Traceability in Supply Chains: A Cyber Security Analysis

Naeem Firdous Syed, Syed W. Shah, Rolando Trujillo-Rasua, and Robin Doss
Deakin University, Geelong, Australia
Centre for Cyber Security Research and Innovation (CSRI)
{naeem.syed, syed.shah, rolando.trujillo, robin.doss}@deakin.edu.au

Abstract

Digital technologies are increasingly adopted in modern supply chains for product traceability, enabling data sharing amongst trading partners, quick availability of product data, and end-to-end visibility of products. This adoption increases the system attack-surface and the number of cyber threats capable of harmful business impact, such as leak of business data, disruption of business operations, and lost of reputation, intellectual property and financial assets. A supply chain network thus needs an effective cybersecurity and threat management strategy, which requires reaching a thorough understanding of the most important assets and resources in a supply chain traceability system, the cyber threats that may impact them, and potential countermeasures.

This article contributes a comprehensive threat modeling report on supply chain traceability systems, where we make explicit more than a hundred relations between assets, threats and countermeasures of relevance to supply chain traceability. Our analysis is reproducible, extensible and falsifiable. Reproducibility is achieved by following a systematic asset-centric threat modeling approach and adopting the STRIDE threat model to present a description of common threats; extensibility by using a layered-architecture for supply chains which the analyst can accommodate to a concrete implementation; and falsifiability by providing the sources used to establish the relation (asset, threat, countermeasure). Albeit the focus of the analysis is on technology, for the sake of completeness, the article briefly analyses secure traceability in supply chains when people and processes are made part of the system, resulting in a generic list of recommendations and best practices.

1. Introduction

Traceability has emerged as a cornerstone in supply chain management and is increasingly mandated by governments across the globe as a means of ensuring product safety and consumer protection. It enables product related information to be accessible throughout its entire lifetime by leveraging Information and Communications Technology (ICT) [1], allowing companies to track the movement of a product at least one step back and one step forward at any point in the supply chain. A traceability system should be capable of identifying the sources of all inputs, such as raw materials, additives, and packaging. The origins of raw materials and the various transformations that a product undergoes need to be tracked and traced from source to destination, making it possible for different actors in a supply chain to access and identify a product's source, transit point(s), and destination.

Historically, traceability systems depended on paper-based recording, providing a cheap, albeit lengthy, tedious, and error-prone, alternative to ICT based traceability systems [2]. In contrast, ICT solutions are capable of performing on-the-fly updates of product information and instantaneous communication of those changes to trading partners and consumers alike. This brings significant advantages in terms of competitiveness, profitability, and transparency. Moreover, emerging ICT technologies for low-powered devices, such as Radio Frequency Identification (RFID) and Internet of Things (IoT), have opened the door to further improve the way products are tracked across a supply chain [3–5].

The downside of introducing computer systems is that they are hackable [4, 6, 7]; understanding to what extent is the goal of this article. We do so by performing the first threat analysis of a generic traceability system for supply chains. Although various cyber security analysis on traceability systems [8–11] exist, they mainly focus on the risk of data sharing and the use of Blockchain technology to mitigate risks. We, instead, advocate for a comprehensive and systematic end-to-end analysis that assesses threats operating at different levels of the traceability system, from data capture (e.g., barcodes or RFID tags) through to end-user application (e.g., a mobile app). We argue for an end-to-end threat analysis given the complexity of supply chain systems, which involve multiple ICT components and impact many stakeholders. Vulnerabilities can arise due to complex interconnection, and are often missed when analysed in isolation, as in previous studies. We also advocate for security analyses that are reproducible, extensible and falsifiable. This can be achieved by following a systematic threat modelling approach, committing to a generic system architecture, and making explicit and easily identifiable the relation between assets, threats, and countermeasures.

Contributions. This paper presents a detailed asset-centric threat analysis of supply chain traceability. This is achieved by first defining a traceability architecture, largely based on the GS1 traceability standards [12] and the architecture presented in [2]. The proposed architecture consists of four layers, each detailing how traceability data is captured, carried, shared, and consumed by end-users. The architecture describes the main assets used at each layer for data collection, storage, manipulation and sharing. This makes our analysis extensible by considering additional layers or assets.

To systemically identify system vulnerabilities, we adopt the popular Microsoft's STRIDE threat model, which offers a methodology to categorise vulnerabilities based on their contribution to a broad list of threats including including Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, and Elevation of privileges. This means that our analysis can be navigated through two dimensions, by focusing on a specific layer and reading its full analysis report, or by understanding how a specific threat, such as tampering, impacts the system as a whole. The vulnerabilities of individual system components were identified by a comprehensive review of the scientific literature and vulnerability databases. Visually, we display the relation between assets, threats and countermeasures in a tabular form and categorise them by asset types. The result is a collection of more than a hundred (asset, threat, countermeasure) relations which, to the best of our knowledge, form the most comprehensive threat modeling report on supply chain traceability.

Acknowledging the role of people and processes in traceability systems, we briefly discuss their roles, additional threats they bring, and best practices and mitigation strategies. This aids the understanding of weaknesses at different levels of a traceability system and helps stakeholders in selecting the most appropriate security controls.

Organisation. The rest of this article is organized as follows. Section 2 reviews previous threat modeling efforts on supply chain traceability systems. Section 3 describes the threat modelling process that we adopt for our analysis and introduces a layered-architecture for supply chain traceability systems. Sections 4, 5, 6 and 7 report on our analysis results, each section dedicated to a system layer. Section 8 wrap ups the information provided in earlier sections and lists best security practices. A short note of concluding remarks is given in Section 9.

2. Related Work

Various research articles have considered data tampering and information disclosure in traceability systems. For example, Lin et al., [8] propose a traceability system that utilizes blockchain and Electronic Product Code Information Services (EPCIS) and leverage enterprise-level smart contracts to counter data tampering and information leakage shared amongst partners. Likewise, Lu et al., [10] and Tian [9] introduce a blockchain solution focused on addressing information falsification and problems stemming from the centralized nature of traditional traceability systems. In [9], it is shown how blockchain and RFID technology can contribute to food safety by collecting and sharing accurate data amongst different units of the supply chain, such as production, processing, warehousing, distribution and selling points [11]. We observe that there exists no security analysis for the blockchain solutions described above.

One of the earliest steps on the direction of providing traceability systems with a threat analysis was given by Urciuoli et al. [13], who surveyed cyber security threats, attack patterns and motivations with the potential to impact supply chain systems. Impact to society was also included in their analysis, such as the possibility of trafficking with weapons in pre-cleared containers of reputed logistics companies, which may not be subjected to heavy inspections by the relevant authorities. Albeit informative, the exact relation between threats, vulnerabilities and assets is missing in their work, which is essential for organisations to conduct risk analysis. Recently, Yeboah-Ofori and Islam in [7] proposed the use of Structured Threat Information Expression (STIX) to perform threat analysis. STIX is a threat information expression language used for exchanging threat intelligence between supply chain partners. Like the work by Urciuoli et al. [13], this analysis neglects the relation between assets, threats and vulnerabilities.

A study conducted in [14] presents a generic description of cyber risks to supply chains and lists best practices for protecting against potential threats. Similarly, Pandey et al. [15] presented a conceptual framework of cyber security risks in global supply chains. The authors identified sixteen cyber security risks categorised into supply, operational and demand risks. Their analysis is based on case studies, rather than on a system architecture as we do in this article. That is, this type of analysis does not follow a systematic threat modelling methodology and is not extendable and adaptable to other architectures.

Other analysis reports on supply chain traceability are narrower in scope in comparison to ours. For example, Divyan et al. in [16] assesses the security threats to the EPCGlobal standard, as defined in 2006. A study conducted in [17] evaluated a framework that models the security threats of software supply chain of Swedish armed forces. A similar study is conducted in [18], where authors point out the counterfeiting problems and their potential countermeasures in global semiconductor supply chains. Lastly, authors in [19] analysed vulnerabilities of barcodes and RFID. All these analyses are valuable, and some of their results are reused in our analysis, but they are clearly limited in scope.

This summary of the literature suggests that previous works mostly aim at either solving the problems associated with tampering of traceability data or identifying security threats to a particular component of the traceability system. This means that a systematic analysis of threats is missing. Because many threats emerge with the interconnection of various components, and they may remain hidden when analysing individual components in isolation, we argue for a systematic analysis that comprehensively identifies the vulnerabilities of the entire end-to-end traceability system. This article provides the first of such analysis.

Systematic threat analysis can only be performed by following a well-defined threat modelling methodology. There exist several such methodologies, such as Microsoft's STRIDE [20], Open Web Application Security Project (OWASP) [21], Process for Attack simulation and Threat modelling (PASTA) [22], Operationally Critical Threat, Asset and Vulnerability Evaluation (OCTAVE) [23], Attach Trees [24], amongst others. They all help to evaluate the security posture of a system and help organizations in preparing defence strategies for safeguarding their systems from various threats. In this work, we choose the STRIDE model due to its simplicity and broad threat categories, which are applicable to various systems [25]. A description of how we implemented threat modelling and used STRIDE for the security analysis of traceability in supply chains is given next.

3. Threat Modelling Process

Threat modelling is a proactive cyber security approach to secure computer systems from potential breaches. It is an iterative process which can be performed before the system is developed as well as throughout the system life-cycle to accommodate potential changes [26]. The threat modelling process we have adopted in this work, based upon the framework presented in [27], is presented in Figure 1. It comprises three main steps, system modelling, threat identification, and threat analysis. Next we detail how we approach each of those steps.

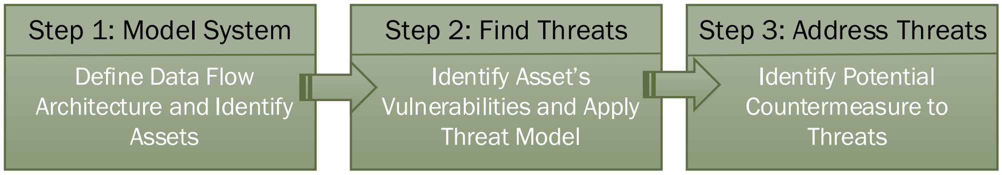
Step 1: Model System
Define Data Flow Architecture and Identify Assets
Step 2: Find Threats
Identify Asset's Vulnerabilities and Apply Threat Model
Step 3: Address Threats
Identify Potential Countermeasure to Threats

Figure 1: Threat modelling steps followed in identifying threats and their countermeasures.

3.1. Traceability System modelling

Modelling the system under analysis is essential to identify the system assets and the interaction between the system components [28, 29]. We define a generic data flow architecture for supply chains that splits the system into four abstraction layers, and identifies the key assets of the system at each layer. The system we describe identifies individual units (i.e., products/items) that are to be traced, and records the transformations those units may have gone throughout their life-cycle [5]. A change of a product's state triggers an event, which contains the product attributes, involved actors (e.g., manufacture, distributor etc), and the geospatial location of the product [30].

Figure 2 shows our *four layered data-flow architecture* for a supply chain's traceability system. This is not a new architecture, but an abstraction from existing traceability architectures, such as [2, 31–34], and the GS1 traceability standards [12]. Our architecture generalises the four layer architecture by Appelhanz et.al., [2], which focuses on wood furniture supply chains. We simplified the generic traceability architecture by Wang and Liu in [33], from five layers to four logical layers. And, we leveraged the GS1 standards to describe the common method(s) and language(s) for data encoding, reading, and exchanging traceability information amongst stakeholders.

