Master Thesis

Master's Programme in Network Forensics, 60 credits

Enhancing Supply Chain Cybersecurity with Blockchain

Thesis in Digital Forensics, 15 credits

Halmstad 20.05.2022
Ari Hämäläinen
Rekha Nadesan

HALMSTAD UNIVERSITY

Enhancing Supply Chain Cybersecurity with Blockchain

Authors:

Ari Hämäläinen
Rekha Nadesan

Supervisor:

Sundas Munir

Abstract

Supply chains have become targets for hostile cyber actors. Motivations for cyber crimes include intellectual property theft, customer data theft and industrial espionage. The cyber threat landscape in which businesses operate is constantly evolving. The consequences of a successful cyber attack can be devastating for a business. Increasing the resilience of the supply chain in the digital environment is a complex task because the supply chain consists of different organisations with varying levels of cybersecurity defence capability. Orchestrating cybersecurity improvement in a supply chain requires visibility into the security posture of each participating organisation and this is generally lacking. This thesis studies the potential use of blockchain for enhancing the cybersecurity of the supply chain. The study simulates a permissioned blockchain among supply chain members to monitor digital assets important for cybersecurity. The blockchain is analysed to extract insights from the perspective of a supply chain cybersecurity oversight role. The study finds that a blockchain can provide visibility by sharing cybersecurity-related information among supply chain members. It can also provide a digital forensic record for incident response and forensic investigations.

Keywords:
Cybersecurity, blockchain, supply chain, information sharing

List of Abbreviations

APT	Advanced persistent threat
AWS	Amazon Web Services
CEA	Council of Economic Advisers (USA)
CISA	Cybersecurity and Infrastructure Security Agency (USA)
DDCMS	Department for Digital Culture Media & Sport (UK)
DLT	Distributed ledger technology
DNS	Domain name system
DOC	Department of Commerce (USA)
DOHS	Department of Homeland Security (USA)
DOJ	Department of Justice (USA)
DSB	Defense Science Board (USA)
ENISA	European Union Agency for Cybersecurity
HSM	Hardware security module
IBM	International Business Machines
ICS	Industrial control system
ICT	Information and communications technology
IOC	Indicators of compromise
IOT	Internet of things
IP	Internet protocol
ISO	International Organization for Standardization
IT	Information technology
MSP	Managed service provider
NCSC-UK	National Cyber Security Centre (UK)
NCSC-US	National Counterintelligence and Security Center (USA)
NERC	North American Electric Reliability Corporation
OT	Operational technology
PCI-DSS	Payment Card Industry Data Security Standard
PII	Personally identifying information
PKI	Public key infrastructure
POTUS	President of the United States
RFID	Radio frequency identification
SCADA	Supervisory control and data acquisition
SIEM	Security information and event monitoring
SME	Small and medium enterprise
SSH	Secure shell
TLS	Transport layer security
UDP	User datagram protocol
UK	United Kingdom
USA	United States of America
VPN	Virtual private network

Table of Contents

1. Introduction

A supply chain is an agreed upon set of relationships established between trading business partners to facilitate the efficient development, production and distribution of products in profitable volumes. This interdependence creates a logical and contractual supply chain consisting of a set of business partners. Together, they perform market-driven development, manufacturing and distribution processes to provide services or products to the market. The cybersecurity of the supply chain becomes relevant because of the interconnections and links between partner organizations. Blockchain's potential use as an extra layer of security on top of existing cybersecurity systems used by the supply chain is our area of interest.

The idea for this research work had its genesis in the application of our minds to the problem of cybersecurity in supply chains. We were curious to find a realistic, practical way to contribute to addressing a complex and pressing challenge. Blockchain technology has developed into a useful means of sharing data, for example, in digital currencies. In the context of the supply chain, blockchain technology is being leveraged, for financial aspects such as transaction processing and distribution aspects such as product tracking. We wanted to explore properties of blockchain that could be leveraged to enhance security of each partner's computer systems and in this way contribute to the overall cybersecurity of the supply chain. The principle is that if the individual links in a chain of security can be strengthened, then the whole chain becomes stronger. The aim of this thesis was to find a solution that could be implemented over and above any existing cybersecurity practices employed in securing the computer systems of supply chain partners. This thesis explores "permissioned" blockchain instead of other types which are designed to be deployed in an untrusted environment. Supply chain relationships are usually founded on sound business and legal contracts and a permissioned blockchain architecture seems appropriate in this context.

Modern supply chains operate in an environment that could be described as a cyber wild-west replete with hostile actors seeking to plunder the digital territory of organizations that operate in today's digitalized global economy. The consensus is that "Security incidents against international supply chains are threats to international trade and the economic growth of trading nations" (ISO, 2007). It was estimated that "malicious cyber activity cost the U.S. economy between $57 billion and $109 billion in 2016" (CEA, 2018).

Software supply chain attacks are those where hostile cyber actors compromise a supplier to exploit an existing trusted connection that the

supplier has with the final target, the supply chain customer (CISA, 2021). Attacks on the software supply chain are dangerous because many software products facilitate privileged access to the customer's system where the software is installed. The software product installed in the customer's premises frequently establishes routine communications with the supplier's remote systems thus creating a convenient infiltration vector if the supplier is compromised. Network defenders face a challenging task because organisations seldom control their entire software supply chain. Our proposal may help in meeting this challenge. Two well-known examples of supply chain attacks are described next.

Solarwinds, which produces a digital software product used by many customers including government agencies, recently suffered a supply chain cyber attack (NCSC-US, 2021). The development environment of this product was compromised by attackers who gained unauthorised access into this part of SolarWinds' manufacturing process. The attackers inserted malicious modifications into the product which, when used by Solarwinds' customers, allowed the same attackers to gain unauthorised access to customers' systems. More than 100 customers (private-sector companies and nine federal agencies) were compromised in follow-on attacks enabled through their unwitting use of the compromised Solarwinds product. This supply chain attack was attributed to the Russian Foreign Intelligence Service, a remnant of the former Soviet Union's oppressive communist regime's KGB. This well-resourced state-sponsored APT actor specialises in nation state cyber espionage. The potential to gain access to systems and data used by government agencies motivated these threat actors to meticulously plan, stage and execute a supply chain attack against Solarwinds. The strategy of the attackers was to leverage the supply chain as a vector to gain access to their intended final targets, any Solarwinds customer.

A similar supply chain attack occurred against Kaseya (NCSC-US, 2021). In this case the main consequence was loss of data and business productivity due to ransomware infection. Kaseya, a critical MSP for many organizations was compromised and infected with ransomware. About 1500 customers were affected. This attack targeted a software product manufactured by Kaseya, a tier-2 supplier, and used by upstream tier-1 suppliers to provide services to those 1500 customers. One of these tier-1 suppliers is an MSP for the Swedish supermarket chain COOP. The compromised Kaseya software resulted in the ransomware infection spreading into COOP systems and rendering the supermarket chain unable to conduct business and disrupting the food supply chain. This attack demonstrates the potential of a supply chain attack to propagate to end customers who also suffer the

fallout. It also demonstrates the exposure to cyber risks that supply chain interconnectivity brings; an unfortunate side-effect of Industry 4.0.

The UK government offers guidance for supply chain primary organizations: "Understand the risk posed by your supply chain. Build an understanding of what your suppliers' security looks like. Build security considerations into your normal contracting processes" (NCSC-UK, 2018, p.14).

Foreign state-sponsored cyber actors use supply chain attacks as an avenue to undermine confidence in democratic institutions and erode democratic values (NCSC-US, 2021). These actors also perpetrate theft of intellectual property such as research which can undermine economic competitiveness of democratic nations. Theft of intellectual property is the most damaging type of cyber attack with victim organizations losing over 6% of their market value on average (CEA, 2018, p.12). SolarWorld AG became insolvent after suffering a cyber attack and theft of its intellectual property by hostile Chinese state-sponsored cyber actors (CEA, 2018; DOJ, 2014). This cyber attack enabled China to assume a dominant position in the global solar panel market.

The US government published a strategy document recognizing that well-resourced hostile cyber actors seek to exploit key supply chains which underpin the operation of US national critical infrastructure (NCSC-US, 2020). This kind of cyber warfare is expected to increase in the future because it can succeed without necessarily crossing the threshold of an act of war.

In May 2021 the UK government called for views from industry to inform the government's understanding of supply chain cybersecurity and later in the year published their response (DDCMS, 2021). In presenting the context of this thesis, we wanted to include these views from the front lines of the cyber threat landscape. These are the industries, businesses and organisations currently suffering heavy losses due to cyber crime and advanced hostile cyber attacks perpetrating industrial espionage and intellectual property theft. The corpus of responses to the call revealed a number of barriers to effective supply chain cybersecurity of which "limited visibility into supply chains" was seen the most severe barrier (DDCMS, section 3.1). Even though the UK National Cyber Security Centre provides guidance for businesses, respondents emphasised that barriers such as the one mentioned above "make implementation of the guidance challenging" (DDCMS, section 1). It is clear that organisations require platforms, tools, and processes to help them manage supply chain cybersecurity risk. Other issues and challenges identified in the responses included complexity of

modern global supply chains, multiple layers, increasing numbers of suppliers and lack of transparency. Oversight of the supply chain was identified as a critical element in managing the cybersecurity risks of a supply chain. This is where our work can also help by providing a possible mechanism and platform that can assist in supply chain oversight. The government response document also refers to cybersecurity's multi-dimensional nature as the "manifold facets of cybersecurity" (DDCMS, section 4.6). This thesis echoes the need to appreciate that adequate cybersecurity involves many elements often with different scientific and technological underpinnings and assumptions. Combining these into a practical cybersecurity system that meets supply chain cybersecurity objectives demands an approach which uses engineering judgement to create a workable solution which adds value to the supply chain and effectively reduces its cybersecurity risk.

1.1 Research Question

The primary research question of this thesis is framed as "Can blockchain technology be leveraged to improve supply chain cybersecurity?". If this is found to be the case, we then ask a further question: "What features of a blockchain implementation are the most useful to this end?". These questions are qualified by specifying that we seek practical, cost-effective and non-disruptive implementations that can co-exist with existing cybersecurity controls. The findings are critically evaluated with the question: "Can a supply chain oversight role extract useful insights from a blockchain that will assist in achieving common cybersecurity objectives?".

1.2 Definition of Supply Chain Cybersecurity

For the purpose of this study, we define the cybersecurity of the supply chain as the evolving set of strategies, policies, controls, methods, and processes employed to protect the confidentiality, integrity and availability of the supply chain ecosystem. This encompasses all suppliers and customers involved in the design, production, delivery, handling, deployment, use and maintenance of created products or services over their entire life-cycle. This definition is intended to capture the notion that the cyber threat surface is formed by the participation of many elements and stakeholders in the supply chain. What needs to be protected is the whole threat surface that defines the cybersecurity posture of the supply chain.

2. Literature Review

2.1 Scope

In order to highlight key aspects relevant to this thesis, we intersperse comments throughout this literature review. The cited works serve to inform our efforts by describing important supply chain elements. We contrast the diversity of views concerning the extent of blockchain's potential benefit in this context. Given the "infancy" of research into blockchain's application to supply chain cybersecurity (Cole et al., 2019, p.479) these works are reviewed to understand the broader context and to seek more specific context for this thesis. Issues that have bearing on this thesis are highlighted to illustrate the various inter-related perspectives. We find many unanswered questions in a field of research and application that is in its infancy. Our perspective is grounded in the knowledge and understanding of cybersecurity which we have acquired and developed during this degree programme.

This review begins by describing the situation and context in which supply chains find themselves by using a well-known example of a cyber attack on a supply chain. We touch on the history of computer security and how insufficient attention to this field coupled with the rapid digitalization of economies have translated into supply chain vulnerability. We then review the application of blockchain in this area and how the previous work informs our own exploration.

2.2 Methodology

This review was conducted by searching via the Halmstad University library search portal. Search terms "supply chain cybersecurity" were used to seek works at the intersection of two knowledge domains, supply chain and cybersecurity. This approach was refined using keywords "supply chain cybersecurity blockchain" to seek works that investigate the application of blockchain in this inter-sectional domain. These domain intersections are represented in Figure 1 below. We used the same search criteria to seek additional published knowledge from both the cybersecurity practitioner and industry domains using the main search engines. We deemed this approach beneficial because research in supply chain cybersecurity is a relatively new field where the real-world need is urgent and great.

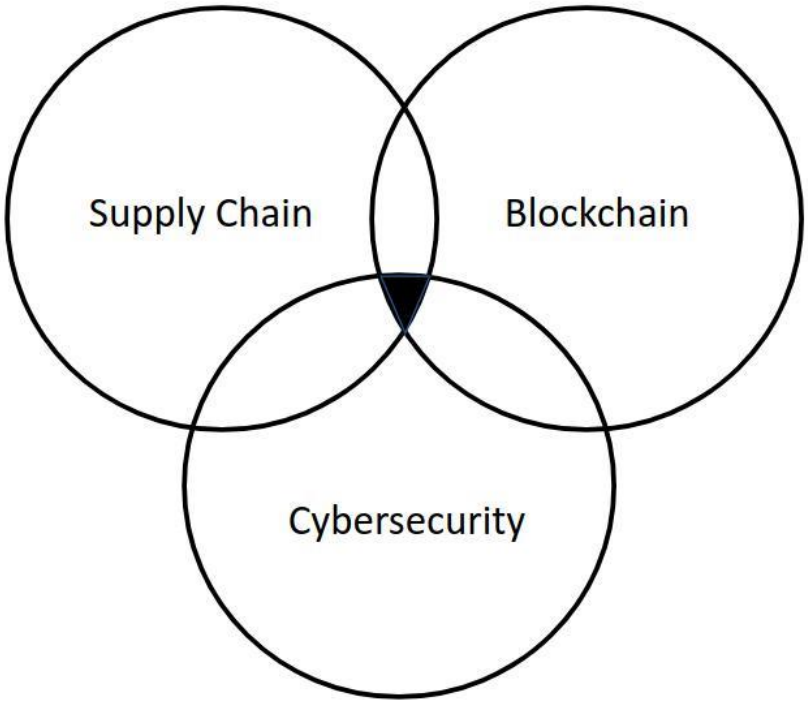

Figure 1: The focus of research in this thesis is located at the intersection of three knowledge domains. These are Supply Chain, Cybersecurity and Blockchain. The focus can be described as the application of blockchain for improving the cybersecurity of the supply chain.

Figure 1 above shows that the literature review is focused at the intersection of three knowledge domains. These are Supply Chain, Cybersecurity and Blockchain. Due to a lack of research in this narrow area, this thesis also included reviews of past research in the pairwise intersections of Supply Chain - Cybersecurity, and, Supply Chain - Blockchain. This allowed us to incorporate important issues identified by past research and better understand the context of our work.

Cyber threats challenge supply chains globally. We set the stage by describing one notable example of a successful supply chain attack in the following section.

2.3 The Threat Context

Plachkinova and Maurer (2018) and Middleton (2017) describe a notable example of a supply chain cyber attack which drew much attention to cybersecurity of supply chains. Target, a giant in the retail industry, was attacked and compromised in 2013 by cyber-attackers who gained unauthorised access to the personal and financial data of over 70 million

customers. Retail companies rely on customer trust and a positive brand image and this attack cast the spotlight on Target's exposure to cyber risk. It highlighted the notion of an organisation's attack surface and the need to be aware of this surface of exposure in the context of today's cyber threat landscape which is an unfortunate consequence and feature of the modern digitally inter-connected economy. The fact that Target had recently passed a PCI-DSS compliance audit, serves to illustrate that complete safety against cyber threats is an illusive goal. This is a fact widely recognised by cybersecurity domain experts. Following security best practices and implementing industry-accepted security controls is no guarantee of safety from cyber threats. Target was not adhering to good cybersecurity practices in many of their internal processes and these deficiencies were not exposed by the PCI-DSS audit. Compliance audits are one measure in the effort to secure an organisation's digital assets and operations but these cannot be effective by themselves. This incident supports the view that regulatory compliance does not guarantee protection from well-resourced attackers. Our own view is that cybersecurity is a multi-faceted problem which requires a multi-dimensional engineering approach to design security policies and systems tailored to the operations, processes and assets of a specific organisation. Effective cybersecurity is often more practically achievable using a layered approach where different security controls and methods are harnessed together in an orchestrated cybersecurity strategy (this approach is described later in the Discussion chapter of this thesis report). Target's business systems were compromised via a supply chain cyber attack in which a service provider was attacked and compromised. The vendor had access rights to Target's electronic billing systems. It is believed that the attackers used this vector to gain initial access into Target's computer systems from where they expanded their access laterally, exploiting other weaknesses. Eventually the attackers compromised Target's Point-of-Sale systems. The direct cost to Target of this breach was USD 200 million.

The next section describes some significant responses to supply chain cyber attacks by national governments including examples of their efforts to understand the scope and extent of the threat.

2.4 Responses to the Threat

The US government called for reviews of various key supply chains with the aim of strengthening the resilience of supply chains (POTUS, 2021). In response, government agencies conducted an assessment of supply chains supporting the ICT industry (DOC and DOHS, 2022). While the assessment focused attention on the integrity of the product throughout its journey through the supply chain, it highlights the risks and threats facing

participants and relying parties. Compromise of goods, components, services and parts made by the supply chain begins with a compromise of security at the suppliers which are in control of the production and distribution processes. A large organization can have many suppliers from which it purchases components directly, and each of those suppliers, in turn, can have multiple suppliers. Thus the complete supply chain of a large corporation can include many thousands of direct and indirect supplier organizations. A vulnerability at any one supplier has the potential to impact the end product or service produced by the supply chain and consumed by its customers. The report puts this into perspective: "Every company, organization, and individual that relies on ICT products is part of a global supply chain" (DOC and DOHS, 2022, p.63). All modern supply chains rely increasingly on ICT products as part of their business and operational processes. Today's digitized world and its constantly evolving threat landscape creates a large attack surface for any supply chain. This systemic exposure to cyber threats is often under-appreciated. The ICT supply chain is critical for a nation's economy and national security due to reliance by critical infrastructure sectors on ICT products. The assessment identified risks that threaten these supply chains. A key finding of the assessment was the lack of transparency into suppliers' cybersecurity posture. Complexity of supply chains makes securing them a difficult and complex exercise. A consequence of complex supply chain relationships is that the primary organisation, which procures goods and services from the suppliers, may not have visibility into the cybersecurity hygiene of their suppliers. Cyber threat actors try to exploit vulnerabilities in software and ICT systems used by global supply chains. Compromise of ICT products can render customers using the compromised products vulnerable to cyber attacks, exploitation, data theft, data destruction and espionage. Successful attacks can lead to a disruption of the supply chain or can enable subsequent attacks against the supply chain participants, in particular the end-customers and users of the products produced by the supply chain. The assessment identified the ICT industry as one of six key industries that are most likely to be targeted by foreign state-sponsored cyber actors given the advanced nature of research and development in this sector. Intellectual property theft by state-sponsored cyber actors has the potential to erode the national economic stability of victim nations. The report notes that "As ICT supply chains have become more extended, the number of threat vectors that can be exploited to access proprietary information flows along the supply chain has grown exponentially" (DOC and DOHS, 2022, p.70). The assessment recommends that the private sector build more transparency into their supply chains and that the public sector identify ways to improve cybersecurity for SME's that are part of supply chains.

The European Union Agency for Cybersecurity studied supply chain cyber attacks from January 2020 to July 2021 as part of its mandate to help improve cybersecurity across the Union (ENISA, 2021). The agency's report attributes the importance of supply chains to the ripple effect that successful attacks may have on many customers affected by a single compromised supplier. 62% of attacks on customers in 2020 took advantage of their trust in their supplier and hence the report identifies a need for new protective measures that "incorporate suppliers" (ENISA, p.3). This finding is a key guiding principle in this thesis. The cascading effect of supply chain attacks was evidenced recently in both the Solarwinds and Kaseya attacks which we described earlier. Among the impacted organisations were government agencies and corporations dependent on these supply chains. Most of the supply chain attacks documented in the ENISA report show that the initial target was software code, a kind of digital asset. This highlights the importance and priority that should be given to protection of digital assets such as code produced in a supply chain for use by its customers. The purpose of a supply chain attack is to compromise the customers who depend on the product or service created in the supply chain. The supply chain attack is a stepping stone for a subsequent attack against the dependent customers who use the products of the supply chain. An attractive target for a supply chain attack is any software asset created by a supplier which is used by customers of the supply chain. Such a compromised asset will be routinely downloaded by customers due to their trusted relationship with the supplier. The compromised software, usually infected with malicious code, will be executed by customers on their computer systems. Such an attack, while only affecting one supply chain digital asset, typically leads to compromise of many customers. Herr et al. (2020) offer the sobering perspective that agile software development processes can create new opportunities for risk propagation throughout the codebase of a project ultimately introducing vulnerabilities into trusted software supply chains. Due to the increasing trend of supply chain attacks the ENISA report calls for "novel protective measures" to be developed by the security community and policymakers (ENISA, p.4). This thesis tries to provide a way for supply chains to implement some of the recommendations of the ENISA report. These recommendations include specifying contractual obligations for suppliers to share relevant cybersecurity information and submit to routine security audits.

Half of the cyber attacks studied in the ENISA report were attributed to well-known, well-resourced, state-sponsored APT actors. These APT groups are mainly based in Russia, China, North Korea and Iran (CISA, 2022). Supply chains are an attractive target for APT groups whose primary goals are not the supply chain itself but rather the customers of the supply chain. APT groups view the supply chain as a potentially exploitable avenue

or attack vector to gain unauthorised access to their final target, the customers' computer systems. An important part of cybersecurity is understanding the adversary and the kinds of threats they pose. State-sponsored APT groups exploit cybersecurity weaknesses to conduct cold-war espionage and to steal intellectual property in technology areas where their own progress is inferior. In so doing they destabilise the economies of democratic nations and undermine the prosperity and progress of the free world. State-sponsored APT groups view their hostile actions as being below the threshold of an act of war thus entailing a low risk of military reprisal by the free world. In most of the attacks on the supply chains, customers were oblivious to the reasons and details of how their suppliers were compromised. This blind spot shines the light on what we consider to be a key shortcoming in the cybersecurity of supply chains, namely, a lack of visibility into suppliers' cybersecurity posture. Procuring organizations do not have access to potential indicators of compromise and forensic data of the supply chain upon which they depend. This constitutes a weakness in the cybersecurity of the supply chain and undermines the stability and resilience of the supplier-customer business relationship.

Sweden's National Cyber Security Strategy (Government of Sweden, 2017) calls for enhanced collaboration and information sharing between organisations and stakeholders who have a common interest in national cybersecurity. Supply chains can have an effect on national security.

