, point-to-point, point-to-multipoint and hybrid. In order to create a truly decentralised distributed system, the mesh topology is best suited to this project. In a mesh topology, each device is connected to every other device, which provides the highest level of redundancy possible, as if one device were to crash, or a cable be disconnected, communication can continue via the other devices.

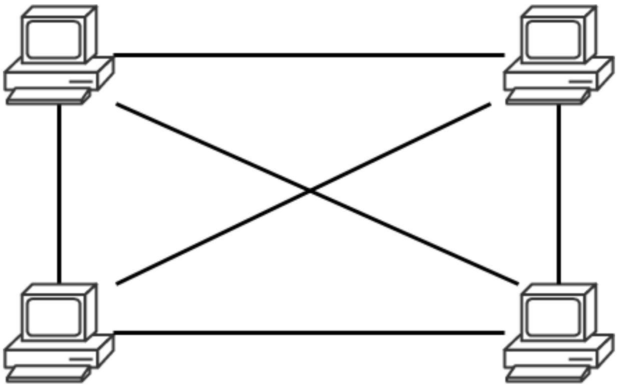

Figure 3.7: Mesh Topology

3.8 Inter-Process Communication (IPC)

Tanenbaum and Woodhull provide an extremely detailed breakdown of their UNIX based operating system in "Operating systems: Design and Implementation" [17]. This book covers all aspects of an OS, providing insightful explanations of the decisions made at every step in the design process. As this project is designed to run as an application, many of the details covered in this publication are irrelevant, however Tanenbaum and Woodhull cover one essential aspect of this project: Inter-Process Communication (IPC). IPC is a function within many Operating System which enables multiple processes to communicate by passing messages to each other. IPC will be the technique used to interact with the distributed system, as a client process will use IPC to issue commands to the stub.

Chapter 4

Design and that

Glossary

blockchain A growing list of records, called blocks, that are linked using cryptography. 2, 5–7, 10

cryptocurrency A digital currency produces by a public network. 5

hashing The practise of taking data and representing that data as a fixed-length string. 6

ledger A record of all transactions executed on a particular cryptocurrency. 5

malware Malicious computer software that interferes with normal computer function or sends personal data about the user to unauthorised parties. 4

middleware Software that functions at an intermediate layer between applications and the operating system to provide distributed functions. 2, 5, 8

phishing Sending an email that falsely claims to be from a legitimate organisation, usually combined with a threat or request for information. 4

smishing Sending a text message via SMS that falsely claims to be from a legitimate organisation, usually containing a link to a malicious website. 4

stub A piece of code that is used to marshal parameters for transmission across the network. 5, 7–9, 13

Acronyms

APT Advanced Persistent Threat. 4

DMZ Demilitarised Zone. 2, 11, 12

IPC Inter-Process Communication. 13

NCSC National Cyber Security Center. 4

OS Operating System. 5, 9, 13

RPC Remote Procedure Call. 2, 9, 10

TCP Transmission Control Protocol. 11, 13

UDP User Datagram Protocol. 11, 13

Bibliography

[1] A. S. Tanenbaum and M. van Steen, *Distributed Systems: Principles and Paradigms.* Pearson Prentice Hall, 2007.

[2] M. Nofer, P. Gomber, O. Hinz, and D. Schiereck, "Blockchain," *Business & Information Systems Engineering*, vol. 59, pp. 183–187, 2017.

[3] R. L. Rivest and B. Lampson, "SDSI-a simple distributed security infrastructure," Crypto, 1996.

[4] R. Guerraoui and A. Schiper, "Fault-tolerance by replication in distributed systems," in *International conference on reliable software technologies*, pp. 38–57, Springer, 1996.

[5] NCSC and CSE, "Advisory: APT29 targets COVID-19 vaccine development." `https://www.ncsc.gov.uk/files/Advisory-APT29-targets-COVID-19-vaccine-development-V1-1.pdf`, 2020.

[6] P. K. Ozili and T. Arun, "Spillover of covid-19: impact on the global economy," *SSRN 3562570*, 2020.

[7] H. S. Lallie, L. A. Shepherd, J. R. Nurse, A. Erola, G. Epiphaniou, C. Maple, and X. Bellekens, "Cyber security in the age of COVID-19: A timeline and analysis of cybercrime and cyber-attacks during the pandemic," *Computers & Security*, vol. 105, 2021.

[8] BBC, "Newcastle university cyber attack 'to take weeks to fix'." `https://www.bbc.co.uk/news/uk-england-tyne-54047179`, 2020. Accessed: 01/04/2020.

[9] BBC, "Northumbria university hit by cyber attack." `https://www.bbc.co.uk/news/uk-england-tyne-53989404`, 2020. Accessed: 01/04/2020.

[10] D. P. Reed, *Naming and synchronization in a decentralized computer system.* PhD thesis, Massachusetts Institute of Technology, 1978.

[11] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system." `https://bitcoin.org/bitcoin.pdf`, 2008.

[12] V. Buterin, "A next generation smart contract & decentralized application platform." `https://whitepaper.io/coin/ethereum`, 2013.

[13] J. N. Gray, "An approach to decentralized computer systems," *IEEE Transactions on Software Engineering*, vol. SE-12, no. 6, pp. 684–692, 1986.

[14] E. S. Al-Shaer and H. H. Hamed, "Modeling and management of firewall policies," *IEEE Transactions on network and service management*, vol. 1, no. 1, pp. 2–10, 2004.

[15] E. Dulaney and C. Easttom, *CompTIA Security+ Study Guide: Exam SY0-501.* John Wiley & Sons, 7 ed., 2018.

[16] T. Lammle, *CompTIA Network+ Study Guide: Exam N10-007.* John Wiley & Sons, 4 ed., 2018.

[17] A. S. Tanenbaum and A. S. Woodhull, *Operating systems: design and implementation.* Pearson, 3 ed., 2015.