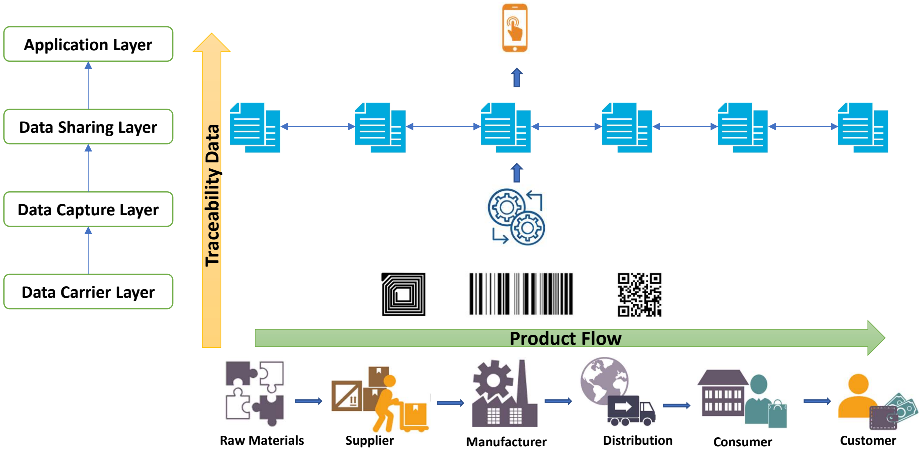

Figure 2: A four-layered generic traceability architecture.

Generally, the entity that is to be traced is referred to as the Traceable Resource Unit (TRU) [5]. Because the TRU is used to track the product as it moves along the supply chain, it must be uniquely identifiable with an identification code that contains sufficient product-related information [2]. Some of the most common examples of identifiers used in supply chain for holding the TRU's information are barcodes and RFID tags [30]. The product's data generally contains information related to the source of raw materials, the compositions of products, and their shelf life. Therefore, the traceability process begins by labelling products (or units) so as to identify each unit that enters the supply chain. This forms the *data carrier layer*, where data related to the product is carried using the various identifiers. Components that enable capturing the product data from physical identifiers form the *data capture layer*. Examples of these components include data readers and middleware applications. The *data sharing layer* is then responsible for the manipulation and exchange of product identifying information and system events amongst stakeholders. This layer allows different partners to create, exchange and store events related to the movement of products [35]. For example, when a product undergoes a transformation, such as splitting or joining [5], an event is created and shared for end-to-end traceability. Finally, the *application layer* establishes the communication protocols and interfaces that different applications can be used to implement their business cases. We note that these layers may be present in individual trading partners operating in both downstream and upstream direction.

Figure 2 is useful to understand our system architecture at a high level, but does not specify the main assets and components of the system. To identify those, we rely on the GS1 traceability standards, which are globally recognised and widely used for product identification and event tracking. The result is depicted in Figure 3, providing a more fine-grained view of our system architecture. Under the *data carrier layer*, we primarily consider three widely adopted data carriers or product identification standards that include barcodes, RFID tags, and IoT devices (e.g., various sensors such as temperature and humidity sensors). In the *data capture layer*, components such as barcode and RFID readers, air interfaces, and middleware applications that facilitate the recording and transformation of traceability information are regarded as relevant assets. The *data sharing layer* includes the central repository of master data of products, event data repository, and other technologies used for sharing information amongst trading partners in upstream or downstream (e.g., EPCIS capture and query interfaces, GS1 XML, EDI, Applicability Statement2 (AS2)/AS4). Similarly, the *application layer* considers any application that accesses traceability data for tracking and analysis purposes as an asset. In addition, B2B applications can directly communicate with trading partners using the EDI and customer facing applications will have access to public data related to the products.

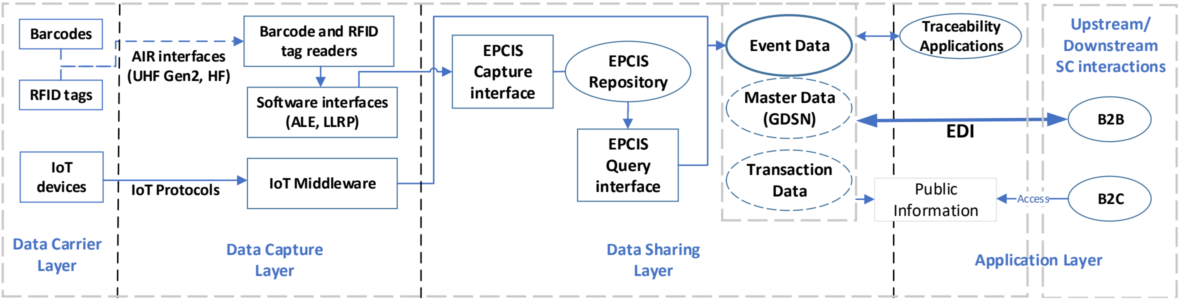

Figure 3: A traceability data flow architecture based on the GS1 standard. SC - Supply Chain

In this work we have adopted the terminology used by the following GS1 standards: GS1 Identification Numbers, GS1 Data Carriers [36], and GS1 Data Exchange [37]. For example, the standards for encoding different information (e.g., global trade item numbers (GTIN), global location numbers (GLN), and others), are generally used for carrying the product related information (e.g., barcodes, RFID). The movement and transformation of products results in Critical Tracking Events (CTEs) which record the various events or business processes that occur in the object's lifecycle [38]. Individual events within the CTEs are described using the Key Data Elements (KDEs), which capture the 'who', 'what', 'where', 'when', and 'why' of the event. This product relevant tracking data is exchanged among the trading partners to enable end-to-end supply chain traceability. The GS1 Data Exchange standard describes how product information can be exchanged amongst trading partners. It is independent of standards used in lower layers and thus provide a way to exchange information irrespective of the methods used for product identification and information capturing.

3.2. Threat identification in Traceability Systems

In order to secure traceability systems from cyber threats, it is essential to identify the potential threats to the system. This, along with implementable prevention strategies, can ensure all entities of the system remain protected from cybersecurity attacks. To accomplish this, we begin by identifying vulnerabilities of assets at each layer of our data flow architecture by consulting the relevant literature. Then, we map these vulnerabilities to potential threats by adopting the STRIDE model, which consists of the following six threat types:

- ***Spoofing***: These threats aim to subvert the authentication mechanism of the system by using fake or cloned credentials.
- ***Tampering***: These attacks target the various components of the system where the ICT components and data stored in them are tampered with to cause adverse impact to the system.
- ***Repudiation***: These attacks target the system's vulnerability in logging and tracing activities to prevent detection and identification of malicious activities.
- ***Information Disclosure***: These threats aim to access unauthorized information from the system and disclose to unauthorized entities.
- ***Denial of Service***: These attacks target the availability of the system in order to deny service to valid entities.
- ***Elevation of Privileges***: This occurs when an unprivileged user or adversary gains unauthorized access to the system and performs actions that they are not authorised to do.

The remainder of this article is dedicated to conduct the threat analysis methodology described above. We proceed by analysing each layer of the supply chain architecture independently. For each layer, to which we dedicate an entire section, we identify the relevant assets and, by conducting a comprehensive literature review, pinpoint vulnerabilities and mitigation strategies. This allows us to present, for every asset, a STRIDE table containing relevant threats and corresponding remedies (See Tables 1-11 for details).

4. Data Carrier Layer

This layer focuses on prevalent standards that are used to carry the product related information so as to facilitate their identification and thereby ensure end-to-end visibility of products in supply chains. Generally, this layer encompasses physical objects attached to the traceable product that capture information such as product identification key, its origin and destination, environmental conditions and so on. The technologies used in this layer provide unique identities to the traceable objects and contribute to the *'What'* part of the KDEs. The three main assets in the data capture layer are barcodes, RFID tags, and IoT devices, whose vulnerabilities we analyse next.

4.1. Vulnerabilities of Barcodes

Various barcodes types exist such as EAN/UPC, GS1 DataBar, ITF-14, GS1-128, GS1 DataMatrix, and GS1 QR Codes which can encode a variety of product related information which are elaborated in [39]. However, barcodes can be physically tampered with by either damaging it or replacing it with a malicious barcode [40]. Barcodes can be easily cloned or reprinted to make fake goods look legitimate. In addition, a motivated adversary can embed malicious data into barcodes using methods, such as Advanced Data Formatting [41], or encode a malicious URL in QR codes or 2D barcode to launch a phishing attack or buffer overflow attacks [42]. This allows an attacker to redirect the scanner to an illicit web address to capture sensitive data [43–45]. These malicious sites can in turn inconspicuously install malware on the system by leveraging exploit kits that fingerprint the device and selects the appropriate exploit and malware [46]. It is also possible to integrate one barcode into another (e.g., QR in QR or DataMatrix in QR), and since readers generally conforms to multiple standards, the embedded barcode may be read by the reader which can potentially carry harmful commands or may direct it to malicious sites [47]. Security researchers have demonstrated the possibility of encoding SQL statements in barcodes to attack or fully erase a database system [43] [48]. We refer the reader to [19] for more information on vulnerabilities associated with barcodes along with their potential countermeasures.

Table 1 presents the STRIDE threat analysis of barcodes and elaborates different threats along with potential countermeasures. Note that, due to space limitation, all the tables beyond this point denote spoofing as '*S*', tampering as '*T*', repudiation as '*R*', information disclosure as '*I*', Denial of service as '*D*', and elevation of privileges as '*E*'.

Table 1: STRIDE Threats in Barcodes

	Threats	**Mitigation strategies**
S	Cloning or reprinting barcodes	Use anti-copying and anti-cloning methods [40, 49, 50]. Use special printing material, physical unclonable functions (PUF), digital water marking [51] and high-density black and white blocks for preventing illegal copying of barcodes.
T	Physical tampering, embedding malicious data or attaching unauthorized barcodes.	Use durable materials for manufacturing barcode labels [52]. Implement tamper detection for QR codes [53] by integrating the digital signature of a barcode content in the error correcting area. These countermeasures can be complemented with market vigilance on the part of the brand owner and trading partners.
R	Denying malicious actions due to lack of logging capability.	Repudiation attacks are difficult to defend as barcodes are non-electronic.
I	Leaking sensitive information due to weak encryption.	Use security-enhanced barcodes, such as Secret-function-equipped QR codes (SQRC) which are only accessible by scanners with the correct cryptographic key [54].
D	Buffer overflow attacks by embedding malicious codes.	Input validation, code verification, keep up with bugs reports, apply patches if necessary.
E	SQL injection attacks and QR Code Phishing attacks.	Incorporate security features in scanners or host device to block the execution of malicious commands or loading of malicious URL – e.g., incorporate threat signature library as indicated in [55].

4.2. Vulnerabilities of RFID tags

Electronic Product Code (EPC) / Radio Frequency Identification (RFID) standards make use of Ultra High Frequency (UHF) or High Frequency (HF) for enabling radio frequency identification. EPC enabled RFID is being increasingly adopted by retailers and product owners for a quick and efficient identification, capturing, and sharing of information. This provides many benefits across the supply chain, such as inventory accuracy, simultaneous counting, reduced out-of-stock incidences, and enabling electronic proof of delivery. EPC may contain product information such as identification key, date/place of manufacturing, batch/lot number, and date shipped, and date delivered [39].