2.5 Perspectives from the Literature

2.5.1 Historical Perspectives

A 1970 report by the DSB Task Force on Computer Security, known as the Ware Report after its main editor, is regarded as a pioneering work in introducing the concept of computer security (Misa, 2016). It introduced the notion of computer security as a specialised discipline in computer science or computer engineering and highlights the multi-dimensional nature of computer system security. The Ware Report concluded that "Providing satisfactory security controls in a computer system is...a system design problem. A combination of hardware, software, communications, physical, personnel and administrative-procedural safeguards is required for comprehensive security" (DSB Task Force on Computer Security, 1970, p.vi).

More than two decades ago, Warren and Hutchinson recognized the increasing dependence of supply chains on "electronic systems" (Warren and Hutchinson, p.710). They succinctly described a growing and systemic characteristic of supply chains, namely, the "new dependence upon

computers and the data they contain" (Warren and Hutchinson, p.710). They wisely advised that companies should expect to make additional investments towards cybersecurity.

2.5.2 Supply Chain Cybersecurity

The term "digital assets" has been used to describe things in the supply chain that should be protected (Melnyk et al., p.168). Cyber threat actors seek illegal access to digital assets in order to gain economic and political advantage. Cyber criminals motivated by financial gain see cyber crime as low risk with potentially high rewards. The authors mentioned that academic research on supply chain cybersecurity is relatively limited compared to general cybersecurity literature due to the "novelty of the subject" with most studies being conceptual and qualitative in nature (Melnyk et al., p.164). The authors' efforts to integrate findings from supply chain and cybersecurity literature helps articulate the context of this thesis in the "relatively new development" of supply chain cybersecurity research (Melnyk et al., p.178). Notably, the authors point out that researchers, still trying to make sense of this fusion of complex knowledge domains, lag behind the cybersecurity practitioners dealing with the challenges on the front lines. Melnyk et al. found that the practitioner literature's overall findings were that the supply chain is the weakest link in an organisation's cybersecurity. They point out that SME suppliers, which play a vital and important role in most supply chains, pose a special challenge because they usually lack the resources and capital to make effective investments in cybersecurity. This makes them a risk to the supply chain. Melnyk et al. observed that no adequate solutions have been found by either the academic or practitioner communities and that the urgency of the need requires contributions from both communities. The authors likened the current state of research into supply chain cybersecurity as being analogous to Saxe's parable of the six blind men and the elephant (Saxe, 1873, p.135). While there are a number of different perspectives on supply chain cybersecurity, a comprehensive and holistic understanding remains elusive. In the context of multi-tier supply chains, the authors noted that it remains an ongoing challenge to achieve adequate visibility into the participants which is critical to effective supply chain cybersecurity. This critical issue of supply chain visibility is a primary factor that this thesis attempts to address. Melnyk et al. emphasized that a supply chain is only as strong as the weakest member. This view is a motivation for cybersecurity solutions and approaches that provide visibility and transparency across the supply chain. A systemic weakness of SME's as suppliers is that their limited resources impede their ability to achieve adequate cybersecurity and this affects the risk exposure of the whole supply chain. This is a fundamental problem of modern, digitalised, interconnected supply chains. Procuring organisations at the

head of the supply chain must consider how the other participant suppliers affect their risk exposure. According to Melnyk et al. finding the right balance of coordinated investments across the supply chain to increase its cybersecurity is still not adequately understood because this is a relatively new research area.

Windelberg (2016) describes the concept of supply chain resilience as the ability or capacity of a supply chain to recover from a disturbance, disruption or interruption of its activities and processes. This encompasses protection measures and responses to threats that are "unknown or highly uncertain" (Aven, 2011, as cited in Windelberg, 2016). This provides further context for this thesis in that protective cybersecurity controls and incident response (during and after an inevitable cyber attack) are both important components of building supply chain resilience in the face of a constantly evolving threat landscape.

Ghadge et al. (2020) found that cybersecurity considerations and responses have lagged behind the rapid digitalization of supply chains noting that "Extant literature has failed to address the implications of cyber threats at the level of supply chains" (Ghadge et al., p.224). The authors noted that a possibility exists for cyber attacks to discourage collaboration between supply chain partners. They emphasized that it is important for participants to embrace the collaborative effort as a shared security objective worthy of collective pursuit because it helps manage the cyber risks of the supply chain as a whole. Cybersecurity risk management is about protecting the stakeholders as much as it is about protecting digital assets such as data. Unfortunately, due to lack of awareness, in practice there has been a gap between standard business practices and the need for continuous attention to cybersecurity. The authors describe the concept of risk propagation whereby a risk in one part of the supply chain propagates to affect other participants by virtue of the interconnectedness between partners. They reiterate the realization that perfect cybersecurity is unattainable. Instead, a pragmatic approach is to have a diverse set of security controls and methods. This creates what we term an onion-like layered security approach which is effective and implementable. They also found that information sharing and collaboration are key elements for supply chain security, features that are highlighted in this thesis. The authors identified the need for "highly context-specific studies" (Ghadge et al., p.236). This finding served as motivation in this thesis to create a working prototype implementation.

2.5.3 Blockchain Use in Supply Chains

Agrawal et al. (2021) discuss blockchain enhanced traceability in a textile supply chain. While this use-case is not the focus of this thesis, it confirms

the general understanding that information shared via a permissioned blockchain can enhance visibility and reduce risks in the supply chain.

Chang & Chen (2020) outline the generally perceived advantages of blockchain such as traceability, transparency and immutability. They express the apparent contradiction that blockchain may "provide a better foundation of trust" via a "trustless operating environment without traditional trusted authorities" (Chang & Chen, p.62479-62478). As they explain, this concept can be a hard sell to supply chains which have been used to traditional trust mechanisms based on centralised trust authorities. The authors found that private (permissioned) blockchain may be more suited to a supply chain because of its greater degree of centralization as an entity with a defined boundary and constituency.

Pournader et al. (2020) convey the realization that among the many challenges faced by global supply chains, cybersecurity is one of the biggest. They explain that blockchain's relevance to supply chain management is due to its potential to enhance trust, transparency and traceability of inventory/goods. They cite IBM's blockchain as an example application for a specific aspect of cybersecurity, namely, identity management. The findings support the view that sharing critical information in a secure and trustworthy way is vital for a resilient and efficient supply chain. Investigating blockchain's information sharing feature as a way to contribute to supply chain resilience is a focus of this thesis.

Etemadi, Van Gelder et al. (2021) discuss barriers to blockchain adoption in supply chain cyber risk management and note that this particular use-case of blockchain is in its infancy. Barriers mentioned include immaturity of technology, lack of trust and suitability. They also find that a significant challenge is the reliance and trust needed by a blockchain system on external data sources called oracles. The authors describe blockchain as a decentralized database providing data transparency and security. Our view is that, in some respects, blockchain is no different to a distributed database of any kind which uses digital signatures and other cryptographic methods to achieve integrity of data and authentication of users. The main difference is that blockchain concatenates all the records into a list and maintains an ongoing measure or indicator of its overall integrity. Etemadi et al describe the generally perceived advantages of blockchain such as transparency and auditability. The auditability aspect is a feature that is highlighted in this thesis because of its potential value to supply chain cybersecurity. They remark that cyber-attacks have exposed the "fragility of the supply chain" (Etemadi, Van Gelder et al., p.3).

Organizations may view blockchain innovation as a low priority because a clear path to value is not readily apparent (Choo et al., 2020).

Cole et al. (2019) articulate the importance of understanding the characteristics of a supply chain to see if blockchain adoption can substantially add value or enhancement. The authors found that there is not a clear link between trust and potential blockchain application in supply chains. They describe a blockchain system making use of special nodes which are assigned a higher level of trust to act as monitors of user behaviour (Cole et al. citing Zhu et al. 2019). They note that this detracts from the perceived advantage of the trustless nature of blockchain. There seems to be a widespread perception that blockchain does away with the need for trust. While this may be true for cryptocurrency networks, we question if this assumption holds in a supply chain context designed on the basis of trusted contractual relationships between suppliers and procuring organizations. The authors also note this apparent dichotomy. Bayramova et al. (2021) caution against a headlong rush into blockchain adoption because technology project failures usually occur because of a mismatch between business requirements and the proposed technology. Cole et al. also bemoan the lack of empirical evidence to support many claims about blockchain. They identify trust and governance as a research gap.

Etemadi, Borbon-Galvez et al. (2021) note that information asymmetry may weaken the supply chain due to only one member having access to vital information. Their perspective is that trust between supply chain partners can be strengthened by sharing of important information. Our thesis was guided by this perspective to address information inconsistency by leveraging blockchain's transparency as a medium of shared information. Current literature tends to associate cybersecurity together with IoT use-cases reflecting the great interest in solving IoT issues (Etemadi, Borbon-Galvez et al., 2021, Figure 17, p.15). In our thesis we take a different perspective and investigate an application of blockchain which is complementary to established cybersecurity practices in an organizational context. The authors concluded that most blockchain proposals are at the academic stage and that more "applicative" tools are needed (Etemadi, Borbon-Galvez et al., p.20).

van Hoek (2019) highlights the need for governance in the application of blockchain to supply chains. The author explored previous lessons learned from RFID pioneers because of similarities in the challenges faced and the current limited empirical research on blockchain in the supply chain context. The author's findings support the view that blockchain can find application in augmenting existing supply chain systems, not necessarily replacing

them. In our view, the governance aspect demands significant planning and design effort for sustainability of a blockchain solution in a supply chain.

Hellani et al. (2021) describe the various attributes which can facilitate effective collaboration among supply chain partners. These include transparency, traceability and trust. All the existing DLT based solutions that the authors enumerate and describe are related to enhancing traceability of products manufactured by the supply chain by tracking their journey through the supply chain. While this is not the focus of our thesis, the authors raise important considerations worth mentioning here. They assert that blockchain guarantees trust among untrusted parties by virtue of, amongst other properties, its transparency as a shared store of data and therefore it is not necessary to evaluate participants' trustworthiness in a decentralized supply chain network. However, we take the view that while this is generally the case in the context of cryptocurrency applications, it is perhaps more nuanced in supply chain applications. The authors regard the smart contract as "a trusted tool" (Hellani et al., p.18). As we note further below in our review of the work by Al-Jaroodi and Mohamed, trusting software implies acceptance of certain risks because software is authored by humans who do not necessarily have a stake in the supply chain using the blockchain. Hellani et al. (2021) raise the issue of privacy by noting that sometimes it may be necessary to introduce a degree of opacity due to privacy issues. This thesis assumes that participants in the permissioned supply chain blockchain are trusted based on pre-established business contracts which specify terms and responsibilities of data sharing. In our suggestions for further work we describe how a degree of opacity can be added to an implementation.

Jabbar et al. (2020, p.788) found that opinions on blockchain are divided. In their analysis they found some arguments against blockchain including its slowness compared to traditional databases. This is one reason why this thesis does not focus on a real-time application. The authors note that the decentralized nature of blockchain is at variance with the notion and objectives of governance which is usually centrally administered. Privacy issues such as the right-to-be-forgotten can also be at variance with blockchain's immutability property. They also argue that correcting data errors on an immutable blockchain is not possible. New data could be appended that overrides erroneous data but the old data would still remain on the chain and could pose privacy issues. They define the important supply chain attribute of traceability as "knowledge of the history and location of an entity" (Jabbar et al., 2020, p.788). Traceability of supply chain assets can be achieved by recording all the transactions pertaining to an asset on a ledger such as a blockchain. Such a historical record can serve to document the integrity of the final product of a supply chain provided the

integrity of the ledger and its data can be guaranteed. In our thesis we adapt this concept to encompass the history of any data which may change over time and that has relevance to a supply chain's cybersecurity. However, a key distinction between our thesis and the reviewed research is that we do not focus on data which tracks progress of products through the supply chain. The authors pointed out that quality of data cannot be guaranteed by the immutability feature of blockchain alone implying that there is still a reliance on other business processes which determine data quality. These kinds of questions led us to moderate our expectations of blockchain by adopting a more cautious, measured approach. Our thesis is therefore positioned more as an effort to augment existing cybersecurity measures in a supply chain and not to replace them.

Dutta et al. (2020) identified enhancement of supply chain performance as a potential benefit for Internet of Things (IoT) systems. Although we do not focus on IoT devices, the notion of enhancement, or incremental improvement, of cybersecurity is an attractive proposition that, if successful, could meet real and urgent needs in a supply chain context. According to the authors, "Blockchain is much more secure than...traditional security services" (Dutta et al., p.8). Business process management was also identified as a more general area where blockchain can provide enhancements and our thesis touches on an aspect of this relating to governance or oversight. The authors' perspective is that inadequate identity management and cybersecurity in the technology sector can be resolved by integrating platforms with blockchain. We take a different perspective with a view that existing solutions could work well if properly designed and correctly implemented. Deficiencies in cybersecurity are often due to incorrect design and engineering decisions, and implementation mistakes.

Lesavre et al. (2020) describe the application of blockchain in identity management systems as a new field where its perceived advantages are often unclear. While our research does not focus on the use of blockchain for identity management, the authors note some important aspects of such applications which are relevant to cybersecurity. Traditional identity management systems are characterised by users' credentials being stored by the businesses they interact with or by third party organizations to which this task has been delegated. A perceived disadvantage is that users are not the custodians of their own identifying credentials. The authors state that blockchain can allow users to control the custody of their own credentials. In our mind this serves to restate the fact that credentials bound to an identity are the crucial foundation for any secure computer-based ecosystem. For businesses, the administrative burden of being a custodian of the customers' identity credentials constitutes a business process and cost burden that blockchain can potentially help to alleviate. Lesavre et al.

explain that while blockchain can be used to share information about cryptographic keys and their mapping to identities, some functions can be separated from the blockchain. They give examples like user-controlled wallets or third-party identity hubs which are off-chain datastores for storing secrets. In all cases, mechanisms and processes will have to be established for recovery of user secrets in case of loss if this is deemed necessary for a particular use-case. These findings lead us to question whether blockchain can be a disruptive replacement for many well-established business practices. Their discussion of governance models motivates for more research into creative methods of supply chain governance, especially if such governance can be practically automated in a way that adds value. They draw attention to the importance of an appropriate governance structure for blockchain-based systems for managing critical user identity tokens or credentials. They note that governance is needed to foster trust in the blockchain system and that centralized governance by the system owner can enhance control over the scheme. This seems to fit the supply chain scenario. Since blockchains are basically a collaborative record of participants' interactions and their common shared data, it is paramount that the digital representations of participants' identities are always secure, trusted and verifiable. In our view this objective is readily achievable in a permissioned blockchain established by agreement between a set of supply chain participants where governance is agreed upon.

Mylrea and Gourisetti (2018) proposed a conceptual framework for the use of permissioned blockchain to enhance cybersecurity of field devices used by electric utilities for operation of an electricity grid. These devices are comprised of a mix of IT and OT devices. Their interconnection with ICS systems via IT networks and the internet creates a threat surface for cyber attacks. The NERC issues requirements for cybersecurity of such devices and specifies requirements such as vulnerability assessments and inventory management. A key motivation of the authors' work was to assist utilities in meeting NERC compliance requirements. In the context of the authors' paper the term supply chain encompasses the set of all such devices and the vendors and processes for procuring and maintaining them (e.g. software updates). The scope of the authors' proposal is a real-time system appropriate for the OT use-case. This differs from our proposed use-case which is not intended for real-time applications nor for OT uses. Nevertheless, the authors highlight key features of blockchain which we also try to leverage. The authors identify the use of blockchain for monitoring state of devices and critical assets. They note that by recording this information on a blockchain it can enhance collaboration among participants of the supply chain by virtue of the blockchain's transparency. We adopt a similar perspective but focus our efforts on monitoring state of computer systems within supplier organisations with the objective of

enhancing supply chain cybersecurity via visibility of the shared blockchain data. Auditability is another benefit of the blockchain record. The authors' proposal does not include test results from an implementation. Their proposed patch management and software development use-case is similar to git, the well-known collaborative software development environment. Many developers collaborate on development and maintenance of a codebase with all updates cryptographically logged and recorded thus giving an auditable history of the software life-cycle. Mylrea and Gourisetti mention that it may be necessary to rely on traditional databases to solve interoperability problems with their approach. This highlights the complexity and difficulty of adapting blockchain technology to different use cases especially in an OT environment with outdated (but still operational) SCADA equipment, a common occurrence in long-lifecycle OT industries.

Kshetri (2017) evaluated blockchain's role in strengthening cybersecurity. The author noted the general view that a blockchain is secure by design and that attacks are usually directed against systems that hold private keys. This observation supports our view that the security of the implementation depends not only on the blockchain itself but also on the security of ancillary systems used to hold secrets used for blockchain transacting. Kshetri points out that there is insufficient empirical research to support assertions that blockchain is superior to other cybersecurity methods and approaches. Roles identified for blockchain in cybersecurity include its use as a controlled datastore for protecting sensitive healthcare records and for enhancing management of IoT devices by secure patch distribution and access controls. Supply chain related roles include tracking of products as they move through a supply chain and tracing and verifying their provenance. Kshetri notes that blockchain's ability to enforce legally binding contracts is an unresolved issue. In our view this hinges partly on the use of digital signatures which depend on cryptographic tokens bound to an identity. These identity bindings must be created, stored and maintained. If this is done by a central authority like a certificate authority then the decentralised advantage of blockchain is reduced by this dependence on other centralised systems. The identity bindings must be effected somewhere in the blockchain ecosystem and by some responsible party who maintains a record of each and every identity binding. Furthermore, the processes for verifying assertions of identity and storing the identity bindings need to be sound from a legal perspective if they are to form the basis of legally enforceable smart contracts. Kshetri describes an example of this where a bank performs the role of identity verification and binding which then enables subsequent use of associated identity tokens in a blockchain ecosystem. This shows that comparing blockchain approaches solely on the basis of decentralised versus centralised may not always be appropriate.

Taylor et al. (2020) conducted a systematic literature review of blockchain applications for cybersecurity and framed one of their research questions as, "How is blockchain used to improve cyber security?" (Taylor et al., 2020, p.148). The authors found that blockchain offers no "silver bullet" for cybersecurity but that it has potential to augment existing security approaches (Taylor et al., p.153). This confirms our view that blockchain should prove its utility as a value-add to existing cybersecurity controls before being presented as a replacement of existing systems and practices. Taylor et al. describe state-of-the-art applications of blockchain which demonstrate its utility for various cybersecurity purposes. Most of these relate to IoT ecosystems for authentication and firmware deployment (a type of supply chain process). Other uses include creating data storage redundancy leveraging blockchain's decentralised nature. They contrast existing security approaches that rely on a single trusted authority versus blockchain's perceived trustless nature because the ledger is distributed and all participants hold a copy. In their conclusions they allude to an associated aspect that we feel can moderate and contextualise the trust versus trustless debate. They note that the relatively recent global adoption of secure World Wide Web technology (in the form of secure hypertext protocol connections between clients and servers), has created a growing need to manage the associated cryptographic identity certification schemes that underpin the technology. Hence the outcome, a decentralised internet ecosystem requiring centralised oversight for practical security.

A survey paper by Berdik et al. (2021) consolidates numerous expressions of excitement concerning blockchain's perceived potential benefits. Expressions range from "ensuring that data is accurate...an incredibly easy task to implement through blockchain" (Berdik et al., p.3) and "potential to take over nearly any domain" (p.1), to "replacing the modern database" (p.6). These are balanced by more realistic reflections regarding the need to tailor its application to particular use cases by, inter alia, fully understanding the business requirements, processes and the consequences of process modification. In discussing the use of smart contracts to ensure transaction integrity, the authors note the reliance "on the blockchain to verify that the signatures on the contract are legitimate" (Berdik et al., p.15). In our view the smart contract cannot vouch for its own integrity.

Al-Jaroodi and Mohamed (2019) emphasize the importance of trust in the integrity of blockchain for industrial applications by its industry and business participants. Parties agree to trust the blockchain ledger as the common record of their interactions. In our thesis we assume such a prior agreement which specifies the data to be shared on the blockchain along with access rights and rules to the sources of the data. They expand on the

issue of trust by noting that blockchain relies on verified digital identities and they suggest that these identities be verified by a governmental organization. This suggestion is juxtaposed against their introduction of blockchain which mentions one of it's advantages as enabling parties to reach agreement without the need for a regulatory authority. The authors allude to potential applications of blockchain in governance but do not offer specifics. Governance is an important focus in our thesis. Al-Jaroodi and Mohamed (2019) point out that the legal enforceability and correctness of an agreed upon smart contract depends on the underlying code and therefore on the code writers or developers. They contrast the technical binding created by code logic versus legal binding created by laws. The code developers thus become a hidden dependency or participant of the contract.

Patil et al. (2020) state that "Among all technologies available for data security and privacy, blockchain is the most efficient one" (Patil et al., p.1815) and that "blockchain will offer many superior results" for the supply chain (Patil et al., p.1816). These perspectives are based on the authors' focus on the IoT use case where the term "supply chain" often refers to the supply of data to the devices (e.g. software updates). More importantly, they draw attention to some key issues that need to be fully appreciated when considering blockchain applications for cybersecurity. The authors identify the primal need to keep private keys safe. In our mind, this is arguably the weakest link and thus blockchain is subject to the same dependence as other systems on the security of secrets. Hwang, D. et al., 2018 (as cited in Patil et al.) describes the concept of a "management hub", responsible for managing downstream IoT devices, which checks and verifies access control permissions stored in a blockchain. This hybrid approach is appealing because verification and authentication of access is so important that it may warrant delegating some trust to special nodes or servers. Cybersecurity is difficult and challenging and such hybrid approaches are promising strategies for engineering practical solutions.

In discussing avenues of future blockchain research, Ault (2018, p.16) proposes an interesting concept of blockchain "witness" nodes. These are envisaged to be trusted nodes with a vested interest in the blockchain network and respected among the community of participants. Witness nodes receive funding for hosting trusted, secure servers. The idea is to reduce the mining and computational burden on the network by offloading some of the validation work to such trusted witness nodes. Such hybrid approaches are practically appealing for businesses because incremental security improvements are less risky. Ault mentions a successful pilot project in a Brooklyn electricity microgrid where residents within a neighborhood can sell excess photovoltaic energy to each other via blockchain transactions. The blockchain is used to facilitate trading among partners who have agreed

to trade a commodity (energy) that they each produce from their own solar panels. If a resident is not consuming energy being produced by the resident's solar panel, then this excess energy can either be stored in a battery for later use, or a decision can be made to allow this excess energy to flow to another resident who is able to use it and willing to buy it in a real-time transaction. The technical aspects of controlling real-time electric energy flows require the expertise of an electric utility to ensure that these are done safely and without adverse effect on the stability of the greater connected electric grid. This particular combination of electrical engineering and blockchain is almost as exciting as the convergence of blockchain and cybersecurity.

