Applying LINDDUN Framework for Privacy Threat and Response Analysis in Disaster Relief System Architecture

Alex Lappin
Department of Cyber Security Engineering
George Mason University
Fairfax, VA, USA
alappin@gmu.edu

Matt Young
Department of Cyber Security Engineering
George Mason University
Fairfax, VA, USA
myoung45@gmu.edu

Sumayah Alomari
Department of Cyber Security Engineering
George Mason University
Fairfax, VA, USA
salomar3@gmu.edu

Mahi Khan
Department of Cyber Security Engineering
George Mason University
Fairfax, VA, USA
mkhan209@gmu.edu

Sherok Neamaalla
Department of Cyber Security Engineering
George Mason University
Fairfax, VA, USA
sneamaal@gmu.edu

***Abstract*—Every country deals consistently with natural disasters and tragedies that cost trillions of dollars in damage and millions of lives lost. During these disasters it is crucial to establish communication during a time when people are at most in need without internet, food, shelter, and basic resources. Due to the lack of telecommunication links between Crisis Management Centers (CMC) and those affected, there is a unique and innovative secure solution using drones which bring communications with an ad-hoc network style to relay messages. These logs and drones are monitored for repudiation and a secure algorithm. The current solution focuses on general security, whereas this paper aims to leverage the LINDDUN framework to accurately analyze the possible threats and vulnerabilities within the system's privacy and then provide valuable mitigation techniques with detailed threat modeling encompassing the scope of the secure solutions. In doing so, we realize that this system requires a multi-layered zero trust architecture (ZTA) while including user consent, regulatory obligations, and end-end encryption of Personally Identifiable Information (PII) through the recommended Privacy Enhancement Technologies (PET) that mitigate the lack of privacy by design.**

Keywords—Cyber-Security, natural disasters, disaster relief, Crisis Management Centers (CMC), LINDDUN, Privacy, zero trust architecture (ZTA), Personally Identifiable Information (PII).

I. INTRODUCTION

Natural disasters will never be a problem that can be solved, only mitigated through cooperation in the form of technology and communication. In many cases of disaster, traditional technological infrastructure proves to be insufficient due to the extremities in place. This calls for an additional solution that will take effect when everything fails.

Disasters have caused irreparable damage to many nations including Central and West Africa where 7.2 million people are currently affected by flooding and need disaster relief solutions [1]. Climate change is said to play a significant role in this disaster with an increase of global temperatures of just 2 degrees Celsius being a risk factor. In 2024 alone, flooding in West and Central Africa uprooted 1.1 million people and wrecked 642,000 homes across 13 nations which could cause regular torrential downpours of this magnitude annually. The flooding was enough to completely overwhelm all local disaster resources at hand and cause people in charge like Governor Babagana Zulum of Borno State to advocate for global assistance due to the humanitarian crisis causing loss of thousands of lives [2]. Beyond the toll taken on human individuals, 960,000 hectares of cropland were submerged which ruined the economic structure and food in which citizens rely on. This becomes a problem of technology when local resources are used up and the internet becomes unavailable after the flooding.

A smaller scale example of this same exact occurrence would be in Hunga Tonga-Hunga Ha'apai, an island in the South Pacific, where an underwater volcano erupted on January 15, 2022, and was so intense that the atmosphere was changed globally. These people were then impacted by a resulting tsunami and complete severance of internet due to the damage [4]. Around 85,000 people or 80% of the population were impacted by these events and about 600 structures were compromised making the urgency of the incident skyrocket. The direct damage bill rose to US $ 90.4 million which severely impacted the economic status of the nation.

A rise of only 10% in broadband access lifts GDP growth by 1.38% in developing nations, making resilient networks just as important to the economy as the humanitarian component [5]. The solution is to provide internet in the case of a disaster causing terrestrial internet implementations such as fiber optic cables, cell towers, and local routers to become

unavailable. Rescue teams will direly need this internet to know victim's whereabouts and situational status. Flooding will guarantee that this is the case since it has the most impact on ground electronics versus other natural disasters. Using drones to relay local traffic to responders using a lightweight protocol will provide this operational need of the internet.

A. Current Technical Limitations

Clearly current technical infrastructure is not capable of handling the extent of these disasters as seen in the case of Central and West Africa as well as the island of Tonga. In Africa, flooding caused damage to crucial terrestrial infrastructure points connecting submarine cables to onshore landing stations, which led to physical fiber optic disconnections. Additionally, there was water infiltration into cable landing stations, combined with debris and sediment movement from intense flooding which destroyed the integrity of the fiber optic links themselves. Countries including Nigeria, Chad, Niger, Mali, and Cameroon were all impacted by the severing of critical emergency communication channels and humanitarian efforts [3]. These outages illustrate the failure to meet the ≥ 99.99% service availability target as well as the Mean Time to Repair (MTTR) being ≤ 72 hours presented in ITU-T Y.1540 and ITU-T Supplement 35 (Network Resilience & Recovery – NRR) [6]. Although the area had some infrastructure already in place for this scenario, the sheer amount of flooding ultimately proved the terrestrial optic cable to be inadequate regarding redundancy and resilience.

This case is also reflected in our second example where Hunga Tonga–Hunga Ha'apai caused a huge tsunami after a volcanic eruption. The fiber optic cable line connecting Tonga to Fiji was destroyed, which was the main form of communication. A 55-kilometer segment of this cable was either buried beneath thick volcanic sediment or completely severed due to the eruption's intense seismic activity and resulting underwater landslides. The result was an inaccessible cable that was completely unfunctional. The Time to Restore (TTR) was 38 days, which is well above the ≤ 168-hour restoration objective recommended regarding submarine backbones in ITU-T L.1700 ("Low-cost resilient backbone networks") and L.1500 ("ICT adaptation to climate-change effects"). Repair efforts became challenging when specific underwater vehicles were required (ROVs) in relaying the groundwork causing the time to recover (TTR) to be much longer than anticipated [4]. Starlink was involved in deploying specialized ground stations which had to be sourced internationally but breached the E.108 requirement of availability after 24 hours. Transport constraints, damaged infrastructure, customs clearance processes, and the need for specialized personnel to oversee installation and operations also delayed the amount of time that this temporary solution was able to be implemented.

The proposed solution of a drone-implemented ad-hoc network will bridge the gap between the affected ground circumstances, the specialized personnel needing to be called in, as well as the victims themselves as they will use a personal device. Drones will provide reliability and scalability due to them being easily replaceable as well as rapid deployment which will satisfy ITU-T Supplement 35. MQTT will also conserve bandwidth and power in this IoT context without sacrificing reliability of transmitted messages, meeting the ITU-T Y.1540 end-to-end delay of < 150 MS and packet-loss of < 0.1 % threshold [6]. However, this solution currently does not support much security and privacy due to the lightweight protocol being used to link the victims' phone with the Crisis Management Center (CMC) [7]. This reliability is only the case when there is security in the context of privacy, thus the need for LINDDUN.

B. LINDDUN Framework for Threat Modeling

Due to the privacy requirements of the disaster response system utilizing MQTT-based drone communications in IoT, we need a structured approach, where the Linking, Identifying, Non-Repudiation, Detecting, Disclosure, Unawareness, and Non-compliance Framework (LINDDUN) is particularly suited to this scenario, since it integrates the necessary operational objectives into cybersecurity privacy planning, creating a relationship between the secure technical solution and privacy of each department within the disaster-response scenario.

LINDDUN provides a very detailed seven category threat methodology of starting with clear system goals about the underlying technical infrastructure and ties them to associated privacy threats and weaknesses that are uncovered. By modeling threat scenarios through DFDs, it creates clear guidelines for engineers of the system so that vulnerabilities and proactively discovered and dealt with early into the design process. Ultimately this should improve the detection and response rate of any cybersecurity issues which may impact the privacy of the disaster recovery and response system.

Given the need to adapt a security centric system to the needs of privacy within the disaster scenario, LINDDUN is the clear framework since PASTA and OCTAVE both focus on detailed methodologies within unsecure components and mitigations. The disaster system has a well implemented security system when looking at traditional threats regarding confidentiality, availability, and integrity, however until this point, the user's privacy has been a secondary concern. These differences are highlighted within Table 1.

Table 1: LINDDUN and PASTA Formal Comparison

Aspect	**LINDDUN**	**PASTA**
Assumes Security?	**Yes**, it works the best when applied on secure systems.	**No**, focuses in-depth on unsecure system components.
Use Case	Creating trust and transparency regarding legal compliance and privacy by design.	Enhancing audit results, within a high stress security environment.
Compliance	Legal and ethical compliance including; GDRP, HIPPA, or CCPA	Focusing on business risk and less on regulatory concerns.
Threats	Looks at internal threat actors and misconfiguration as well as external.	Looks primarily on external threat actors within the STRIDE context.
Output	Privacy-specific mitigations.	Mitigations for specific attack paths realized within STRIDE.

C. Objectives and Purpose

Regarding this solution, we will be utilizing LINDDUN to perform a systematic and detailed privacy threat analysis which will assess the privacy and security posture of the MQTT based drone communication system for disaster response. These objectives specifically include:

1. Mapping and identifying vulnerabilities specific to the disaster response system architecture in its entirety. This will include the users and their associated devices, the drones acting as publishers and responders, and the CMC cloud as an endpoint and hub for all communications.
2. Proposing and simulating realistic cyber-attack scenarios to evaluate potential impacts on communication integrity, confidentiality, and availability. This will also include authentication and authorization of proper individuals in the context of information privacy [7].
3. Recommending security improvements based on identified privacy risks to effectively mitigate potential privacy related attacks down the line. These will be accessed and procured from standard practice and implementations within the industry [7] [8].
4. Incorporation of business risk frameworks such as the NIST Risk management framework (RMF) to evaluate the risks from an organizational perspective. This includes the likelihood of threat impacts within critical objectives such that both technical and privacy concerns are properly addressed. This will also align with regulatory compliance within LINDDUN and enable business impact to be contextualized within the threats of privacy and security [8].

The conjunction of these security and privacy standard methods aims to ensure that the designed communication system remains secure, reliable, and trustworthy, which will create a more effective environment for disaster response teams to then respond to the crisis at hand without worry of cyber related problems on top of the already given problem. This will minimize financial damage, loss of life, and further improve infrastructure in place for disaster affected areas. This is a basic standard in IoT since there is usually a lack of privacy focus early in the design process and IoT can cause damage beyond the scope of normal software due to its interaction with the physical environment as well as associated regulations crashing down onto an organization [7].

II. System Architecture

The system is an innovative solution to ensure reliable, secure, and lightweight transmission within the event of a disaster where traditional forms of communication are unavailable. It uses sole air and elevated traffic transmission where terrestrial events that occur do not play a role in the reliability of the messages being sent. This IoT drone-based solution employs an ad-hoc aerial network of drones to facilitate communication. Victims that are being affected by the disaster, or in this case the flood, will utilize a smartphone application to send distress signals which include location and the severity of their situation via MQTT protocol. The messages are sent to drones safely nearby and then dynamically forwarded to the Crisis Management Center. At the CMC these messages are visualized on a platform called Grafana which enhanced the situational awareness of the response team based on the victims' needs. Additionally, there are secure components implemented within a separate analysis that ensure additional role-based controls and repudiation within the system.

A. System Purpose

The core objective of the project is to develop a relationship between reliable, secure, and private between the victims in a flooding scenario and rescue operations which need vital information. Terrestrial communication often fails within this context so there are additional measures in order to subvert this flaw. Traditionally there are things underground or underwater which have countermeasures against flooding, however, the failing rate of this is too high to consider usable. There are also efforts such as Starlink which require large teams to input groundwork first and cannot cover an area as large since the ground isn't necessarily available. This solution fills the need for crisis teams to be able to view information that will help the victim depending on their own personal situation without interference from disaster or technological flaws getting in the way. The solution implements this reliability within a secure context regarding the flow of data between external entities, drones, and databases. This original architecture is built from a security threat analysis that systematically applied PASTA through threat modeling and thorough mitigations to help create a more resilient version of the system, although it is missing the privacy component which is enumerated within section III.