There exist several advantages of RFID tags over barcodes, such as faster and accurate scanning of products and larger storage for product information. RFID tags introduce additional vulnerabilities, though. For example, RFID tags can be detached from a genuine product and attached to fake products, known as tag snatching [16]. This can be used to insert fake products to the supply chain by using the identity of a genuine product. They are also prone to counterfeiting [56]. Physical access to RFID tags may enable an adversary to create a replica of a tag by reverse engineering or revealing the confidential information stored in the tag. Spoofing attacks are also possible [57], which allows a product to be forged whereby an expensive item can be replaced with a cheap item. Like barcodes,

Table 2: STRIDE Threats in RFID Tags

	Threats	**Mitigation strategies**
S	Attaching cloned, forged counterfeited tags.	Use tamper evident RFID tags that alerts if tags are detached from an item and render them unusable [60]. Use anti-counterfeiting techniques, such as PUFs, chip-less RFID tags [56], and distance bounding protocols that utilize broadcast and collisions to find cloned tags [61].
T	Physical tampering, modifying tag memory or unauthorized tag killing.	Use RFID authentication protocols, such as [62], and write protect memory banks to prevent data manipulation. Use malicious tag data detection techniques in the RFID middleware [63].
R	Denying malicious actions due to limited memory on tags to store logs.	Allocate sufficient memory to log tag manipulations.
I	Reverse engineering stolen or discarded tags to access data due to weak encryption and weak authentication.	Use mutual Reader/Tag authentication and encryption to prevent unauthorized access [64]. Use shielded enclosures to protect against any unauthorized access to tag data outside the legitimate access area.
D	Causing buffer overflow by inserting malicious tag data.	Perform exception handling at the reader applications. Increase data buffer size.
E	Accessing tags and installing malware on RFID tags.	Incorporate security features in reader / middleware to detect the presence of malicious viruses and malware in the tag's memory.

RFID tags can be used as a medium to propagate viruses and execute SQL injection commands to target back-end servers and bring entire systems down [58]. Moreover, tag data may be corrupted or manipulated by an adversary by using a malicious tag writer, allowing an adversary to penetrate an unprotected scanning application by inserting malicious code into the tag's memory. Lastly, memory limitations of RFID tags make them vulnerable to DoS attacks by using blocker tags [59]. Other type of vulnerabilities, such as key desynchronisation in key-updating RFID identification protocols, can be found in [19].

4.3. Vulnerabilities of IoT Devices

Due to the pervasive nature of Internet of Things (IoT) devices, they are expected to play a major role in traceability at various stages of the supply chain. For example, in a food supply chain, IoT devices can be used effectively at the farm level to monitor various environmental factors that may impact food quality [65, 66]. On the other hand, IoT devices used on production lines can improve the sorting and packaging of food products and ensure food standards are met and maintained. IoT devices also find application in transportation to monitor optimal environmental conditions to prevent any spoiling of food products during transportation [66].

Although computationally stronger than RFID tags, IoT devices can be resource-constrained too, making them vulnerable to various attack vectors [67]. Adversaries with physical access to an IoT device can tamper with the device to monitor or modify sensing data, such as temperature, moisture levels and location [65]. Reverse engineering has also proven possible [68] and used to obtain hardcoded credentials, which are used to perform lateral attacks that target other parts of the supply chain system. Insecure user practices such as using weak passwords and use of default passwords for Internet exposed IoT devices have resulted in many recent malware targeting the consumer IoT devices such as the BASHLITE and Mirai [69]. Poor IoT security measures built into IoT firmware as well as faster time to ship devices results in vulnerable IoT devices prone to cybersecurity attacks. Software backdoors installed by manufacturers (or any other adversary) can enable remote access to IoT devices [70]. Lack of IoT firmware support is also listed as one of the reasons of easy cybersecurity attacks on IoT devices

Table 3: STRIDE Threats in IoT Devices

	Threats	Mitigation strategies
S	Impersonating device credentials. Cloning IoT devices to bypass authentication.	Use strong authentication mechanism e.g., device-characteristics-based mutual authentication. [74]. Use multi-factor authentication.
T	Tampering firmware to access sensitive information. Physically damaging and disabling IoT devices.	Encrypt and digitally sign the firmware binaries to preserve their confidentiality and integrity. Enforce a secure boot process to prevent modification of a back-doored firmware. Authenticate and encrypt device update process. [71]. Physically secure access to IoT devices.
R	Denying malicious actions by erasing or depleting the memory dedicated to log data.	Use cloud or external storage for critical event logs.
I	Reverse engineering to extract digital certificates. Side-channel attacks.	Encrypt data. Use cryptographic operations resistant to side-channel analysis. Avoid hardcoding encryption keys in firmware [71].
D	Sending unnecessary communication requests to drain a device's battery [67].	Use middleware platform that can detect DoS attacks.
E	Installing malware or adding IoT devices to botnets.	Protect against unauthorized access to firmware by encrypting the binaries [71]. Detect and disable malicious or compromised IoT devices. Enable strong access control. Disable unnecessary services running on IoT devices.

in [71]. This generally results in unpatched devices connected to the supply chain network thereby leaving the supply chain system vulnerable. A detailed description of IoTs vulnerabilities falls outside the scope of this article. We refer the interested reader to [72, 73] for more details.

The overall threat analysis report of IoT devices, including security controls, is depicted in Table 3.

5. Data Capture Layer

The data capture layer focuses on assets and interfaces that facilitate the collection of product data associated with a traceable object. It also ensures that the captured data is converted to a format suitable for different applications and storage. The prevalent assets in a data capture layer are barcode scanners, RFID reader/writer, RFID air interfaces, RFID software interface and IoT middleware.

5.1. Vulnerabilities of Barcode Scanners

These scanners are used to read the data encoded in a barcode attached to products. Barcode scanners are either wired (attached to computers Universal Serial Bus (USB)) or wireless (communicating either over Bluetooth, 2.4GHz or 433 MHz wireless channel). Barcode host system applications are used to decode and act on the data read from barcodes.

Barcode scanners that are directly connected to the host system pose a threat to the traceability system. For example, unauthorized scanners interfaced with the host system can send malicious commands to the back-end system. Malicious scanner manufacturers inserting malware/spyware to either the firmware or the operating system (OS) of wireless scanners, which in-turn facilitates the unauthorized monitoring of scanning events and associated data was reported in [75]. Similarly, WiFi capable scanners are generally connected to the organisation's WiFi network, thereby posing an additional security threat, whereby a scanner compromise may lead to a network compromise [76]. Likewise, a failure to promptly apply security updates and patches to a scanner's OS/firmware can be exploited by adversaries to compromise the device. Wireless scanners are also susceptible to various attacks targeting THE wireless channel, such as MiTM attacks, replay attacks, and side channel attacks.

In addition to the scanner, vulnerabilities found in the reader applications can also be exploited using customised barcodes to compromise the system. Examples of scanners being used to insert keystrokes in the host system are presented in [42, 77]. The STRIDE threat analysis of barcode scanners and host applications is presented in Table 4.

Table 4: STRIDE Threats in Barcode Scanners/ Host Applications

	Threats	**Mitigation strategies**
S	Using unauthorized scanners.	Mutual authentication between barcode scanners and the host computer system to prevent unauthorized connections.
T	Tampering of software/firmware and installing backdoor on scanners.	Implement access control policies. Digitally sign and encrypt firmware updates [78]. Physically secure WiFi connected and hand-held barcode scanners from unauthorized usage.
R	Denying malicious actions by clearing logs or scanner events on reader software application.	Logging needs to be enabled and secured on all scanning devices and reader applications.
I	Using compromised scanner and host applications. Eavesdropping attack on WiFi connected hand-held scanners.	Encrypt the communication between scanners and host applications. Patch the scanner firmware and and apply security updates to reader applications regularly.
D	Jamming attacks or sending unnecessary Bluetooth messages to wireless scanners.	Implement anti-jamming techniques and use secure Bluetooth communication using fingerprinting techniques [79]. Secure and patch the communication mechanisms used by wireless barcode scanners to prevent DoS attacks [80].
E	Deploying malware to scanners/readers.	Incorporate the access control levels on the host systems to prevent reader applications from having privileged access.

5.2. Vulnerabilities of Tag writers, Readers and Air Interfaces

RFID Air Interfaces provide a common Radio Frequency (RF) operating range and a standard communication protocol to facilitate the tag and reader to communicate. The readers identify the tags and access the stored data using the air interfaces. GS1 provides two air interface standards:

- *UHF Gen2 Air Interface*: The latest UHF Gen2 standard proposed by GS1 is the Gen2 V2.0 which defines an operating range of $860 - 960$ MHz UHF range.
- *HF Air Interface*: An Interrogator-talks-first (ITF) protocol operating at 13.56 MHz frequency with passive-backscatter.

The most common vulnerability observed in readers and writers is an adversary using compromised readers to read data from a genuine tag and write it into fake tags [81]. Likewise, an attacker can use a malicious writer to manipulate a tag's data, or kill the tag as pointed out in [82]. Similarly, an adversary can insert hardware Trojans during the fabrication process of the chips used in these devices to obtain unauthorized access to the data [83].

The majority of the threats to RFID systems emerge from the air interfaces between the tag and reader [84]. An attacker can conduct jamming attacks to disrupt communication and also achieve denial of service through desynchronisation of the authentication keys stored on back-end systems and tags [82]. An adversary can also eavesdrop on the data transferred between the tag and reader, and record the communication to launch replay attacks [85]. Furthermore, an attacker can use an illicit reader to hijack the session between a legitimate reader and a tag to launch MiTM attacks as highlighted in [85]. Power emission analysis of a tag can also be conducted to retrieve the tag information as demonstrated in [35]. The relevant STRIDE threat analysis is presented in Table 5.

Table 5: STRIDE Threats in Readers/Scanners/Air Interfaces

	Threats	**Mitigation strategies**
S	Using malicious readers and writers. Replay attacks.	Use strong authentication between readers, writers and tags. Use timestamps, counters, and challenge response protocols to protect against replay attacks [86].
T	Tampering or reverse engineer the reader/writer's firmware.	Enable authentication and access control to prevent tampering. Encrypt and digitally sign the firmware updates [78].
R	Denying malicious actions due to limited logging capability.	Enable logging on all readers/writers.
I	Eavesdropping unencrypted communication. Extract authentication keys by analysing power fluctuations. Divert communications to malicious devices by impersonation attacks.	Encrypt communication between tag and reader. Filter the power signal or delay the computation randomly to make power analysis difficult [85].
D	Carrying out RF jamming attacks on communication or desynchronization attacks between endpoints. Unauthorized killing of tags using malicious writers.	Use mutual authentication to prevent desynchronization [82] attacks and launching kill command attacks. Use external noise/radio shielded enclosure to protect against the RF jamming attack [85].
E	Compromising readers/writers using malicious data.	Built-in security features in reader to detect malicious data from tags.

5.3. Vulnerabilities of RFID Software Interfaces

These interfaces form the middleware between the RFID tags and the applications that access RFID data and help in transforming the RFID stored data into a format required by the upper layer applications. These interfaces include: Low level Reader protocol (LLRP) which defines the control and delivery of raw tag reads from readers to the Filtering and Collection (F&C) role. Application Level Event (ALE) which defines the control and delivery of filtered and collected tag read data from F&C to the EPCIS Capturing Application role. In accordance with EPC network architecture defined by EPC global, LLRP resides between the F&C and the reader. The smallest communication unit exchanged between a reader and client is in the form of a message that includes a query about readers, configuration read from or written to readers (client to reader), and reader status reports (from reader to client).

LLRP has various security vulnerabilities [87]. For example, the lack of a mutual authentication mechanism for connections from client applications to the reader can result in unauthorized access of readers [87]. The lack of fault-tolerance [88] in LLRP can be exploited by an adversary to cause DoS and disable the software interfaces required to scan RFID tags on large scale product supply chains. Infected readers can close the protocol session resulting in failed communication. This can result in loss of information from readers and disrupt the supply chain processes. Lack of encryption in LLRP can lead to eavesdropping or deception attacks [87]. The protocol does not support encryption natively and depends on Transport Layer Encryption. LLRP cannot not resist replay attacks as it does not use serial numbers or timestamps [87]. LLRP does not ensure data integrity as it does not incorporate digital signatures of transmitted message [87]. Improved versions of LLRP [89] referred to as TLS-LLRP and secured version of LLRP [90] support lightweight encryption of messages, mutual authentication between client and reader, and authentication of reader before accepting control commands. However, both solutions also suffer from data integrity attacks. Secured LLRP also suffers from vulnerabilities associated with constant session keys, as indicated in [87].