3. Method

The scientific method used in this thesis is an experiment. The type of experiment performed is a simulation. To explore blockchain's potential use for cybersecurity of a supply chain, we designed and created a simulation to evaluate the ideas proposed in this thesis and to answer the research questions. Intuitively, it seems that the sharing of security-relevant data among partners may be beneficial for their common security objectives. This is compatible with the inherent transparency and shared nature of blockchain data. A carefully selected set of cybersecurity information may be useful if captured onto a shared blockchain. The simulation is used to test this idea by capturing a reduced set of available data with a narrow focus tailored to be useful to a cybersecurity oversight role or a forensic process such as incident response. Testing this idea in a real supply chain was not realistic within the constraints of this thesis project. This would have required the agreement and participation of numerous parties. Instead, we chose to create an experiment where we simulate a supply chain as a set of interconnected systems. The simulation is an experimental environment where these ideas can be explored. The simulation is described further below.

We define various types and sources of digital data (assets) relevant to cybersecurity which we record onto a common permissioned blockchain. Our focus is on data from the computer systems used within each participant organization in the performance of their business operations. We simulate these systems and record specified data from these systems onto the common permissioned blockchain. We design a data model which specifies the data to be captured. This specification is based on the data's relevance to the cybersecurity of the systems used within the supply chain member organizations. The selection of pertinent cybersecurity data is partly based on industry practices and cybersecurity domain knowledge of threats, and partly on the limitations of a simulation. We monitor and record the status of identified digital assets, such as important digital files, at specified regular intervals. We record either the entire data or a synopsis depending on the size of the data and how much of it can be useful on the blockchain. The identification of relevant cybersecurity data sources and digital assets results in a data model which forms the basis of the data recorded onto the blockchain. To simulate changes over time to these monitored assets we automate a process that perturbs the state of these assets during the simulation. We also record some dynamic system-generated data which contains an element of randomness. Some perturbations are performed manually to generate more data for analysis.

For the simulation, a data model is defined consisting of the following data types:

- Directory: a directory containing files. A hash is computed over all files in the directory.
- FileHashOnly: a file for which we record just the hash.
- FileWhole: a file for which we record the hash, and the content when the hash has changed.
- FileWholeRemote: a file located on a cloud-hosted system which we access remotely.
- Command: various pre-defined commands that are executed by agent machines and the outputs recorded.
- BlockChain: a data structure encapsulating all properties that define a blockchain.
- Block: a data structure that holds block headers and data.
- BlockData: a data structure that hold the data portion of a block.

These asset types are enumerated in detail in Appendix A. For the simulation assets were chosen whose compromise will manifest as changes of state (indicators of compromise) on the blockchain and which can be detected by analysis of the blockchain. These include suspicious artifacts such as modified files, suspicious network connections, altered system configuration files, malicious changes to administrative software programs, and insertion of malicious files into important directories to undermine system security.

We created a blockchain application using Python which is the core component of the simulation. This is a monolithic programme containing functionality for use both by the blockchain administrator and by the agent machines which represent suppliers and create new blocks. We show how a selected set of cybersecurity data is recorded onto the blockchain and how changes to the status of digital assets specified in the data model propagate onto the blockchain. We analyze the resulting blockchain from the perspective of an oversight role, auditor, or higher-level security analyst to extract useful insights. The blockchain records and presents a view of each supplier’s enterprise computer system represented by a virtual machine (or agent) in the simulation.

The simulation consists of six virtual machines (agents) which together represent a supply chain. These are shown in Figure 2 below.

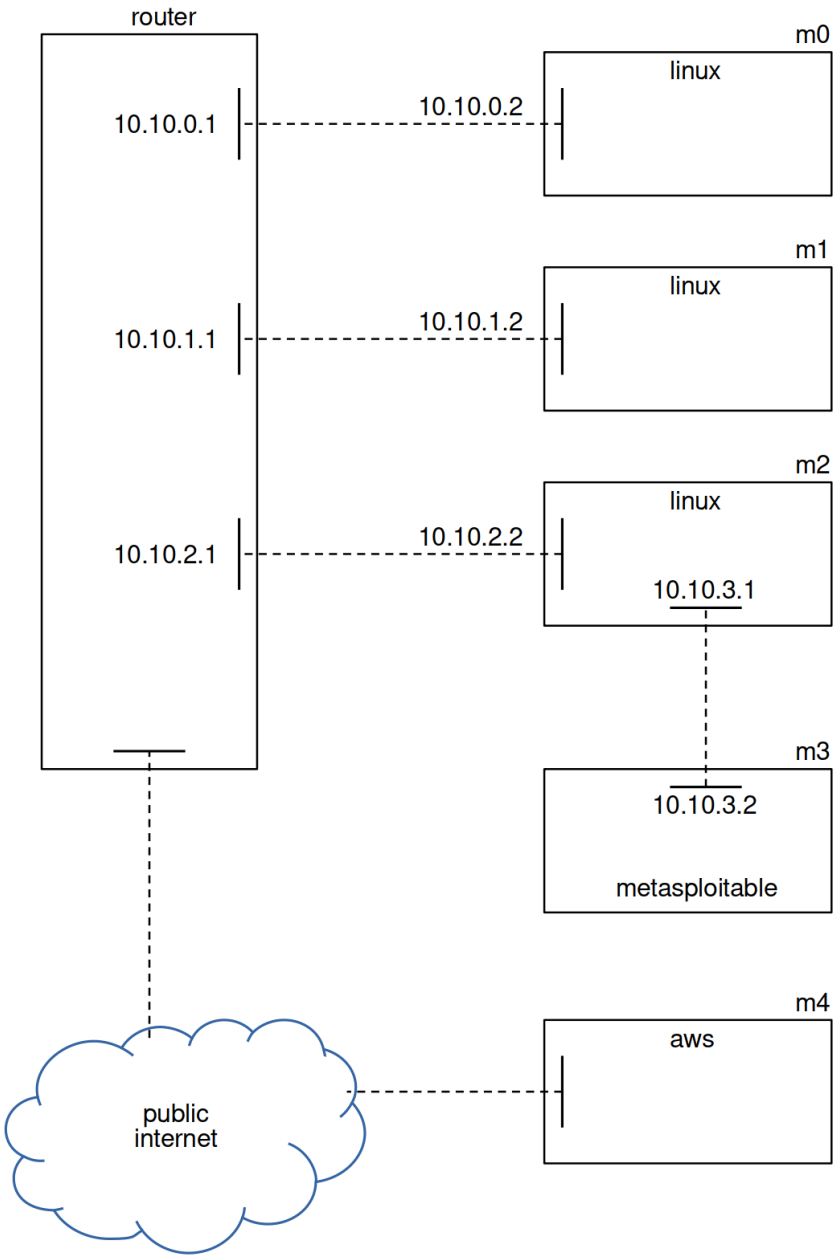

Figure 2: Simulation network schematic depicting the networking arrangement between virtual machines. Machine m0 is the administrative machine which has access to the three agent machines, m1, m2 and m4. Machine m3 simulates a standard office workstation computer and is connected only to machine m2.

In Figure 2 above, Machine m0, represents the primary organization at the head of the supply chain. Agent machines m1, m2, m4 represent downstream organizations which are suppliers to the dependent primary organisation. Each agent machine represents an actual physical computer placed in a supplier's premises and which performs the sole function of monitoring specified digital assets in that organisation and creating new blocks. The blocks created by the agent machines contain the results of their

monitoring actions. Machine m3, represents any non-agent machine (e.g. ordinary office workstation) downstream of agent machine m2 and monitored by it. Machines m2 and m3 are in the same supplier organisation. Machine m3 is configured with the metasploitable framework and has open ports which can be probed from machine m2. Machine m4 is located in the Amazon AWS cloud and access to this machine passes through the real internet. Machines m1, m2 and m4 are called agent machines because they act as agents of the primary organization performing the monitoring functions specified by the primary organization. The three agent machines create new blocks recording the status of specified assets. Each agent is granted network read access to the specified assets of the organisation in which it is located. This access is contractually agreed upon by the suppliers participating in the blockchain system. Each agent machine creates a new block according to a pre-defined, controlled round robin schedule (Yaga et al., 2018). The primary machine or blockchain administrator m0, is responsible for synchronizing the blockchain among the agents so that each agent has the latest version of the blockchain before writing a new block. A separate simulation script is used to automate perturbations that simulate changes to the status of assets. The perturbations have been designed to mimic the kinds of threat vectors that attackers may exploit and assets they might target. The goal of the simulation is to simulate digital assets that are considered valuable from a cybersecurity perspective and a business perspective. These may be termed the "crown jewels" of a business in the sense that their compromise could result in serious negative consequences for the overall supply chain.

The agent machines are connected via a secure network to the single administrative machine, m0, at the primary organisation's premises which has the oversight and administrative role in the blockchain scheme. Each agent machine has a host-based software firewall that does not allow network connections to the machine. Only outbound connections initiated by the agent machines are allowed. To simulate a real supply chain where geographically dispersed suppliers are connected over the untrusted internet, we implemented a virtual router to segregate the participants onto separate IP sub-networks. The IP subnets for the simulation were chosen arbitrarily.

In order to maximize confidentiality and protection of transmitted data between the machines it was decided to implement a VPN overlaid on top of the IP network. The secure VPN interconnects only the machines that participate in the blockchain scheme. No other machine has access to the VPN. To secure the VPN, an associated PKI was created as the basis for identity authentication and encryption in this secure network. The VPN secures the network access to the blockchain. Access is restricted to machines which are trusted, and the trust model is implemented by the PKI.

The blockchain is administered by the primary top-level participant in the supply chain (the dependent organization). The primary participant in the simulation performs the distribution and synchronization of the blockchain among all machines via the secure VPN. The VPN was implemented using the open source OpenVPN software and the PKI was created using OpenSSL. Both OpenVPN and OpenSSL are included in standard Linux distributions. Figure 3 below shows the VPN schematic.

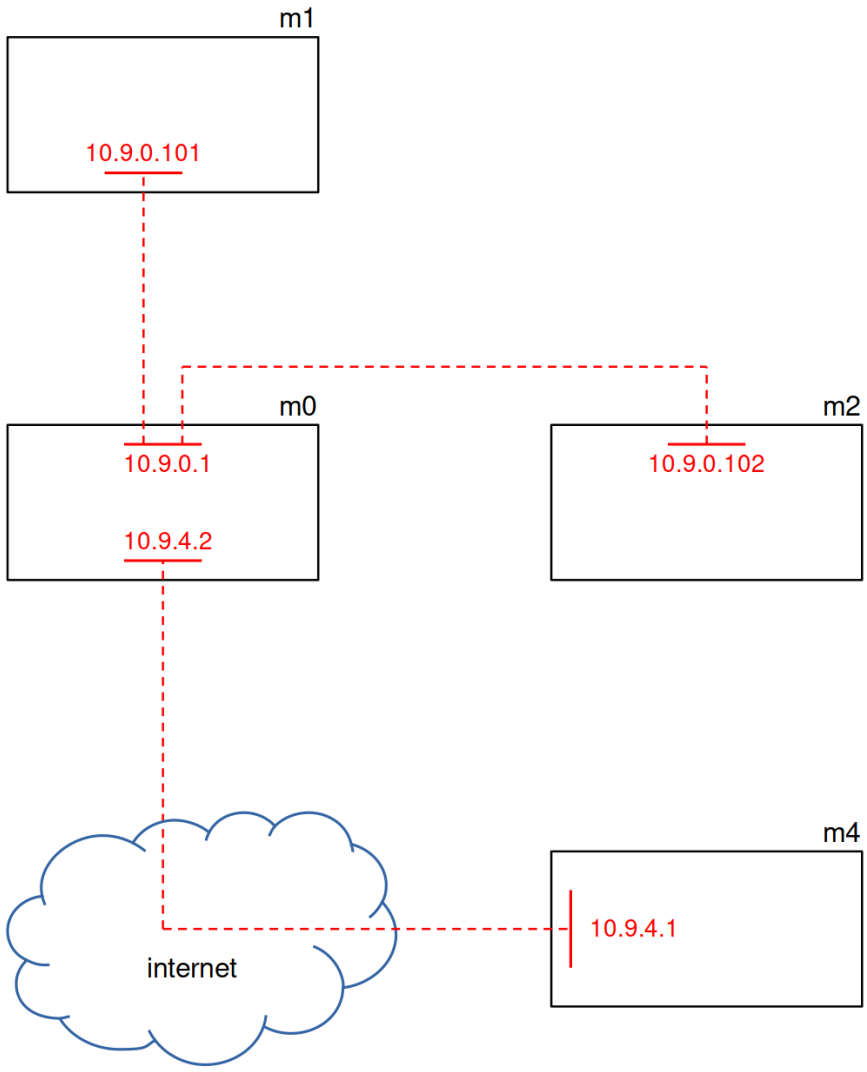

Figure 3: Network schematic depicting the networking arrangement of the secure VPN. Machine m0 has secure connectivity to all three agent machines m1, m2 and m4. Access to machine m4 passes through the real internet.

In Figure 3 above it can be seen that the IP subnets for the VPN are different to the actual IP subnets indicated in Figure 2. This is because each machine

has an additional virtual network adapter acting as its interface to the secure VPN. The encrypted VPN traffic is transmitted over the normal IP network. This is an example of a virtual network overlaid on a real network. Agent machine m4 performs a dual role in the simulation. It acts both as an agent machine monitoring a dynamic asset, and as a remote workstation computer with a file asset remotely monitored by another agent, m1. It is implemented as a remote agent connected over the real internet. This simulates a typical real-world scenario where the secure VPN between the administrative machine and the agents is established over the untrusted internet. Machine m4 is allowed to be subjected to internet based attacks by having an SSH server publicly listening on port 22. This lends some realism to the simulation. In its role as a standard workstation computer, m4 represents a geographically remote machine located anywhere within a supplier's enterprise infrastructure. Assets on that machine are monitored remotely by an agent (m1) located at that supplier headquarters.

Blockchain synchronization is performed via the VPN using key-based SSH communication sessions within the VPN.

Digital signatures based on asymmetric cryptographic keys are used to verify the authorship of each blockchain record. The verification of signatures is performed by each agent and also by the administrative machine m0. The simulation programme does not permit a new block to be written unless the signature of the most recent block can be verified as well as the hash of the previous block. The blockchain simulation makes use of the ED25519 digital signature scheme via the PyNaCl Python binding to libsodium. Libsodium is a fork of the public domain, high-speed, Networking and Cryptography library (NaCl). Development of NaCl was funded by the European Commission's Seventh Framework Programme. This signature scheme uses state-of-the-art elliptic curve asymmetric keys of smaller size than RSA based keys with equivalent cryptographic strength (Brendel et al., 2021). The public keys are distributed to all agents while each agent retains sole possession of its own private key stored on the agent machine.

4. Results

Figure 4 below shows regular network activity between the administrative machine, m0, and the agent machines, m1, m2 and m4. The activity shown consists of UDP traffic used to keep the connections alive.

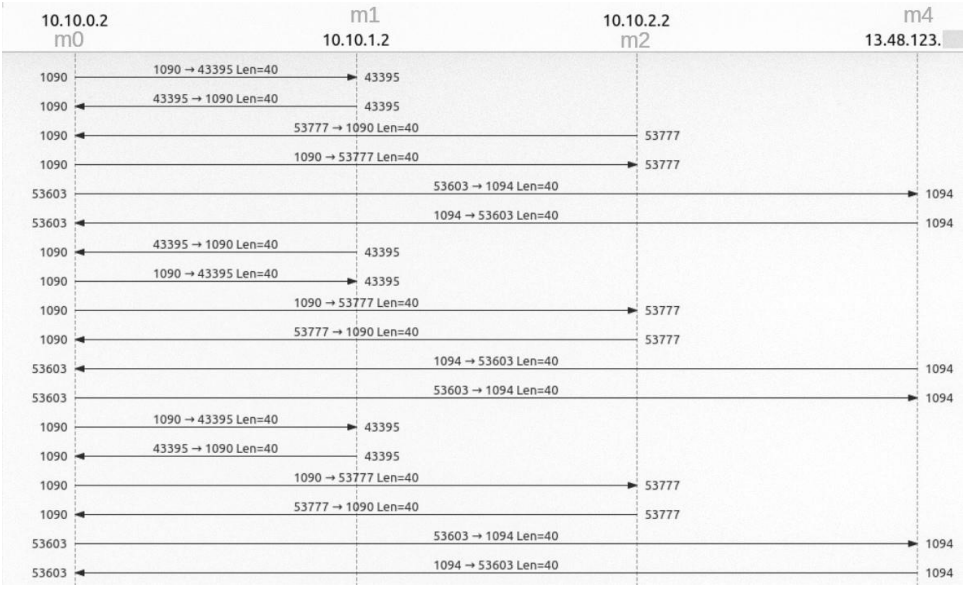

Figure 4: Network traffic between the administrative machine m0 and each agent machine. This depicts low volume background traffic used to maintain the active VPN sessions. Machine m4 is located in the public IP space.

In Figure 4 above, the network activity is represented by horizontal arrows indicating the direction of traffic flow from one machine to another. The IP addresses of the machines are indicated and correspond to the machines m0, m1, m2 and m4. A vertical dashed line represents each machine in this diagram. Thus there are four vertical dashed lines with the machine IP address shown at the top of each line. The vertical lines represent, from left to right, machines m0, m1, m2 and m4. The horizontal arrows link pairs of vertical lines thus representing bidirectional traffic between m0 and each agent machine. The diagram shows that the administrative machine, m0, is maintaining active VPN connections to each of the agent machines, m1, m2 and m4.

Figure 5 below shows the synchronization traffic when machine m0 synchronizes the blockchain across all machines at regular intervals according to a pre-defined schedule.

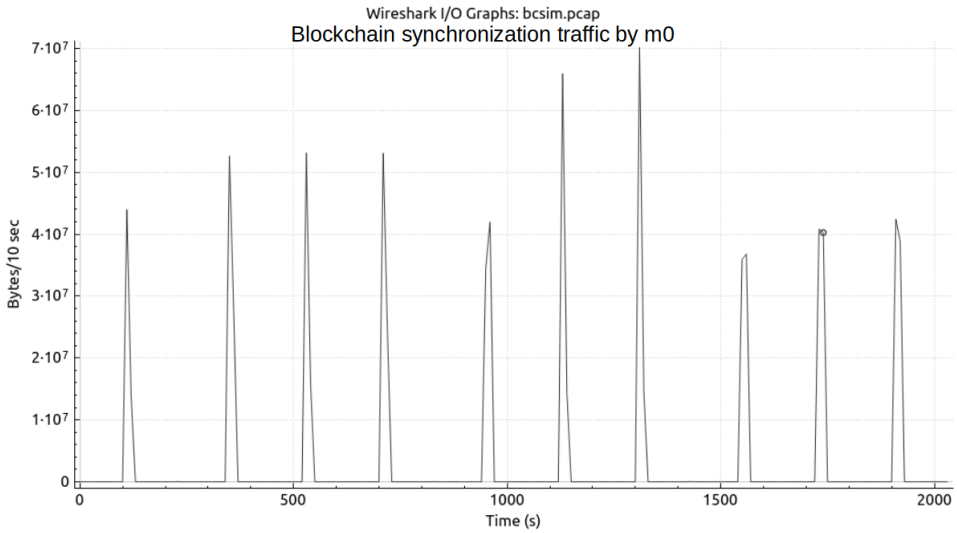

Figure 5: Blockchain synchronization traffic initiated by administrative machine m0 when it synchronizes the blockchain across participating machines. The graph shows data transfer rate versus elapsed time.

The graph in Figure 5 above shows the network traffic rate in bytes per 10-second interval. The vertical scale of this graph is such that the background traffic used to keep the VPN sessions active is insignificant. The purpose of this graph is to show the blockchain synchronization traffic which occurs each time machine m0 synchronizes the blockchain across all machines. The synchronization happens at regular intervals as specified by the simulation schedule and shows up as traffic spikes in the graph. The height of the spikes varies between machines because each machine writes a different volume of block data. This is due to the different asset specifications for each machine resulting in different volumes of data.

Figure 6 below shows an example UDP packet. All VPN traffic is encrypted (as can be seen in the packet bytes) and the VPN uses the UDP protocol.

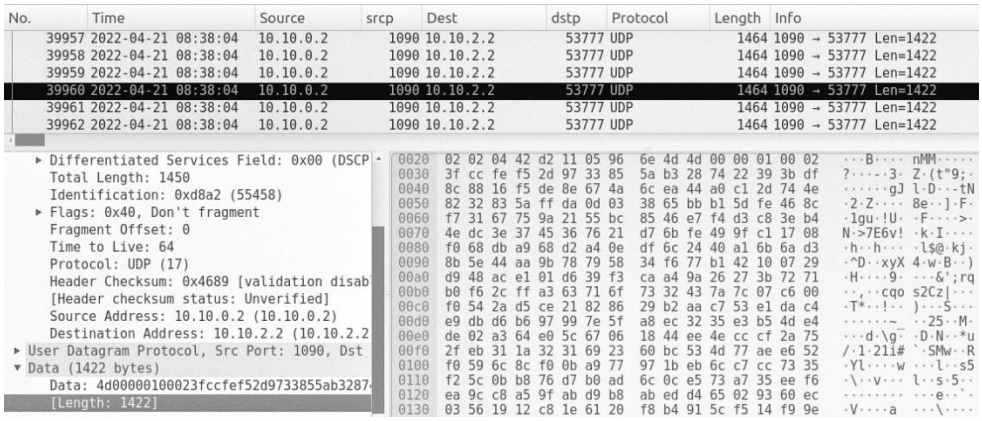

Figure 6: Encrypted VPN traffic encapsulated using UDP protocol. A single packet from machine m0 to machine m2 is selected to show the encrypted packet payload bytes confirming that the traffic is encrypted.

Figure 6 above is an extract from a packet capture which was used to verify the correct operation of the secure VPN network. One of the traffic packets is selected in order to view its payload. On the far right it can be seen that the payload bytes are unintelligible because the payload is encrypted so that the payload cannot be represented as human-readable text. On the left the source and destination IP address of the selected packet can be seen, in this case, being from machine m0 (10.10.0.2) to machine m2 (10.10.2.2).

In this section we present views into the resulting blockchain, the data it captured, and how we can interpret what we see to gather useful insights.