B. Key Components and Architecture

This system operates as a secure yet lightweight traffic forwarding and management architecture which has some key technical components [8].

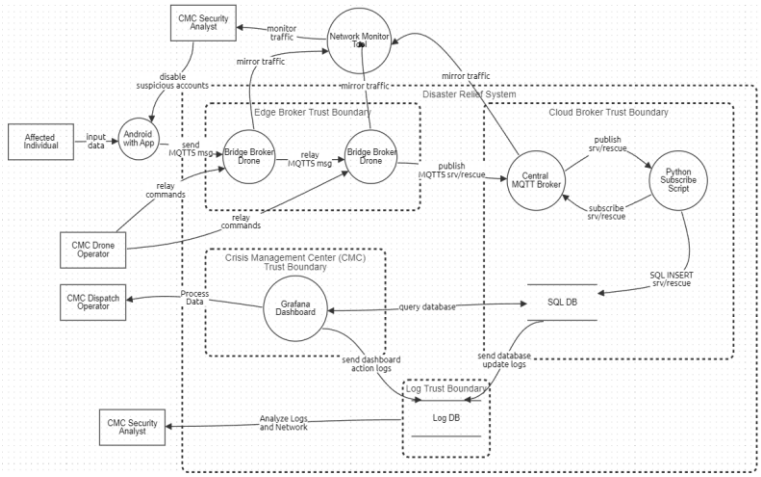

Figure 1: Entire System Architecture DFD

Figure 1 illustrates the entire system's key components working in unison with every stakeholder, boundary, process, data flow, and database included.

1. Stakeholders: These are all on the outside of the boundaries since they are considered external entities. They can be affected by outside influence and control and are usually considered the most vulnerable regarding security and privacy. These will also be the ones getting impacted if things malfunction.

   The victims here are the affected users in the disaster scenario who are non-technical users and

carry a smartphone with them. These are the people in need of rescue and will be preoccupied by the gravity of the disaster [2]. They will be publishing their own distress information which includes the levels of severity, low, medium, or high as well as the description of the condition they are in and the location of themselves in the form of GPS coordinates. They will also update their situation according to their needs and wants [8].

The emulated victim is less of a stakeholder itself but will mimic the stakeholder in a testing scenario for development purposes so they will be regarded in the same category as the victims themselves. These are simply coming from python scripts using Mininet WIFI which establishes false victims and drone nodes to forward similar information to the real scenario.

Finally, there is the CMC Operators, Dispatchers, and Security Analysts which are the people who are committed to helping the victims, whether volunteer or for a career. This includes the drone operators, who will be flying the drone bridge brokers around remotely from a command center and will be responsible for the correct positioning of these drones to properly capture the victims' traffic within an Ad-Hoc mesh network, as well as avoid the natural disasters at hand. There are also the operators responsible for the crisis management center aggregation and processing of data. These operators will be focusing on rescue planning and management based on the data flowing in from the Grafana dashboard. They will be looking specifically for trends or individuals in the most need and may also coordinate with drone operators and rescue teams to further assist them. The last operator of note is the CMC Security Analyst, who is responsible for analysis of the logging database to keep repudiation and historical records of the SQL logs which are flowing in and out as well as the Grafana network activity. They are additionally responsible for monitoring the traffic within the bridge and central brokers to keep security within these and disabling user accounts that violate any of the brokers themselves.

2. Processes: These are the ongoing software elements within the system that provide key roles in connecting stakeholders with one another based on the flow of data given. They are reprogrammable and must be resilient.

   The drone bridge broker is a lightweight MQTT (Message Queuing Telemetry Transport) broker running on Eclipse Mosquitto and Ubuntu within a drone that is able to take controls from an operator and cover aerial support. This is responsible for receiving the published messages from the victims and then forwarding them to the central broker, which is the subscriber. They may also be responsible for relaying rescue data to one another to then better reach the central broker. This will be using a topic, which is a virtual channel used to filter and route messages; thus, publishers and subscribers do not need to know each other directly. The MQTT protocol that it is running on is a machine-to-machine protocol designed to be lightweight and for IoT specifically [7]. This causes low bandwidth and power consumption. This broker specifically uses the bridging feature, which requires a hostname and credentials, to send it to the central broker. Once this topic has been republished or forwarded, the drone's duty has been completed by subverting any need for a cell tower.

   The Central broker is using the exact same MQTT technology, however it is in a much more stable technology of being in the cloud environment. This makes it easily reachable for all incoming messages from the drone bridge brokers. There will be multiple drones so there must be bandwidth and processing power available on this cloud broker. This requires high availability and scalability. It will be looking for authentication within its incoming bridge messages as well as the same hostname and credentials when the python subscriber reaches out to it. Once the subscriber has been confirmed to receive a topic, the central broker will then forward the MQTT data over to the python subscriber function.

   The network monitoring tool is responsible for the mirroring of all drone broker and central broker data and then processing it for viewing by a CMC Security Analyst. This will be done through Zeek technology which inspects the MQTT network traffic and can be scriptable to pick up on specific attack patterns. This is lightweight and ideal for connection in IoT ad-hoc networks and may even be connected to an existing IDS or SIEM.

   The Android application is used by victims to reach out to the drone system. It has a simple UI interface that allows the user to publish their location and distress severity in the form of an MQTT message which is sent to the nearest drone broker. These users will log in with individual accounts which are registered and viewable by the CMC Security Analyst for processing and possible termination given a violation of terms of service or malicious behavior.

   The python subscriber will have a mqtt_subscribe function which will be responsible for subscribing to the target topic in the central broker. Then to authenticate with the main broker. After a proper handshake, the central broker in the cloud will then be forwarding this MQTT data over to the python cloud lambda function. This will be a function called mqtt_publish where it accepts the data and then parses the significant fields such as the IDs, severity, location, and timestamp. Once this data is extracted it will persist into the SQL database by using a SQL insert expression. It will also be responsible for error handling within any missing logs or fields that the central broker has received [8].

Finally, the Grafana dashboard is an open-source data visualization and monitoring tool. This will be connected to the SQL database and be presenting real-time data in accessible formats for operators to read in the form of the ID, severity, geo-location, and timestamp. There will be a dynamic map where you identify the risk situation colors the node positions by severity, a bar chart shows the number of nodes classified by risk, and a bar chart shows the number of nodes classified by severity. It will be running on HTTPS/HTTP to display on operators' screens. This will provide operators with the necessary information to perform their duties to full efficiency [8].

3. Database: The main SQL database is a persistent storage of all the details forwarded by the python subscriber including IDs, severity, location, and timestamp. This allows for historical records of any published data as well as repudiation in the event of tampering. Operators may also use this to analyze the trends in an area over time. It is also responsible for real-time SELECT queries from Grafana where any amount of information may be taken out or modified. This is stored in the cloud somewhere to prevent any blackouts where even if local internet is down, the cloud is still active. There can also be calculations of KPI involved within this technology.

   The Log database is a central database within its own trust boundary. This database is used for storing Grafana action logs which are tied to a historical session and provide additional repudiation within the Grafana executable environment. These include operator button interactions as well as data accessed by the operator. Additionally, the SQL database sends update logs that inform a CMC Security Analyst of srv/rescue python script updates within the SQL database itself. This data is then used for forensics and anomaly detection within both of these boundaries.

4. Trust boundaries: These are the zones that transition one domain of security to another in a readable format and encompass everything aside from the stakeholders. This looks specifically at risks and authentication.

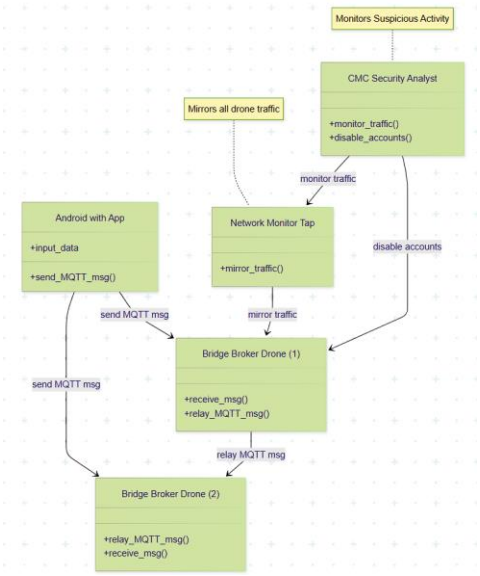

Figure 2: Edge Broker UML Trust Boundary

The edge broker trust boundary represented in Figure 2 is the ad-hoc mesh network of aerial drones that occupies the middle point between the victims and the cloud environment with the network monitor in between for security. It includes drone bridge brokers. This boundary focuses on authentication and authorization of smartphones, emulated and real, as well as accessibility to the cloud. This environment is the most physically dangerous since drones will be flying near disaster events and may receive physical harm out in the field. Data integrity is critical in this area since it can easily be tampered with or destroyed given any interference. Local untrusted networks will be involved out in the field as well as attackers that are far away from management, so physical compromising is a concern here.

The cloud broker trust boundary utilizes Software as a Service (SaaS) as well as Infrastructure as a Service (IaaS). This means that the resources are provided on the cloud without management of any hardware. It will be highly scalable and include plenty of redundancies, especially with a budget. It includes the central broker, python subscriber, and SQL database. All these processes will be looking for credentials from one another hence the longer process of receiving information without having the hardware private on site. There will also be a third party managing these so there must be trust in place regarding security and privacy. There will also be access controls in place regarding what can and cannot be subscribed to further prevent any excess load of messages or corrupted data.

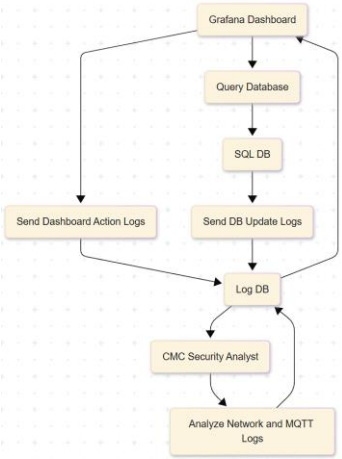

Figure 3: Data Logging SysML Trust Boundary

The logging trust boundary described in Figure 3 only contains the log database due to the special access restrictions that are only given out to a CMC Security Analyst. This emphasizes zero trust in the event of auditing or secure logging that cannot be historically modified. This is an example of separation of privilege between the CMC Dispatch Operator and the Security Analyst. The SysML diagram presents the exact workflow beginning with the Grafana dashboard receiving the SQL logs and displays the exact secure relationship that the logging database has within this boundary for added repudiation.

The crisis management trust boundary is the physical location boundary of the CMC headquarters. This will be occupied by CMC operators of all types as well as anyone else involved in rescue operations. This boundary only includes the Grafana dashboard process which will be locally hosted and connected to the cloud environment. There will be computers and hardware servers involved at this physical location as well as physical restrictions such as badges and guards to prevent unauthorized users from accessing the area. There may also be compliance with the local government here or any HIPAA since workers are involved.

5. Data flows: These are the actual streams of bits going over some sort of connection and are the heart of the operations' integrity. These connect processes to users and other processes as well as the SQL database.

   The MQTT data is originally sent from the victim to the central broker, and then to the python subscriber. MQTT transmits binary data in small packets. This technology makes it very lightweight since there are fewer bytes associated [7]. This TCP stack also allows it to run on something like a raspberry pi in cases such as drones. The packets have fixed headers and may contain an associated payload. This operates on a subscriber-publisher model where instead of HTTP, it uses a topic channel publishers send a message and subscribers receive a message. The topic is a string which serves as a routing key. The brokers will then use these topics to determine which subscriber will receive the message. This then works perfectly in an ad-hoc environment. In the case of this scenario, it will be carrying the data of IDs, severity, locations, and timestamps.