The F& C middleware is an important component of the EPCglobal network architecture, which uses a single interface to numerous distributed readers and capturing applications that may be interested in the collected data. This interface is called the Application Level Event (ALE). ALE 1.1 is comprised of five standard APIs - i.e., Reading, Writing, Tag Memory, Logical Reader and Access Control APIs. Reading API allows clients (i.e., capturing applications) to specify the tag data they are interested in and gives the corresponding reports in a variety of ways. Some vulnerabilities of ALE middleware are described below.

As RFID tags carry more information than just the EPC, the information flowing via the ALE middleware to the ERP systems will be prone to data manipulation and eavesdropping attacks [91]. Role based access control in the access control API restricts access to critical methods (i.e., subscription to capture tag data) but cannot cope with data aggregation details and filtered/combined reports in a request specification [91]. ALE provides an abstraction layer between RFID readers and the applications that depend on RFID data. The more commonly used RBAC in RFID middleware do not support dynamic access control policies required to secure RFID data [92]. Due to large volumes of RFID data processing occurring at the ALE middleware, DoS attacks will disrupt the supply chain system [93]. As ALE translates raw RFID data from tags to a format understood by the ERP interface, adversaries can target the vulnerabilities in ALE's data translation or filtering methods. Lack of mutual authentication in ALE can be exploited by adversaries to use unauthorized readers to connect to ALE middleware [92]. The relevant STRIDE analysis is presented in Table 6.

Table 6: STRIDE Threats in RFID Middleware

	Threats	**Mitigation strategies**
S	Connecting to RFID middleware using malicious readers or scanners. Replay attacks.	Mutual authentication between readers and ALE middleware. Use sequence numbers and timestamps to protect against the replay attacks.
T	Tampering LLRP protocol communication or middleware applications by deploying backdoors and unauthorized code.	Build security feature in middleware that can check for insertion of any malicious data. Enable authentication and access control between the reader and ALE.
R	Improper logging of configuration or deleted logs.	Enable activity logs that track modifications and communication in middleware.
I	Eavesdropping unencrypted LLRP communications. MiTM attacks diverting reader and middleware communications to malicious nodes.	Use of encryption in LLRP communication as well as the middleware to protect against the eavesdropping between a reader and ALE.
D	Using malicious values in the reader protocols to cause buffer overflow attacks. Corrupting the ALE interface with malicious reader values.	Use programming languages that offer bound checking to protect against buffer overflow [94]. Allow only data from authenticated readers to flow through the ALE interface. Use load-balanced ALE middleware to prevent availability issues.
E	Compromising ALE interface to gain unauthorized access. Launching cross-site scripting or SQL injection attacks.	Build security features in readers and middleware to only accept data in a pre-defined format to protect against code injection.

5.4. Vulnerabilities of IoT Middleware

The IoT protocols play an important role in collecting sensing data from IoT sensor devices and delivering it to the upper application layers. The middleware enables translation of message formats or inter-device communication between heterogeneous IoT devices and the protocols used to communicate between them [95]. Due to the critical role played by IoT middleware platforms, they can be susceptible to various cyber security threats. Some of the examples of these attacks include MiTM attacks, SQL injection attacks, signature wrapping attacks, malware injection and flooding attacks, DoS attacks, sinkhole attacks, cryptanalysis attacks and side channel attacks as highlighted in [95, 96].

A compromised IoT middleware can also become a launch pad for attacks on IoT devices and gaining control of the edge devices as reported in [97]. The detailed analysis of IoT vulnerabilities falls outside the scope of this article. We refer interested readers to [72, 73, 95, 98] for an overview of the security challenges and mitigation strategies relevant to securing of data flowing through IoT middleware.

6. Data Sharing Layer

In order to support traceability of products, GS1 provides global traceability standards that aid in the identification, capturing and sharing of traceability data. According to GS1 [38], traceability data is composed of master data, transactional data and event data, related to the traceable objects, which is shared among the trading partners. In addition, GS1 also categorises traceability data into public and private. Accordingly, organizations are not required to share all the traceability data with the trading partners. All organizations taking part in traceability must have all the relevant information that can be searched and accessed internally (private) as well as share agreed upon information among the traceability partners (public) without compromising their intellectual property rights [99]. Among the GS1 standards, Global Data Synchronization Network (GDSN) is used for sharing master data and relational data, Electronic Product Code Information Services (EPCIS) is used for sharing visible-event Data, and Electronic Data Interchange (EDI) is used for sharing transactional data. The different data categories are briefly explained below:

- **Master Data** - refers to data that is shared by one trading partner with many others and contains the description of attributes of real-world entities (product information, product catalog and product prices) identified by GS1 identification keys.
- **Transactional Data** - refers to data created by execution of a business process function such as a supply contract, custom processing, order placement, and final settlement using the GS1 identification keys.
- **Visibility-Event Data** - refers to data associated with physical movement of products (or other assets) identified by keys within the supply chain, detailing where and why products are at a given time within and across organizations. This persistent data contains all EPCIS events generated internally within the organization and received from other trading partners. This data is made available to all the EPCIS accessing applications.

The main assets in the data sharing layer are data repositories, data sharing interfaces and data sharing protocols whose vulnerabilities are elaborated next.

6.1. Vulnerabilities of Data Repositories

GDSN uses XML messages to exchange and update product information [100]. Consequently, data pools can be vulnerable to various XML based attacks that target vulnerabilities in XML message formats. Attacks such as XML injection where malicious characters are embedded in XML messages, as indicated in [101], can be launched. Adversaries can launch data manipulation attacks against these repositories targeting the XML message format. Furthermore, an adversary can flood GDSN registry with false registration requests or subscription requests causing DoS.

The EPCIS deployment contains a web server which serves as an interface for query functions and a database sever component which is used to store the various types of EPCIS events. The database server and the web server can be prone to various well-known attacks such as SQL injection attacks, intrusions and virus. An attacker can make use of weak trading partners to lodge enormous data requests or insert malicious payloads in XML files leading to DoS. This data sharing system can also be prone to hardware failures leading to data loss. The STRIDE analysis of data repositories is presented in Table 7.

Table 7: STRIDE Threats in Data Sharing Repositories

	Threats	**Mitigation strategies**
S	Using stolen/spoofed credentials. Using stolen authorization token to access event data repositories.	Enable strong authentication (e.g., multi-factor authentication) prior to giving access to critical data stored in data pools.
T	Tampering the master data. Compromising a single weak trading partners.	Allow only authorized individuals to make any changes to product related information after verifying their identity.
R	Denying malicious activity due to improper logging or log configuration errors.	Enable logging on repositories.
I	Carrying out data breaches at the global registry servers. Malware infected repositories may lead to data leakages.	Ensure that correct data is being shared only with authorized partners. Enable protection against virus and malware. Accept data only in predefined format to protect against any malicious data fed to event data repositories.
D	Flooding master/event data queries. Malicious payload in XML files, or oversized XML documents.	Protect against different types of XML attacks as mentioned in [102].
E	Compromising master and event data repositories. SQL injection attacks on repositories. Malicious payload of AS2 or from RFID tags.	Ensure that access tokens for event data is shared with correct partners. Allow data pool access only to authorized partners. Protect against malicious XML payloads that may lead to unauthorized data retrieval as indicated in [102].

6.2. Vulnerabilities of Data Sharing Interfaces

With this interface, visibility event data in accordance with EPCIS data model is delivered from capturing applications to a receiver. Similarly, EPCIS query interface is used when event data is requested by and delivered to a business application or a trading partner. The communication channels between the middleware, EPCIS repository, and query interface can be targeted by an adversary to access the sensitive EPCIS data. MiTM attacks [16] can be launched on unencrypted communication between middleware to EPCIS repository, or to other trading partner. Similarly, replay attacks can be launched if an attacker can record the communication between middleware, repository, and other trading partners as highlighted in [16]. Likewise, without mutual authentication, malicious applications may succeed in sending data to repositories or trade partners causing data corruption as discussed in [16].

As the ECPIS standards [103] have limited authorization mechanisms, adversaries can perform unauthorized actions by elevating their privileges. Furthermore, with EPCIS allowing master data to be updated using methods such as Instance/Lot master data (ILMD), the GDSN data pools can be susceptible to data corruption by exploiting the EPCIS interfaces. The STRIDE analysis for data sharing interfaces is summarized in Table 8.

6.3. Vulnerabilities of Data Sharing protocols

Electronic Data Interchange (EDI) is a widely used method among trading partners to exchange business documents. Most commonly used EDI communication methods are: Email, File Transfer Protocol (FTP) / Odette File Transfer Protocol (OFTP), AS2/AS4, Web services, Web-based [105, 106]. In this work we focus on vulnerabilities of AS2/AS4, which are widely used communication protocols in supply chains.

AS2. Since AS2 is leveraged for both GDSN synchronization and EPCIS event data transfer amongst the trading partner, identifying the vulnerabilities to AS2 and the subsequent threats on AS2 communications are paramount. Even though AS2 provides secure transmission messages, vulnerabilities do exist. The AS2 protocol is payload agnostic and allows various data formats which are then compressed

Table 8: STRIDE Threats in Data Sharing Interfaces

	Threats	Mitigation strategies
S	Carrying out impersonation attacks or replay attacks due to lack of mutual authentication between a middleware and repository.	Enable mutual authentication between capture application, accessing application, repositories, and trade partners.
T	Tampering data in transit, sending tampered content.	Accept data only from authenticated readers/trade partner in proper format to prevent data corruption. Allow only authorized individuals to makes changes to master and event data.
R	Denying malicious actions due to improper logging or log configuration errors.	Enable activity logging.
I	Carrying out information theft due to lack of encryption and mutual authentication between capture application, repositories, and trade partners.	Encrypt the communication between middleware, accessing application, repositories, and repository and trade partners.
D	Sending malicious data and corrupting capture and query interface. Sending unnecessary queries to exhaust resources.	Accept data only in pre-defined form from data carriers and enable anomaly detection to detect and prevent malicious data and anomalous requests.
E	Exploiting weak authorisation mechanism.	Mutually authenticate capture application and repositories. Accept data only in pre-defined a format from data carriers and trade partners. Incorporate proper redaction to alleviate the unauthorized access to data (redaction refers to denying a data request or restricting the amount of data requested by a trade partner) [104].

and digitally signed. The payload encryption is achieved by the S/MIME format and communication channel encryption is achieved using HTTPS protocol. The sender organisation can also request a message delivery receipt in the form of Message Disposition Notification (MDN) messages. Such services can potentially attract various forms of attacks. For example, application and network layer DoS attack on the exposed internet services can occur. Similarly, adversaries can target the AS2 software vulnerabilities to gain access to the system. Furthermore, a compromised AS2 software vendor can also push malicious updates to the AS2 server. Since DNS entries are used to discover and translate the Fully Qualified Domain Name (FQDN) pointing to AS2 servers, adversaries can use DNS poisoning to redirect the AS2 connections to malicious AS2 servers. Furthermore, stolen encryption certificates can be used to decrypt EDI messages. Use of weak digital HASH algorithms can be used to forge signatures and subvert authentication and integrity checks. The default encryption algorithms and digital signatures supported by AS2 are obsolete and have been proven to be insecure. Adoption of newer and stronger algorithms depends on the algorithms supported by the trading partner's AS2 software [107]. Lack of logging facilities on AS2 data flow events in the servers can lead to repudiation attacks. Compromised AS2 servers can also be used to send malware payloads in the AS2 messages. Several XML based attacks as indicated in [101] can be launched with the use of malicious XML messages supported in the AS2. Many free AS2 client software are available on the Web and a malicious client application distributed among the third-party trading partners can put the entire supply chain at risk.