4.1 Summary of Blockchain

The following simulator programme command was performed,

```
./bcsim.py admin --printsummary
```

which prints a summary of the created blockchain. A condensed version of the output is shown below depicting salient details of the first two blocks and the last block.

```
------------------------------------------------
BlockChain(217 blocks, head = 216)
{'_blocks': [Block(0, m0), Block(1, m1), Block(2, m2), Block(3,
m4),
...
Block(214, m1), Block(215, m2), Block(216, m4)], '_blockcount':
217, '_totaldatasize': None, '_head': 216}
------------------------------------------------
Block(0, m0)
{'_blocknumber': 0, '_authorid': 'm0', '_previousblockhash': 0,
'_genesisblock': True, '_unixtime': 1651342691.3100054,
'_datetime': 'Sat Apr 30 20:18:11 2022', '_tz': 'CEST',
'_tzoffset': '+0200', '_nonce': 13420632891802009885, '_data':
BlockData(0 assets, 0 commands), '_comment': 'Genesis block created
by admin on Sat Apr 30 20:18:11 2022', '_sig':
b'Q7scTbGZWa3eH+NcdiiHmrJyCsoLrdXlJ4z8QJkcrHWoMeI+ta2zFo5IpSlM5R5Rt
0MEBJBhDSVImEVuQnAvAg==''}
------------------------------------------------
Block(1, m1)
{'_blocknumber': 1, '_authorid': 'm1', '_previousblockhash':
'c57b5c75a089d66c8069e03612f94119d1fdb2cb21a648b95ab60e4a',
'_genesisblock': False, '_unixtime': 1651342923.2609017,
'_datetime': 'Sat Apr 30 20:22:03 2022', '_tz': 'CEST',
'_tzoffset': '+0200', '_nonce': 14341385218504313226, '_data':
BlockData(6 assets, 3 commands), '_comment': 'Block created by
root@m1 on Sat Apr 30 20:22:03 2022', '_sig':
```

```
b'kYQT3BVlfoGdZEvCRa2YBGOzE/I7WLrQ31Bjm/lhkswXb/sEMFavVzZ5cbJ2PCmjI
dtzsk6OiMaNkMh32rNlBQ==''}
------------------------------------------------
...
------------------------------------------------
Block(216, m4)
{'_blocknumber': 216, '_authorid': 'm4', '_previousblockhash':
'cf6765aba0e6d74f4c0bf03010570d9f74b23a6b57c224c7343964ee',
'_genesisblock': False, '_unixtime': 1651385881.554614,
'_datetime': 'Sun May  1 06:18:01 2022', '_tz': 'UTC', '_tzoffset':
'+0000', '_nonce': 402166979440898786, '_data': BlockData(1 assets,
1 commands), '_comment': 'Block created by root@m4 on Sun May  1
06:18:01 2022', '_sig':
b'XlsmFXAk+5EFtaMQg/SgXC1C+AZW5hv2osExBNo15lTJMXfJEWEjpV5peYV8KY/6V
WLcZC34/WcD4AhOvOv5Dg==''}
------------------------------------------------
```

The above result shows that the blockchain consists of 217 blocks numbered from zero to 216. Block 0 is the genesis block i.e. the first block created. The term "head" is used as a name indicating the number of the last (most recent) block written. The blockchain attribute "totaldatasize" is currently not used and is intended for further development of the programme. Block 1 is the first block written by an agent machine, m1. The last block on the blockchain is block 216 written by machine, m4. Each block header has attributes that describe its author, the block number, the timestamp of when it was written. Each block header contains the hash of the previous block in the chain and a signature, by the block author, over all the block contents. The "nonce" attribute is currently not used and is populated with a random number generated at block creation. Use of this attribute is discussed later in this report in the context of possible further expansion of this programme. The block attribute "data" contains the data portion of a block. It records the various assets and commands being monitored by the simulation. For example, in block 1 the data attribute records the state of 6 file system based assets and 3 command type assets.

The following command was performed,

```
./bcsim.py admin --checkintegrity
```

which checks the integrity of the whole blockchain. The result below shows that block signatures and previousblockhash attribute values were verified. The verification process begins from the last block until the genesis block. Only the signature is verified for the genesis block. Integrity checking is performed automatically during creation of new blocks.

```
----------------------------------------
checking blockchain integrity...
Block(216, m4).previousblockhash verified.
```

```
Block(216, m4) signature verified
Block(215, m2).previousblockhash verified.
Block(215, m2) signature verified
...
Block(1, m1).previousblockhash verified.
Block(1, m1) signature verified
Block(0, m0) signature verified
----------------------------------------
```

4.2 Summaries of Specific Blocks

The following command was performed,

```
./bcsim.py --summarizeblock 1
```

to reveal more granular insight into block 1. Hashes and signatures are truncated for clarity throughout the rest of this section. The result below reveals the properties of block 1. The header contains the hash of block 0 in the attribute "previousblockhash". Also listed are the assets monitored by agent machine, m1. These include 6 filesystem based assets numbered 1000-1005, and 3 command type assets numbered 10000-10002. For filesystem type assets (directories and files) the data model class name is listed along with the assetid and the filesystem path to the asset. For command assets, the actual command parameters are listed.

```
------------------------------
summarizing Block(1, m1)
------------------------------

blocknumber:  1
authorid:  'm1'
previousblockhash:  'c57b5c75a089d66c8069e03612f94119d1fdb2...'
genesisblock:  False
unixtime:  1651342923.2609017
datetime:  'Sat Apr 30 20:22:03 2022'
tz:  'CEST'
tzoffset:  '+0200'
nonce:  14341385218504313226
data:  BlockData(6 assets, 3 commands)
comment:  'Block created by root@m1 on Sat Apr 30 20:22:03 2022'
sig:  b'kYQT3BVlfoGdZEvCRa2YBGOzE/I7WLrQ31Bjm...'

Assets
1000  Directory(1000, /usr/share/ca-certificates/mozilla/)
1001  FileHashOnly(1001, documents/reactorspec.odt)
1002  FileWhole(1002, /etc/cron.monthly/awsiplist)
1003  FileWhole(1003, documents/supplierdata)
1004  FileWholeRemote(1004, aws_hosts.temp)
1005  Directory(1005, files/)
```

```
Commands
10000  CmdLast(10000, ['last', '--ip'])
10001  CmdGeneric(10001, echo "$PATH")
10002  CmdNetstat(10002, ['netstat', '-tupan'])
------------------------------
```

4.3 Cybersecurity Analysis

Here we present results showing analysis of the blockchain from a cybersecurity perspective. For clarity, some of the output data in this report is truncated. For more details of data captured on the blockchain and further analysis see Appendix C: Block Data Details. For the syntax of all simulator commands, see Appendix B: Simulator Synopsis.

The following command was used,

```
./bcsim.py admin --hashchanges m1
```

to identify, for a specified supplier, all blocks where asset state has changed. Below is the output from this command for machine m1. The blocks authored by machine m1 are listed, the assets monitored, and for each asset, the blocks where the asset hash changed. It can be seen that assetid 1004 changed from block 34 to block 49.

```
----------------------
    Blocks by supplier m1:
[1, 4, 7, 10, 13, 16,...]
----------------------
    Assets by supplier m1:
('1000', 'Directory')
('1001', 'FileHashOnly')
('1002', 'FileWhole')
('1003', 'FileWhole')
('1004', 'FileWholeRemote')
('1005', 'Directory')
----------------------
    Supplier m1 asset hash changes:
_____________________________________________________
asset ('1000', 'Directory') changed in blocks:
    [1, 4, 31, 55, 82, 106, 133, 157, 184, 208]
...
asset ('1004', 'FileWholeRemote') changed in blocks:
    [1, 16, 34, 49, 67, 82, 100, 115, 133, 148, 166, 181, 199, 214]
...
```

The details of the change in asset 1004 from block 34 to block 49 was revealed using the Linux diff command,

```
diff <(./bcsim.py admin -fw 34 1004) <(./bcsim.py admin -fw 49 1004)
```

The result below shows that one line, "10.124.125.126 yahoo.com", was added to the file. The "+" sign at the start of this line is an indicator showing that this is a new line that was not present in the previous version of this file (in block 34).

```
----------------------
-Block(34, m1), fileasset: 1004
+Block(49, m1), fileasset: 1004
assetpath = aws_hosts.temp
-hash = 5f061e393e59349bee1791a7021d4a385a734d510100076f7c580a35
+hash = 65e136e6c03f04e09b01f7b3519f690e85f2cb3ddd20f5f1c0842a3e
filecontent:
127.0.0.1 localhost
+10.124.125.126 yahoo.com
...
----------------------
```

The file asset monitored above is a system "hosts" file and the added line creates an association between the specified ip address and the specified domain name. This will cause any DNS queries for yahoo.com to be resolved to the ip address 10.124.125.126. This example illustrates how the hosts file can be compromised. Such a compromise can lead to a system accessing a resource at a server under attacker control possibly leading to malware infection.

The following command was used

```
./bcsim.py admin --dirdiff 2000 2 5
```

to identify changes to directory asset 2000 from block 2 to block 5. The result below indicates (using the plus sign) that the listed file hash is new.

```
----------------------
Directory asset 2000 changed from block 2 to block 5
----------------------
+ ('d139588b3883d843ecd5f8a21a29fd69d98cc4ebf1c65f3d03fa0276',
'/usr/bin/hanoi.py')
----------------------
```

The above result shows that a file, hanoi.py, was added to the system directory /usr/bin (which usually contains executable files). The addition of an unknown file to a system directory is an example of suspicious cyber activity that can be revealed by inspection of the blockchain.

The following command was used

```
./bcsim.py admin --dirasset 5 2000
```

to view the detailed directory listing of asset 2000 in block 5. The resulting directory listing shown below confirms that the added file, hanoi.py, is an executable file. An analyst could conclude that a new executable file was added to the system's /usr/bin directory which often appears in the PATH environment variable. This suggests that further detailed analysis is required since it may be a malicious file because of its sudden appearance among the system's executables.

```
----------------------
Block(5, m2), dirasset: 2000
Dir path = /usr/bin/
Dir hash = c620ce114cd646267ca3180d7e1f8fdd0ffaf9f9a5706b9f552128c4
Dir listing = total 180712
drwxr-xr-x  2 root root         65536 2022-04-30 20:26 .
drwxr-xr-x 14 root root          4096 2021-10-12 22:37 ..
...
-rwxr-xr-x  1 root root           759 2021-04-21 20:17 hanoi.py
...
----------------------
```

The following command was performed

```
./bcsim.py admin --dirdiff 2003 47 62
```

to reveal a change to an important system directory. The result shown below indicates that a file, trustme.gpg, was added to the certificate trust store used by the system package manager. This warrants further investigation because of its implications for system security.

```
----------------------
Directory asset 2003 changed from block 47 to block 62
----------------------
+ ('b776d5535fae7d73653483255a8bf7d531cfe1aad04f4d9337037468',
'/etc/apt/trusted.gpg.d/trustme.gpg')
----------------------
```

It is possible to extract a list of all unique file hashes per agent machine on the blockchain, i.e. per supplier in the supply chain. These consist of pairs of hash, file values and can be used for further forensic analysis, for example, by checking these hashes against databases of known safe hashes, or against known malware hashes.

The following command was used

```
./bcsim.py admin --hashlist
```

to extract the hashes. A sample of the hashes for machine, m1, is given below.

```
----------------------
3beb86f4b2fe6a55cfe91cf46a64c8015dfc858b617d040b3f8811b9
/etc/apt/sources.list
dc2953f86fa9665de717c2dcd8baa35a2829d752642254fe62446e02
/etc/apt/sources.list
81a6ee8164dc2fbd06c187c823279a3faa541b8165e761276d259329
/home/bcadmin/.ssh/authorized_keys
d6b355a5e1cb3c817ef4efecff4f883c12b0a495bf3d169f6bb0861a
/home/bcadmin/.ssh/authorized_keys
17d91d30fb8f7c4190a7b1ed9f24edb9a86dd1bd5fc5514d8b253cba
/usr/bin/411toppm
b98d57d73b8837b0c29b42e2af7b9a97815808bd8b1d7804b16b2107
/usr/bin/7z
...
----------------------
```

In the above result, a repeated file name indicates that there are two different versions of the file on the blockchain. This is revealed by the hashes being different while the filename has not changed.

The following command was performed

```
diff <(./bcsim.py admin -fw 1 1002) <(./bcsim.py admin -fw 10 1002)
```

to discover what changed in asset 1002 from block 1 to block 10. The result below shows the changes that resulted in a different hash for file asset 1002 from block 1 to block 10.

```
----------------------
-Block(1, m1), fileasset: 1002
+Block(10, m1), fileasset: 1002
assetpath = /etc/cron.monthly/awsiplist
-hash = 91a968522f5f7f06369db789c9114f35e1f0dd540e21a4cc3b8eb230
+hash = a4390e87743126482a996744cdccfb255fb726403650ff6bba2b4683
filecontent:
-wqet https://ip-ranges.amazonaws.com/ip-ranges.json
+wqet https://hackers.50cent-army.cn/ip-ranges.json
----------------------
```

In the above result, a "-" prefix on a line corresponds to the earlier block, block 1, while the "+" prefix corresponds to the later block, block 10. It can be seen (under the filecontent) that a line of text in this file was changed

from, "wqet https://ip-ranges.amazonaws.com/ip-ranges.json" to "wqet https://hackers.50cent-army.cn/ip-ranges.json". This change effectively causes the system which depends on this file to use a different target domain for its action. This result illustrates how an important system file could be compromised to redirect host resource requests to an attacker-controlled domain.

The following command was performed

```
./bcsim.py admin --cmdhist 20003
```

to extract the history of command asset 20003. The result is shown below. Command asset 20003 captured network firewall alerts onto the blockchain.

```
...
----------------------
block 5: CmdUfwBlock(20003, ['journalctl', '-o', 'short-unix', '--
no-pager', '-n', '100000', '--quiet', '-g', 'ufw block', '--since',
'@1651343103.1204717'])

1651343684.950113 BLOCK SRC=10.10.3.2 DST=10.10.3.1 PROTO=TCP
SPT=34154 DPT=80 WINDOW=5840 RES=0x00 SYN URGP=0
1651343687.937939 BLOCK SRC=10.10.3.2 DST=10.10.3.1 PROTO=TCP
SPT=34154 DPT=80 WINDOW=5840 RES=0x00 SYN URGP=0
----------------------
block 8: CmdUfwBlock(20003, ['journalctl', '-o', 'short-unix', '--
no-pager', '-n', '100000', '--quiet', '-g', 'ufw block', '--since',
'@1651343701.9061954'])

1651344256.418295 BLOCK SRC=10.10.3.2 DST=10.10.3.1 PROTO=TCP
SPT=52393 DPT=22 WINDOW=5840 RES=0x00 SYN URGP=0
1651344259.398706 BLOCK SRC=10.10.3.2 DST=10.10.3.1 PROTO=TCP
SPT=52393 DPT=22 WINDOW=5840 RES=0x00 SYN URGP=0
1651344263.476739 BLOCK SRC=10.10.3.2 DST=10.10.3.1 PROTO=TCP
SPT=42663 DPT=23 WINDOW=5840 RES=0x00 SYN URGP=0
1651344266.468179 BLOCK SRC=10.10.3.2 DST=10.10.3.1 PROTO=TCP
SPT=42663 DPT=23 WINDOW=5840 RES=0x00 SYN URGP=0
----------------------
...
```

The above result shows that block 5 recorded two attempts to connect to machine, m2, at ip address 10.10.3.1 and port 80. In block 8, four attempts were made to connect to ports 22 or 23. All connection attempts were blocked by the security controls implemented by the host based firewall.

The following command was performed

```
./bcsim.py admin --cmdhist 40000 > sshd_hist
```

to extract the history of command asset 40000 (which monitored SSH connection attempts to machine, m4) and write the output into a file. The resulting file was analyzed further using Linux tools to count unique ip addresses and a geoip lookup was performed. This revealed the countries from which these connection attempts originated. This is currently not a feature of the simulator itself. An extract of the result is shown below. The first field represents the number of intrusion attempts originating from the given source ip address and country.

```
----------------------
...
54          35.189.4.165            AU          Australia
64          27.124.46.42            SG          Singapore
191         61.177.173.19           CN          China
----------------------
```

From the above result we observe that most of the unauthorised connection attempts originated from China. Repeated unauthorised connection attempts to a server constitutes an attack and warrants further investigation and attention.

We performed the following command

```
diff <(./bcsim.py admin -fw 40 1003) <(./bcsim.py admin -fw 64 1003)
```

to identify changes to an important non-system file. The result below shows that supplier financial data was changed by changing the account number and payment limit. The effect of such changes could result in larger financial payments being made to an incorrect bank account.

```
----------------------
-Block(40, m1), fileasset: 1003
+Block(64, m1), fileasset: 1003
assetpath = documents/supplierdata
-hash = 984cc9075cb8fef29bc21cefb32a97063c398af23f269112d76c5ecb
+hash = bbd964b156476347e6fb89d515ad76931e674006e7cade44a104300b
filecontent:
Supplier 1 data:
----------------
SupplierID: X509404
Name: J.Doe, Inc.
Contract: C2105
-Payment-limit: USD20000
+Payment-limit: USD200000
Bank: Fargo Wild West
-Account Number: 239840213
```

```
+Account Number: 239109220
----------------------
```

4.4 Log Files

To discover the history of log files which grow over time we performed the following command

```
./bcsim.py admin --fileassethist 2005
```

and the resulting output (see below) shows the history of file asset 2005.

```
Extracting history of file asset 2005
---------------------
block 2: FileWhole(2005, syslogs/commslog)
12ccfc3ac87b7fdf220dd30659d4f147ca51710e1d3c593bfb409f85
Source Org,Destination Org,Timestamp,Dest IP Address,Action type
SuperIT,Target,2013-02-02 09:00:08,192.168.1.1,Review Invoice
Vending4All,Target,2013-02-03 15:00:15,192.168.1.1,Submit DeliveryOrders
Pallets Inc,Target,2013-02-11 16:00:09,192.168.1.1,Order Sales
...
Pallets Inc,Target,2013-05-23 18:10:25,192.168.1.1,Review Seawaybill
Vending4All,Target,2013-06-10 09:00:44,192.168.1.1,Submit Packinglist
Pallets Inc,Target,2013-06-11 16:00:40,192.168.1.1,Review Sales
---------------------
block 5: FileWhole(2005, syslogs/commslog)
12ccfc3ac87b7fdf220dd30659d4f147ca51710e1d3c593bfb409f85
None
---------------------
...
---------------------
block 17: FileWhole(2005, syslogs/commslog)
12ccfc3ac87b7fdf220dd30659d4f147ca51710e1d3c593bfb409f85
None
---------------------
block 20: FileWhole(2005, syslogs/commslog)
dbea1912cf3667e84b85bff04b4a239f691c28eae3b1a88980866741
Source Org,Destination Org,Timestamp,Dest IP Address,Action type
SuperIT,Target,2013-02-02 09:00:08,192.168.1.1,Review Invoice
Vending4All,Target,2013-02-03 15:00:15,192.168.1.1,Submit DeliveryOrders
Pallets Inc,Target,2013-02-11 16:00:09,192.168.1.1,Order Sales
...
Pallets Inc,Target,2013-05-23 18:10:25,192.168.1.1,Review Seawaybill
Vending4All,Target,2013-06-10 09:00:44,192.168.1.1,Submit Packinglist
Pallets Inc,Target,2013-06-11 16:00:40,192.168.1.1,Review Sales
HVAC Inc,Target,2013-06-12 08:00:55,192.168.1.1,unknown
HVAC Inc,Target,2013-06-22 09:20:45,192.168.2.1,unknown
SuperIT,Target,2013-06-24 09:30:09,192.168.1.1,Order Invoice
HVAC Inc,Target,2013-07-03 16:00:34,192.168.2.200,Admin System
Pallets Inc,Target,2013-07-16 14:30:24,192.168.1.1,Order Sales
HVAC Inc,Target,2013-08-18 12:00:55,192.168.1.1,Review Invoice
HVAC Inc,Target,2013-08-23 08:30:05,192.168.1.1,Order Invoice
---------------------
```

The above result is an extract of the history of asset 2005, a file named commslog. It simulates a log that records actions by authorised remote users of a primary organisation's billing system. Block 2 contains the first record of the contents of this file. The file does not change in subsequent blocks; therefore only the hash value is recorded. In block 20, the hash has changed and therefore the file content is recorded. We see that more records were appended to the log file. Besides visual inspection of the above result, we can gain insight of what exactly changed in the log file by performing the following command

```
diff <(./bcsim.py admin -fw 2 2005) <(./bcsim.py admin -fw 20 2005)
```

The result is shown below.

```
----------------------
-Block(2, m2), fileasset: 2005
+Block(20, m2), fileasset: 2005
assetpath = syslogs/commslog
-hash = 12ccfc3ac87b7fdf220dd30659d4f147ca51710e1d3c593bfb409f85
+hash = dbea1912cf3667e84b85bff04b4a239f691c28eae3b1a88980866741
filecontent:
Source Org,Destination Org,Timestamp,Dest IP Address,Action type
SuperIT,Target,2013-02-02 09:00:08,192.168.1.1,Review Invoice
@@ -27,4 +27,11 @@
Pallets Inc,Target,2013-05-23 18:10:25,192.168.1.1,Review Seawaybill
Vending4All,Target,2013-06-10 09:00:44,192.168.1.1,Submit Packinglist
Pallets Inc,Target,2013-06-11 16:00:40,192.168.1.1,Review Sales
+HVAC Inc,Target,2013-06-12 08:00:55,192.168.1.1,unknown
+HVAC Inc,Target,2013-06-22 09:20:45,192.168.2.1,unknown
+SuperIT,Target,2013-06-24 09:30:09,192.168.1.1,Order Invoice
+HVAC Inc,Target,2013-07-03 16:00:34,192.168.2.200,Admin System
+Pallets Inc,Target,2013-07-16 14:30:24,192.168.1.1,Order Sales
+HVAC Inc,Target,2013-08-18 12:00:55,192.168.1.1,Review Invoice
+HVAC Inc,Target,2013-08-23 08:30:05,192.168.1.1,Order Invoice
----------------------
```

The above result shows that seven new lines were appended to the log file, and two of these have a value of "unknown" in the "Action type" field. If this field is normally used to record the type of action taken, then a value of "unknown" means that a non-standard or unsupported action was performed. This could be due to an attempt to subvert or bypass the system's security controls and warrants further investigation.