   The python subscriber will create an MQTT client in python using the python library Paho [8]. This will then prompt an authentication to be sent to the central broker. Once it has been accepted, there are callbacks set for on_connect and on_message. The script then attempts to connect to srv/temperature or srv/rescue which will be the topic in this case of the central broker [8]. It then takes the messages in on_message and decodes them and inserts them into the SQL database. The script will also be pulling these credentials from the SQL database itself.

   Grafana supports SQL queries by default and after configuration of connection information of the host, database name, credentials, and TLS settings, Grafana will be able to access the data from the SQL Server [8]. Once this is done it queries the data in real-time from the database itself. This is then ready for operators to view and interpret live. They may categorize the data or make trends as the situation evolves. They will also probably forward the data to other rescue operators based on their interpretations.

   The dataflow between the CMC Analyst and the drone would be through the MQTT data coming into Zeek. This data includes the MQTT topic, user, payload, timestamp of the message, and drone IP addresses. The verbosity of this data is crucial in monitoring drone traffic for any security threats and halts any blind spots within the field. The CMC Analyst would also be receiving a separate flow of data through the accessing of the log within the log trust boundary after confirmation of specific credentials regarding this role. This data would include any and all actions performed on Grafana within the form of PCAP traffic and a keylogger. The SQL database would also be included in this flow regarding any updates of MQTT traffic being transferred into it.

   Finally, drone operators will be using a local interface at the CMC. Since the drones are running Ubuntu, they can run a lightweight SSH command session which allows them to control the drone remotely after giving a proper authentication [8]. They will be navigating through flight paths and waypoints. The operators will also be modifying the Mosquitto file to set bridge parameters such as the topic, credentials, and bridge target.

C. Current limitations and Areas for Development

Despite the tight knit security and lightweight nature of the system as a whole, its biggest flaw revolves around the assumption that there will not be any personal privacy attacks or regulations included in the system. Strong TLS encryption and robust certification management will be an entirely difficult operation on its own and may even ruin the simple lightweight nature of MQTT. Bridge rules will also be equally frustrating to effectively manage since they are error prone

and may not be used by trained operators. Each broker is also prone to being a bottle neck since under high volume, lightweight nature will not help. The privacy of data is not safe when the data is not being properly encrypted. This framework is known to be insecure to many types of attacks or leaks within personally identifiable information due to these reasons [7].

The drones themselves are also an extremely fragile link within the system. Drones have limited battery life and flight time and act as a single point of failure. Without an excess of extra drones and skilled operators they will completely shut down the system if the signal is somehow lost. The weather is a huge variable that the drones need to operate properly as well as sunlight giving heavy restrictions. Data will be completely lost in this case with no recovery or redundancy. Keeping a proper ad-hoc multi hop connection between these drones is also not an easy task as this will be flakey and should be regarded going into the design process as such. Additionally, these drones will be out in the field and subject to anybody flying a counter-offensive drone or measure to capture and ruin the drone. Finding the proper number of operators in order to use these things also may prove challenging.

Finally, using Mininet as a substitute for drone operation and victim messaging will not prepare the operators for real world scenarios of attempts to steal personal information out of the drones. Additionally, it is notably complex to set up properly in a large-scale topology.

III. Threat Modeling Overview

The LINDDUN model is trying to solve the problem of privacy risks in software systems. This issue matters because many people today are using systems that collect, share, or store their personal data without them knowing exactly what's happening to that data. This creates concerns not just about security, but about transparency, control, and fairness. The problem goes beyond technical errors—it also affects people's rights and trust in digital services. As personal data becomes more central to how governments, companies, and apps operate, the way systems handle privacy becomes a critical part of responsible technology design. The issue is especially important in light of new privacy regulations like the GDPR, which require systems to handle user data carefully and give users more control. LINDDUN aims to help system designers, developers, and policy makers build technology that protects privacy from the beginning rather than trying to fix it later.

LINDDUN is different from traditional models that focus mainly on security threats. For example, STRIDE, one of the best-known threat modeling frameworks, looks at problems like spoofing, tampering, or denial of service—all of which are important but centered around keeping attackers out. In contrast, LINDDUN focuses on privacy-specific threats that can happen even without a malicious hacker. These threats include situations where users are identifiable without their knowledge, or where systems are collecting more data than necessary. It also covers whether users are properly informed about what happens to their data, and whether they have any say in the process. A system can be very secure but still violates user privacy, for example by secretly tracking someone's behavior or sharing their data with third parties. That's why LINDDUN focuses on privacy as a separate goal from security—because the two are related but not the same.

At the center of the LINDDUN method is the idea of looking at how data flows through a system. It starts by creating a data flow diagram, or DFD, which shows where data comes from, where it goes, and how it is processed. This helps identify points where personal data could be exposed or misused. Once the data flow is clear, LINDDUN uses a set of categories to look for specific types of privacy threats. These include things like whether people can be linked to their actions, whether they are identifiable, and whether they are aware of how their data is being used. The method then helps the team think through solutions, such as encrypting data, minimizing how much personal data is collected, or making users more aware of what's happening behind the scenes. This process allows privacy to become part of the system's structure, not just a policy passed on afterward.

LINDDUN also supports fairness and equity in technology design. Some privacy problems affect everyone, but others can affect certain groups more—especially if they have fewer choices or less understanding of how digital systems work. By helping teams look at the system from a user's point of view, LINDDUN makes it easier to see where certain users might be overlooked or exposed to higher risks. It supports privacy-by-design thinking, which means making sure the system gives people control, keeps them informed, and treats their data with care—right from the start. This is important because once personal data is leaked or misused, the harm is often permanent. LINDDUN's structured steps help prevent this by making privacy planning a normal part of software development, not just an afterthought for legal teams.

In summary, LINDDUN is a helpful model for making sure software systems respect people's privacy. It offers a way to look at privacy threats early, and helps teams fix those issues before they become real problems. It is different from other models because it doesn't just focus on attackers or technical errors—it looks at how the system affects users and their rights. It helps teams build systems that are both useful and trustworthy. In a world where digital services are growing fast and privacy concerns are rising, models like LINDDUN are becoming more important for creating technology that works for everyone, not just the developers who build it.

IV. Threat Modeling Applied

The strategic objectives are as follows:

- Provide reliable means for affected individuals to send locational data and status to the CMC personnel in an area of operation with limited to no communications infrastructure
- Ensure confidentiality, integrity, and availability of communication channels from affected user to bridge broker drone, bridge broker drone to bridge broker drone, and bridge broker drone to central MQTT broker
- Ensure confidentiality, integrity, and availability of backend cloud services
- Ensure compliance with privacy regulations/laws is upheld
- Maintain public trust in system through privacy by design principles and maintaining visibility transparency with users
- Protect privacy of users through a proactive and preventative posture, embedding privacy into the system design, having full functionality, providing end to end security, and fundamentally respecting user privacy

Beginning at the highest level of objective definition, strategic objectives describe the overall purpose and requirements of our system in a crisis scenario. From a functionality standpoint, the system needs to provide a reliable means for affected individuals in operation to communicate with Crisis Management Center (CMC) personnel via a pre-installed Android application. It is expected that the area of operation will have limited to no communication infrastructure due to the disaster, so the communication system needs to be easily deployable, redeploy able, and must utilize lightweight communication protocols to compensate for limited resources. The system needs to have a method of processing and storing data that CMC personnel can reliably access so that they can relay information to search and rescue units.

In addition to providing functionality, the system needs to possess security features that provide resilience and reliability in the event of malicious activity. This means ensuring the confidentiality, integrity, and availability of communication channels throughout the system. Namely the communication channels from affected user to bridge broker drone, bridge broker drone to bridge broker drone, and bridge broker drone to central MQTT broker. Data stored in the system must retain confidentiality, integrity, and availability to ensure operational effectiveness and public trust. Both internal and external threat actors must be considered in security design.

From a privacy perspective, the system's architecture needs to incorporate privacy-by-design principles. This involves protecting sensitive data, such location, status, or identifying characteristics, from unwanted disclosure by encrypting all important data flows from the time the user application captures them until they are stored in the backend database. Before giving their consent, users must be fully informed about the types of data being collected, how they will be used, and how long they will be kept preserving transparency and public trust. Users should be able to access and examine their stored data, among other rights under applicable privacy regulations. Additionally, audit procedures must be established to show responsible data management, and compliance with national and international privacy requirements (such as the CCPA and GDPR) must be guaranteed. This proactive, preventative approach guarantees that privacy is not considered an afterthought but rather as a core design objective, on par with usefulness and security.

Operational Objectives:

- Ensure reliable means of distributing app to affected individuals in area of operation
- Process and store data acquired from central MQTT broker in an SQL database
- Maintain database queries to store field data
- Ensure data processing and storage actions are resilient to tampering disruptions
- Ensure CMC dispatch operators can effectively and securely receive and relay situational data to first responder units
- Ensure CMC drone operators have secure and reliable means to operate bridge broker drone field assets
- Ensure asset manipulation tracking is implemented
- Ensure critical data flows containing sensitive data, such as between the affected individual and the central broker, have end to end security
- Apply pseudonymization and data minimization principles to prevent user privacy violations from external and internal threats
- Ensure unauthorized or excessive querying of sensitive data in the SQL database is prohibited to preserve user privacy
- Maintain audit trails/logs to keep CMC personnel accountable
- Ensure transparency with users of data collected and usage of collected data

Next operational objectives are defined; these state the system level functions that support the strategic objectives. From a functionality standpoint, this includes utilizing MQTT communication protocol due to its low overhead and low bandwidth and power requirements. The system must relay data–which will need to be processed via cloud-based services–from bridge broker drones to the central MQTT broker. All data is stored and queryable via an SQL database. The system must utilize a dashboard interface that presents a visualization of situational data for CMC dispatch operators to process. Furthermore, CMC dispatch operators must also coordinate search and rescue units. CMC drone operators must be able to reliably communicate to fielded drone assets for redeployment based on changes to operational needs.

From a security standpoint, the system must secure the MQTT protocol to ensure the integrity and confidentiality of messages from affected users. Communication channels from the CMC drone operators to fielded drone assets must also be secured to ensure resilience to external threats. The SQL database must have methods in place to mitigate attack via external incoming malicious data and internal CMC personnel manipulation. The system must also have methods in place to identify and track internal and external anomalous behavior.

From the standpoint of privacy, the system must guarantee that the situational and personal information it gathers from impacted individuals is treated with the highest care. To lower the danger of privacy infractions, this entails encrypting sensitive data at every stage of its lifetime and using data minimization and pseudonymization strategies. Strict access controls and usage limitations must be in place to safeguard any personal information kept in the SQL database, prohibiting excessive or unauthorized querying of sensitive fields. To hold CMC staff members responsible for their data handling procedures, privacy-aware audit logging should be put in place to document when, by whom, and for what reason data is accessed. Users' consent must be actively sought in accordance with legal and ethical norms, and they must be openly informed at the time of interaction about the types of data being collected and how they will be used. Additionally, users should be able to access or request the erasure of their personal data, if possible. These initiatives all support the system's public trust while guaranteeing adherence to pertinent data protection laws and humanitarian privacy standards.