AS2 also do not mandate Certificate Authority (CA) validation, many implementations use self-signed certificates which can result in MiTM attacks in the EDI process. In addition, as many Small and Medium Enterprises (SMEs) are involved in supply chain process which do not implement strict security guidelines, this can result in security breaches that affect the entire supply chain [108]. This is especially challenging when SMEs need to deploy Internet exposed AS2 servers for EDI communications. As third-party supply chain trade partners are not rigorously audited for their security measures, they can be compromised to launch lateral attacks on other trade partners [108, 109]. One example of a third party breach resulting in cyber security attacks on larger organizations was launched against US retailer Target which suffered a data breach in 2013 with a loss of 40 million customer payment card information. The cybersecurity attack was carried out by using stolen credentials from a third party HVAC company contracted to service Target locations.

AS4/SOAP/XML. The AS4 protocol is dependent on Simple Object Access Protocol (SOAP) which is a messaging protocol that can be leveraged for sending structured messages between trading partners. In addition, EPCIS also allows SOAP bindings for query control interface. SOAP is XML-based protocol running over HTTP. In addition, GDSN and EPCIS also use the XML format to share information. Due to the wide application of XML in AS4, GDSN, EPCIS as well for other EDI protocols, we discuss the XML vulnerabilities that need to be considered when implementing traceability system. Recently, NIST indicated a significant increase in security vulnerabilities that exploit XML based technologies [110]. Attacks can be executed via XML External Entities (XXE) by exploiting the XML processors that allow specification of an external entity (e.g., URI) that is de-referenced and evaluated during the processing of XML documents. An adversary can include crafted payloads in XML documents that exploit the vulnerabilities in code leading to data leakage or DoS attacks on servers. As indicated in [101], an attacker can insert malicious characteristics/strings in XML documents which can lead to XQuery, shellcode, or other attacks if the supplied data is not validated. Similarly, an attacker can create oversized XML documents inserting large data/number of characteristic or long names of XML node elements which can lead to DoS attacks [101]. A poorly configured XML parser can process an XML document that references a large external data source, which can also lead to DoS [101]. Likewise, XML injection attacks through the SOAP web-service interface or using search-based techniques to find malicious combinations of input data to compromise [103] the EPCIS web-service can be launched. An attacker can also potentially tamper the XML schema that serves as a template for validating the XML documents. The STRIDE threat analysis for the data sharing layer is presented in Table 9.

Table 9: STRIDE Threats in Data Sharing Protocols

	Threats	**Mitigation strategies**
S	Using stolen digital certificates for bypassing authentication.	Keep digital certificates in secure locations such as on an encrypted device and hardware security module. Keep the OS and antivirus up to date and avoid running any suspicious program.
T	Tampering HTTP/S based communications or tampering the payload. Tampered XML schema, malicious characters in XML documents.	Use strong hash algorithms so that collision attacks are not possible and any attempt to tamper the sent data be detected (e.g., SHA-2 instead of SHA-1 which is recommended AS2 transport communication guidelines available on GS1 official website [103]). Use strong XML payload encryption and validation of XML documents.
R	Performing malicious actions on communication servers and removing traces of adversarial actions.	Enable logging of all operations.
I	Poisoning DNS cache leading to MiTM attacks or using stolen digital certificates to get access to the EDI data. Accidentally or with malicious intentions revealing private keys or credentials.	Ensure that digital certificates are kept in secure locations. Use strong public-private keys for asymmetric encryption. For example, consider using 2048 bits keys instead of 1024 bit keys. Weak keys are likely to be compromised as demonstrated in [111]. Protect against the MiTM attacks through DNS poisoning by enabling DNSSEC [112].
D	Flooding based application layer attacks against server Large size XML documents and exploiting XML parser.	Prevent HTTP/S or application message flooding attacks by incorporating techniques such as traffic profiling, computational challenges, firewall, and constant monitoring of threats [113].
E	Compromising weak supply chain trading partner's EDI servers to gain privileged access. Sending malicious communication payload such as malware infected Excel files, or XML based attacks.	Check for malicious payloads such as XML injection attacks as mentioned in [102].

7. Application Layer

The application layer is the uppermost layer in the data flow architecture where end-user applications access the traceability data in order to perform various tasks. The traceability data is accessed by various end-user applications such as Enterprise Resource Planning (ERP) tools, Supply Chain Management (SCM), audit applications, consumer applications, monitoring and analytic tools [38, 114]. These applications that access the traceability data can be broadly categorised into business to business (B2B), business to government (B2G) and business to customer (B2C). Due to complex interrelationships between the producers, supply chains, consumers, financial institutions and government organizations, weaknesses or vulnerabilities in any one domain can cascade cyber security risks to the entire food traceability system [115]. In order to simplify the analysis of vulnerabilities and threats to the Application Layer, we broadly categorise all the applications that businesses utilise internally and with other business partners as B2B and those business applications that are accessed by end customers as B2C.

7.1. Vulnerabilities of B2B Applications

ERP/SCM. Various ERP systems of manufactures to supply chain traders access the traceability data for inventory management, order management, shipping, transportation and financial transactions related to products [116]. SCM systems are used to manage the flow of products from source to destination [117]. Since ERP refers to a type of software that businesses generally use to conduct their usual business activities (e.g., manufacturing, sales, and finance etc), their security vulnerabilities can adversely impact the success and survival of the business. Similarly, SCM systems manage activities such as procurement, supply chain planning (e.g., inventory planning, and product lines maintenance), and logistics, etc [117]. Therefore, SCM systems need to interact with numerous other partners and businesses, which can lead to numerous problems such as unintentional revealing of important information and other cybersecurity attacks [117]. Ransomware/Malware can impact ERP/SCM systems leading to disruption of critical business activities and often cause significant financial losses to the business [118][117]. Similarly, ERP/SCM systems protected with weak authentication such as simple passwords can be vulnerable since passwords are easy to hack [119]. Allowing full data access and edit privileges to everyone in an ERP system can lead to unauthorized information disclosure [119]. Failing to keep up with the software updates to alleviate the impact of known security issues can also impact ERP system [119]. ERP systems may also be impacted by application vulnerabilities such as XSS and SQL injection attacks as indicated in [117]. Similarly, an attacker can exploit the vulnerabilities in ERP/SCM software applications to conduct different types of attacks [117]. Likewise, insecure communication between ERP/SCM applications and back-end servers can lead to eavesdropping/MiTM attacks.

7.2. Vulnerabilities of Data Analytics (B2B)

Such applications are used in the supply chain industry for exploratory analysis of traceability data. The use of various applications in the traceability system will result in a huge amount of data being generated from various sources which relate to consumers, traceability of products, various events, manufacturing processes and various physical devices involved in product movement from source to destination. In addition, such large volumes of data also require high performance computing solutions or cloud computing services where data from various stages in supply chains and retail operations need to be integrated to implement smart traceability systems [120]. Due to the sensitive nature of data in this domain, several vulnerabilities exist as indicated in [121]. For example, poor access control can lead to unauthorized access and possible data leakage. Third-party client applications can collect user's data without their consent. Privacy of consumer information due to lack of privacy measures can be precarious when analysing sensitive data which may contain personal identifiable information. Similarly, privacy compromise of the manufacturer and supply chain trading partners can have serious consequences (e.g., disclosure of trade secrets and financial information). Adversarial attacks to data analysis tools or data tampering attacks can impact the integrity of data. The cloud computing infrastructure is also prone to cyber threats and the most common threats to cloud computing are: data breach, user account compromise, DoS attacks, injection threats, API vulnerabilities [122]. Table 10 lists the STRIDE analysis of B2B applications.

Table 10: STRIDE Threats in B2B Applications

	Threats	**Mitigation strategies**
S	Using compromised credentials to access the ERP/SCM and Cloud computing systems.	Protect ERM/SCM systems with strong authentication mechanisms such as 2FA/MFA [117].
T	Exploiting weak access control measures or causing shared memory attacks on cloud nodes to tamper data.	Allow only authorized individuals to modify the traceability data using access control measures.
R	Denying malicious actions by deleting logs and associated traces to cause non-repudiation attack.	Enable logging of all operations.
I	Exploiting weak authentication, weak access control or insecure communication between B2B applications. Causing XSS or SQL injection attacks. Causing data breaches in cloud computing infrastructure.	Protect all communication with encryption to alleviate chances of any unauthorized access to data. Enable strict query checking to prevent SQL injection attacks. Enable robust authentication and using it enable fine-grained access control. Use of privacy preserving techniques such as differential privacy.
D	Launching ransomware attacks or exploiting the vulnerabilities in the ERP/SCM software to disable the software.	Train staff on ransomware and how they impact system. [118]. Scan systems regularly with good anti-virus.
E	Circumventing weak application authorisation controls. Infecting ERP/SCM server with virus or malware. An adversary exploiting known software vulnerabilities.	Incorporate fine grained access control. Protect against virus/malware. Update software as soon as they are made available to patch any known security vulnerabilities.

7.3. Vulnerabilities of B2C Applications

The B2C applications allow customers to access the traceability data related to the products they purchase. Organizations provide applications often in the form of mobile applications that allow customers to access traceability related information. GS1 provides the resolver services that allow customers to directly access traceability information using GS1 digital link standard [123]. The product is either equipped with QR code or a Near Field Communication (NFC) tag which contains the GS1 Digital Link Universal Resource Identifier (URI). This 2D barcode can then be used by either the consumer to fetch product information or can be used by supply chain partners to get the GTIN. These consumer facing applications can be vulnerable to various vulnerabilities as indicated in [124]. For example, consumer errors could lead to accidental deletion of data, accidentally revealing access credentials or falling victim to social engineering attacks which harvest sensitive information leaving the consumer-facing software applications vulnerable to cybersecurity attacks. Similarly, consumer related data stored in backend servers without adequate privacy measures can reveal sensitive user information. The consumer facing client mobile applications, or the backend servers can be vulnerable to malware attacks which could result in infected mobile applications or backend server applications. Malicious characters in the digital link resolver queries can result in undesirable effect on the resolver servers. Similarly, insecure communication between the client application and back-end servers can result in eavesdropping attacks. The Internet facing traceability application servers can be targeted with DoS attacks to cause availability issues. Table 11 present the STRIDE analysis of B2C applications.

Table 11: STRIDE Threats in B2C Applications

	Threats	**Mitigation strategies**
S	Using stolen client credentials or forged client identities	Enable multi-factor authentication to protect against any potential subversion of first factor of authentication
T	Tampering client and backend consumer facing applications (e.g., leaving a backdoor or inserting a malware) or tampering traceability data	Allow only authorized individuals to modify the traceability data. Protect against the insertion of malware
R	Denying malicious actions due to insufficient logging capabilities or by deleting logs on client applications and the backend servers	Enable logging of all operations
I	Exploiting insecure communication between client applications and the backend servers	Encrypt communication with the back-end server
D	Adversaries launching spamming or DoS attacks by sending unnecessary requests (i.e sending large number of digital link queries to resolver service) to the backend servers to exhaust resources	Incorporate a reliable DoS detection and mitigation solution as indicated in [125]
E	Using compromised client applications or backend servers to launch attacks	Update client application and server software regularly to fix any know security vulnerabilities. Regularly scan the system for malware/ransomware

8. Potential Impact of threats on Traceability Systems

The above presented threats to technology indicate that a traceability system can be prone to several cybersecurity attacks. The involvement of multiple stakeholders and an heterogeneous system landscape exacerbates the attack surface and their potential impact. In this section, we present some of the potential impacts of cyber security threats to the traceability system. The data carrier layer which consists of various TRU identification schemes, is prone to more physical tampering attacks compared to other layers. The data carriers such as barcodes and RFID tags carry primary identifiers for the TRU and are physically attached to the products. This makes them susceptible to tampering attacks that aim to introduce counterfeit goods or divert goods to unauthorized third-parties. Whereas the upper layers are more prone to communication channel-based attacks that aim to either gain unauthorized access to the traceability system or cause disruptions and delays in supply chains. The potential impact of various threats to traceability systems can be summarised as:

- *Spoofing and Tampering attacks*: At lower layers these attacks results in loss of quality and financial loss due to insertion of fake goods. These attacks render physical identifiers unusable for traceability and would require significant human intervention to process and facilitate movement of goods in supply chains. The implementation of proposed mitigation techniques to making these physical identifiers tamper-proof is a challenging task due to the ease with which these identifiers can be accessed and organizations need to consider the cost versus benefit trade-off before implementing these methods. At higher layers, these attacks can result in gaining unauthorized access to the system and can result in tampering of traceability data.