The following command was performed

```
./bcsim.py admin --fileassethist 2006
```

to show the history of a log file, invoicelog. The result is shown below.

```
...
---------------------
block 26: FileWhole(2006, syslogs/invoicelog)
e101b8d7aad9d1c0ac14e94dd6a228a94099e3b3ae32ed308d1cd8c8
None
---------------------
block 29: FileWhole(2006, syslogs/invoicelog)
034c1be8b34da7012d84049b348a81f96cda95615d88456be337f792
Invoice Number,Billed By,Date of Issue,Invoice amount
856475-023,SuperIT,2013-02-02 09:00:08,"11000,50"
237865-001,Vending4All,2013-02-03 15:00:15,"56000,00"
...
387591-001,Vending4All,2013-04-16 18:10:20,"15000,50"
901175-023,SuperIT,2013-04-19 10:00:04,"27000,60"
447851-020,HVAC Inc,2013-04-20 15:30:15,"50000,90"
228530-030,Pallets Inc,2013-04-20 17:30:35,"62390,00"
662901-001,Vending4All,2013-05-11 14:00:06,"52709,40"
118629-020,HVAC Inc,2013-05-11 14:40:20,"30000,25"
---------------------
block 32: FileWhole(2006, syslogs/invoicelog)
12c659675827c7b6dc2bfcd52e2628dd2ca8f1cf888c922a740d9461
Invoice Number,Billed By,Date of Issue,Invoice amount
856475-023,SuperIT,2013-02-02 09:00:08,"11000,50"
237865-001,Vending4All,2013-02-03 15:00:15,"56000,00"
...
387591-001,Vending4All,2013-04-16 18:10:20,"15000,50"
901175-023,SuperIT,2013-04-19 10:00:04,"27000,60"
447851-020,HVAC Inc,2013-04-20 15:30:15,"500000,90"
228530-030,Pallets Inc,2013-04-20 17:30:35,"62390,00"
662901-001,Vending4All,2013-05-11 14:00:06,"52709,40"
118629-020,HVAC Inc,2013-05-11 14:40:20,"30000,25"
345981-023,SuperIT,2013-05-19 12:00:16,"65000,00"
417892-030,Pallets Inc,2013-05-23 18:10:25,"42045,30"
893371-001,Vending4All,2013-06-10 09:00:44,"39086,20"
219731-030,Pallets Inc,2013-06-11 16:00:40,"10500,40"
227531-020,HVAC Inc,2013-06-12 08:00:55,"16500,30"
690132-020,HVAC Inc,2013-06-22 09:20:45,"27942,60"
856475-023,SuperIT,2013-06-24 09:30:09,"89213,00"
---------------------
```

The above result shows that the log file grows over time as more entries are appended to it. However, from a digital forensic perspective it is necessary to confirm that nothing else changed besides the new lines. To do this, we performed the following command

```
diff <(./bcsim.py admin -fw 29 2006) <(./bcsim.py admin -fw 32 2006)
```

and the result is shown below.

```
----------------------
-Block(29, m2), fileasset: 2006
+Block(32, m2), fileasset: 2006
assetpath = syslogs/invoicelog
-hash = 034c1be8b34da7012d84049b348a81f96cda95615d88456be337f792
```

```
+hash = 12c659675827c7b6dc2bfcd52e2628dd2ca8f1cf888c922a740d9461
filecontent:
Invoice Number,Billed By,Date of Issue,Invoice amount
856475-023,SuperIT,2013-02-02 09:00:08,"11000,50"
@@ -19,8 +19,15 @@
452389-001,Vending4All,2013-03-27 10:00:55,"27840,25"
387591-001,Vending4All,2013-04-16 18:10:20,"15000,50"
901175-023,SuperIT,2013-04-19 10:00:04,"27000,60"
-447851-020,HVAC Inc,2013-04-20 15:30:15,"50000,90"
+447851-020,HVAC Inc,2013-04-20 15:30:15,"500000,90" !!!!!!
228530-030,Pallets Inc,2013-04-20 17:30:35,"62390,00"
662901-001,Vending4All,2013-05-11 14:00:06,"52709,40"
118629-020,HVAC Inc,2013-05-11 14:40:20,"30000,25"
+345981-023,SuperIT,2013-05-19 12:00:16,"65000,00"
+417892-030,Pallets Inc,2013-05-23 18:10:25,"42045,30"
+893371-001,Vending4All,2013-06-10 09:00:44,"39086,20"
+219731-030,Pallets Inc,2013-06-11 16:00:40,"10500,40"
+227531-020,HVAC Inc,2013-06-12 08:00:55,"16500,30"
+690132-020,HVAC Inc,2013-06-22 09:20:45,"27942,60"
+856475-023,SuperIT,2013-06-24 09:30:09,"89213,00"
----------------------
```

The above result shows seven new lines were appended to the file (from block 29 to 32). This is indicated by the leading "+" character of the last seven lines. The result also reveals that an existing line was modified by changing the "Invoice amount" from 50000,90 to 500000,90. The old version of this line has a leading "-" character, while the new version has a leading "+" character. We highlight the new version of this line with exclamation marks.

4.5 System Configuration Files

Here we show results of system configuration file modifications by extracting insights from the blockchain. Configuration files are used to tailor the operation of a computer. An example is the sources.list file in a Linux system which is used to specify the source repository from where the operating system downloads system updates. The following command was performed

```
./bcsim.py admin --fileassethist 2002
```

to list the history of this file on machine, m2. We condensed the output shown below to the essential parts needed to illustrate the point. The above command produced the following result.

```
----------------------
block 44: FileWhole(2002, /etc/apt/sources.list)
dc2953f86fa9665de717c2dcd8baa35a2829d752642254fe62446e02
```

```
deb https://ftp.acc.umu.se/mirror/ubuntu/ impish main restricted
deb https://ftp.acc.umu.se/mirror/ubuntu/ impish-security universe
deb https://ftp.acc.umu.se/mirror/ubuntu/ impish-security multiverse
...
block 59: FileWhole(2002, /etc/apt/sources.list)
3beb86f4b2fe6a55cfe91cf46a64c8015dfc858b617d040b3f8811b9
deb https://malware.hackersrus.ru/mirror/ubuntu/ impish main restricted
deb https://malware.hackersrus.ru/mirror/ubuntu/ impish-security universe
deb https://malware.hackersrus.ru/mirror/ubuntu/ impish-security
multiverse
----------------------
```

The above result shows (by inspection) that the configured repository domain, ftp.acc.umu.se, was changed to, malware.hackersrus.ru. The changes between these two versions of the configuration file were also viewed using the following command,

```
diff <(./bcsim.py admin -fw 44 2002) <(./bcsim.py admin -fw 59 2002)
```

The result of the above command is shown below where the old (indicated by leading "-") and new (indicated by leading "+") lines are evident under "filecontent".

```
----------------------
-Block(44, m2), fileasset: 2002
+Block(59, m2), fileasset: 2002
assetpath = /etc/apt/sources.list
-hash = dc2953f86fa9665de717c2dcd8baa35a2829d752642254fe62446e02
+hash = 3beb86f4b2fe6a55cfe91cf46a64c8015dfc858b617d040b3f8811b9
filecontent:

-deb https://ftp.acc.umu.se/mirror/ubuntu/ impish main restricted
+deb https://malware.hackersrus.ru/mirror/ubuntu/ impish main restricted
...
-deb https://ftp.acc.umu.se/mirror/ubuntu/ impish-security main restricted
-deb https://ftp.acc.umu.se/mirror/ubuntu/ impish-security universe
-deb https://ftp.acc.umu.se/mirror/ubuntu/ impish-security multiverse
+deb https://malware.hackersrus.ru/mirror/ubuntu/ impish-security main
restricted
+deb https://malware.hackersrus.ru/mirror/ubuntu/ impish-security universe
+deb https://malware.hackersrus.ru/mirror/ubuntu/ impish-security
multiverse
----------------------
```

4.6 Access Control Configuration Files

Configuration files are also used to configure access controls. Here we show the results of our analysis of a configuration file used to configure a specific access control, namely, authorised remote access via SSH.

The following command was performed

```
./bcsim.py admin --fileassethist 2001
```

to discover the history of file asset 2001 which is the SSH authorized_keys file on machine, m2. The result below shows the change in the file hash which was recorded in block 56.

```
----------------------
...
block 50: FileHashOnly(2001, /home/bcadmin/.ssh/authorized_keys)
81a6ee8164dc2fbd06c187c823279a3faa541b8165e761276d259329
----------------------
block 53: FileHashOnly(2001, /home/bcadmin/.ssh/authorized_keys)
81a6ee8164dc2fbd06c187c823279a3faa541b8165e761276d259329
----------------------
block 56: FileHashOnly(2001, /home/bcadmin/.ssh/authorized_keys)
d6b355a5e1cb3c817ef4efecff4f883c12b0a495bf3d169f6bb0861a
----------------------
block 59: FileHashOnly(2001, /home/bcadmin/.ssh/authorized_keys)
d6b355a5e1cb3c817ef4efecff4f883c12b0a495bf3d169f6bb0861a
...
----------------------
```

The above result shows that the file changed from block 53 to block 56. This asset was defined as one where we chose to record the hash but not the content. In this case, an analyst or forensic investigator would need to inspect the file outside of the blockchain system to see the nature of the change. This means access to both versions of the file is needed, as well as access to the system where the file is located. The blockchain system can make this easier by capturing the content of both versions on the blockchain. To accomplish this, the authorized_keys file may be defined as an asset type, FileWhole. This definition will cause the simulator to capture the contents of the file whenever the hash changes. This highlights the importance of careful consideration of how much data and what kind of data the blockchain should record in a specific implementation scenario.

For completeness we show below that the authorized_keys file changed because a new public key was added to the file.

```
----------------------
ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIMyFk8g5o4JE1Dto0IQVhqPz/xIXj0NrKVODMEX3ehrX m0
+ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIGo8nxNk7UH7f5L5ajkDb5YL1EUO6cSlbox9hu2Ew3bR
unknown
----------------------
```

4.7 File Metadata

The blockchain simulation currently does not detect changes to file metadata. This could be a possible further extension of the simulator. We performed various commands (shown below) to illustrate this current limitation by comparing the Directory asset 1005 from block 178 to block 181.

The following command was performed

```
./bcsim.py admin --dirassethist 1005
```

to extract the history of Directory asset 1005 as shown below.

```
----------------------
block 178: Directory(1005, files/)
dirhash = 43644a4d01bb2d08a16eb854cea9c5c92d564799c48881c48456d6f6
----------------------
block 181: Directory(1005, files/)
dirhash = 43644a4d01bb2d08a16eb854cea9c5c92d564799c48881c48456d6f6
----------------------
```

The directory hash in the above result is identical from block 178 to block 181. This indicates that, as far as file contents are concerned, nothing has changed. A comparison of the directory listings, however, reveals that metadata has changed.

The following command was performed

```
./bcsim.py admin --dirasset 178 1005
```

to obtain the directory listing of asset 1005 as recorded in block 178, with the result shown below.

```
----------------------
Block(178, m1), dirasset: 1005
Dir path = files/
Dir hash = 43644a4d01bb2d08a16eb854cea9c5c92d564799c48881c48456d6f6
-rw-r--r-- 1 root    root      1445 2022-04-30 20:17 ntfslog
-rw-r--r-- 1 bcadmin bcadmin 103571 2022-04-30 20:17 secret
----------------------
```

The following command was performed

```
/bcsim.py admin --dirasset 181 1005
```

to obtain the directory listing of asset 1005 as recorded in block 181, with the result shown below.

```
----------------------
Block(181, m1), dirasset: 1005
Dir path = files/
Dir hash = 43644a4d01bb2d08a16eb854cea9c5c92d564799c48881c48456d6f6
-rw-r--r-- 1 root    root      1445 2022-04-30 20:17 ntfslog
-rwxr-xr-x 1 bcadmin bcadmin 103571 2022-04-30 20:17 secret
----------------------
```

The above two results show that although the directory hash is the same in both blocks, the file permissions metadata for the file, secret, has changed. Hashes are computed only over the file contents and do not capture the file metadata. This file's permission mode bits were changed to make it an executable file. While the simulation does not monitor changes to metadata, such changes may be discovered by inspection of the captured directory listings.

Further detailed results from interrogating the blockchain are shown in Appendix C: Block Data Details.

5. Discussion

From the results of the experimental simulation we answer the research questions as follows.

"Can blockchain technology be leveraged to improve supply chain cybersecurity?"

We find that this research question can be answered affirmatively. The results show that we can interrogate the blockchain and identify changes to the status of the monitored digital assets which were defined. We are able to find out how these assets changed and infer, using cybersecurity domain knowledge, whether the observed changes could be related to malicious cyber activity. We demonstrated how to extract useful insights from the blockchain which can inform an oversight role and thus improve the cybersecurity of the supply chain. The value of this approach is that the blockchain constitutes a view into all the participating suppliers and this visibility is key to improving the cybersecurity of the supply chain. We also demonstrated how vulnerability scanning can be performed under the auspices of a blockchain system with the results captured onto the blockchain. A single agent machine in a supplier's premises can be used to monitor assets on many other machines within the supplier's IT infrastructure.

"What features of a blockchain implementation are the most useful to this end?"

From our results, the features of blockchain that were found to be most useful and seem to offer most potential in a supply chain context are:

1. visibility
2. shared forensic record
3. independence from other systems

Leveraging blockchain's visibility as a shared record together with a well-specified data model of key assets to monitor has potential to add value to supply chain cybersecurity. This thesis studied the benefits of information sharing and transparency afforded by a blockchain in which the members of a supply chain participate to share information. The blockchain use case that we investigated may serve as an affordable way to enable a supply chain to achieve better cybersecurity for its SME participants via the oversight capability afforded by this visibility and transparency. Increased visibility of

supply chain data among its participants has potential to enhance collaboration, a need identified in Sweden's national cyber security strategy. The second feature is discussed in more detail in section 5.2 below relating to the usefulness of the resulting blockchain as a forensic record.

The third feature above stems from the separation of the blockchain system from other systems. This implementation aspect involves the use of a distinct, secured VPN for blockchain communications, and dedicated physical machines as agents. This increases the robustness of the blockchain record and its resilience to survive an attack that renders other business systems inoperable. This independence makes an implementation less disruptive to other systems enabling the blockchain system to supplement other existing security measures. The blockchain effectively adds another type of security layer. The concept of layered security is depicted below in Figure 7.

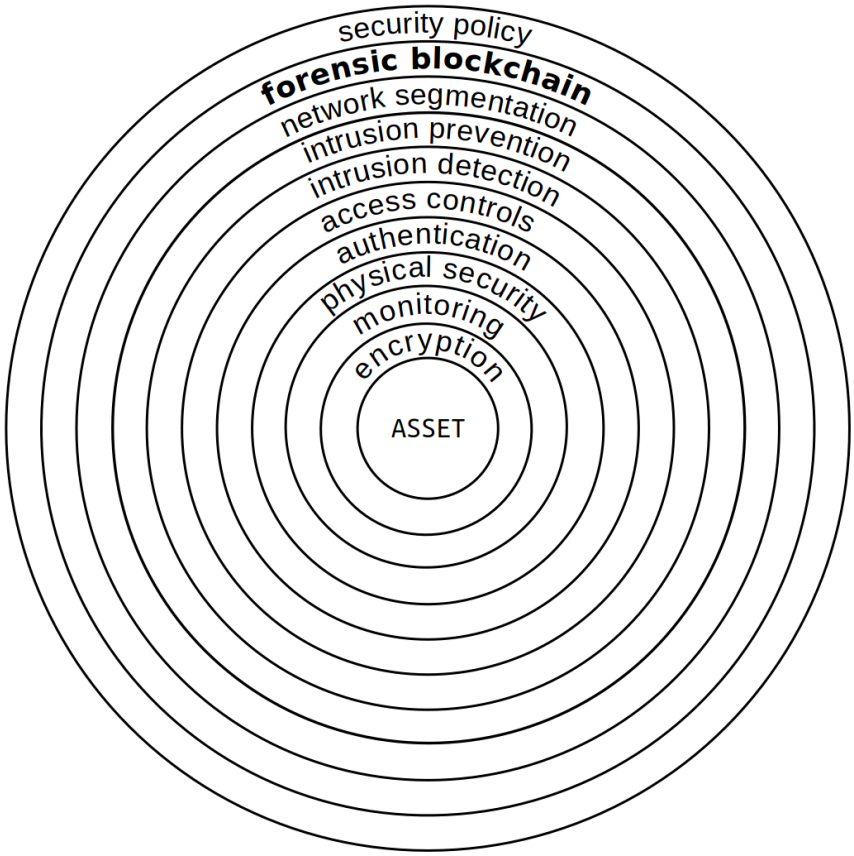

Figure 7: The concept of layered security. Each circle represents a different type of security control. Together these various security measures protect the asset in the centre. An attacker must penetrate all layers to reach the asset. The concept studied in this thesis is shown as "forensic blockchain".

Figure 7 above depicts the onion-like layered security approach where each layer represents a different type of security control. Each layer adds to the total security of the asset to be protected. This asset is shown in the centre surrounded by all the security layers. The order of these layers is not as important as their presence. The complete implementation of security measures consists of the simultaneous application of all these security layers. A hostile attacker must penetrate all the layers in order to get to the asset. Selecting the type and quantity of layers for any given application is a cybersecurity engineering design task. This should result in a cost-effective orchestration of various selected security measures to achieve a desired level of security for an asset. The concept investigated in this thesis is shown as a layer labeled as "forensic blockchain".

"Can a supply chain oversight role extract useful insights from a blockchain that will assist in achieving common cybersecurity objectives?"

The above research question is also answered affirmatively. Aspects have already been mentioned above. The advantages of using the approach investigated in this thesis lie both in its use for detection and monitoring of malicious behaviour, and as a forensic record for post-incident investigations and incident response triage. We emphasized the importance of specifying which assets to monitor as this has a direct bearing on the usefulness of the blockchain data. This specification can be tailored to the characteristics and properties of a particular supply chain to maximise its usefulness.

The motivation and viability of adopting blockchain can vary substantially between businesses that collaborate within a supply chain and between different types of supply chains. It is important to consider existing security requirements of a supply chain and whether these are also met by a blockchain implementation. This consideration led us to formulate our research questions around blockchain's potential to supplement existing security systems. The hope is that this work contributes an enhancement to cybersecurity practices which is not disruptive to existing business processes but effectively adds incremental value. We elaborate further below and discuss other findings of interest.

5.1 Limitations

The simulation has some limitations. It records snapshots of asset state at a frequency specified by the implementation. There is no continuous monitoring, only periodic. This limitation exists because the system is not designed to be a real-time system. Since this blockchain system is not a real-

time monitoring or detection system we do not know when an asset changed. This is a trade-off between frequency of blockchain updates, the data capacity of the blockchain, and the needs of this kind of blockchain-based cybersecurity enhancement. The trade-off is analogous to the trade-off made in the Netflow method of network traffic capture and analysis where packet capture volume is reduced by capturing a reduced subset of traffic data. The result is that meaningful insights can still be extracted from analysis of less data. The extent of such trade-offs is the focus of design and engineering efforts to tailor solutions to specific organisational contexts. Our main purpose is to create a record of the state of assets which can be analysed by the supply chain oversight which we assume is the primary organisation - the implementer of the scheme.

The simulation cannot determine the nature of changes in the content of binary files (a more general problem in computing). This would require use of systems outside the blockchain, for example, the applications used to create the binary files. The simulation also cannot detect changes in file metadata but this functionality can be added to the simulator.

We suggest that blockchain technology could be viewed as a set of primitives, a toolbox of parts and components. How these are woven together and orchestrated into a workable and useful solution to meet a specific need is an engineering and systems design problem. The large gap between expectations of blockchain's disruptive potential and fit-for-purpose solutions is perhaps an indicator that insufficient attention has been given to the engineering stage of developing a practical solution. Software development usually takes place without warranties of any kind let alone fitness-for-purpose. This is a long established characteristic of the software industry. Adopting blockchain may introduce weaker guarantees of fitness-for-purpose than other traditional, tailored enterprise resource management platforms. On the other hand, the peculiar characteristics of blockchain can work to mitigate such general weaknesses. These are elements of supply chain risk management that need to be considered.

5.2 Monitoring and Forensics

The analysis of the simulation results confirms that the proposed concept can be useful for monitoring purposes, albeit not real-time. We readily identified assets that changed and were able to interrogate the blockchain to determine the nature of the changes for most filesystem assets except binary files. The blockchain was able to capture a diversity of asset types from static data such as disk boot sector bytes to dynamic data such as network state and environment variables. We performed this analysis manually to demonstrate the functionality but automation of analysis is possible.

The potential use of the resulting blockchain as an independent digital forensic record is a useful feature that manifested its utility as our research progressed. After having performed analysis of the blockchain we are of the view that the blockchain fits this additional role quite well. The blockchain record constitutes a sound digital forensic record bounded by the explicit and clear limitations of the implementation. This is due to the integrity checking feature of the blockchain, the digital signing of blocks, and implementation design aspects such as the secure VPN communication channel which all add to the credibility of the blockchain as a forensic record. This means that forensically sound, useful inferences can be drawn by examining the blockchain. We think that its use as a supplemental source of corroborating digital evidence in a forensic investigation or incident response process is apt and justified. For example, the analysis of the access control file, authorized_keys, revealed that a new access key had been added. This could be used to corroborate other findings in an investigation. If a forensic investigator found a private key on a suspect's computer then it can be compared with the blockchain data. This comparison is possible even if traces of the modified file had been wiped from the victim machine post-attack and replaced with the original version. This highlights the value of the blockchain as an independent, immutable forensic record.

The blockchain may also be viewed as a method of capturing and recording indicators of compromise (IOC). If we take this perspective, we might use the term, IOC-Blockchain. In a supply chain context, the proposed concept can provide even more value by using the shared common record to correlate suspicious activity across members of the supply chain. This is suited to supply chains which are often targeted by attackers in a systematic and holistic manner as they seek to infiltrate the supply chain at multiple points.

5.3 Blockchain versus other methods

The state of monitored assets could be recorded onto a different medium such as a database or document. This led us to question why a blockchain should be preferred over such other recording methods. To achieve the same degree of integrity other methods would also need to implement similar cryptographically sound approaches for authenticating the authorship of data submissions and verifying the integrity of the entire record of captured data. While a document or database can be shared among participants the associated proofs and attestations of integrity would also need to be shared so that readers can be sure they are viewing an unaltered document or database. We feel that the need to share these different components makes for a more complex implementation. The blockchain, as a vehicle for

information sharing, conveniently provides these components in a flexible and customisable framework readily tailored to specific implementation scenarios. Our simulation is entirely based on proven, well-tested open source software providing alternatives to proprietary databases and document formats.

5.4 Challenges

The research team faced a challenge in the lack of previous work in this specific application of blockchain. Most of the body of research work has been aimed at using blockchain for tracking the status and movement of assets produced and manufactured by the supply chain. The status and location of these assets are tracked and recorded on a shared blockchain as they move through the supply chain from manufacture to end consumer thus providing a history of their provenance and handling. We were curious about monitoring the business IT assets used for day-to-day business operations which underpin the entire supply chain's operational cybersecurity. In other words, this thesis focused on protecting the organisational systems and tools used in the performance of business. This thesis focused on gaining visibility into the overall cybersecurity posture of the supply chain by monitoring the aforementioned assets. We simulated the three agent machines working together in concert and orchestrated by administrative machine, m0. Each agent machine provides a partial view of the supply chain's security posture. Combining these views on the blockchain presents a more complete picture analogous to the tale of The Good, The Bad and The Ugly.