Tactical Objectives:

- Ensure data processing and storage actions are resilient to tampering disruptions

- Ensure CMC operators cannot affect integrity of the database
- Utilize Python subscribe client in backend cloud services
- Implement hardening mechanisms in MQTT communication channels
- Develop security policies that define permissions, roles, and response plans
- Encrypt critical data flows to protect sensitive data related to user privacy
- Incorporate privacy focused logging
- Ensure appropriate access controls are in place for the SQL database
- Detailed information on data collection should be clearly provided to the user prior to gaining user consent
- Ensure users are allowed access to their stored data
- Maintain compliance with privacy standards and local privacy laws

Finally, tactical objectives define the needed functionalities that support real-time operations. These objectives ensure the system remains constantly operable to meet a larger goal. The system must utilize a publish/subscribe Python process that persists the messages received from the central MQTT broker to an SQL database. The system must then use Grafana as the dashboard service that visualizes the data for CMC dispatch operators.

From a security standpoint, the system must integrate methods of end-to-end encryption and authentication into MQTT to ensure confidentiality and integrity of the data. The system must ensure the central MQTT broker is resilient to a denial-of-service scenario which would negatively impact availability. The data coming from affected individuals must be checked for malicious content to prevent SQL database compromise. Methods of communication used between CMC personnel and field assets must implement encryption and authentication. A log system and intrusion detection system must be utilized to monitor anomalous activity and identify external attacks and insider threats in real-time. Security policies and settings must be put in place that follow industry standards.

All crucial data flows, especially those including sensitive field data or personally identifiable information, must be encrypted by the system to respect privacy standards at the tactical level. Logging systems should prioritize privacy by just recording the metadata that is required and preventing needless disclosure of sensitive information. The SQL database needs granular access restrictions that are rigorously enforced to guarantee that only authorized queries may retrieve sensitive data and to stop user data from being extracted in bulk or profiled. To obtain meaningful consent, the system must also give users comprehensive and unambiguous information about the types of data it collects, its intended use, and its storage duration. Users must have access to their stored data, if appropriate, and be able to request deletions or adjustments in compliance with their right to privacy. Last, every system component needs to abide by all applicable privacy standards and local legislation such as GDPR, and any data protection laws unique to a given nation which is mapped in Table 2. When taken as a whole, these actions support a privacy-by-design strategy that guarantees individual rights are upheld even in demanding, mission-critical settings. Table 3 further details the business side which provides an entire picture into the operational objectives.

Table 2: Regulatory Mapping

Regulation	Requirement	Impact on design
GDPR	Minimize PII collection	Apply data minimization and pseudonymization to field-level user data
GDPR	Provide clear consent and data usage transparency	Present data collection notices in the Android app before collecting user data
GDPR	Enable user data access and correction rights	Implement user interface or endpoint for data access requests
GDPR	Ensure end-to-end security of user data	Encrypt all critical data flows, especially user-to-broker communication
NIST 800-53	Access control and system integrity	Enforce RBAC and implement database access logging
NIST 800-53	Maintain audit trails and system accountability	Log data creation, modification, deletion, and access events in a tamper-evident way
ISO/IEC 27001	Secure processing and data lifecycle management	Use encrypted storage, define data retention policies, and ensure secure deletion

Table 3: Business Impact Matrix

ASSET	CIA PRIORITY	BUSINESS IMPACT IF COMPROMISED
Affected Individual	High (C)	Identity or location data leaks violate user privacy and potentially expose subject to harm
CMC Dispatch Operator	Medium (I/A)	Tampering with data could compromise mission completion. Could violate privacy of users by exposing/collecting stored sensitive data
CMC Drone Operator	Medium (I/A)	Loss of control to fielded drone assets

		could lead to mission failure
CMC Security Analyst	Medium (C/I)	Compromised security analyst could disclose traffic/logs being monitored, violating user privacy
Android with App	Low (C/I)	Compromised user device could lead to user data disclosure, but component is outside of system scope and ability to reliably control
Network Monitoring Tool	Low (I/A)	Misconfiguration/improper use can lead to missed detections; security can be placed on the third-party vendor
Bridge Broker Drones	High (A/C)	Presents a critical point in the communication channel, compromization could lead to false data relaying or data exposure, limiting mission effectiveness and violating privacy
Central MQTT Broker	High (A)	Central broker failure could halt data flow and lead to mission failure
Python Subscribe Script	Medium (I/A)	Compromise could result in malicious data being inserted into SQL database or failure to process incoming data
SQL Database	High (C/I)	Compromise of storage unit could result in exposure of sensitive user data or tampering of data to limit mission effectiveness
Log Database	Medium (C/I)	Compromise of logs could lead to cover up of attacks
Grafana Dashboard	Medium (C/I)	Compromise could lead to unauthorized entities viewing sensitive data and violating user privacy

A. LINDUNN STAGE 1: Describe the Existing System

In this section, the technical scope of the system is defined by identifying components involved in data transmission, processing, and visualization in the IoT architecture. First it is important to establish trust boundaries within the system. There are three main trust boundaries (not including the external boundary) within the Disaster Relief System, those being the Edge Broker, the Cloud Broker, and the CMC. The initial component in the system is the Android device with the Disaster Relief System application installed. This component is technically outside the system's external trust boundary and acts as the initial data sender. Data flows from this external source into the Edge Broker trust boundary. Within the Edge Broker trust boundary there are bridge broker drone components. They act as sensors in the system, collecting data from the external Android devices in operation. There are multiple bridge broker drones within this boundary since they communicate with each other to forward MQTT messages to a central MQTT broker located in the Cloud Broker. During this process data flows from the Edge broker trust boundary into the Cloud broker trust boundary. It is important to note that throughout the process of the affected user's Android sending data and the central MQTT broker receiving it, the MQTT protocol is used. Eclipse Mosquitto is the message broker that implements the MQTT protocol. Also, there is an external component which is a network monitoring tool, which actively monitors the exact traffic that flows between the bridge broker drones and the central MQTT broker. That traffic is then analyzed by a CMC security analyst to look for any unusual patterns, signals or data coming from any suspicious device from the field. From this point on in the system, the MQTT protocol is not being used.

Within the Cloud Broker Boundary there are three components: the central MQTT broker, the Python subscribe script, and the SQL database. When the central MQTT broker receives data, it is then forwarded to the Python subscribe script through a publish/subscribe process. The nature of the publish/subscribe relationship means that data is flowing bidirectionally between the central MQTT broker, and the Python subscribe script. The data, then persisted to the SQL database. From there the data flows from the SQL database trust boundary to the Grafana dashboard located within the CMC trust boundary. This is done via SQL queries. There is also the log database component which stores logs of the conversations that are actively taking place between the Grafana dashboard and the SQL database. The log database collects what is being updated in the SQL server through the Grafana dashboard or from the Python subscribe script. It also collects what is being sent to the Grafana dashboard from the SQL database. Once these logs are collected in the log database, a CMC security analyst then analyzes the logs and looks for any suspicious or unusual log activities and traffic patterns. To note, a CMC security analyst is considered an external actor.

Data flows from the Cloud Broker trust boundary into the CMC trust boundary when Grafana dashboard queries the SQL database. Grafana dashboard helps CMC personnel visualize, and process data related to location and status of affected individuals through an interface. The Grafana dashboard is the only component within the CMC trust boundary. As previously mentioned, CMC personnel are the ones accessing the Grafana dashboard, so the data flow ends when it exits the CMC and external trust boundary and flows to the CMC operator who is considered an external actor. Additionally, a CMC drone operator is sending commands to fielded drone assets. This CMC drone operator is classified as an external actor to the system.

To summarize, the components of the system that are identified in the technical scope are listed in Table 4 with their respective boundaries. This helps us to better understand the components that may have security concerns and how they have their place within the system.

Table 4: System Components

COMPONENT	BOUNDARY	DESCRIPTION
Android Device	External	Sends affected user data to the nearest drone
Network Monitor Tool	External	Monitors traffic that is being captured by drone and the traffic flow that goes until Central MQTT Broker
Bridge Broker Drones	Edge Broker	Collects data and sends the information to the MQTT broker
Central MQTT Broker	Cloud Broker	Receives information from the drones and sends it to the database
Python Subscribe Script	Cloud Broker	Service within the cloud that receives data from the central broker and forwards to SQL database
SQL Database	Cloud Broker	Database where all the data from MQTT broker gets stored
Log Database	Log	Database where all the updated logs from Grafana dashboard and SQL database is being stored
Grafana Dashboard	Crisis Management Center (CMC)	Dashboard that visualizes data from SQL database

As for the technologies that are being used with their appropriate dependencies, they can be categorized into five layers. The first layer is the front-end portion, where React and the Android app are being used, throughout the entire operation. JavaScript ecosystem and mobile OS API are the dependencies for this layer, since they are directly dependent on React and the Android app. Second layer is the backend layer, where we are using Python as our main language and Wireshark as our network monitoring tool. To use Python to operate the entire backbone of the operation, the Python libraries and runtime environments become dependencies of the technology in the second layer. As for Wireshark, dependencies are C runtime library, GLib, libcap and QT. Communication is a crucial part of the operation, which is our third layer. Technologies like MQTT Protocol and Eclipse Mosquitto are being used in our system, since these are network components that are being used, network libraries can be considered as dependencies of these components. Fourth layer is the Data Storage layer, where MQTT is uploading data in SQL databases in our system, Grafana Dashboard and SQL Database storing logs into log database and since SQL databases are being used, our key technology here is MySQL. To operate SQL databases using MySQL, SQL components like SQL connectors and cloud storage are being used as the dependencies of MySQL. This is because the database for our system is in the cloud and would require SQL connectors to connect to the database and storage to store all the data. Our last layer is the dashboard, where Grafana is being used as a Graphical User Interface for users to read and interact with the data from the database. The Data Source Plugins allow the user to connect and interact with the data source, since they act as bridges which allow Grafana to retrieve and display data from the database. Therefore, data source plugins are the dependencies of Grafana. Table 5 showcases the layers and technologies that are being used within the layers and their dependencies.

Table 5: Layers of the System

LAYER	TECHNOLOGIES	DEPENDENCIES
Frontend	React, Android App	- JavaScript ecosystem - Mobile OS API
Backend	Python, Wireshark	- Python libraries - Python runtime environment - C runtime libraries - GLib - libcap - Qt
Communication	MQTT Protocol, Eclipse Mosquitto	- Network libraries
Data Storage	MySQL	- SQL connectors - Cloud storage
Dashboard	Grafana	- Data source plugins

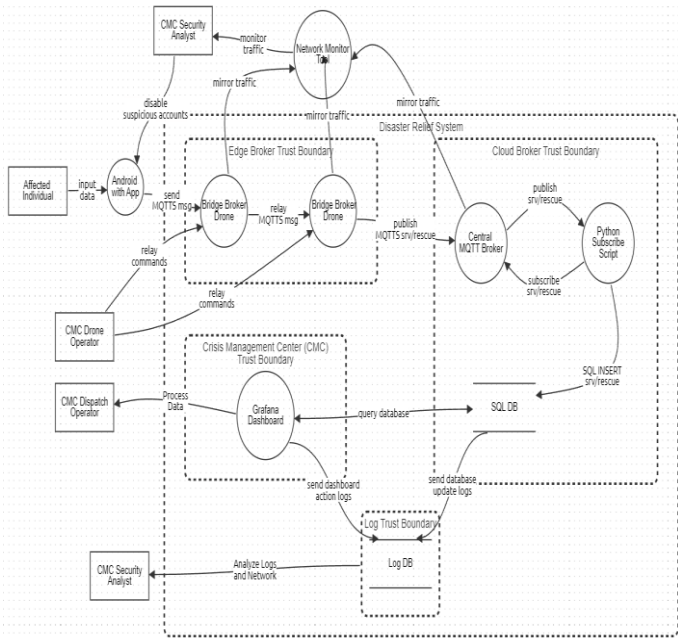

Figure 4: DFD of the System

In this section, the system is broken down into assets that are identified and categorized which are detailed in Figure 4. This mainly consists of going into more detail of how components work, how the system views them from a trust perspective, and how different users will interact with them.