- *Repudiation attacks*: This can lead to a lack of evidence in identifying the sources of attack and affect the correct implementation of the security controls in the traceability system. In the event of a fraud and the subsequent tampering of evidence, it becomes challenging to trace the stolen goods or identify the source of fake goods. Even though the severity of these attacks is low and is often overlooked by the system administrators, the information obtained by correct logging of system changes provides useful means in identifying the gaps.

- *Information disclosure*: This attack can lead to unauthorized access to sensitive data and can cause financial loss as well as loss of reputation for supply chain organizations. In addition, organizations may also have legal obligations to protect the individual's data (e.g., personal identifiable information if any, or credit card information). Therefore, the privacy of such data may need to be ensured. The contemporary privacy techniques (e.g., anonymisation, differential privacy, and federated learning) can prove to be helpful in this regard.

- *DoS attacks*: These attacks are the most common form of cybersecurity attacks and are often difficult to protect due to the large volumes of attack nodes used to launch a distributed DoS. These attacks can result in disruptions and delays in supply chains which can cause financial losses. As highlighted in the previous sections, these attacks can have a significant impact at higher layers especially at the data sharing layer which contain various traceability data repositories.

- *Elevation of privileges*: These attacks can result in a compromised traceability system. Using these attacks, adversaries can cause significant impact to the traceability information and also leverage these attacks for lateral movement with in the supply chain. Weak links (i.e., small trading partners with weak security measures) with in the supply chain are more susceptible to these attacks and can become the entry point for adversaries to launch attacks on other supply chain partners.

The threats and the potential impact on supply chains and traceability systems call for strong mitigation measures to prevent successful cybersecurity attacks. The preceding sections dealt with the modelling of various attacks and the appropriate countermeasure for all technological assets at different levels of our data-flow architecture. Therefore, in subsequent discussion, we enlist the recommendations that are applicable to supply chain organisations to better plan, prepare and act to counter cyber security incidents. Note that, the below-mentioned recommendations are provided as a guide to supply-chain stakeholders to plan their security strategies, however it is the task of organizations to continuously evaluate their cyber security posture for long term benefit.

Recommendations for Managing Technology Related Cyber Security Risks in Supply Chains:

- *State-of-the-art Defense Systems* – Organizations must adopt the latest defense mechanisms such as firewalls, endpoint security, Intrusion Detection System (IDS), AI based anomaly detection system, defense-in-depth, and automated and continuous vulnerability and penetration testing. Ensure that software is updated regularly to patch any known vulnerability that can lead to data breaches, critical systems are installed with latest anti-virus/anti-malware and updated regularly to protect against virus/malware, and Incorporate IDS for detecting any potential cybersecurity attacks.

- *Wise Use of Contemporary Technologies* – Modern businesses are heavily reliant upon AI and ML for analyzing the enormous amount of data to provide insights to the business leaders. Similarly, they are often used within the cyber strategy – e.g., in intrusion detection systems. However, these technologies open a whole new vector of cyber threats that may be considered and mitigated accordingly. Similarly, federated learning may be considered where appropriate so as to alleviate the need of data sharing for distributed model-learning if needed [126].

- *Manage Security of IoT devices and CPS Systems* - As automated supply chains heavily rely on IoT devices and CPS systems it is essential that organizations have effective strategies to manage and establish security policies to safeguard devices and the data stored in them. As traditional security tools cannot be implemented on constrained devices, it is essential to use IoT specific measures that can play a critical role in securing access and communication to these devices, such as light-weight authentication protocols and encryption schemes suited to IoT devices [74] [127]. Physical security of all the devices used within the supply chains is also an important aspect that needs to be considered.

- *Data Protection* - Always encrypt data be it in rest or in transit. Especially the use of secure multi-party computation (MPC) is recommended for securing data between several trading partners [128]. This necessitates that all the trading partners update their security mechanisms and adopt the same security standards as their counterparts.
- *Decentralisation* - Decentralised data sharing techniques such as blockchains provide a secure network to share data with added security of immutability, resilience to cryptographic attacks and updated only with peer consensus [129]. The use of such technologies can allow supply chain partners to share data related to traceability in a transparent way.
- *Data Cleanrooms* - Sharing sensitive product information among peers for demonstrating the competitive advantage over others is a challenging task. Solutions such as data cleanrooms and digital marketplaces have been suggested as means to securely share such sensitive information [130, 131]. These methods can enhance the quality of the shared data and introduce transparency among the supply chain peers which are essential for competitive intelligence.
- *Secure Data Storage* - Store critical data on secure locations with proper protections (i.e., authentication and access control). Destroy any data that is not used anymore and maintain its record.

Limitations

Although this article adopts a layered data-flow architecture for generic supply chains and comprehensively identifies the assets and their vulnerabilities, it is worth mentioning that supply chains may also suffer from various other vulnerabilities due to interaction of business processes. For the sake of completeness, we provide a high-level assessment on people and processes in the **??** and the related best practices in **??**. Making such an analysis as detailed as the technology-oriented analysis we provide in the paper is left for future work.

9. Conclusion

This paper undertook an investigation of supply chains and traceability systems with an emphasis on identifying different cyber threats and vulnerabilities. To accomplish this, we adopt a generic data flow architecture that represents different layers and associated assets of a typical supply chain system. We conduct an extensive and systematic security assessment of each and every layer of the data flow architecture by employing the popular STRIDE threat model. To the best of our knowledge, such systematic security assessment does not exist in current extant literature, and thus will assist relevant stakeholders (e.g., companies, distributors, retailers, and solution providers, etc) to understand the potential cybersecurity risks and vulnerabilities in different elements of supply chains and traceability systems. We also present potential mitigation measures for countering the identified threats to assist stakeholders in safeguarding their traceability systems. Finally, the paper presents a list of strategies for the protection of supply chain systems and important traceability data. In order to maintain their competitive advantage, reputation, and avoid any financial losses incurred due to data leakages it is imperative that stakeholders adopt a strong cyber security posture.

Disclaimer: Although the architecture considered in this work refers strongly to GS1, the threats highlighted are only related to the underlying technologies and not in the GS1 standard itself. We emphasize that this analysis is intended to be used only as a reference for security risk assessment.

References

[1] P. Olsen, M. Borit, How to define traceability, Trends in Food Science & Technology 29 (2) (2013) 142 – 150. doi:https://doi.org/10.1016/j.tifs.2012.10.003.
URL `http://www.sciencedirect.com/science/article/pii/S0924224412002117`

[2] S. Appelhanz, V.-S. Osburg, W. Toporowski, M. Schumann, Traceability system for capturing, processing and providing consumer-relevant information about wood products: system solution and its economic feasibility, Journal of Cleaner Production 110 (2016) 132–148, special Volume: Improved resource efficiency and cascading utilisation of renewable materials. doi:https://doi.org/10.1016/j.jclepro.2015.02.034.
URL `https://www.sciencedirect.com/science/article/pii/S095965261500147X`

[3] T. G. Karippacheril, L. D. Rios, L. Srivastava, Global Markets, Global Challenges: Improving Food Safety and Traceability While Empowering Smallholders Through ICT, 2017, pp. 283–308. doi:10.1596/978-1-4648-1002-2_Module11.
URL `https://elibrary.worldbank.org/doi/abs/10.1596/978-1-4648-1002-2\_Module11`

[4] M. Bogaardt, K. Poppe, V. Viool, E. van Zuidam, Cybersecurity in the agrifood sector, Tech. rep., Capgemini Consulting (2016).

[5] P. Olsen, M. Borit, The components of a food traceability system, Trends in Food Science & Technology 77 (2018) 143 – 149. doi:https://doi.org/10.1016/j.tifs.2018.05.004.
URL `http://www.sciencedirect.com/science/article/pii/S0924224417304107`

[6] V. S. C. Jakes, Addressing cybersecurity vulnerabilities in fresh and cold supply chain amid covid-19, `https://www.foodlogistics.com/technology/article/21129735/addressing-cybersecurity-vulnerabilities-in-fresh-and-cold-supply-chain-amid-covid19` ((Accessed: 02-12-2020)).

[7] A. Yeboah-Ofori, S. Islam, Cyber security threat modeling for supply chain organizational environments, Future Internet 11 (3). doi:10.3390/fi11030063.
URL `https://www.mdpi.com/1999-5903/11/3/63`

[8] Q. Lin, H. Wang, X. Pei, J. Wang, Food safety traceability system based on blockchain and epcis, IEEE Access 7 (2019) 20698–20707. doi:10.1109/ACCESS.2019.2897792.

[9] F. Tian, A supply chain traceability system for food safety based on haccp, blockchain & internet of things, in: 2017 International conference on service systems and service management, IEEE, 2017, pp. 1–6.

[10] Q. Lu, X. Xu, Adaptable blockchain-based systems: A case study for product traceability, IEEE Software 34 (6) (2017) 21–27. doi:10.1109/MS.2017.4121227.

[11] F. Tian, An agri-food supply chain traceability system for china based on rfid & blockchain technology, in: 2016 13th international conference on service systems and service management (ICSSSM), IEEE, 2016, pp. 1–6.

[12] GS1, Traceability, `https://www.gs1.org/standards/traceability` (Accessed: 12-12-2020).

[13] L. Urciuoli, T. Männistö, J. Hintsa, T. Khan, Supply chain cyber security–potential threats, Information & Security: An International Journal 29 (1).

[14] G. S. C. Institute, Managing cyber risks in global supply chains: The four fundamentals, `https://haslam.utk.edu/` ((Accessed: 26-10-2020)).

[15] S. Pandey, R. K. Singh, A. Gunasekaran, A. Kaushik, Cyber security risks in globalized supply chains: conceptual framework, Journal of Global Operations and Strategic Sourcing.

[16] D. M. Konidala, K. Woan-Sik, K. Kim, Security assessment of epcglobal architecture framework, `https://cocoa.ethz.ch/downloads/2014/06/` ((Accessed: 23-10-2020)).

[17] B. A. Sabbagh, S. Kowalski, A socio-technical framework for threat modeling a software supply chain, IEEE Security Privacy 13 (4) (2015) 30–39. doi:10.1109/MSP.2015.72.

[18] U. Guin, K. Huang, D. DiMase, J. M. Carulli, M. Tehranipoor, Y. Makris, Counterfeit integrated circuits: A rising threat in the global semiconductor supply chain, Proceedings of the IEEE 102 (8) (2014) 1207–1228. doi:10.1109/JPROC.2014.2332291.

[19] Yu-JuTu and Wei Zhou and Selwyn Piramuthu, Icritical risk considerations in auto-id security: Barcode vs. rfid, Decision Support Systems.