An agreement among supply chain partners to share cybersecurity information via the blockchain should consider how to avoid exposing personally identifying information (PII) on the shared record. This can be avoided by only selecting those digital assets used for system operations and excluding data containing names, addresses, bank account information etc. Exceptions to this approach would need to be justified in terms of privacy laws such as the GDPR. Since the monitoring performed by the blockchain system amounts to agreed surveillance of specified assets the need to also protect sensitive confidential data should be considered. If it is decided and agreed that certain sensitive supplier data should be recorded onto the blockchain then measures such as encryption can be used to protect it. A supplier can encrypt certain block data with its own public key. However this would render the data opaque to the blockchain oversight and impede its use in a forensic investigation. A supplier might agree to make this data visible only to the primary organisation (blockchain oversight). In that case the supplier can encrypt the data with both its own public key and the public key of the oversight. In this way the data can only be decrypted and viewed

by those two parties since they are in possession of the corresponding decryption keys.

5.5 Implementation Aspects

The blockchain use case explored in this study can augment other existing cybersecurity systems by adding value over and above those. In the case of SME's who often do not have the budget for costly cybersecurity controls, the proposal of this thesis may be a cost-effective alternative. Some security monitoring is better than none. For the primary organisation, the proposed concept is easy to justify for suppliers who have little or no cybersecurity. It is also justifiable in general because it is not intrusive and does not interfere with existing business operational processes. Agents only need read access to agreed-upon digital assets. The common benefit to the whole supply chain can be motivated accordingly and contractually specified in the primary organisation's procurement requirements. The proposed concept can be considered low cost because it involves the cost of one agent machine (a standard modern desktop computer) and associated networking per supplier. Each agent machine should have a degree of physical security specified by the primary organisation. This is a consideration in establishing the forensic soundness of the blockchain data. This aspect may result in a higher implementation cost. In the simulation, the signing keys are stored on the agent machines' storage disks. In a real implementation these could be stored in a physical HSM attached to each machine.

The fact that an asset has changed and that the time of change occurred between two block timestamps constitutes useful information that can be of value in a forensic investigation. Supply chains are increasingly the target of APT actors whose tactics are characterised by slow, methodical infiltration. Thus a non-real-time system can still provide effective monitoring of valuable assets and detection of compromise as we have demonstrated through manual analysis. The common blockchain record can also allow correlation of suspicious activities or asset status changes across participating suppliers. This has obvious value to the primary organisation which has a vested interest in the security of the whole supply chain upon which it depends.

The common visibility into the shared data can have different uses, for example, security monitoring, anomaly detection, periodic auditing, and to support other compliance efforts and common cybersecurity objectives. While many businesses operate as silos even though they are part of a supply chain, the blockchain could add value by mandating transparency into a tailored set of data relevant to the common cybersecurity of the supply chain. Such a mandate could form part of the contractual

relationships between the business partners and could specify what set of data is to be recorded. This could be useful for organizations and companies that provide important services where cybersecurity requirements are high and where the regulatory cybersecurity compliance burden on the primary organization is high. In this way the primary organization could specify cybersecurity requirements for its suppliers to be implemented as an IOC-blockchain. The blockchain may provide benefits such as enabling limited, periodic security audits more frequently than more comprehensive, costly formal audits.

A comprehensive view of supply chain cybersecurity acknowledges not only the need to ensure integrity of the final product but also the integrity and security of equipment, tools, processes and people employed in its production. As large, well-resourced primary organisations are improving their cybersecurity, attackers are changing their tactics to target weaker, smaller suppliers in the supply chain. Since supply chains can be large and extensive the aggregate attack surface presented collectively by all suppliers in a supply chain is much broader than that of the attackers' final target, the customer. The chain of security is only as strong as its weakest link.

The constantly evolving nature of supply chain business and operational relationships makes supply chain incident response a difficult task. The sophistication of the attackers and the cascading effects on many dependent organisations complicates forensic analysis. Impediments to forensic investigation and lack of transparency can undermine trust in the supply chain. The blockchain use case studied in this thesis can help in addressing these challenges by providing a forensic record that remains immutable and can serve as an independent source of forensic data during incident response and investigation. By adding value to incident response the proposed blockchain adds another layer of capability to incident responders and cybercrime investigators.

While the simulation is a simplified implementation, it could be used as a platform for further experimentation and research. It could serve as a learning tool in an educational context perhaps as a small project to expand some of its features, or as a laboratory excercise. The proposed concept adds to the pool of cybersecurity techniques, tools, methods and experience.

5.6 Security

All cybersecurity systems ultimately rest on the same security primitives and principles which are well understood. In our view, the strength of applied cybersecurity controls is where huge variances are to be found because these are based on the quality of design and engineering of the

complete system which depends on business factors such as budget. In cybersecurity it is imperative that security solutions are proven over time and hence we think an incremental approach holds more potential to add value. We chose to focus our work on using blockchain to augment existing security controls rather than advocate for a disruptive replacement of proven, but imperfect, cybersecurity practices. We believe that incremental changes that prove themselves over time can offer practical and sustainable added value and this is how we positioned this research work. Due to the constantly evolving threat landscape, it is the custom and practice of cybersecurity practitioners to continually seek improvements to existing cybersecurity methods and strategies. We all benefit from the diverse and complimentary knowledge contributions from both the academic and practitioner communities in our quest to improve cybersecurity.

The use of cryptographic schemes in blockchain technology must ensure the secure management of secrets and their certified representations which act as proxies for participants' identities. We believe this is not a simple matter and has significant ramifications for trust given the large number of potential participants in a blockchain system. This is another reason to consider governance as a key factor in the application of blockchain in a supply chain context because at stake is the business of the entire supply chain and its members. Since data integrity and authenticity rely on digital signatures, participants rely on the blockchain implementation to verify those signatures. In a sense, the blockchain is both the verified and the verifier. This aspect of self-contained trust is appealing but obscures the fact that the crucial secrets underlying the processes of verification and attestation are stored and managed outside the blockchain. We compare this to a traditional ledger where the ledger may be a trustworthy record but it is ensconced by an organisation of processes and people who are responsible for implementing the trust. An auditor examines not only the trustworthy record but also evaluates how effectively and correctly the surrounding organisation implemented the trust which the ledger purports to exhibit. If the methods of trust implementation and the exhibit of the ledger concur then it follows that organisational activities were conducted correctly and in compliance with a definition of best practice.

Much like the authenticity of any electronic commerce transaction, the authenticity of blockchain data rests upon cryptographic primitives correctly engineered together to achieve the required security objectives. In particular, the blockchain relies on participant secrets to attest to the authenticity of data and the identity of participants via digital signatures. This dependence relies upon the integrity of the processes and systems used to manage and control each participant's own secrets. The blockchain assumes that all those secrets are correctly and securely managed by every participant. The

compromise of a single participant's secrets undermines the integrity of the blockchain. The actual control of a secret is hidden from view, unknown and cannot be guaranteed. If a participant's secrets are compromised then the data and identity attested by those secrets is no longer under the control of the participant assumed to be represented thereby. Control of secrets can be achieved to some degree by a key infrastructure. This usually introduces an aspect of centralized control. The blockchain is only as strong and trustworthy as the security of each participant's secrets. This could be a reason for the debate around technical validity of digital signatures versus legitimacy in a legal sense.

5.7 Future Work

We suggest the following ideas to add more functionality to the simulator implementation:

- Making use of the totaldatasize attribute as a measure of the total size of data stored on the blockchain.
- Metadata detection functionality can be added so that changes to file metadata can be captured.
- Geoip lookup functionality can be incorporated (if relevant or needed).
- New assets to be monitored could be specified to agents via the blockchain. This requires significant changes to the simulation protocol.
- Comprehensive vulnerability scanning could be done using instructions sent to agents via the blockchain or specifying this as a dynamic asset.

A test implementation on a cyber range may also provide valuable insights for further development by simulating an active hostile environment.

The administrative machine m0, could perform a more active role in blockchain operation by also writing blocks. For example, instead of a round robin time-based schedule which agents use to determine their turn to create a new block, they could wait for a special block written by m0 instructing a specific agent to write a new block. We term such blocks "go blocks" because it would communicate to an agent the message: "go ahead, create a new block". This does away with the need for agents to adhere to a schedule. Each agent would simply monitor the blockchain for a block containing a token encrypted with the agent's public key. If the agent is able to decrypt the encrypted token it is interpreted as a "go" instruction to create a new block. The agent would determine the authenticity of the instruction

from a signature created by m0, the author of the go block. The concept of go blocks allows the oversight to have more control of the blockchain and may be suited to a supply chain scenario where the primary organisation needs a high degree of control. Implementation of go blocks requires significant changes to the simulation programme.

An area for substantial improvements to the simulator would be to automate some of the analysis which we have conducted manually. Whether this blockchain application could be developed into a SIEM-like system is debatable as such systems are designed as real-time systems. Automation of analysis may have value for auditing purposes. Hence a non-real-time equivalent of a SIEM system specially designed for supply chain oversight and auditing may be an interesting avenue of research.

When selecting assets to be monitored by this blockchain it may be beneficial to rank them according to a priority of importance or value to the organisation. This may facilitate more focused attention by oversight when the status of critical assets have changed. An extra header can be defined for a block to signal that the block contains such changes, and another header or asset attribute can be used to flag and identify the assets.

Use of this blockchain system for vulnerability scanning seems a very interesting avenue for further research and application prototypes. This would introduce a vulnerability scanning capability inside each supplier organisation, the results of which will end up on the blockchain. We can think of this as a more active monitoring of the supply chain cybersecurity posture. Also worth pursuing is adding extra analysis functionality to perform correlation of suspicious activity across the supply chain, perhaps even including associated metrics on the blockchain.

The use of blockchain studied in this thesis can be generalizable to any supply chain that uses IT infrastructure in the performance of their business operations. The flexibility of the approach studied in this thesis is appealing. Supply chains in developing nations based on traditional, informal trading and markets are likely to have less IT penetration and thus less scope for this kind of cybersecurity. Those supply chains are also likely to be less vulnerable to cyber attacks. Nevertheless, as those economies grow there is likely to be scope for low-cost concepts such as that proposed in this thesis. In general, we expect any supply chain operating in the global digital economy (and thus connected to the internet) to be within the scope of the proposed use case. This project made exclusive use of open-source software and can thus be considered viable from a cost perspective.

6. Conclusions

We find that the blockchain concept explored in this thesis serves two functions equally well. It provides a periodic monitoring capability suitable for cybersecurity oversight of a supply chain in the form of an auditable record of the state of assets. It also provides an independent forensic record useful in forensic investigations and incident response. The use of blockchain to provide a broad overview of the cybersecurity posture of suppliers in a supply chain seems well suited in that context. Its cost effectiveness makes it an interesting candidate as a cybersecurity value-add because it can potentially provide another layer of effective cybersecurity for little cost.

In the regional context, Sweden's national cyber security strategy calls for increased collaboration among organizations to improve national cybersecurity. This thesis contributes to such efforts by providing visibility, transparency and vital information sharing for supply chains via a shared blockchain. A supply chain has a similar goal which is to protect the entire supply chain against cyber attacks. The value that increased collaboration adds to the total supply chain cybersecurity is apparent from the usefulness of sharing critical security data for the common good. This is provided by the blockchain use case studied in this thesis. The principle is similar to that used by open source software development processes where the more eyes are on the asset, the more attention it receives, and the greater the likelihood of detecting a compromise. The concept is a way to implement collaborative initiatives in a practical way.

The application of blockchain demonstrated in this thesis promotes trust between supply chain partners by sharing valuable data for a common objective. It can help to cover SME suppliers under the supply chain cybersecurity umbrella. It provides visibility into a curated set of data relevant to the common cybersecurity of collaborative supply chain participants. The data shared via the blockchain is intended for consumption by an oversight function e.g., compliance officer, auditor or analyst. It is not intended to replace or duplicate existing backup systems. The primary organization is the one driving the scheme and there should be a real benefit to justify an implementation. The approach demonstrated in this study can help a supply chain to meet compliance or other cybersecurity mandates in an affordable manner while retaining flexibility for enhanced oversight functions.

The cybersecurity threat landscape has been evolving more rapidly than the ability of legislation, standards, and organisational expertise to adequately

address this growing sphere of business risk. Even large established service providers like Microsoft have failed to timeously respond to threats against customers of their services (e.g. proxylogon vulnerability). It has become vividly apparent that achieving effective organisational cybersecurity is a best effort approach and compliance certifications do not guarantee protection. When acknowledged, this realization can motivate for an engineering approach to designing cybersecurity defenses where tradeoffs and compromises are made to achieve a practical, transparent solution with known boundaries and limitations. Creative design constrained by defined, specified boundaries and limitations has been the hallmark of successfully engineered construction in the built environment. Adopting a similar philosophy in approaching the challenge of cybersecurity may facilitate more sustainable and achievable supply chain cybersecurity.

References

(referencing style APA7)

Agrawal, T., Kumar, V., Pal, R., Wang, L. & Chen, Y. (2021) Blockchain-based Framework for Supply Chain Traceability: A Case Example of Textile and Clothing Industry. Computers & Industrial Engineering 154. https://doi.org/10.1016/j.cie.2021.107130

Ault, J. (2018). Advancing the Science and Impact of Blockchain Technology at Oak Ridge National Laboratory. Computational Sciences and Engineering Division, Oak Ridge National Laboratory. https://info.ornl.gov/sites/publications/Files/Pub118487.pdf

Al-Jaroodi, J. & Mohamed, N. (2019). Blockchain in Industries: A Survey. IEEE Access, 7. https://doi.org/10.1109/ACCESS.2019.2903554

Bayramova, A., Edwards, D. & Roberts, C. (2021). The Role of Blockchain Technology in Augmenting Supply Chain Resilience to Cybercrime. Buildings 11(7). https://doi.org/10.3390/buildings11070283

Berdik, D., Otoum, S., Schmidt, N., Porter, D. & Jararweh, Y. (2021). A Survey on Blockchain for Information Systems Management and Security, Information Processing & Management, 58(1). https://doi.org/10.1016/j.ipm.2020.102397

Brendel, J., Cremers, C., Jackson, D. and Zhao, M. (2021). The Provable Security of Ed25519: Theory and Practice. IEEE Symposium on Security and Privacy. https://doi.org/10.1109/SP40001.2021.00042

Chang, S. & Chen, Y. (2020). When Blockchain Meets Supply Chain: A Systematic Literature Review on Current Development and Potential Applications. IEEE Access 8. https://doi.org/10.1109/ACCESS.2020.2983601

Choo, K., Ozcan, S., Dehghantanha, A., Parizi, R. (2020). Editorial: Blockchain Ecosystem-Technological and Management Opportunities and Challenges. IEEE Transactions on Engineering Management, 67(4). https://doi.org/10.1109/TEM.2020.3023225

Cole, R., Stevenson, M. & Aitken, J. (2019). Blockchain technology: implications for operations and supply chain management. Supply Chain Management, 24(4). https://doi.org/10.1108/SCM-09-2018-0309

Council of Economic Advisers (CEA). (2018). CEA Report: The Cost of Malicious Cyber Activity to the U.S. Economy. Executive Office of the President. https://trumpwhitehouse.archives.gov/articles/cea-report-cost-malicious-cyber-activity-u-s-economy/

Cybersecurity and Infrastructure Security Agency (CISA). (2021). Defending Against Software Supply Chain Attacks. National Institute of Standards and Technology (NIST). https://www.cisa.gov/sites/default/files/publications/defending_against_software_supply_chain_attacks_508_1.pdf

Cybersecurity and Infrastructure Security Agency (CISA). (2022). Cybersecurity. https://www.cisa.gov/cybersecurity

Defense Science Board Task Force on Computer Security. (1970). Security Controls for Computer Systems. U.S. Office of The Director of Defense Research and Engineering. https://doi.org/10.7249/R609-1

Department for Digital Culture Media & Sport. (2021). Policy Paper: Response to the call for views on supply chain cyber security. UK Government. https://www.gov.uk/government/publications/government-response-on-supply-chain-cyber-security/government-response-to-the-call-for-views-on-supply-chain-cyber-security

Departments of Commerce and Homeland Security. (2022). Assessment of the Critical Supply Chains Supporting the U.S. Information and Communications Technology Industry. https://www.dhs.gov/publication/assessment-critical-supply-chains-supporting-us-ict-industry

Department of Justice. (19 May 2014). U.S. Charges Five Chinese Military Hackers for Cyber Espionage Against U.S. Corporations and a Labor Organization for Commercial Advantage. https://www.justice.gov/opa/pr/us-charges-five-chinese-military-hackers-cyber-espionage-against-us-corporations-and-labor

Dutta, D., Choi, T., Somani, S., Butala, R. (2020). Blockchain technology in supply chain operations: Applications, challenges and research opportunities. Transportation Research Part E, Logistics and Transportation Review, 142. https://doi.org/10.1016/j.tre.2020.102067

Etemadi, N., Van Gelder, P. & Strozzi, F. (2021). An ISM Modeling of Barriers for Blockchain/Distributed Ledger Technology Adoption in Supply

Chains towards Cybersecurity. Sustainability 13(9). https://doi.org/10.3390/su13094672

Etemadi, N., Borbon-Galvez, Y., Strozzi, F., Etemadi, T. (2021). Supply Chain Disruption Risk Management with Blockchain: A Dynamic Literature Review. Information 12(2). https://doi.org/10.3390/info12020070

European Union Agency for Cybersecurity (ENISA). (2021). Threat Landscape for Supply Chain Attacks. European Network and Information Security Agency. https://data.europa.eu/doi/10.2824/168593

Ghadge, A., Weiß, M., Caldwell, N. & Wilding, R. (2020). Managing cyber risk in supply chains: a review and research agenda. Supply Chain Management, 25(2). https://doi.org/10.1108/SCM-10-2018-0357

Government of Sweden. (2017). A national cyber security strategy. (Regeringens skrivelse 2016/17:213). https://www.government.se/4ada5d/contentassets/d87287e088834d9e8c08f28d0b9dda5b/a-national-cyber-security-strategy-skr.-201617213

Hellani, H., Sliman, L., Samhat, A. & Exposito, E. (2021). On Blockchain Integration with Supply Chain: Overview on Data Transparency. Logistics 2021, 5(3). https://doi.org/10.3390/logistics5030046

Herr, T., Lee, J., Loomis, W. & Scott, S. (2020). BREAKING TRUST: Shades of Crisis Across an Insecure Software Supply Chain. Scowcroft Center for Strategy and Security. Atlantic Council. https://www.atlanticcouncil.org/in-depth-research-reports/report/breaking-trust-shades-of-crisis-across-an-insecure-software-supply-chain/

International Organization for Standardization (ISO). (2007). Security management systems for the supply chain - Best practices for implementing supply chain security, assessments and plans - Requirements and guidance (ISO 28001:2007). https://www.iso.org/obp/ui/#iso:std:iso:28001:ed-1:v1:en

Jabbar, S., Lloyd, H., Hammoudeh, M., Adebisi, B. & Raza, U. (2020). Blockchain-enabled supply chain: analysis, challenges, and future directions. Multimedia Systems, 27. https://doi.org/10.1007/s00530-020-00687-0

Kshetri, N. (2017). Blockchain's roles in strengthening cybersecurity and protecting privacy. Telecommunications Policy, 41(10). https://doi.org/10.1016/j.telpol.2017.09.003

Lesavre, L., Varin, P., Mell, P., Davidson, M., & Shook, J. (2020). A Taxonomic Approach to Understanding Emerging Blockchain Identity Management Systems. Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology (NIST), Cybersecurity White Paper. https://doi.org/10.6028/NIST.CSWP.01142020

Melnyk, S., Schoenherr, T., Speier-Pero, C., Peters, C., Chang, J. & Friday, D. (2022). New challenges in supply chain management: cybersecurity across the supply chain. International Journal of Production Research. 60(1). https://doi.org/10.1080/00207543.2021.1984606

Middleton, B. (2017). A History of Cyber Security Attacks: 1980 to Present. Taylor & Francis Group. https://doi.org/10.1201/9781315155852

Misa, T. (2016). Computer Security Discourse at RAND, SDC, and NSA (1958-1970). IEEE Annals of the History of Computing, 38(4). https://dl.acm.org/doi/10.1109/MAHC.2016.48

Mylrea, M. & Gourisetti, S. (2018). Blockchain for Supply Chain Cybersecurity, Optimization and Compliance. Proceedings of Resilience Week, USA, 70-76. https://doi.org/10.1109/RWEEK.2018.8473517

National Counterintelligence and Security Center. (2020). National Counterintelligence Strategy of the United States 2020-2022. Office of the Director of National Intelligence (ODNI). https://www.dni.gov/files/NCSC/documents/features/20200205-National_CI_Strategy_2020_2022.pdf

National Counterintelligence and Security Center. (2021). SolarWinds Orion Software Supply Chain Attack. Office of the Director of National Intelligence (ODNI). https://www.dni.gov/index.php/ncsc-features/2762

National Counterintelligence and Security Center. (2021). Bulletin: Kaseya VSA Supply Chain Ransomware Attack. Office of the Director of National Intelligence (ODNI). https://www.dni.gov/index.php/ncsc-features/2762

National Counterintelligence and Security Center. (2021). Fact Sheet: Protect Your Organization from the Foreign Intelligence Threat. Office of the Director of National Intelligence (ODNI). https://www.dni.gov/index.php/ncsc-features/2762

National Cyber Security Centre. (2018). Principles of supply chain security. Centre for the Protection of National Infrastructure. https://www.ncsc.gov.uk/collection/supply-chain-security

Patil, P., Sangeetha, M. & Bhaskar, V. (2021). Blockchain for IoT Access Control, Security and Privacy: A Review. Wireless Personal Communications, 117. https://doi.org/10.1007/s11277-020-07947-2

Plachkinova, M. & Maurer, C. (2018). Teaching Case Security Breach at Target. Journal of Information Systems Education, 29(1). http://jise.org/Volume29/n1/JISEv29n1p11.html

Pournader, M., Yangyan, S., Seuring, S., & Koh, S. (2020). Blockchain applications in supply chains, transport and logistics: A systematic review of the literature. International Journal of Production Research, 58(7), 2063-2081. https://doi.org/10.1080/00207543.2019.1650976

President of the United States. (24 February 2021). Executive Order on America's Supply Chains. (EO14017). https://www.whitehouse.gov/briefing-room/presidential-actions/2021/02/24/executive-order-on-americas-supply-chains/

Saxe, J.G. (1873). The Poems of John Godfrey Saxe: Complete Edition. James R. Osgood and Company.