The Android app is used by affected individuals to input a severity level and their geographical location which can later be reviewed by CMC personnel. It is assumed that affected individuals in operations will have this app installed before the disaster. To connect to a local fielded drone asset acting as a bridge broker, the app uses the bridge broker's IP address and port to connect. This will most likely be done either automatically as the drone gets within range or there will be another method relaying that information to the affected individual for manual input. The user can then enter positional data latitude and longitude format. The user can then select a severity level ranging from either low, medium, or high. The Android device then publishes a message containing the user's device ID, severity, and location via MQTTS protocol (TCP/IP) every 10 seconds. The user can also update information as needed.

The app is considered an external untrusted component to the system and represents a primary entry point into the system's data flow. In addition, the data contained within the message sent to the bridge broker drone can be classified as PII and thus is considered sensitive information. Considering the nature of data that is being sent, maintaining a secure communication channel is vital to maintaining user privacy. To that end, all data is sent via MQTTS protocol for secure communication.

The bridge broker drones are deployed in groups over an area of operation and serve as relays for incoming messages from affected individuals. They utilize MQTTS protocol for communications and implement this through Eclipse Mosquitto, acting in bridge mode. This means they receive messages from a user node, the Android devices, and forward them to a central MQTT broker. In an operation, the bridge broker drones are remotely given instructions by a CMC drone operator who positions them over affected individuals.

Bridge broker drones are considered an internal trusted system component located within the Edge Broker trust boundary. They are forwarding messages that contain user PII and possess a direct link to the Cloud Broker trust boundary. MQTTS continues to be utilized to maintain confidentiality of data in transit.

The central MQTT broker is the primary node that receives all data from the field via the bridge broker drone network. It is located within the Cloud Broker trust boundary and marks the entry point to the boundary. The central MQTT broker publishes data to clients within the cloud with a subscribe/publish and topic approach. Essentially the clients receiving data subscribe to a topic string which filters what information the central broker will publish to them. In the case of the system, this topic would be srv/rescue. Through this architecture the central MQTT broker acts as a message router for the Python subscriber script.

The central MQTT broker is considered an internal trusted system component. The contents of the data being received is user PII and the component itself is the only entry point for data from the field to the Cloud Broker boundary. This makes the component a high priority target for malicious actors since it acts as a major entry point that can either be bypassed or blocked. Adversaries may attempt to use this attack surface to compromise system confidentiality, integrity, and availability which would lead to a privacy breach. It can be expected that many attacks will be directed at this component to either access stored data or disrupt the flow of data.

In addition to the MQTTS protocol used in the communication line from the Android App to the central MQTT broker, the system has a CMC Security Analyst operating a network monitoring tool. This network tool monitors live traffic flow for the security analyst to detect anomalies. The security analyst can then delete suspicious user accounts that are identified as threats and report anomalies. Security vulnerabilities of the network monitoring tool are transferred to the tool's respective vendor. It is also important to note that due to the content of the data being monitored, non-compliant behavior from the security analyst could put user privacy at risk.

The Python subscriber script is a trusted component within the Cloud Broker trust boundary that processes data from the central MQTT broker which the script is subscribed to. The received message is parsed for device ID, coordinates, and severity level. Data is sent to the SQL database using an SQL insert operation. The Python subscriber script itself does not represent a major attack surface for adversary actions. It can be seen more as an obstacle that must be bypassed to reach the SQL database. The component does not have a public facing interface and is not an entry point. By utilizing best practices such as input validation/sanitization, the SQL database is protected from SQL injections that would cause a breach that could result in data tampering or disclosure.

The SQL database is a trusted component within the Cloud Broker trust boundary that serves as a data storage unit and is implemented via MYSQL. It receives the processed data from the central MQTT broker and can be queried by the Grafana dashboard by CMC dispatch operators. The database stores the device IDs, positional data, and severity level of all affected users in the area of operation. Because of the queryable nature of the component, it can be considered a controlled entry/exit point from the Cloud Broker trust boundary to the CMC trust boundary.

Similar to the Python subscriber script, the SQL database is not a public facing component and is not directly exposed to external users. Due to the high value data being stored it represents a critical asset in the system's operation. Adversaries may attempt to compromise confidentiality and integrity of the component which would degrade operational reliability. However, by implementing server-side input verification/sanitization, as outlined in Database Query String Analysis, the risk of an SQL injection scenario is mitigated.

The Grafana dashboard is the final major component of the system. It is located within the CMC trust boundary and acts as the interface between the CMC dispatch operator and the SQL database. The dashboard visualizes data to assist the dispatch operator in processing data and make decisions on where to allocate search and rescue units. Grafana utilizes a plugin that allows it to query MYSQL data sources, such as the SQL database in the cloud. The Grafana dashboard can be considered an entry point into the system and the CMC boundary and is considered trusted. The external users that are able to interface with the dashboard are CMC personnel who are currently considered external but trusted users.

The Grafana dashboard is not exposed to public external networks but interfaces with sensitive backend systems. For the system design, it is assumed that the only users who can access the dashboard will be onsite CMC staff. There is no mention of remote workers accessing the dashboard. Due to the interaction with the SQL database, it is possible for insider threat actors to breach the system. However, there is a log database which stores Grafana dashboard and action logs in real time and updated logs from the SQL database. The log database is under its own log boundary, which is under the entire system, but the CMC security analyst is a trusted third-party actor who actively looks for anomalies within the logs that are being stored.

In addition to identifying components and entry points that potentially face threats, it is important to identify system users. This means identifying roles and permissions of the users. The users of the system are currently the CMC dispatch operator, CMC drone operator, affected individuals and CMC security analysts for network monitoring tool and log database.

The CMC dispatch operator is considered an external but trusted user whose role is to process data and allocate search and rescue units. Their current permissions are to interface with the Grafana dashboard and query the SQL database. It is important to note that they are expected to read from the database and not modify it. The CMC security analyst for network monitoring tool, is also considered an external but trusted user who is expected to monitor and detect any anomalies within data and the data flow from drones receiving messages to the central MQTT broker. Their current permissions are to disable any suspicious accounts that may be sending in suspicious messages to the drones and report the anomalies within the data flow. Another third party trusted user is CMC security analyst for log database, their role is to monitor the action logs from Grafana dashboard that are being sent to SQL database and the updated logs that took place within the SQL database. Their permission is to report any unusual behavior pattern within the logs and report them to the organization. Given these components we begin with labelling the threat targets for further analysis in Figure 5.

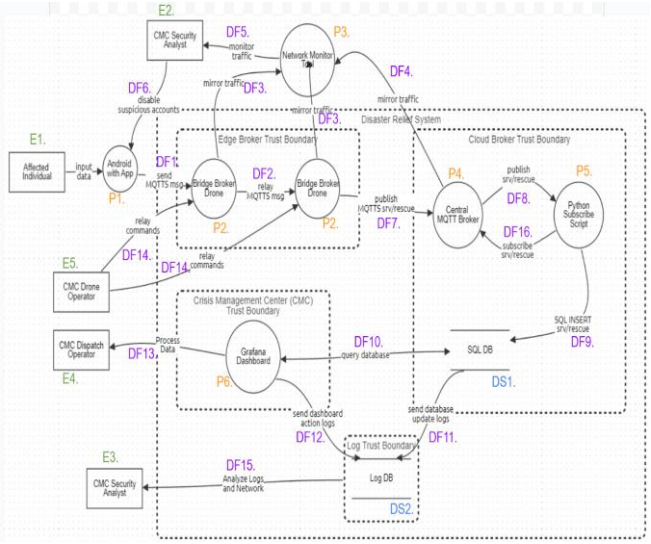

Figure 5: DFD with Threat Targets

B. LINDUNN STAGE 1: Framework

Table 6: LINDDUN Threat Mapping of System

	Threat Target	L	I	N R	D	D D	U	N C
Data Store	DS1. SQL Database	X	X		X	X		
	DS2. Log Database	X	X	X	X	X		
Data Flow	DF.1 (Android App - Bridge Broker Drone)	X	X	X	X	X		
	DF.2 (Bridge Broker Drone - Bridge Broker)	X	X	X	X	X		
	DF.3 (Bridge Broker Drone - Network Monitor Tool)	X	X	X	X	X		
	DF.4 (Central MQTT Broker - Network Monitor Tool)	X	X	X	X	X		
	DF.5 (Network Monitor Tool - CMC Security Analyst)	X	X	X	X	X		
	DF.6 (CMC Security Analyst - Android App)	X	X	X	X	X		

	DF.7 (Bridge Drone Broker - Central MQTT Broker)	X	X	X	X	X		
	DF.8 (Central MQTT Broker - Python Subscribe Script)		X		X	X		
	DF.9 (Python Subscribe Script - SQL Database)		X		X	X		
	DF.10 (SQL Database - Grafana Dashboard)	X	X		X	X		
	DF.11 (SQL Database - Log Database)		X		X			
	DF.12 (Grafana Dashboard - Log Database)		X		X	X		
	DF.13 (Grafana Dashboard - CMC Dispatch Operator)	X	X		X	X		
	DF.14 (CMC Drone Operator - Central Broker Drone)	X	X	X		X		
	DF.15 (Log DB - CMC Security Analyst)	X	X		X	X		
	DF.16 (Python Subscribe Script - Central MQTT Broker)		X		X	X		
Proce ss	P.1 Android App	X	X		X	X		
	P.2 Bridge Broker Drone	X	X	X		X		
	P.3 Network Monitor Tool (NMT)	X	X	X		X		
	P.4 Central MQTT Broker	X	X	X	X	X		
	P.5 Python Subscribe Script	X	X		X	X		
	P.6 Grafana Dashboard	X	X	X	X	X		
Entity	E.1 Affected Individual	X	X				X	
	E.2 CMC Security Analyst (NMT)	X	X					
	E.3 CMC Security Analyst (LDB)	X	X					
	E.4 CMC Dispatch Operator	X	X					
	E.5 CMC Drone Operator	X	X					

The LINDDUN framework, as mapped in Table 6, is an organized approach to modeling privacy threats that aids analysts in locating and reducing privacy threats in software systems. Through an analysis of data processing, storage, and transmission throughout a system, LINDDUN directs the technical and regulatory analysis of possible privacy issues. In contrast to security-focused frameworks like STRIDE or PASTA, LINDDUN ensures that systems manage personal data in a fair, transparent, and legal manner by concentrating exclusively on safeguarding user privacy rights. Architectures are frequently aligned with data protection laws such as the GDPR at an early stage of system design.

DS1 SQL Database stores sensitive and structured operational data such as drone commands or incident reports because repeated items such as drone IDs can be linked to profile behavior, which raises linkability concerns. The storage of sensitive data, such as operator credentials or GPS-linked data, without sufficient confidentiality puts identifiability at risk. If records cannot be changed or deleted, non-repudiation can unintentionally surface, subjecting people to unfair suspicion without enough time to defend themselves. The significance of detectability stems from the possibility that attackers or unauthorized users may utilize timing or query replies to deduce the existence of specific data records. Sensitive operational data may leak if the database is compromised or improperly accessed, making information disclosure a serious risk. Last, failure to implement appropriate access control, data retention, and consent methods may result in non-compliance with requirements such as the GDPR.

The system and activity logs kept in the Log Database DS2 are essential for diagnostics, security monitoring, and auditing. Recurring user behaviors patterns that can be connected over time to certain people give rise to linkability issues. When logs include IP addresses or other information that could be used to uniquely identify a system, identifiability

becomes a problem. Here, non-repudiation is frequently done on purpose, but it can lead to excessive traceability if privacy protections are not balanced with it. Because even encrypted logs can show the time and structure of events, potentially indicating sensitive activity, detectability exists. Since the log database frequently includes data, information disclosure is very likely if it is not adequately safeguarded. Last, non-compliance risks include keeping logs longer than is permitted by law which can result in privacy law violations.