[20] M. Howard, S. Lipner, The security development lifecycle, Vol. 8, Microsoft Press Redmond, 2006.

[21] OWASP, Application threat modeling, `https://www.owasp.org/index.php/Application\_Threat\_Modeling` (2018).

[22] T. UcedaVelez, M. M. Morana, Risk centric threat modeling, Wiley Online Library, 2015.

[23] C. Alberts, A. Dorofee, J. Stevens, C. Woody, Introduction to the octave approach, Tech. rep., Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst (2003).

[24] V. Saini, Q. Duan, V. Paruchuri, Threat modeling using attack trees, Journal of Computing Sciences in Colleges 23 (4) (2008) 124–131.

[25] R. Khan, K. McLaughlin, D. Laverty, S. Sezer, Stride-based threat modeling for cyber-physical systems, in: 2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe), 2017, pp. 1–6. doi:10.1109/ISGTEurope.2017.8260283.

[26] A. Hajrić, T. Smaka, S. Baraković, J. Baraković-Husić, Methods, methodologies, and tools for threat modeling with case study, Telfor Journal 12 (1) (2020) 56–61.

[27] A. Shostack, Threat modeling: Designing for security, John Wiley & Sons, 2014.

[28] W. Xiong, R. Lagerström, Threat modeling – a systematic literature review, Computers & Security 84 (2019) 53–69. doi:https://doi.org/10.1016/j.cose.2019.03.010.
URL `https://www.sciencedirect.com/science/article/pii/S0167404818307478`

[29] P. Torr, Demystifying the threat modeling process, IEEE Security Privacy 3 (5) (2005) 66–70. doi:10.1109/MSP.2005.119.

[30] M. M. Aung, Y. S. Chang, Traceability in a food supply chain: Safety and quality perspectives, Food Control 39 (2014) 172 – 184. doi:https://doi.org/10.1016/j.foodcont.2013.11.007.
URL `http://www.sciencedirect.com/science/article/pii/S0956713513005811`

[31] J. I. San Jose, R. Zangroniz, J. J. de Dios, J. M. Pastor, Four-layer architecture for product traceability in logistic applications, in: Big Data and Internet of Things: A Roadmap for Smart Environments, Springer, 2014, pp. 401–423.

[32] L. B. Campos, C. E. Cugnasca, Towards an iot-based architecture for wine traceability, in: 2015 International Conference on Distributed Computing in Sensor Systems, IEEE, 2015, pp. 212–213.

[33] Z. Wang, P. Liu, Application of blockchain technology in agricultural product traceability system, in: Artificial Intelligence and Security, Springer International Publishing, 2019, pp. 81–90.

[34] P. B. Purwandoko, K. B. Seminar, et al., Development of a smart traceability system for the rice agroindustry supply chain in indonesia, Information 10 (10) (2019) 288.

[35] K. M. Karlsen, B. Dreyer, P. Olsen, E. O. Elvevoll, Literature review: Does a common theoretical framework to implement food traceability exist?, Food Control 32 (2) (2013) 409 – 417. doi:https://doi.org/10.1016/j.foodcont.2012.12.011.

[36] GS1, Data carriers and gs1, https://www.gs1.org/sites/default/files/docs/architecture/2021-03-17_RFF_DataCarriersAndGS1_RFF.pdf ((Accessed: 6-6-2021)).

[37] GS1, Gs1 edi, https://www.gs1.org/standards/edi ((Accessed: 6-6-2021)).

[38] GS1, Gs1 global traceability standard, ttps://www.gs1.org/standards/ (Accessed: 21-11-2020).

[39] GS1, Rfid standards, https://www.gs1au.org/ (Accessed: 1-12-2020).

[40] N. Xie, Q. Zhang, J. Hu, G. Luo, C. Chen, Low-Cost Anti-Copying 2D Barcode by Exploiting Channel Noise Characteristics (2020). arXiv:2001.06203.

[41] S. Moore, A. Roos, Is a barcode security threat ooming in your company's future?, https://id-integration.com ((Accessed: 22-10-2020)).

[42] H. Ma, Badbarcode: How to hack a starship with a piece of paper, https://pacsec.jp/psj15/ ((Accessed: 22-10-2020)).

[43] Krombholz, Katharina and Frühwirt, Peter and Kieseberg, Peter and Kapsalis, Ioannis and Huber, Markus and Weippl, Edgar", editor="Tryfonas, Theo and Askoxylakis, Ioannis, QR Code Security: A Survey of Attacks and Challenges for Usable Security, in: Human Aspects of Information Security, Privacy, and Trust, Springer International Publishing, Cham, 2014, pp. 79–90.

[44] P. Kieseberg, S. Schrittwieser, M. Leithner, M. Mulazzani, E. Weippl, L. Munroe, M. Sinha, Malicious pixels using qr codes as attack vector, in: Trustworthy ubiquitous computing, Springer, 2012, pp. 21–38.

[45] T. Vidas, E. Owusu, S. Wang, C. Zeng, L. F. Cranor, N. Christin, Qrishing: The susceptibility of smartphone users to qr code phishing attacks, in: A. A. Adams, M. Brenner, M. Smith (Eds.), Financial Cryptography and Data Security, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 52–69.

[46] A. Kharraz, E. Kirda, W. Robertson, D. Balzarotti, A. Francillon, Optical delusions: A study of malicious qr codes in the wild, in: 2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, 2014, pp. 192–203. doi:10.1109/DSN.2014.103.

[47] Dabrowski, Adrian and Krombholz, Katharina and Ullrich, Johanna and Weippl, Edgar R., Qr inception: Barcode-in-barcode attacks, in: Proceedings of the 4th ACM Workshop on Security and Privacy in Smartphones & Mobile Devices, SPSM '14, Association for Computing Machinery, New York, NY, USA, 2014, p. 3–10. doi:10.1145/2666620.2666624.
URL https://doi.org/10.1145/2666620.2666624

[48] P. Kieseberg, M. Leithner, M. Mulazzani, L. Munroe, S. Schrittwieser, M. Sinha, E. Weippl, QR Code Security, in: MoMM, MoMM '10, Association for Computing Machinery, New York, NY, USA, 2010, p. 430–435. doi:10.1145/1971519.1971593.
URL https://doi.org/10.1145/1971519.1971593

[49] N. Xie, J. Hu, J. Chen, Q. Zhang, C. Chen, Detection of Information Hiding at Anti-Copying 2D Barcodes (2020). arXiv:2003.09316.

[50] N. Acton, M. Caton, P. White, D. Wicker, Barcode copy protection system, uS Patent 9,311,583 (Apr. 12 2016).

[51] X. J. Zhang, J. M. Luan, L. N. Ni, Design of pdf4172d barcode watermarking system based on sopc, in: Advances in Manufacturing Technology, Vol. 220 of Applied Mechanics and Materials, Trans Tech Publications Ltd, 2012, pp. 2903–2907. doi:10.4028/www.scientific.net/AMM.220-223.2903.

[52] M. F. Roseman, Anti-tamper label and item embodying the same, uS Patent 8,678,289 (Mar. 25 2014).

[53] A. Mukherjee, Authentic barcodes using digital signatures (U.S. Patent US20120308003A1, Dec. 2012).

[54] Delivr, What is sqrc?, `https://delivr.com/faq/1468/what-is-sqrc` ((Accessed: 22-10-2020)).

[55] E. Todeschini, T. Meier, Barcode reader with security features (U.S. Patent US 9.262,633 B1 , Feb. 2016).

[56] Khalil, G.; Doss, R.; Chowdhury, M. A, A Comparison Survey Study on RFID Based Anti-Counterfeiting Systems, J. Sens. Actuator Netw 8 (37).

[57] H. Li, Y. Chen, Z. He, The survey of rfid attacks and defenses, in: 2012 8th International Conference on Wireless Communications, Networking and Mobile Computing, ieee, 2012, pp. 1–4.

[58] atlasRFID, 7 types of security attacks on RFID systems, `https://www.atlasrfidstore.com/` ((Accessed: 25-11.2020)).

[59] A. Khattab, Z. Jeddi, E. Amini, M. Bayoumi, RFID Security Threats and Basic Solutions, Springer International Publishing, Cham, 2017, pp. 27–41. doi:10.1007/978-3-319-47545-52.

[60] atlasRFID, Tamper evident rfid tags, `https://www.atlasrfidstore.com/` ((Accessed: 22-10-2020)).

[61] K. Bu, X. Liu, B. Xiao, Approaching the time lower bound on cloned-tag identification for large RFID systems, Ad Hoc Networks 13 (2014) 271 – 281. doi:https://doi.org/10.1016/j.adhoc.2013.08.011.

[62] U. Mujahid, M. Najam-ul Islam, S. Sarwar, A new ultralightweight rfid authentication protocol for passive low cost tags: Kmap, Wireless Personal Communications 94 (3) (2017) 725–744.

[63] B. Ray, S. Huda, M. U. Chowdhury, Smart rfid reader protocol for malware detection, in: 2011 12th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, IEEE, 2011, pp. 64–69.

[64] H. Xu, J. Ding, P. Li, F. Zhu, R. Wang, A lightweight rfid mutual authentication protocol based on physical unclonable function, Sensors 18 (3) (2018) 760.

[65] I. F. All, Food traceability and the iot solutions improving it, `https://www.iotforall.com/` (Accessed: 3-3-2021).

[66] J. Lin, Z. Shen, A. Zhang, Y. Chai, Blockchain and iot based food traceability for smart agriculture, in: Proceedings of the 3rd Int. Conf. on Crowd Science and Eng., 2018, pp. 1–6.

[67] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, N. Ghani, Demystifying iot security: An exhaustive survey on iot vulnerabilities and a first empirical look on internet-scale iot exploitations, IEEE Communications Surveys Tutorials 21 (3) (2019) 2702–2733. doi:10.1109/COMST.2019.2910750.

[68] A. Costin, Iot malware : Comprehensive survey , analysis framework and case studies, 2018.
URL https://i.blackhat.com/us-18/

[69] A. Marzano, D. Alexander, O. Fonseca, E. Fazzion, C. Hoepers, K. Steding-Jessen, M. H. Chaves, Í. Cunha, D. Guedes, W. Meira, The evolution of bashlite and mirai iot botnets, in: 2018 IEEE Symposium on Computers and Communications (ISCC), IEEE, 2018, pp. 00813–00818.

[70] S. News, Iot devices easily hacked to be backdoors: Experiment, https://www.securityweek.com/iot-devices-easily-hacked-be-backdoors-experiment ((Accessed: 20-12-2020)).

[71] Infosys, Firmware security for iot devices: Discovering key risks through reverse engineering and best practices in risk mitigation, https://www.infosys.com/ ((Accessed: 22-10-2020)).

[72] Sharu Bansal and Dilip Kumar , Iot ecosystem: A survey on devices, gateways, operating systems, middleware and communication, International Journal of Wireless Information Networks (2020) 340–364.

[73] X. Jiang, M. Lora, S. Chattopadhyay, An experimental analysis of security vulnerabilities in industrial iot devices 20 (2). doi:10.1145/3379542.
URL https://doi.org/10.1145/3379542

[74] S. W. A. Shah, N. F. Syed, A. Shaghaghi, A. Anwar, Z. Baig, R. Doss, Towards a lightweight continuous authentication protocol for device-to-device communication, in: 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 2020, pp. 1119–1126. doi:10.1109/TrustCom50675.2020.00148.

[75] J. Scharr, China-made handheld barcode scanners ship with spyware, https://www.tomsguide.com/ (Accessed: 5-3-2021).

[76] K. Marko, How a scanner infected corporate systems and stole data: Beware trojan peripherals, https://www.forbes.com/ (Accessed: 10-3-2021).