Taylor, P., Dargahi, T., Dehghantanha, A., Parizi, R. & Choo, K. (2020). A systematic literature review of blockchain cyber security. Digital Communications and Networks, 6(2). https://doi.org/10.1016/j.dcan.2019.01.005

van Hoek, R. (2019). Exploring blockchain implementation in the supply chain: Learning from pioneers and RFID research. International Journal of Operations & Production Management, 39(6/7/8). https://doi.org/10.1108/IJOPM-01-2019-0022

Warren, M. & Hutchinson, W. (2000). Cyber attacks against supply chain management systems: a short note. International Journal of Physical Distribution & Logistics Management, 30(7/8). https://doi.org/10.1108/09600030010346521

Windelberg, M. (2016). Objectives for managing cyber supply chain risk. International Journal of Critical Infrastructure Protection, 12. https://doi.org/10.1016/j.ijcip.2015.11.003

Yaga, D., Mell, P., Roby, N., & Scarfone, K. (2018). Blockchain Technology Overview. National Institute of Standards and Technology (NIST), Internal Report 8202. https://doi.org/10.6028/NIST.IR.8202

Appendix A: Asset Definitions

In this section we describe the various digital assets monitored by the simulation and recorded onto the blockchain. Their relevance to cybersecurity is explained by considering the normal expected use of such assets in a business environment and describing possible implications of their compromise. All assets chosen for this simulation serve as examples of the kinds of business assets that could be identified as part of a cybersecurity risk assessment. Such an assessment can lead to a tailored security strategy and policy suited to a specific organisation. Cybersecurity controls can then be designed to protect the identified valuable digital assets.

The assets are specified in text files with one line per asset. The lines for file assets and command type assets have the following forms respectively,

```
AssetID, AssetType, AssetPath
CmdID, CmdType, Command
```

where the ID's are unique to each asset and the Types are as defined in the implementation. AssetPath is the filesystem path to an asset. Command is a specified command or it is omitted, in which case, it is defined as part of the Type definition. The first digit of each ID corresponds to the agent machine responsible for monitoring that asset. All defined assets are enumerated below, each followed by a brief description of its relevance to cybersecurity.

```
1000, Directory, /usr/share/ca-certificates/mozilla/
```

This system directory is an important part of a standard Linux computer desktop. It contains X.509 certificate files of Certificate Authorities (CA) used by the system to verify the authenticity of other identity assertions. This directory acts as a root of trust for TLS based communications from this computer to the internet. The unauthorised addition of a CA certificate file to this directory would result in any other certificate signed by that CA to be trusted by the system. This directory is a potential target for an attacker wishing to plant a malicious certificate file to subvert the system's trust mechanisms.

```
1001, FileHashOnly, documents/reactorspec.odt
```

This binary file is a simplified representation of a technical specification document. Such documents contain technical parameters, design tolerances, and other contractually specified details of products to be manufactured.

Unauthorised and undetected modification of such a file can have disastrous consequences if, for example, a product is manufactured to tolerances or parameters that have been maliciously altered.

```
1002, FileWhole, /etc/cron.monthly/awsiplist
```

This file is an example of a cronjob file used by Linux-based computer systems to automate the execution of system, administrative and operational tasks performed by computers. Unauthorised malicious modification of such files can subvert the proper operation of a system and compromise its security.

```
1003, FileWhole, documents/supplierdata
```

This file is an example of a digital asset, the compromise of which, could affect business financial systems and transactions. This specific file simulates a record or database of supplier payment data used for making financial payments for services procured from suppliers.

```
1004, FileWholeRemote, aws_hosts.temp
```

The "hosts" file can be used on Linux systems to provide hostname to ip address mappings which can override those from DNS. Hence malicious modification of this file can be used to subvert the cybersecurity of a computer system. This hosts file resides on a real remote server located in the AWS cloud. Agent machine, m1, monitors the status of this file remotely. This simulates an agent machine monitoring digital assets on another computer located elsewhere within a supplier's enterprise network (e.g. warehouse or satellite branch office).

```
1005, Directory, files/
```

This directory serves to simulate any directory of important files used by a supplier in the normal course of business. We perturb the contents of this directory in various ways to simulate files being modified, deleted or renamed. We attempt to detect these changes when analysing the blockchain.

```
2000, Directory, /usr/bin/
```

This directory is used in Linux computer systems to store executable files. The presence of malicious executable files in this directory effectively amounts to system compromise. In the simulation, the blockchain records a hash over all the files in this location as well as the hashes of the individual

files. In this way any modification of the directory's contents appears on the blockchain as a forensic record.

```
2001, FileHashOnly, /home/bcadmin/.ssh/authorized_keys
```

On Linux and other computers where remote access is enabled using SSH protocol, this file contains a list of public keys associated with users authorised to access the system. Any public key listed in this file enables the possessor of the associated private key to remotely connect to the system. The contents of this file must be closely guarded.

```
2002, FileWhole, /etc/apt/sources.list
```

This file specifies the source repositories for operating system updates including security updates. Malicious modification of this file can enable an attacker to subvert system cybersecurity by directing the system to obtain its updates from an attacker-controlled server.

```
2003, Directory, /etc/apt/trusted.gpg.d/
```

This directory contains the cryptographic public keys of a Linux operating system's software publishers. This directory acts as a trust anchor for all system updates. Insertion of a malicious key into this directory can allow an attacker to cause the system to install malicious software.

```
2004, FileWhole, syslogs/userlog
```

This artificial log file is used to simulate the monitoring of log files which have the purpose of recording user activity of any sort. This is used to illustrate how changes to file contents can be viewed as part of the analysis of the blockchain.

```
2005, FileWhole, syslogs/commslog
```

This file serves as an example of a communications log where certain communication events are appended to the log file as they occur.

```
2006, FileWhole, syslogs/invoicelog
```

This file is a simple example of an invoicing record captured onto the blockchain. The choice to monitor such digital assets would be made based on the value of information contained therein and the potential risk to business operations if the integrity of such files are compromised.

```
4000, FileWhole, /etc/passwd
```

This file is a system file which underpins access control and authorisation for users of a Linux based computer system. It contains a record of all user accounts authorised to access and use the computer system. This file is located on agent machine, m4, hosted in the AWS cloud.

For the simulation, we define another type of digital asset which we term a "command asset" or a "command type asset". This means that a defined command is executed on a machine and the command output is recorded. The output of the command constitutes the digital asset which is monitored and captured by the blockchain. In the simulation, these instructions have been pre-defined on the agent machines and only the results (output) of their execution is captured by the blockchain.

```
10000, CmdLast,
```

This command type asset is pre-defined as the execution of the Linux system command "last". It outputs a history of all user login activity on a Linux machine. In the simulation this command is executed on the agent machine, m1.

```
10001, CmdGeneric, echo "$PATH"
```

This is a command asset of the generic sub-type where we can define any command to be executed by an agent machine. In this example case, we use it to capture the state of a special kind of dynamic system asset, an environment variable. Environment variables are used by systems to store real-time operational state information which affects system operation at any time the environment variables are in use. Malicious modification of environment variables can give an attacker the ability to affect the operation of the system in real-time.

```
10002, CmdNetstat,
```

This pre-defined command provides a view of transient and active network connections on a Linux computer and is useful for gaining insight into the current network state of a machine. This command is useful because it shows the state of any established network connections. Capturing its output can serve as a valuable cybersecurity forensic record of network activity, and most importantly, the IP addresses of the remote endpoints involved in such activity.

```
20000, CmdNmap,
```

The pre-defined "nmap" command is useful for probing connected computer systems for vulnerabilities, specifically open ports. In the simulation, we execute this command on agent machine, m2, to probe the downstream connected machine, m3, which is not an agent machine. We configured machine, m3, with the publicly available metasploitable operating system. This system conveniently has numerous open ports and we can probe the state of these from agent machine, m2. This type of record on the blockchain can be used to assess the networking state of computers within a supplier network. Any discovered open ports can be compared against security policies and controls to identify machines that have vulnerabilities or unnecessary network exposure. Open ports that are not needed for business operations can constitute an increased cybersecurity risk and an unnecessary enlargement of a supplier's threat surface.

```
20001, CmdGeneric, route -n
```

This generic command type asset executed on agent machine, m2, provides a view of the system's routing table. The routing table is a vital networking component as it specifies the paths for IP packet-based communications. Malicious modification of the routing table can subvert machine-to-machine communications and thus compromise cybersecurity of a supplier's computer system.

```
20002, CmdGeneric, dd if=/dev/sda count=2 | sha224sum | awk '{print
$1}'
```

This pre-defined system command outputs the hash of the Basic Input Output System (BIOS) boot sector of a computer system's persistent storage medium (or disk). This vital piece of software code is stored in the first sectors of a hard disk and is used for system initialization on bootup. Malicious compromise and modification of this software code undermines the integrity and confidentiality of the entire system.

```
20003,CmdUfwBlock,
```

This pre-defined command is executed on agent machine, m2, and monitors host firewall alerts on the machine. This is an example of a defined command asset which extracts event data from log files generated by a computer system. In the simulation, we defined this command to extract data from the Linux system journal logging facility on agent machine, m2. All agent machines have been configured with an active host firewall configured to only allow authorised connections. We simulate unauthorised

connection attempts to agent machine, m2, by manually issuing netcat connection commands on the metasploitable machine, m3.

```
40000, CmdSshd,
```

This pre-defined command asset extracts event data from Linux journal system logs on agent machine, m4. The extracted data is a record of all unsuccessful SSH connections attempts to machine, m4, since the previous execution result of this command on the blockchain. In other words, only events that have occurred since the timestamp of the previous block associated with this asset, are recorded. For the simulation, we configured a real server hosted in the AWS cloud with an SSH server listening on a publicly accessible open port 22. As expected in today's hostile internet, regular and numerous unauthorised connection attempts were made in real-time against this server. The data we capture onto the blockchain using this command constitutes a random source of data.

Appendix B: Simulator Synopsis

This is the usage help text provided by the simulation programme.

```
usage: bcsim.py admin [-h] [--sync] [-p] [-i] [-s BLKNUM] [-d
BLKNUM] [-r BLKNUM ASSETID] [-f BLKNUM ASSETID] [-fw BLKNUM
ASSETID] [-fr BLKNUM ASSETID] [-c BLKNUM ASSETID] [-rh ASSETID] [-
fh ASSETID] [-ch CMDID] [-hc AUTHOR] [-dd ASSETID BLKNUM1 BLKNUM2]
[-hl]
optional arguments:
-h, --help            show this help message and exit
--sync                determine latest blockchain for syncing
-p, --printsummary    Print blockchain summary
-i, --checkintegrity  Check blockchain integrity
-s BLKNUM, --summarizeblock BLKNUM
      Summarize a block
-d BLKNUM, --blockdata BLKNUM
      Show block data details
-r BLKNUM ASSETID, --dirasset BLKNUM ASSETID
      Show dir asset
-f BLKNUM ASSETID, --fileasset BLKNUM ASSETID
      Show file asset
-fw BLKNUM ASSETID, --filewholeasset BLKNUM ASSETID
      Show filewhole asset
-fr BLKNUM ASSETID, --filewholeremote BLKNUM ASSETID
      Show filewholeremote asset
-c BLKNUM ASSETID, --commandasset BLKNUM ASSETID
      Show command asset
-rh ASSETID, --dirassethist ASSETID
      Show dir asset history
-fh ASSETID, --fileassethist ASSETID
      Show file asset history
-ch CMDID, --cmdhist CMDID
      Show command asset history
-hc AUTHOR, --hashchanges AUTHOR
      Show where hashes have changed
-dd ASSETID BLKNUM1 BLKNUM2, --dirdiff ASSETID BLKNUM1 BLKNUM2
      Show directory change
-hl, --hashlist
      Extract all unique file hashes
```

Appendix C: Block Data Details

In this appendix more detailed results are presented of the data recorded onto the blockchain with further insights and analysis.

The following command was performed

```
./bcsim.py admin --blockdata 12
```

to view all data contents of block 12 in the Python format as specified by the data model. The output is shown below.

```
----------------------
details Block(12, m4)...

    Assets
    4000  FileWhole(4000, /etc/passwd)
    {'_assetid': '4000', '_assettype': 'FileWhole', '_assetpath':
'/etc/passwd', '_hash':
'156745f668a707cf8e62a94d33346039c5f6e9af8aa293cb6ab77179',
'_filecontent': None}

    Commands
    40000  CmdSshd(40000, ['journalctl', '-o', 'short-unix', '--no-pager',
'-n', '1000', '--quiet', '_SYSTEMD_UNIT=ssh.service', '--since',
'@1651344481.711378'])
    {'_cmdid': '40000', '_cmdtype': 'CmdSshd', '_command': ['journalctl',
'-o', 'short-unix', '--no-pager', '-n', '1000', '--quiet',
'_SYSTEMD_UNIT=ssh.service', '--since', '@1651344481.711378'], 'host':
'm4', 'user': 'root', '_starttime': None, '_endtime': None, '_returncode':
0, '_cmdoutput': ''}
----------------------
```

Details of Filesystem Assets

The following command was performed

```
./bcsim.py admin --dirasset 1 1000
```

to print the content of a Directory asset 1000 in block 1. The result of the above command is shown below.

```
----------------------
.Block(1, m1), dirasset: 1000
Dir path = /usr/share/ca-certificates/mozilla/
Dir hash = dbdb4d6018a2c601bcfa6ad092f8aa24cb810e43ae99e9e13ad65a42
Dir listing = total 528
drwxr-xr-x 2 root root 12288 2022-04-24 14:13 .
drwxr-xr-x 3 root root  4096 2021-10-12 22:37 ..
...
-rw-r--r-- 1 root root  1870 2021-09-22 13:46 TeliaSonera_Root_CA_v1.crt
```

```
-rw-r--r-- 1 root root  1493 2021-09-22 13:46 TrustCor_ECA-1.crt
...
-rw-r--r-- 1 root root  1883 2021-09-22 13:46 TWCA_Global_Root_CA.crt
-rw-r--r-- 1 root root  1269 2021-09-22 13:46
TWCA_Root_Certification_Authority.crt
...
7d85e5fb31de488f52153d76b48387698c9a7a0405aa51a014696c13 /usr/share/ca-
certificates/mozilla/TWCA_Global_Root_CA.crt
1046e9afc294a92baf64c320f37eabb04eed263caf8a993e360c8ee6 /usr/share/ca-
certificates/mozilla/TWCA_Root_Certification_Authority.crt
85699ff3687f90f1e6cfb0e81bf8e6a6da350b26a152f78e37b53c68 /usr/share/ca-
certificates/mozilla/TeliaSonera_Root_CA_v1.crt
0ff10302a18b850e9ef14fe83164a3f7f701b1ae497efed8d1f5a755 /usr/share/ca-
certificates/mozilla/TrustCor_ECA-1.crt
...
----------------------
```

The above result shows the data recorded for Directory asset 1000 which is the directory located at filesystem path /usr/share/ca-certificates/mozilla/. This directory contains cryptographic system certificate files. Filesystem metadata and file hashes were recorded and captured onto the blockchain. This data provides a complete picture of the directory contents. The current version of the simulator always records a directory listing to capture a record of file metadata although, presently, this data cannot be used in the analysis except by inspection.

The following command was performed

```
./bcsim.py admin --dirasset 1 1005
```

to show the state of Directory asset 1005 in two consecutive blocks, 1 and 4. The results for both blocks 1 and 4 are shown below.

```
----------------------
Block(1, m1), dirasset: 1005
Dir path = files/
Dir hash = 6412b89852777faf9500982b6e87478a77430b8786439249dff464a1
Dir listing = total 1620
drwxrwx--- 2 bcadmin bcadmin   4096 2022-04-30 20:17 .
drwxrwx--- 6 bcadmin bcadmin   4096 2022-04-30 20:18 ..
-rw-r--r-- 1 root    root      1445 2022-04-30 20:17 certlog
-rw-r--r-- 1 bcadmin bcadmin  76657 2022-04-22 23:10 confidential.jpg
...
-rw-r--r-- 1 root    root      2247 2022-04-30 20:17 ntfslog
-rw-r--r-- 1 bcadmin bcadmin 103571 2022-04-30 20:17 secret
-rw-r--r-- 1 bcadmin bcadmin  65873 2022-04-22 23:10 secret.jpg

edc2be52470d5341885df13aa97866387259b2ccff8732c769c6ee44 files/certlog
c7285c6865152308d57da7f63ad2066a2794e72d2c6f9df700a86a3b
files/confidential.jpg
...
f7b80cc588865cb442017f5b445481c6869936d3aa5c55332cbd7dea files/secret
```

```
0035baab6983798bfd1aa5c21301a96869b2b1ceba72b3b486d785aa files/secret.jpg
----------------------

----------------------
Block(4, m1), dirasset: 1005
Dir path = files/
Dir hash = 6412b89852777faf9500982b6e87478a77430b8786439249dff464a1
Dir listing = total 1620
drwxrwx--- 2 bcadmin bcadmin   4096 2022-04-30 20:17 .
drwxrwx--- 6 bcadmin bcadmin   4096 2022-04-30 20:23 ..
-rw-r--r-- 1 root    root      1445 2022-04-30 20:17 certlog
-rw-r--r-- 1 bcadmin bcadmin  76657 2022-04-22 23:10 confidential.jpg
...
-rw-r--r-- 1 root    root      2247 2022-04-30 20:17 ntfslog
-rw-r--r-- 1 bcadmin bcadmin 103571 2022-04-30 20:17 secret
-rw-r--r-- 1 bcadmin bcadmin  65873 2022-04-22 23:10 secret.jpg
----------------------
```

The above results show that because the directory hash (computed over the whole directory) has not changed from block 1 to block 4, the file hashes need not be recorded. Only the directory listing containing the file metadata was recorded.

The following command was performed

```
./bcsim.py admin --fileasset 1 1001
```

to print the data content of asset 1001 in block 1. The output is shown below.

```
----------------------
Block(1, m1), fileasset: 1001
assetpath = documents/reactorspec.odt
hash = 65077421d7d54f8df01890a56d8318ddba261bfdcfc7b8edf1223927
----------------------
```

The above result is the recorded hash for the file asset, in this case, a document file, reactorspec.odt. This is an example of an asset type where we have chosen to record only the hash of the file, not the file content. The reason for this choice is that although this document is an important document and we would like to know what changed, the file is a binary file. The blockchain programme does not have the functionality to determine changes in a binary file. If we chose to record the file content, other tools would be needed to determine the changes in file content.

The following command was performed

```
./bcsim.py admin --filewholeasset 1 1002
```

to print the data content of asset 1002 in block 1. The output is shown below.

```
----------------------
Block(1, m1), fileasset: 1002
assetpath = /etc/cron.monthly/awsiplist
hash = 91a968522f5f7f06369db789c9114f35e1f0dd540e21a4cc3b8eb230
filecontent:
wqet https://ip-ranges.amazonaws.com/ip-ranges.json
----------------------
```

The above file, awsiplist, is a cronjob file and contains a line of text which the host system will execute as a command at a time specified by the cronjob schedule. The content of the recorded file consists of a single command listed under "filecontent".

The following command was performed

```
./bcsim.py admin --filewholeasset 1 1003
```

to print the data content of asset 1003 in block 1. The result is shown below.

```
----------------------
Block(1, m1), fileasset: 1003
assetpath = documents/supplierdata
hash = 984cc9075cb8fef29bc21cefb32a97063c398af23f269112d76c5ecb
filecontent:
Supplier 1 data:
----------------
SupplierID: X509404
Name: J.Doe, Inc.
Contract: C2105
Payment-limit: USD20000
Bank: Fargo Wild West
Account Number: 239840213

Supplier 2 data:
----------------
SupplierID: X455606
Name: Wright Brothers, Inc.
Contract: B747
Payment-limit: USD300000
Bank: Union Bank
Account Number: 49088214
----------------------
```

The above file is a simplified representation of a file containing financial data.

The following command was performed

```
./bcsim.py admin --filewholeasset 1 1004
```

to print the data content of asset 1004 in block 1. The result is shown below.

```
----------------------
Block(1, m1), fileasset: 1004
command = ['scp', '-q', 'm4:/etc/hosts', 'aws_hosts.temp']
hash = 5f061e393e59349bee1791a7021d4a385a734d510100076f7c580a35
---------- filecontent ---------:
127.0.0.1 localhost
...
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
----------------------
```

The above file is a "hosts" file located on machine m4, but monitored remotely by agent machine, m1, under assetid 1004. This file contains ip address to hostname mappings used by the operating system on m4.

The following command was performed

```
./bcsim.py admin --filewholeasset 3 4000
```

to print the data content of asset 4000 in block 3. The result is shown below.

```
----------------------
Block(3, m4), fileasset: 4000
assetpath = /etc/passwd
hash = 156745f668a707cf8e62a94d33346039c5f6e9af8aa293cb6ab77179
filecontent:
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
...
ec2-instance-connect:x:112:65534::/nonexistent:/usr/sbin/nologin
lxd:x:998:100::/var/snap/lxd/common/lxd:/bin/false
----------------------
```

The above file is the user password database file residing on machine, m4. This file lists authorised users on that system and is important for the security of the machine. The choice to record the contents of this file can be made as part of a process of identifying valuable digital assets of a supply chain.

Details of Command Type Assets

This sub-section presents results from command type assets that were captured onto the blockchain.

The following command

```
./bcsim.py admin --commandasset 1 10000
```

shows the data recorded for command asset 10000 in block 1. The result is shown below.

```
----------------------
Block(1, m1), commandasset: 10000
CmdLast(10000, ['last', '--ip'])
command output:
------------------------------
bcadmin  tty7         0.0.0.0          Tue Apr 19 11:15    gone - no
logout
reboot   system boot  0.0.0.0          Tue Apr 19 11:15   still running
bcadmin  tty7         0.0.0.0          Tue Apr 19 10:07 - crash  (01:08)
reboot   system boot  0.0.0.0          Tue Apr 19 10:07   still running
bcadmin  tty7         0.0.0.0          Mon Apr 18 20:07 - crash  (14:00)
reboot   system boot  0.0.0.0          Mon Apr 18 20:07   still running
bcadmin  tty7         0.0.0.0          Mon Apr 18 20:05 - 20:06  (00:00)
reboot   system boot  0.0.0.0          Mon Apr 18 17:36 - 20:06  (02:29)
bcadmin  tty7         0.0.0.0          Mon Apr 18 17:33 - 20:05  (02:32)
reboot   system boot  0.0.0.0          Mon Apr 18 17:33 - 20:05  (02:32)
bcadmin  tty7         0.0.0.0          Sat Apr 16 20:10 - 20:37  (00:26)
reboot   system boot  0.0.0.0          Sat Apr 16 20:10 - 20:37  (00:27)
bcadmin  pts/1        10.9.0.1         Fri Apr 15 20:16 - 20:16  (00:00)
bcadmin  pts/1        10.9.0.1         Fri Apr 15 20:13 - 20:15  (00:01)
...
----------------------
```

The above command asset executed the system "last" command which reports the history of user logins on machine m1. The output of that command was captured in block 1 of the blockchain. This command would be useful in a scenario where the agent machine is monitoring a remote multi-user computer. In that case, the record of user logon activity can be valuable from a security monitoring and digital forensic perspective.