The DF1 data flow entails communication between the Bridge Broker Drone and the Android App, most likely sending user inputs or commands. Because repeated interactions can be linked to certain users or drones, allowing behavioral profiling, linkability is a risk. When passwords, device fingerprints, or unique user identifiers are sent without encryption or anonymization, identifiability occurs and may reveal sensitive information. Transmitted messages that are logged or encrypted create permanent records that may be used against users in the future, making non-repudiation relevant. If an attacker can see or deduce the flow through traffic analysis, for example, indicating when the application is being utilized or when orders are being given, then detectability is applicable. If the data in this flow is not encrypted, disclosure of information poses a risk since it enables attackers to intercept critical operational data. Lastly, if this flow sends sensitive information without following legal requirements for security, consent, or purpose limitation, non-compliance hazards arise.

Through the DF2 data flow, the Bridge Broker Drone sends status updates and drone-collected data to the central Bridge Broker for further distribution and coordination. If several transmissions can be linked to the same drone or action, linkability is important because it allows opponents to map behavior patterns. If the drone contains data that can be linked to certain drones, including GPS locations, device IDs, or mission identifiers, identifiability issues could occur. If the data is logged using integrity measures such digital signatures, non-repudiation concerns arise, making it more difficult for operators to later dispute actions or the origin of the data. If the flow is not encrypted, it is detectable, which makes it possible for an attacker to identify drone-to-broker communication and deduce operational phases. Information disclosure is crucial because if this flow is intercepted, it may expose operational telemetry or key mission details. Threats of non-compliance arise when operational location for instance is transferred without the proper legal justifications or protections, in violation of internal policies or privacy laws.

For real-time monitoring and threat detection, the DF3 data flow entails sending drone network activity data from the Bridge Broker Drone to the NMT. If recurring network data patterns can be connected to certain drone flights or routes, linkability becomes important. The inclusion of identifiers that make it possible to identify specific people or systems, such as drone IDs, operator credentials, or network addresses, creates identifiability risks. If logs from this data flow are utilized as forensic evidence, they can be used to link specific acts to a drone or operator, making non-repudiation risky. If the traffic is unencrypted or exhibits recognizable patterns, detectability issues arise because attackers can determine when surveillance is in progress. Information disclosure is important because this flow may contain sensitive operational data and may be used against you if it were made public. As for non-compliance, it is possible to happen with transferring critical data without managing them appropriately in complying with internal governance or privacy legislation

From the Central MQTT Broker, the DF4 data flow sends MQTT messages to the NMT, including data such as drone status and system alarms. If an attacker is able to link repeated MQTT messages to certain drones, operators, or missions over time, linkability issues become obvious. If the payload contains distinct identifiers such as device IDs that allow for direct linkage with specific devices, identifiability is at risk. If logs from this flow are utilized as proof of actions without giving users a chance to dispute them, non-repudiation can be accidentally introduced. If an attacker can deduce activity patterns without reading the message content by looking at traffic volume or time, then detectability is important. If sensitive mission data are sent without proper encryption or access controls, disclosure can become a serious problem. If privacy laws like the GDPR are broken because of improperly kept communication records, non-compliance may result.

The CMC Security Analyst receives the DF5 data flow from the NMT, which includes processed network activity and system warnings. If activity patterns across sessions can be linked to specific people or devices, linkability becomes an issue. If logs or warnings contain device ID or IP addresses that directly link a device's actions to them, then identifiability occurs. If these reports are used to hold people accountable without providing them with a means to contest the data, non-repudiation could turn into a privacy concern. If attackers are able to deduce certain analytical patterns or continuous surveillance from system behavior, then detectability is a concern. If the transmitted reports contain sensitive or poorly protected data, information leakage poses a serious risk. If the flow includes sensitive data that is exchanged or stored without authorization or appropriate privacy safeguards, non-compliance becomes a problem.

This data flow DF6 involves the CMC Security Analyst using the Android App to send the end-user alerts, threat reports, or instructions. If certain updates or messages may be linked to devices or usage trends, linkability becomes an issue. Messages that contain specific identifiers or are specially customized in a way that discloses identity run the risk of being identified. In the absence of explicit tracking or consent procedures, non-repudiation may inadvertently subject people to accountability. If external actors can determine when the app is receiving crucial updates, detectability is important since it may disclose operating behaviors. If the app receives sensitive operational data or classified threat information over an unsecured or incorrectly authenticated channel, information sharing is crucial. If this flow transmits operational intelligence without complying with data handling guidelines or privacy laws, non-compliance might follow.

This data flow DF7 shows how messages are delivered from the Bridge Broker Drones to the Central MQTT Broker, these messages could include commands or status updates. If recurrent drone communication patterns can be linked to certain missions or drone identities, linkability emerges and behavioral analysis is possible. If distinct drone IDs or data can be linked to a particular operating mission, identifiability becomes an issue. If the broker enforces strict authentication logs without authentication, non-repudiation could jeopardize the confidentiality of the operator or drone. An attacker can determine operational timing or drone presence in a certain area if they can detect when a drone is communicating, raising a detectability issue. In this flow, information disclosure is crucial because if important information such as coordinates are sent without encryption, they may be made public. Finally, if the drone-to-broker data exchange does not adhere to security and privacy laws, including those related to secure

transmissions for example using MQTT vs. MQTTS, non-compliance may ensue.

A backend Python script receives data from the MQTT broker in order to process or store it in the data flow DF8. The ability to correlate signals from several devices gives rise to linkability, which may disclose usage behaviors [7]. When messages contain distinct device IDs or other data that can reveal operator's identity, identifiability issues arise. Unintentional accountability could arise if the messages can be linked to sources without appropriate confidentiality, leading to non-repudiation problems. If an observer can determine when important messages are being sent, detectability is important. If messages include sensitive sensor readings or command data and are not encrypted during transmission, there is a significant danger of information exposure. As for non-compliance, it may arise if regulated operational information or personal data is handled without following privacy and security regulations.

The backend software writes processed MQTT messages into the SQL database for analysis and storage as part of the data flow DF9. To expose behavior across time, linkability is important if many entries may be linked to the same device or session. If the data includes network identifiers or device serial numbers that link records to devices, identifiability becomes a concern. If the data inputs can be linked directly to a person or device, making it impossible for parties to dispute involvement or origin, then non-repudiation is applicable. If an attacker keeps an eye on database activity and determines when sensitive operations occur based on data frequency or content size, detectability may occur. If the script contains raw or unencrypted data, such as command logs, that could be abused in the event of a database breach, information disclosure is a serious risk. If sensitive or regulated data is kept without appropriate encryption, access restrictions, or data retention guidelines in line with GDPR or other legal requirements, non-compliance can occur.

Privacy considerations are crucial since the data flow DF10 makes it possible to visualize stored data from the SQL database on the Grafana dashboard. If visible data enables attackers to correlate several datasets and monitor device activity over time, linkability becomes an issue. Dashboards that display device IDs or other identifiers run the risk of exposing identities known as identifiability. If logs or charts offer indisputable proof of a user's actions non-repudiation may be affected. If delicate patterns or infrequent occurrences are obviously shown on the dashboard, indicating private processes or states, detectability problems could arise. If the dashboard is not access-controlled, there is a significant danger of information disclosure, which could expose data such as commands. Lastly, if visualized material contains regulated data without adhering to the relevant privacy and governance standards, non-compliance may ensue.

The DF11 data flow involves transferring data from the SQL database to the log database for archival or auditing purposes. Linkability is established if it is possible to recreate device behavior across sessions by cross-referencing log database information. Identifiability is compromised if logs reveal certain identities. Non-repudiation becomes important if logs are used to hold people accountable without their knowledge or express permission. Detectability issues may occur if logging reveals the times at which high-value actions were performed, allowing sensitive workflows to be traced. Information disclosure is a serious issue, particularly if logs contain sensitive requests or raw data that hasn't been sufficiently secured. If the log database contains data without the appropriate safeguards or auditability mandated by regulations, non-compliance issues arise.

System logs are retrieved from the Log Database by Grafana Dashboard for analysis and display in the data flow DF12. If recurring log searches or access patterns enable the correlation of user activity or certain system components over time, linkability may arise. When logs include distinct identifiers that can link actions to specific devices, identifiability becomes an issue. If actions or mistakes are recorded and displayed on the dashboard indefinitely without permission or justification, non-repudiation occurs. Detectability is important if commands are found in logs, enabling an attacker to deduce protected processes. A serious risk is information disclosure, particularly if Grafana shows log data. Finally, when sensitive log content is viewed or retained without appropriate control, this leads to non-compliance.

The Grafana Dashboard shares data with the CMC Dispatch Operator in this data flow DF13. If the dashboard shows trends over time connected to certain drones, or operational units, linkability is present. When operator activities or device IDs that potentially link activity to specific devices are included in displayed data, identifiability issues arise. If the dashboard records and shows actions in a way that permanently links them to operators without their knowledge or consent, there are non-repudiation threats. If an attacker can determine sensitive events or system modifications by looking at the dashboard outputs, then detectability is applicable. If the dashboard reveals operational flaws, raw log data, or private communications in plaintext, information disclosure becomes crucial. Non-compliance may arise if the data visualization does not adhere to privacy laws, particularly if private information is accessed in ways that are not intended.

In this data flow DF14 commands are sent from the CMC Drone Operator to the Central Broker Drone. If recurring command patterns can be linked over time to certain drone operators, linkability is an issue. If operator credentials or identifiers are disclosed in the transmission, exposing critical information, identifiability issues occur. If activities are recorded in a way that binds to commands without sufficient knowledge or consent, non-repudiation becomes a problem. If unauthorized attackers determine the frequency, timing, or presence of command messages then detectability issues arise. Information disclosure is essential since revealed commands could provide attackers with the ability to influence drones or deduce mission objectives. Finally, if sensitive information in the transmission is not encrypted or logged in accordance with operational privacy or data protection rules, non-compliance may result.

For the purposes of auditing, threat detection, or system diagnostics, the data flow DF15 is where the CMC Security Analyst retrieves system and activity logs from the Log Database. If logs enable analysts to follow certain individuals or devices over time by comparing activities or identifiers, linkability becomes an issue. When logs reveal distinct information like IP addresses or timestamps associated with certain people, identifiability occurs. If logs are utilized to assign users to acts without their knowledge or capacity to challenge the records, non-repudiation could become problematic. If sensitive system states or behavior are revealed by querying or viewing logs and could be deduced by attackers, then detectability is applicable. If the logs include raw data that contains sensitive operational information,

information disclosure is a serious risk. If processing or accessing log data breaches privacy standards, retention guidelines, or legal requirements such as the GDPR, non-compliance may ensue.

Data flow DF16 shows how the Python Subscribe Script communicates with the Central MQTT Broker, potentially for control signals or status updates. If message patterns or identifiers enable the broker or attackers to reliably link the script to certain devices, or processes, linkability becomes a concern. Identifiability is a risk if the script transmits data such as credentials and IDs that relates its activity to certain individuals or roles. If these messages are recorded and subsequently utilized to hold someone accountable without their knowledge, non-repudiation problems could occur. If processing workflows or sensitive data-handling procedures are revealed by the timing or frequency of script interactions with the broker, then detectability is applicable. If payloads in messages delivered from the script to the broker are not appropriately encrypted or protected, information exposure may occur. Last, non-compliance is an issue if these communications contain sensitive information yet do not follow the encryption, access control, or retention guidelines established by the relevant privacy laws.

A primary target for LINDDUN attacks, the P1 Android App is a crucial interface that directly interacts with important operational and device data. When device identifiers or recurring app usage are able to connect user behaviors across sessions, linkability occurs. If the software gathers or sends personally identifiable information, such as names, locations, or device fingerprints, then identifiability becomes an issue. If user behaviors are recorded or linked to an identity without providing users with control or consent, non-repudiation becomes difficult. When app behavior may be externally watched, exposing operating patterns, detectability problems arise. Because the program handles critical inputs and outputs that could jeopardize system security and privacy if intercepted, information disclosure risks are significant. Finally, if the app violates privacy laws non-compliance risks could materialize.