[77] P. Ducklin, Forget badbios, here comes badbarcode, https://nakedsecurity.sophos.com/2015/11/19/forget-badbios-here-comes-badbarcode/ (Accessed: 15-12-2020).

[78] acob Beningo, 7 tips for securing an embedded system, https://www.designnews.com/ (Accessed: 22-10-2020).

[79] M. Barbeau, J. Hall, E. Kranakis, Detection of rogue devices in bluetooth networks using radio frequency fingerprinting, in: proceedings of the 3rd IASTED International Conference on Communications and Computer Networks, CCN, Citeseer, 2006, pp. 4–6.

[80] C. details, Cve-2018-15453, https://www.cvedetails.com/cve/CVE-2018-15453/ ((Accessed: 23-10-2020)).

[81] T. Halevi, S. Lin, Di Ma, A. K. Prasad, N. Saxena, J. Voris, Tuo Xiang, Sensing-enabled defenses to rfid unauthorized reading and relay attacks without changing the usage model, in: 2012 IEEE International Conference on Pervasive Computing and Communications, 2012, pp. 227–234. doi:10.1109/PerCom.2012.6199835.

[82] Tagra, Deepak, Rahman, Musfiq and Sampalli, Srinivas, Technique for preventing DoS attacks on RFID systems, in: SoftCOM 2010, 18th International Conference on Software, Telecommunications and Computer Networks, IEEE, 2010, pp. 6–10.

[83] T. Reece, W. H. Robinson, Analysis of data-leak hardware trojans in aes cryptographic circuits, in: 2013 IEEE International Conference on Technologies for Homeland Security (HST), 2013, pp. 467–472. doi:10.1109/THS.2013.6699049.

[84] B. Khoo, P. Harris, S. A. Husain, Security risk analysis of RFID technology: A RFID tag life cycle approach, in: 2009 Wireless Telecommunications Symposium, 2009, pp. 1–7. doi:10.1109/WTS.2009.5068991.

[85] Xiao, Qinghan, Gibbons, Thomas and Lebrun, Hervé and Huo, Y, Rfid technology, security vulnerabilities, and countermeasures, Supply Chain the Way to Flat Organization, Publisher-Intech (2009) 357–382.

[86] A. Mitrokotsa, M. R. Rieback, A. S. Tanenbaum, Classifying rfid attacks and defenses, Information Systems Frontiers 12 (5) (2010) 491–505.

[87] Jun Yang, Pengpeng Yang , Ziyue Wang and Jianbin Li , Enhanced secure low-level reader protocol based on session key update mechanism for RFID in IoT, International Journal of Web and Grid Services 13 (2). doi:https://doi.org/10.1504/IJWGS.2017.083386.

[88] Rafik Kheddam, O. Aktouf and I. Parissis, An Extended LLRP Model for RFID System Test and Diagnosis, 2012 IEEE Fifth International Conference on Software Testing, Verification and Validation (2012) 529–538.

[89] S. Qadir, M. U. Siddiqi, Performance evaluation of a secure Low Level Reader Protocol(LLRP) connection, in: IJCSNS International Journal of Computer Science and Network Security, 2009.

[90] B. Zhao, Z. Wang, B. Cui, X. Liang, An Enhanced Secure Mechanism of Low Level Reader Protocol (LLRP) V1.1, in: 2014 Ninth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, 2014, pp. 475–480. doi:10.1109/3PGCIC.2014.97.

[91] Wiem Tounsi, Nora Cuppens-Boulahia, Frédéric Cuppens and Guy Pujolle , Access and privacy control enforcement in RFID middleware systems: Proposal and implementation on the fosstrak platform, World Wide Web (2016) 41–68.

[92] S. Figueroa, J. Añorga, S. Arrizabalaga, I. Irigoyen, M. Monterde, An Attribute-Based Access Control using Chaincode in RFID Systems, in: 2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS), 2019, pp. 1–5. doi:10.1109/NTMS.2019.8763824.

[93] S. Mamyun, Top 10 rfid security concerns and threats, `https://securitywing.com/` ((Accessed: 23-10-2020)).

[94] M. L. Chaim, D. S. Santos, D. S. Cruzes, What do we know about buffer overflow detection?: A survey on techniques to detect a persistent vulnerability, International Journal of Systems and Software Security and Protection (IJSSSP) 9 (3) (2018) 1–33.

[95] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, B. Sikdar, A survey on iot security: application areas, security threats, and solution architectures, IEEE Access 7 (2019) 82721–82743.

[96] L. Wüstrich, M.-O. Pahl, S. Liebald, Towards an extensible iot security taxonomy, in: 2020 IEEE Symposium on Computers and Communications (ISCC), IEEE, 2020, pp. 1–6.

[97] A. Nochvay, Security research: Thingspro suite – iiot gateway and device manager by moxa, `https://ics-cert.kaspersky.com/reports/2019/01/22/security-research-thingspro-suite-iiot-gateway-and-device-manager-by-moxa/` ((Accessed: 22-10-2020)).

[98] M. A. da Cruz, J. J. P. Rodrigues, J. Al-Muhtadi, V. V. Korotaev, V. H. C. de Albuquerque, A reference model for internet of things middleware, IEEE Internet of Things Journal 5 (2) (2018) 871–883.

[99] GS1, The gs1 traceability standard, https://www.gs1si.org/Portals/0/GS1_Dokumentacija/GS1_Resitve/Sledljivost/Global_Traceability_Standard.pdf ((Accessed: 6-6-2021)).

[100] GS1, Gdsn operations manual release 3.1, https://www.gs1.org/docs/gdsn/3.1/ ((Accessed: 23-10-2020)).

[101] C. Gupta, R. K. Singh, A. K. Mohapatra, A survey and classification of XML based attacks on web applications, Information Security Journal: A Global Perspective 29 (4) (2020) 183–198. arXiv:https://doi.org/10.1080/19393555.2020.1740839, doi:10.1080/19393555.2020.1740839. URL https://doi.org/10.1080/19393555.2020.1740839

[102] T. Goh, An in-depth look at xml document attack vectors, https://www.opswat.com ((Accessed: 26-10-2020)).

[103] GS1, Epc information services (epcis) standard, https://www.gs1.org/epcis/epcis/1-1 ((Accessed: 26-10-2020)).

[104] GS1, Epcis and cbv implementation guideline, https://www.gs1.org/docs/epc/EPCIS_Guideline.pdf ((Accessed: 26-10-2020)).

[105] H. Hasan, M. Ali, A modern review of edi: Representation, protocols and security considerations, in: 2019 2nd IEEE Middle East and North Africa COMMunications Conference (MENACOMM), IEEE, 2019, pp. 1–5.

[106] J. Betz, E. Jaskolska, M. Foltynski, T. Debicki, The Impact of Communication Platforms and Information Exchange Technologies on the Integration of the Intermodal Supply Chain, Springer International Publishing, Cham, 2020, pp. 131–141.

[107] J. Radko, As2, https://www.edibasics.co.uk/wp-content ((Accessed: 26-10-2020)).

[108] T. Caldwell, Securing small businesses – the weakest link in a supply chain?, Computer Fraud & Security 2015 (9) (2015) 5 – 10. doi:https://doi.org/10.1016/S1361-3723(15)30083-X.

[109] Ghadge, A., WeiB, M, Caldwell, N.D., Wilding, Managing cyber risk in supply chains: a review and research agenda, Supply Chain Management 25 (2) (2019) 223 – 240. doi:https://doi.org/10.1108/SCM-10-2018-0357.

[110] K. S. Anoop Singhal, Theodore Winograd, Guide to secure web services, https://nvlpubs.nist.gov/ ((Accessed: 26-10-2020)).

[111] C. Gidney, M. Ekerå, How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits (2019). arXiv:1905.09749.

[112] ICANN, Dnssec – what is it and why is it important?, https://www.icann.org/ ((Accessed: 26-10-2020)).

[113] NETSCOUT, What is an http flooding attack??, https://www.netscout.com/ ((Accessed: 26-10-2020)).

[114] Origintrail, First purpose-built protocol for supply chains based on blockchain, https://origintrail.io/storage/documents/ ((Accessed: 26-10-2020)).

[115] Ana Paula APPEL, Ricardo Vieira Borges FRANCO and Marisa Affonso VASCONCELOS, Detecting vulnerabilities in a supply chain, uS 2018 / 0197129 A1 (Jul. 18 2018).

[116] GS1, Gs1 global traceability standard, https://www.gs1.org/standards/ ((Accessed: 26-10-2020)).

[117] I. Stankov, G. Tsochev, Vulnerability and protection of business management systems: threats and challenges, Problems of Engineering Cybernetics and Robotics 72 (2020) 29–40. doi:10.7546/PECR.72.20.04.

[118] E. Quirk, How to keep your erp data safe from ransomware?, `https://solutionsreview.com/` ((Accessed: 26-10-2020)).

[119] M. Nunez, Cyber-attacks on erp systems, Datenschutz und Datensicherheit-DuD, Springer 36 (9) (2012) 653–656. doi:https://doi.org/10.1007/s11623-012-0220-5.

[120] Z. Yu, D. Jung, S. Park, Y. Hu, K. Huang, B. A. Rasco, S. Wang, J. Ronholm, X. Lu, J. Chen, Smart traceability for food safety, Critical Reviews in Food Science and Nutrition (2020) 1–12.

[121] Nnamdi Johnson Ogbuke, Yahaya Y. Yusuf, Kovvuri Dharma and Burcu A. Mercangoz , Big data supply chain analytics: ethical, privacy and security challenges posed to business, industries and society, Production Planning & Controldoi:https://doi.org/10.1080/09537287.2020.1810764.

[122] N. Subramanian, A. Jeyaraj, Recent security challenges in cloud computing, Computers & Electrical Engineering 71 (2018) 28–42. doi:https://doi.org/10.1016/j.compeleceng.2018.06.006.
URL `https://www.sciencedirect.com/science/article/pii/S0045790617320724`

[123] GS1, Gs1 digital link, `https://www.gs1.org/standards/gs1-digital-link` ((Accessed: 28-01-2021)).

[124] E. commerce Security Threats, B2c security threats, `http://bta301.weebly.com/b2c-security-threats.html` ((Accessed: 26-10-2020)).

[125] K. Kalkan, G. Gür, F. Alagöz, Filtering-based defense mechanisms against ddos attacks: A survey, IEEE Systems Journal 11 (4) (2016) 2761–2773. doi:10.1109/JSYST.2016.2602848.

[126] A survey on security and privacy of federated learning, Future Generation Computer Systems 115 (2021) 619 – 640. doi:https://doi.org/10.1016/j.future.2020.10.007.

[127] S. Singh, P. Sharma, S. Moon, J. H. Park, Advanced lightweight encryption algorithms for IoT devices: survey, challenges and solutions, J Ambient Intell Human Computdoi:https://doi.org/10.1007/s12652-017-0494-4.

[128] J. Kembro, D. Näslund, J. Olhager, Information sharing across multiple supply chain tiers: A Delphi study on antecedents, International Journal of Production Economics 193 (2017) 77 – 86. doi:https://doi.org/10.1016/j.ijpe.2017.06.032.

[129] J. W. Yingli Wang, Meita Singgih, M. Rit, Making sense of blockchain technology: How will it transform supply chains?, International Journal of Production Economics 211 (2019) 221 – 236. doi:https://doi.org/10.1016/j.ijpe.2019.02.002.

[130] R. Yanamandra, A framework of supply chain strategies to achieve competitive advantage in digital era, in: 2019 International Conference on Digitization (ICD), 2019, pp. 129–134. doi:10.1109/ICD47981.2019.9105913.

[131] M. R. Nichols, How to share data safely across your supply chain, `https://www.smartdatacollective.com/` ((Accessed: 20-11-2020)).