The following command

```
./bcsim.py admin --commandasset 1 10001
```

shows the data recorded for command asset 10001 in block 1. The result is shown below.

```
----------------------
Block(1, m1), commandasset: 10001
CmdGeneric(10001, echo "$PATH")
command output:
------------------------------
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/u
sr/local/games:/snap/bin
```

```
----------------------
```

The above command asset executed a system command to obtain the value of the system environment variable, PATH. This variable is important from a security perspective because malicious modification can subvert a system's security by causing invocation of executables to make use of executable files in non-standard locations.

The following command

```
./bcsim.py admin --commandasset 1 10002
```

shows the data recorded for command asset 10002 in block 1. The result is shown below.

```
----------------------
Block(1, m1), commandasset: 10002
CmdNetstat(10002, ['netstat', '-tupan'])
command output:
-----------------------------
Active Internet connections (servers and established)
Proto  Local Address       Foreign Address     State       PID/Program name
tcp    127.0.0.53:53       0.0.0.0:*           LISTEN      346/systemd-resolve
tcp    0.0.0.0:22          0.0.0.0:*           LISTEN      611/sshd: /usr/sbin
tcp    10.10.1.2:38378     65.9.49.115:443     ESTABLISHED 64511/firefox
tcp    10.10.1.2:43102     151.101.66.49:443   ESTABLISHED 64511/firefox
tcp6   :::22               :::*                LISTEN      611/sshd: /usr/sbin
udp    127.0.0.53:53       0.0.0.0:*                       346/systemd-resolve
udp    0.0.0.0:43395       0.0.0.0:*                       767/openvpn
----------------------
```

The above command executed the system netstat command on machine, m1, to capture the state of network connections at the time of block 1 creation. We see some processes listening for incoming connections including the sshd process listening on port 22, and three established connections.

The following command

```
./bcsim.py admin --commandasset 5 20000
```

shows the data recorded for command asset 20000 in block 5. The result is shown below.

```
----------------------
Block(5, m2), commandasset: 20000
CmdNmap(20000, ['nmap', '-Pn', '-n', '-sS', '--top-ports', '4', '--open',
'10.10.3.2'])
command output:
------------------------------
Starting Nmap 7.80 ( https://nmap.org ) at 2022-04-30 20:35 CEST
Nmap scan report for 10.10.3.2
```

```
Host is up (0.00078s latency).
Not shown: 1 closed port
PORT   STATE SERVICE
21/tcp open  ftp
23/tcp open  telnet
80/tcp open  http
MAC Address: 08:00:27:4D:B5:C1 (Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 0.13 seconds
----------------------
```

The above command asset executed the nmap command on agent machine, m2, to probe machine, m3, to discover open ports on m3. This network scanning command can be executed on any agent machine to probe the state of any other machine to which it has network connectivity. The above result shows that three open ports were discovered, ports 21,23,80.

The following command

```
./bcsim.py admin --commandasset 2 20001
```

shows the data recorded for command asset 20001 in block 2. The result is shown below.

```
----------------------
Block(2, m2), commandasset: 20001
CmdGeneric(20001, route -n)
command output:
------------------------------
Kernel IP routing table
Destination   Gateway     Genmask         Flags Metric Ref    Use Iface
0.0.0.0       10.10.2.1   0.0.0.0         UG    100    0        0 enp0s3
10.9.0.0      0.0.0.0     255.255.255.0   U     0      0        0 tun0
10.10.2.0     0.0.0.0     255.255.255.0   U     100    0        0 enp0s3
10.10.3.0     0.0.0.0     255.255.255.0   U     101    0        0 enp0s8
----------------------
```

The above command asset executed a command that took a snapshot of the system's routing table and captured this output onto the blockchain.

The following command

```
./bcsim.py admin --commandasset 2 20002
```

shows the data recorded for command asset 20002 in block 2. The result is shown below.

```
----------------------
Block(2, m2), commandasset: 20002
CmdGeneric(20002, dd if=/dev/sda count=2 | sha224sum | awk '{print $1}')
command output:
```

```
------------------------------
7562164b36b14ea5da583137588fe10726379ab4c2f8d8b48f220328
----------------------
```

The above command calculated and recorded the hash of the hard disk's boot sector, an important component of a computer's security.

The following command

```
./bcsim.py admin --commandasset 5 20003
```

shows the data recorded for command asset 20003 in block 5. The result is shown below.

```
----------------------
Block(5, m2), commandasset: 20003
CmdUfwBlock(20003, ['journalctl', '-o', 'short-unix', '--no-pager', '-n',
'100000', '--quiet', '-g', 'ufw block', '--since', '@1651343103.1204717'])
command output:
------------------------------
1651343684.950113 BLOCK SRC=10.10.3.2 DST=10.10.3.1 PROTO=TCP SPT=34154
DPT=80 WINDOW=5840 RES=0x00 SYN URGP=0
1651343687.937939 BLOCK SRC=10.10.3.2 DST=10.10.3.1 PROTO=TCP SPT=34154
DPT=80 WINDOW=5840 RES=0x00 SYN URGP=0
----------------------
```

The above command captured two firewall block events from the system log. These events represent unauthorised connection attempts that were blocked by the host firewall. We simulated such events by attempting to connect to machine, m2, from machine, m3. The two blocked connection attempts were recorded in block 5.

The following command

```
./bcsim.py admin --commandasset 60 40000
```

shows the data recorded for command asset 40000 in block 60. The result is shown below.

```
----------------------
...
Block(60, m4), commandasset: 40000
CmdSshd(40000, ['journalctl', '-o', 'short-unix', '--no-pager', '-n', '1000', '--
quiet', '_SYSTEMD_UNIT=ssh.service', '--since', '@1651354081.8176472'])
command output:
------------------------------
1651354395 sshd[169010]: Invalid user support from 116.105.77.196 port 35002
1651354406 sshd[169015]: Invalid user ubnt from 116.105.77.196 port 41406
1651354425 sshd[169019]: Invalid user admin from 116.105.212.31 port 46578
1651354434 sshd[169023]: Invalid user ftp from 116.105.212.31 port 59186
1651354434 sshd[169021]: Invalid user guest from 171.251.29.152 port 40696
...
```

```
----------------------
```

The above command output shows unauthorised SSH connection attempts to agent machine, m4, by threat actors or malware infected devices out on the internet. There were repeated unsuccessful connection attempts made to login as invalid users at the times recorded (unix time format).

History of a Directory Asset

We performed various simulator commands to extract useful insights of the history of directory assets on the blockchain. Here, in this section, we present these commands together with the results and some brief comments.

The following command

```
/bcsim.py admin --dirassethist 1000
```

displays the history of directory asset 1000. The result is shown below.

```
----------------------
block 22: Directory(1000, /usr/share/ca-certificates/mozilla/)
dirhash = 11b07b46164c7da934b2bd664c87281930a7a6b1d23cace611d6ee0f
----------------------
block 25: Directory(1000, /usr/share/ca-certificates/mozilla/)
dirhash = 11b07b46164c7da934b2bd664c87281930a7a6b1d23cace611d6ee0f
----------------------
block 28: Directory(1000, /usr/share/ca-certificates/mozilla/)
dirhash = 11b07b46164c7da934b2bd664c87281930a7a6b1d23cace611d6ee0f
----------------------
block 31: Directory(1000, /usr/share/ca-certificates/mozilla/)
dirhash = dbdb4d6018a2c601bcfa6ad092f8aa24cb810e43ae99e9e13ad65a42

89c864e0cae7bca0e5f10b423bc692aaa17e19860e049ee2b7a87894 /usr/share/ca-
certificates/mozilla/ACCVRAIZ1.crt
7418973a7f382f78a9a55190298b0c6ab8643d4346c50ae9a37f73e5 /usr/share/ca-
certificates/mozilla/AC_RAIZ_FNMT-RCM.crt
...
e19e158c21901d3376f8108a51301a14d450e93c9522563dade2bb03 /usr/share/ca-
certificates/mozilla/emSign_Root_CA_-_C1.crt
e91c93632ecb36d9e381e7e443ea388e1e37fe6db2e9117d24adfef3 /usr/share/ca-
certificates/mozilla/emSign_Root_CA_-_G1.crt
----------------------
block 34: Directory(1000, /usr/share/ca-certificates/mozilla/)
dirhash = dbdb4d6018a2c601bcfa6ad092f8aa24cb810e43ae99e9e13ad65a42
----------------------
block 37: Directory(1000, /usr/share/ca-certificates/mozilla/)
dirhash = dbdb4d6018a2c601bcfa6ad092f8aa24cb810e43ae99e9e13ad65a42
----------------------
```

The above result shows that from block 22 to block 28 the hash of the monitored directory did not change and hence, only the hash is recorded, not

the contents. At the time that block 31 was created the hash had changed and therefore the directory contents are also captured in the block 31 data. The directory hash, dirhash, is a hash computed over all the individual file hashes, and is therefore not affected by changes to file metadata. Changes to file metadata only, cannot be detected by file hash computations. For this reason, this blockchain always records a directory listing even if the directory hash is unchanged.

The following command was performed

```
./bcsim.py admin --dirasset 25 1000
```

to show the directory listing of block 25 for the same assetid 1000. An extract of the result is shown below.

```
----------------------
Block(25, m1), dirasset: 1000
Dir path = /usr/share/ca-certificates/mozilla/
Dir hash = 11b07b46164c7da934b2bd664c87281930a7a6b1d23cace611d6ee0f
Dir listing = total 532
drwxr-xr-x 2 root root 12288 2022-04-30 20:23 .
drwxr-xr-x 3 root root  4096 2021-10-12 22:37 ..
-rw-r--r-- 1 root root  2772 2021-09-22 13:46 ACCVRAIZ1.crt
-rw-r--r-- 1 root root  1972 2021-09-22 13:46 AC_RAIZ_FNMT-RCM.crt
-rw-r--r-- 1 root root  2049 2021-09-22 13:46
Actalis_Authentication_Root_CA.crt
...
-rw-r--r-- 1 root root   948 2021-09-22 13:46
USERTrust_ECC_Certification_Authority.crt
-rw-r--r-- 1 root root  2094 2021-09-22 13:46
USERTrust_RSA_Certification_Authority.crt
-rw-r--r-- 1 root root  1700 2021-09-22 13:46
VeriSign_Universal_Root_Certification_Authority.crt
-rw-r--r-- 1 root root  1513 2021-09-22 13:46 XRamp_Global_CA_Root.crt
----------------------
```

The above result shows that metadata such as file owner, permissions, size and timestamp have been recorded. Currently, the blockchain simulator only records metadata but does not analyse the file metadata. This functionality is proposed as an extension of the work.

History of a File Asset

We performed various simulator commands to extract useful insights of the history of file assets on the blockchain. Here, in this section, we present these commands together with the results and some brief comments.

The following command

```
./bcsim.py admin --fileassethist 1001
```

was performed to show the history of file asset 1001 which is monitored by agent machine, m1. The result is shown below.

```
----------------------
Extracting history of file asset 1001
----------------------
block 1: FileHashOnly(1001, documents/reactorspec.odt)
65077421d7d54f8df01890a56d8318ddba261bfdcfc7b8edf1223927
----------------------
block 4: FileHashOnly(1001, documents/reactorspec.odt)
65077421d7d54f8df01890a56d8318ddba261bfdcfc7b8edf1223927
----------------------
block 7: FileHashOnly(1001, documents/reactorspec.odt)
32fac8063429120800d93ab3672817d98ea932dd98b2f5c4add0de7f
----------------------
block 10: FileHashOnly(1001, documents/reactorspec.odt)
32fac8063429120800d93ab3672817d98ea932dd98b2f5c4add0de7f
----------------------
```

The above result shows that the file reactorspec.odt was modified sometime between the creation times of blocks 4 and 7 resulting in a changed hash being recorded in block 7. Since the file reactorspec.odt was defined as a digital asset for which we only record the file hash and not the contents, this means that we cannot know what changed, only that the file has changed.

The following command

```
./bcsim.py admin --fileassethist 2006
```

was performed to show the history of file asset 2006 which is monitored by agent machine, m2. The result is shown below.

```
----------------------
Extracting history of file asset 2006
----------------------
block 2: FileWhole(2006, syslogs/invoicelog)
4937463feb13c338adb12932f4612dd4ef23b4a40e71a7888ffcb4df
Invoice Number,Billed By,Date of Issue,Invoice amount
856475-023,SuperIT,2013-02-02 09:00:08,"11000,50"
237865-001,Vending4All,2013-02-03 15:00:15,"56000,00"
779341-030,Pallets Inc,2013-02-11 16:00:09,"13459,50"
847923-020,HVAC Inc,2013-02-13 08:30:45,"34852,80"
----------------------
block 5: FileWhole(2006, syslogs/invoicelog)
4937463feb13c338adb12932f4612dd4ef23b4a40e71a7888ffcb4df
None
----------------------
block 8: FileWhole(2006, syslogs/invoicelog)
4937463feb13c338adb12932f4612dd4ef23b4a40e71a7888ffcb4df
None
```

```
----------------------
...
----------------------
block 17: FileWhole(2006, syslogs/invoicelog)
4937463feb13c338adb12932f4612dd4ef23b4a40e71a7888ffcb4df
None
----------------------
block 20: FileWhole(2006, syslogs/invoicelog)
4937463feb13c338adb12932f4612dd4ef23b4a40e71a7888ffcb4df
None
----------------------
block 23: FileWhole(2006, syslogs/invoicelog)
e101b8d7aad9d1c0ac14e94dd6a228a94099e3b3ae32ed308d1cd8c8
Invoice Number,Billed By,Date of Issue,Invoice amount
856475-023,SuperIT,2013-02-02 09:00:08,"11000,50"
237865-001,Vending4All,2013-02-03 15:00:15,"56000,00"
779341-030,Pallets Inc,2013-02-11 16:00:09,"13459,50"
847923-020,HVAC Inc,2013-02-13 08:30:45,"34852,80"
562390-030,Pallets Inc,2013-02-13 08:30:45,"41903,90"
648975-023,SuperIT,2013-02-18 11:40:45,"22000,00"
330964-030,Pallets Inc,2013-03-04 09:00:15,"68000,50"
558210-030,Pallets Inc,2013-03-10 08:00:08,"43519,90"
369012-020,HVAC Inc,2013-03-11 18:00:15,"12560,30"
562309-020,HVAC Inc,2013-03-15 10:10:05,"70890,00"
----------------------
block 26: FileWhole(2006, syslogs/invoicelog)
e101b8d7aad9d1c0ac14e94dd6a228a94099e3b3ae32ed308d1cd8c8
None
----------------------
```

The above result shows a file with invoicing data which changed at block 2 and again at block 23. This file was defined as an asset type for which we record the file contents in addition to the hash whenever the hash changes. Thus the file contents can be inspected for blocks 2 and 23.

History of a Command Asset

We performed various simulator commands to extract useful insights of the history of command type assets on the blockchain. Here, in this section, we present the results and some brief comments.

The following command

```
./bcsim.py admin --cmdhist 40000
```

was performed to show the history of command asset 40000 which monitors system logs on machine, m4. The result is shown below.

```
----------------------
block 81: CmdSshd(40000, ['journalctl', '-o', 'short-unix', '--no-pager',
'-n', '1000', '--quiet', '_SYSTEMD_UNIT=ssh.service', '--since',
'@1651358282.1396976'])
```

```
----------------------
block 84: CmdSshd(40000, ['journalctl', '-o', 'short-unix', '--no-pager',
'-n', '1000', '--quiet', '_SYSTEMD_UNIT=ssh.service', '--since',
'@1651358881.4223733'])
----------------------
block 87: CmdSshd(40000, ['journalctl', '-o', 'short-unix', '--no-pager',
'-n', '1000', '--quiet', '_SYSTEMD_UNIT=ssh.service', '--since',
'@1651359482.0950484'])

1651359839 sshd[174659]: Invalid user pi from 46.160.140.238 port 58258
1651359839 sshd[174661]: Invalid user pi from 46.160.140.238 port 58266
----------------------
block 90: CmdSshd(40000, ['journalctl', '-o', 'short-unix', '--no-pager',
'-n', '1000', '--quiet', '_SYSTEMD_UNIT=ssh.service', '--since',
'@1651360081.3415139'])

1651360342 sshd[175260]: error: kex_exchange_identification: client sent
invalid protocol identifier "MGLNDD_13.48.123.149_22"
----------------------
block 93: CmdSshd(40000, ['journalctl', '-o', 'short-unix', '--no-pager',
'-n', '1000', '--quiet', '_SYSTEMD_UNIT=ssh.service', '--since',
'@1651360681.6707304'])
----------------------
```

The above result reveals a few unauthorised connection attempts to machine m4 which has an SSH server listening on a publicly accessible port. Block 87 records that two events occurred since the time that block 84 (the previous block with this asset) was created. Block 90 recorded 1 event since block 87 was written.

The following command

```
./bcsim.py admin --cmdhist 20000
```

was performed to show the history of command asset 20000. This is used by agent machine, m2, to probe machine, m3, to discover any open ports on m3. The result is shown below.

```
...
----------------------
block 5: CmdNmap(20000, ['nmap', '-Pn', '-n', '-sS', '--top-ports', '4',
'--open', '10.10.3.2'])

Starting Nmap 7.80 ( https://nmap.org ) at 2022-04-30 20:35 CEST
Nmap scan report for 10.10.3.2
Host is up (0.00078s latency).
Not shown: 1 closed port
PORT   STATE SERVICE
21/tcp open  ftp
23/tcp open  telnet
80/tcp open  http
MAC Address: 08:00:27:4D:B5:C1 (Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 0.13 seconds
```

```
----------------------
block 8: CmdNmap(20000, ['nmap', '-Pn', '-n', '-sS', '--top-ports', '137',
'--open', '10.10.3.2'])

Starting Nmap 7.80 ( https://nmap.org ) at 2022-04-30 20:45 CEST
Nmap scan report for 10.10.3.2
Host is up (0.00021s latency).
Not shown: 119 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
...
MAC Address: 08:00:27:4D:B5:C1 (Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 0.18 seconds
----------------------
...
```

The above result shows block 5 detected three open ports (21,23,80) at the ip address assigned to machine m3. At block 8 there are many more open ports and thus the networking state of machine, m3, has changed from block 5 to block 8. This is an example of monitoring and recording the dynamic state of a computer. It illustrates how vulnerability scanning can be conducted under the auspices of a blockchain system with the results being recorded onto the blockchain for later analysis.

The following command

```
./bcsim.py admin --dirdiff 1000 31 55
```

was performed to show the change in directory asset 1000 from block 31 to block 55. The result is shown below.

```
----------------------
Directory asset 1000 changed from block 31 to block 55
----------------------
+ ('31005d3d3ecee76f2ef32775d227c0afbb59f7cb9944e05c0f8e3a10',
'/usr/share/ca-certificates/mozilla/trustme-certificate-authority.crt')
----------------------
```

The above result reveals that directory assetid 1000 changed because a new file was added (shown by leading "+") to the directory. The path of the new file is given. It appears that a new certificate file, trustme-certificate-authority.crt, was added to the system root certificate store.

The following command

```
./bcsim.py admin --dirdiff 1005 19 22
```

was performed to show the change in directory asset 1005 from block 19 to block 22. The result is shown below.

```
----------------------
Directory asset 1005 changed from block 19 to block 22
----------------------
- ('5c42e9b6c7bc5dba71f2584e452b9a686950f7fccaca8d6af17552d6',
'files/ntfslog')
----------------------
```

The above result shows that directory asset 1005 changed from block 19 to block 22 because the file ntfslog was deleted (shown by leading "-") from the directory.

The following command

```
./bcsim.py admin --dirdiff 1005 25 46
```

was performed to show the change in directory asset 1005 from block 25 to block 46. The result is shown below.

```
----------------------
Directory asset 1005 changed from block 25 to block 46
----------------------
- ('edc2be52470d5341885df13aa97866387259b2ccff8732c769c6ee44',
'files/.temp')
+ ('edc2be52470d5341885df13aa97866387259b2ccff8732c769c6ee44',
'files/certlog')
----------------------
```

The above result shows that a file .temp was deleted from the directory and a file certlog was added. The deletion and addition are indicated by a leading "-" and "+", respectively. Since the hashes of these two files are identical the change amounts to a renaming of file .temp to certlog.

The following command

```
./bcsim.py admin --dirdiff 1005 199 202
```

was performed to show the change in directory asset 1005 from block 199 to block 202. The result is shown below.

```
----------------------
Directory asset 1005 changed from block 199 to block 202
```

```
----------------------
- ('edc2be52470d5341885df13aa97866387259b2ccff8732c769c6ee44',
'files/ntfslog')
+ ('5c42e9b6c7bc5dba71f2584e452b9a686950f7fccaca8d6af17552d6',
'files/ntfslog')
----------------------
```

The above result shows that the hash of a file changed but the filename did not. This means that the same filename is now a reference to different content. The file content was changed.

Appendix D: Sample PKI Certificate

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            9f:c8:0b:bd:78:10:18:f5:4b:45:27:34:dd:fb:75:37
        Signature Algorithm: ED25519
        Issuer: C = SE, L = Halmstad, O = Ari_Rekha_Team,
                OU = Management, CN = bcsim CA 2022
        Validity
            Not Before: Apr 11 14:00:48 2022 GMT
            Not After : Jul 14 14:00:48 2024 GMT
        Subject: C = SE, L = Halmstad, O = Ari_Rekha_Team,
                 OU = Management, CN = bcsim_client_m1_2022
        Subject Public Key Info:
            Public Key Algorithm: ED25519
                ED25519 Public-Key:
                pub:
                    fe:1d:4e:3e:72:7b:38:88:e8:e8:39:cb:02:
                    85:ec:ff:d8:d2:cf:5f:4d:f7:99:96:64:61:
                    90:53:33:0b:f9:78
        X509v3 extensions:
            X509v3 Authority Key Identifier:
                keyid:C5:F6:6D:21:4F:05:15:74:31:95:A8:31:
                76:37:60:D8:31:2F:BE:93
            X509v3 Basic Constraints: critical
                CA:FALSE
            X509v3 Key Usage:
                Digital Signature
            X509v3 Extended Key Usage:
                TLS Web Client Authentication
            X509v3 Subject Key Identifier:
                99:8C:7E:28:37:D9:06:F2:2A:24:F1:98:CE:D2:
                B1:4B:0B:1C:FA:2F
    Signature Algorithm: ED25519
        ce:83:52:ba:02:f1:97:1f:f9:66:ae:62:49:05:7d:ec:9d:95:
        12:56:1f:0d:4b:ec:e7:6e:b8:ad:2a:3d:0f:8c:55:76:fb:03:
        e7:6b:dd:24:e1:ea:14:85:49:12:cd:8e:d4:25:50:45:d5:3e:
        57:3d:84:6e:83:e0:2e:e4:83:04
```

Appendix E: Simulation Source Code

The simulation source code and associated files may be downloaded by following the link below:

https://github.com/totaldefence/forensic-blockchain