The Bridge Broker Drone P2 is a high-value target for LINDDUN threats since it acts as a center for intermediary communication. If communication or drone identifiers let an attacker correlate actions over time, linkability hazards occur. If data contains identifiers linked to operators, missions, or particular drones, identifiability becomes an issue. If commands or other data are recorded without appropriate consent or reversible audit trails, non-repudiation threats arise, exposing operations. Here, detectability is very important because drone activity can be detected by an attacker using radio traffic or the frequency of MQTT messages. Because of the sensitive data being communicated such as mission directives and surveillance data, there are significant risks of information disclosure. Last, non-compliance occurs when this data handling does not adhere to privacy requirements.

The P3 NMT is a key source of privacy risk since it is essential for monitoring and evaluating network data. Because the technology may correlate different traffic sources and uncover patterns that connect devices, linkability is an issue. When IP addresses, device IDs are revealed through traffic analysis, identifiability arises. If logs are kept without anonymization, there is a possibility that they will be used to link specific operations, known as non-repudiation. The NMT itself may produce traffic or signatures that notify attackers of its existence, posing a danger to detectability. Given that the tool handles potentially sensitive payloads, such as commands, information disclosure is a serious risk. If the NMT handles regulated data categories without adhering to legal or policy requirements for monitoring and retention, non-compliance risks arise.

The P4 Central MQTT Broker is a crucial communication hub that handles message routing between all system components, making it extremely sensitive from a privacy perspective. The broker can correlate messages from several sources and sessions, exposing behavioral patterns, making linkability a risk. If subscriber identifiers such as IP addresses are recorded or made public, identifiability issues occur. If records are not adequately anonymized or safeguarded, non-repudiation issues arise since acts can be linked to specific devices. Since the broker's data can make its role clear to an attacker, detectability is crucial. Since the broker frequently handles unencrypted or inadequately protected messages that may contain sensitive payloads, information disclosure is essential. As for non-compliance, it could occur if the broker handles or keeps regulated data without following internal governance guidelines or data protection laws.

Data is extracted from the MQTT Broker and sent to the SQL Database via the P5 Python Subscribe Script, which serves as a receiver. It raises a number of privacy issues under the LINDDUN framework. If the script regularly connects with recognizable patterns such as timing, which can enable correlation across data sessions, linkability is a problem. If the script logs or transmits device-specific identifiers without confidentiality, identifiability occurs. If data pulled or pushed by the script can be linked to users or actions without sufficient controls, non-repudiation becomes an issue. Detectability could be possible if attackers use traffic analysis to deduce the script. Since the script handles potentially sensitive data streams from the broker, information disclosure is a serious concern; if encryption or sanitization is not used, system data may be exposed. If the script handles regulated data without putting in place appropriate logging, access control, or audit procedures that are in line with privacy rules, non-compliance concerns could occur.

Almost all LINDDUN threat categories have privacy issues with the P6 Grafana Dashboard. If the dashboard permits correlations between various datasets or device actions that were not intended to be connected, linkability becomes an issue. When the dashboard shows metrics unique to a device that may either directly or indirectly reveal identities, identifiability occurs. If actions are attributed without user authorization using dashboard activity tracking or access logs, non-repudiation may become a problem. Detectability is important because sensitive processes may be indicated by specific user actions or dashboard notifications. If the dashboard is improperly designed or made publicly available, there is a serious risk of information disclosure, which could result in the leakage of private or sensitive operational data. Risks of non-compliance also exist if the information displayed is in violation of organizational policy or data protection laws, particularly if sensitive logs for example are published without the appropriate privacy measures.

Since the Affected Individual E1 is at the center of the system's privacy issues, they are extremely susceptible to the linkability, identifiability and unawareness of LINDDUN threat types. If data or behaviors across services can be connected without consent, linkability problems occur. If any information gathered such as location can either directly or

indirectly reveal their identity, identifiability is a danger. Unawareness may arise if clear information isn't given regarding data collection procedures, how the data is utilized, or who has access to it.

Since the CMC Security Analyst E2 uses the NMT to monitor and maybe interact with confidential system traffic, they are vulnerable to a number of LINDDUN privacy concerns. If analysts' activity logs reveal behavioral patterns and may be linked across sessions or systems, linkability issues occur. If there are analyst-specific identifiers in the system logs that could reveal identities, identifiability becomes important. If analysts are not completely aware of the scope of logging or data retention pertaining to their own operations, this raises the issue of unawareness.

There are several LINDDUN privacy hazards associated with log access and analysis for the CMC Security Analyst E3 who works with the log database (LDB). If log entries may be linked over time to individuals or actions, linkability is an issue that could reveal behavioral patterns. When logs contain personal identifiers that might be used to link activity to specific people, like device IDs, identifiability issues occur. Unawareness could arise if the analyst is not aware of the regulatory obligations or the privacy-sensitive nature of the data they handle.

The CMC Dispatch Operator E4 is subject to several privacy risks associated with LINDDUN because of their interactions with operational dashboards and real-time data. When their actions across systems, like dispatch choices or login events, can be linked to create operational or behavioral patterns, linkability emerges. When an operator is linked to identifiable logs, particularly if these are available in audit trails, identifiability becomes a risk. If the operator is not made aware of the private consequences of their actions or how they are tracked, unawareness turns into a danger.

The CMC Drone Operator E5 is responsible for controlling drones, exposing them to a variety of privacy LINDDUN risks. If several drone movements over time can be linked to a single operator's identity, linkability becomes an issue as it may expose operational commands. If the operator's behaviors are revealed by system logs or data, identifiability becomes a problem. Last, operators that are unaware of the operational data being captured, processed, or shared pose a risk.

C. LINDUNN STAGE 2: Elicit the Threats

	Threat 1 Enumeration
Nbr	T01
Title	Device Identity Spoofing
Summary	Drone assets within the area of operation receive distress signals from affected users. Adversaries could attempt to create fake user accounts and connect to the drones, feeding false information to the system. Using session cookies from the Android device and message received timestamps, adversaries could link these two to create a group profile of affected users.
DFD Elements	E1, E2, P1, P2, DF1, DF2, DF3
Threat Type	Linking
Tree Nodes	L.2.2.2
Assets Involved	Drones
Priority	Medium
Assumptions	The mirror traffic that is being sent to the network monitoring tool is sent from each and every drone within the field without any delays.

Figure 6: Threat 1 Enumeration

Figure 6 describes the basis of the first threat enumeration. Threat 1, profiling affected users, exposes a critical vulnerability in the system. Adversaries can exploit this vulnerability and create a profile for a group of individuals who share the same area they are sending information from, through linking. The threat is significant in areas with big number of users because adversaries could use the profile they created to send fake confirmation request to the users in the same area and then collect even more data about the users and even affect their android device to not be able to send any signals Since the adversaries are targeting group of individuals, individuals within the same area will lean more towards trusting a confirmation request from an unknown source, assuming the individuals are already in a state of waiting to get rescued. The DFD elements that are involved in the threat are Affected Users (E1), CMC Security Analyst (E2), Android with App (P1), Bridge Broker Drone(P2), Relay MQTTS messages from device to drones (DF1), Relay MQTTS messages within Drones (DF2) and Mirror Traffic from Drones (DF3). These are the elements involved because there are two elements that do not fall directly within our system and depend completely on the user side. Even though they are completely on the user side, they still indirectly play their role as a part of the threat. These elements are Affected Users (E1) and Android with App (P1), they are still part of the threat because if the adversaries were to profile a group of individuals, Hey would in this case profile individuals who are sending their information to a particular drone on the field or by any other similarities they might be after. As for the remaining elements, adversaries are using Relay MQTTS messages from device to drone (DF1) as an entry point and then using the data of individuals that are being sent to one drone to another through Relay MQTTS messages within Drones (DF2), they are linking individuals and specially linking groups of individuals by their geo-location. Although live traffic is sent to CMC Security Analyst (E2), if the security analysts are not trained properly to detect anomalies, then adversaries may be able to link data of individuals within that time frame, thus having a privacy breach in the system.

This threat aligns with the LINDDUN framework especially L.2.2.2 tree node. This specific node talks about profiling a group of individuals, since data of a single

individual leads to insights about a larger group of individuals. Knowledge transfers from one individual to another through known relations of group membership or similarity. Adversaries can use timestamps of when signals were received and geo location of the drone to link data and profile users within the range in the field. They could also use session cookies from the android app to link with the timestamp of when requests were sent to the drone.

In terms of assets that are at risk because of this threat are Affected Individuals E1, Android App P1 and CMC Security Analyst E2. However, affected individuals and android apps are outside our system assets, but CMC Security Analyst however is at risk because of this threat. This is because if the analyst is not trained properly and does not have enough security clearance then not only will it affect the organization's integrity but also affect the user's privacy of their data.

As for the impact and priority of this threat, it is considered medium since it is going to have implications on privacy and security, but it is not the immediate concern. However, this implies that there should be proactive mitigations to prevent adversaries from exploiting linking vulnerability. From a privacy perspective, linking allows adversaries to infer sensitive details about users. Also, linking related vulnerabilities could result in non-compliance with privacy regulations, leading to legal penalties and reputational damage for the system that exists and credibility of the organization.

Mitigation for this particular threat can be taken care of in two aspects. The first one is by making changes to the personnel within the system, therefore making changes within the architecture of the system. This mitigation process where the CMC Security Analyst gets training sessions to be more proactive when detecting analogies in real time and they should also take updated knowledge tests periodically to maintain their credibility and performance within the organization. The second one is by applying the concept of shielding data which is vital to mitigating the threat of data linking. Specifically, by utilizing encryption for data in transit and at rest, the adversary will find it difficult to access sensitive data and create a profile on users. The system already implements this via the MQTTS protocol for data in transit between the affected user and the central MQTT broker. In addition, however, data at rest within the SQL database could also be encrypted. This would prevent adversaries who breach security of backend cloud services from gaining any meaningful information on user PII as mapped in the Figure 7 attack flow.

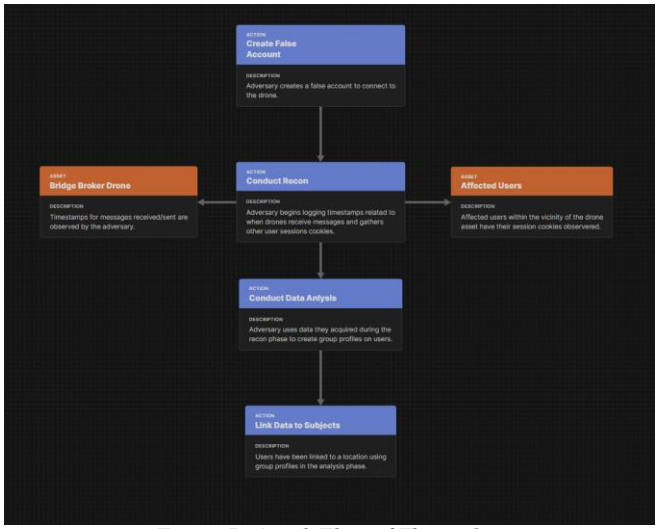

Figure 7: Attack Flow of Threat One

	Threat 2 Enumeration
Nbr	T02
Title	Exposure of Stored Data
Summary	User data is stored within an SQL database for CMC dispatchers to assess. If user information is retained after the rescue operation is completed, there is a potential privacy risk should the system suffer a security breach.
DFD Elements	DS1, P6, E4, DF10
Threat Type	Data Disclosure
Tree Nodes	DD.3.4
Assets Involved	SQL Database, Grafana Dashboard
Priority	Medium
Assumptions	The data that is collected is being stored indefinitely and beyond what is operationally required. The CMC Dispatch operator has no directions or means to delete data when not needed through the Grafana dashboard.

Figure 8: Threat 2 Enumeration

Threat 2, as detailed in Figure 8, elicits exposure to data that is being stored for longer than needed period of time becomes a huge risk for a data disclosure threat. This is because if data were to be collected for a rescue operation and then deleted after the operation is over, the organization cuts out the entire factor of having a risk of exposing user data.

This will cause the organization to have a completely unnecessary consequence, since the entire purpose of collecting the data was to rescue the affected individuals but once the operation is over, there is no means for the organization to hold the data unless there are specific portions of the data that needs to be kept, to help better the rescue missions. In this scenario however, the user's personal data is only the geo-location of the affected individual, so there is no means for the system or the organization to hold that data, since it won't help improve the quality of rescue missions. On the other hand, a specific affected individual might not even be on the same geo-location when there is another disaster that takes place, so keeping the old data is just going to make the organization spend more funding towards cloud storage for irrelevant data. The DFD elements that are involved in this threat are SQL Database (DS1), Grafana Dashboard (P6), CMC Dispatch Operator (E4) and SQL Database to Grafana Dashboard (DF10). They are involved because data is stored in a SQL database which is maintained by the CMC dispatch operator using Grafana dashboard as a graphical user interface and the data flow of SQL database to Grafana dashboard. Since the data flow route of SQL database to Grafana dashboard is bi-directional, the CMC dispatch operator plays an important part since the operator is able to delete, modify and store the data within the server, therefore within the system.

This threat aligns with the LINDDUN framework, especially DD.3.4 tree node. This falls under the Data disclosure threat tree and the specific node talks about keeping personal data longer than functionally needed. Storing data longer than needed serves purpose and only increases the impact of a data breach. Since there is a dashboard and an operator that are part of the operation, they need to be the second verification route for deleting old data, specifically if the system doesn't currently have an existing process to remove the old data when an operation is over.

There aren't necessarily any assets that are at risk here, since the threat consists of a potential future threat due to unhealthy practices. However, it is to be noted that storing old data which does not have any relevance can lead to more payments for cloud storage. So, in this case the asset that is at risk is the proper funds that are being spent.

The threat's priority level is medium, since it is a risk that could happen depending on a potential incident. It is going to have implications for privacy and security if there is a breach but not to the point where it needs immediate action. From a privacy standpoint disclosure of data allows adversaries to commit multiple bad actions, for which the users might need to face the consequence for. The impact of this threat could be significant because this will make the organization and system, an uncredible system or organization because in the future affected users will be reluctant to send signals to be rescued at the moment of disaster, in order to save themselves from huge consequences that they might need to face because of adversaries taking breaching data from the system, and using to their advantage.

As for the mitigation for this threat, there are two elements that can be combined together to mitigate the issue. One of the elements is the Grafana dashboard and the other one is the CMC dispatch operator; this is because the organization can train the CMC operators by practicing deleting user data after the safety operation is successful. The operators should have a periodic check up with the system's database to see whether old irrelevant data still exists or not. Once the operator decides to delete the old data, that is where the second element comes into play, which is the Grafana dashboard. In this element, management interfaces should be implemented into the Grafana dashboard. It should be programmed in a way where once the operator reports that an operation is over, the dashboard will have a pop-up window with the message asking whether the operator wants to delete the user data who just got rescued. Since getting rid of data will make the organization lose the information forever, this mitigation strategy could also act as a two-man verification, where both the dashboard and the operator's response is what deletes the data, thus preventing the exposure of stored data as mapped in the Figure 9 attack flow.

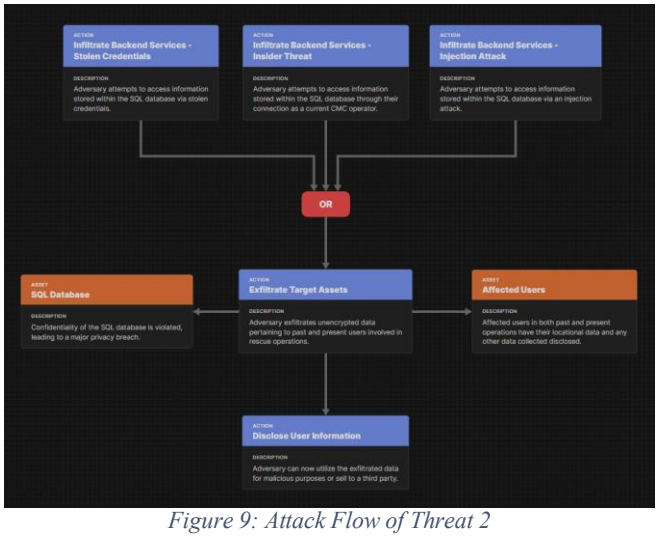

Figure 9: Attack Flow of Threat 2

D. LINDUNN STAGE 3: Manage Threats and Mitigation

This stage is the final stage of the phase of the LINDDUN privacy threat modeling process during which threats that have already been addressed are prioritized and dealt with using the proper mitigation techniques. To identify which risks, require addressing and immediate action this step involves assessing each privacy issue according to its impact and likelihood, in coordination with a Data Protection Officer (DPO). To help choose appropriate mitigations that fit the system's design and privacy objectives, the procedure uses a set of privacy-enhancing technologies (PETs). PETs are chosen to mitigate or eliminate the risk associated with each danger. By the end of this stage, the system design is refined to minimize privacy risks while balancing functionality and compliance with privacy regulations. The mitigations tree is as follows in Figure 10 with the associated strategies.

Mitigation Strategy	LINDDUN Threat Tree
Protect ID	L_p1, I_df
Protect Data	ID_df7
Transactional Data	NR
Contextual Data	D_df7
Awareness	U
Guard Exposure	DD_ p6, DD_ p5, DD_ds1
Compliance	NC
Confidentiality	DD_df1, DD_df2, DD_df15

Minimization	L_ds, I_ds, D_ds
Maximize Accuracy	NR
Review Data	U, NC
Update/Request Deletion	DD_p6

Figure 10: Mitigation Strategy and LINDDUN Threat Tree

Using the formula Risk = Impact × Likelihood, LINDDUN uses a systematic risk assessment approach to prioritize privacy threat mitigation. This approach assesses the potential severity of a threat's impact and its likelihood of happening. Impact takes into account consequences like data leaks, legal infractions like GDPR non-compliance, or harm to one's reputation. Taking into account the ability of the attacker, system flaws, and the existence of current defenses, likelihood indicates how vulnerable a system is to the danger. To determine the priorities for mitigation, threats are mapped onto a risk matrix and rated by combining these two dimensions. While low-risk risks *may* be tracked or lessened over time, high-risk dangers are dealt with immediately. This risk-based approach guarantees that mitigation initiatives are effective, focused, and in line with privacy objectives.

Privacy Enhancement Technology (PET) are technologies chosen to assist in mitigation or elimination of risks associated with each threat. There are three main categories in PET: altering data, shielding data, and systems and architecture. Within each of those categories there are various technologies and techniques that can be applied to the system. Their purpose is to enhance privacy of individuals by reducing or eliminating the processing of personal data or making it less identifiable and linkable.

Altering data is the process of transforming or modifying the original data to protect a user's privacy while maintaining utility of the data. This can help when the goal is to prevent the identification of an individual within a dataset. Altering data is vital when preserving data during analysis operations. There are four methods: anonymization, pseudonymization, differential privacy, and synthetic data. Anonymization is the process of removing PII from data so that individuals cannot be identified, indirectly or otherwise. In theory, re-identification should be impossible once data is anonymized. Pseudonymization is the process of replacing private identifiers with fake identifiers, also known as pseudonyms. Data can be re-identified under this process if there is a lookup table or something similar. Useful when identity recovery is necessary. Differential privacy is the process of quantifying and limiting privacy risks associated with statistical analysis. This is done through the addition of controlled noise to outputs so individual data cannot be inferred. Finally, there is synthetic data which is a process of artificially generating data that mimics real data patterns while not containing real PII. Can help in testing and training machine learning models while maintaining privacy.

Shielding data is the process of protecting sensitive data from unauthorized exposure and access while in transit and in rest. Shielding techniques are essential to preserving the confidentiality and integrity of the data, thus denying privacy breaches. There are three methods: encryption, homomorphic encryption, and privacy enhanced hardware. Encryption is the process of transforming readable data into unreadable ciphertext via a cryptographic algorithm. Decryption keys are required to access the original data, making this an invaluable tool for data in transit and data at rest. Homomorphic encryption process of encryption that allows computation on ciphertexts. The decrypted results match what would have been obtained if operations were performed on the plaintext. Useful when wanting to preserve privacy on analytics related to sensitive cloud data. Privacy enhanced hardware is hardware that supports trusted execution environments, isolating code and data in secure enclaves even from the host OS. Provides confidentiality and integrity for computation of sensitive data.

Systems and architecture, in regard to privacy security, is how a system is structured to support secure data communications, processing, and control. It is through the systems and architecture that privacy principles must be designed and enforced at every layer. There are four methods: Multi Party Computation (MPC), data dispersion, management interfaces, and digital identity. MPC allows multiple parties to compute a function over their inputs without revealing those inputs to each other. Useful when trying to be collaborative on a task while not revealing data. Data dispersion is the process of using sharding, secret sharing, or distributed storage where data is split and spread across multiple nodes. This is done to increase resilience and confidentiality so that no single location holds the complete set of data. Management interfaces are control panels or APIs used for configuring, monitoring, or managing the system. They are secured via authentication, access control, and logging to prevent internal or external privacy breaches. Digital identity includes technologies and protocols used to uniquely identify users and devices. Digital identity frameworks ensure authenticated access, authorization, and include federated identity or decentralized models.

For the purposes of the Disaster Relief System, a key PET was encrypted. In a system that serves as a substitute for a communications network, maintaining confidentiality and integrity of data is a primary concern. Data shielding through encryption of data in transit and rest helps to mitigate privacy breach scenarios where adversaries attempt to access sensitive data. For instance, T01 is an example of how encryption could be applied. Encrypting data flowing between the user and the central MQTT broker and data at rest in the SQL database will create a significant obstacle for the adversary attempting to disclose PII of users. Another useful PET is the management interface which can be applied to T02. By implementing management interfaces into the Grafana dashboard, the CMC dispatch operator can delete data in the SQL database that is no longer relevant to the mission. This lessens the impact of a privacy breach and protects prior users.

V. Final Remarks

This project set out to build a clear, practical understanding of the privacy risks facing a disaster relief communication system, designed to keep working even during crisis situations. The system spans everything from Android devices used by people in need, all the way to a central hub made up of drone-based data relays, a message broker (MQTT), back-end services written in Python, an SQL database, and a Grafana dashboard to give emergency teams a real-time view of the situation.

Through decomposing our objectives into strategic and operational requirements we were able to begin critical evaluation of an already secured system through the lenses of privacy preservation and humanitarian compliance. The assets and data flows within the system had resulting connections across trust boundaries, adversary capabilities were mapped within an in-depth business impact matrix, and the resulting

privacy threats and vulnerabilities were analyzed through the LINDDUN framework which was then further paired with actionable mitigations and privacy-enhancing technologies to provide a grouped analysis of a theoretically secure system with the additional layer of regulatory benefit. This business-oriented rationale allows for a blend of resilience, confidentiality, integrity, availability, and compliance requirements into a vulnerable and bare-bones system. Although our initial model within a previous report helped to secure the privacy of victims within the form or logging databases that monitored the flow of MQTT data as well as the addition of CMC Analysts to review mirrored drone broker traffic, it was found by the end of this report to be insufficient in meeting regulatory needs due to unforeseen threats regarding the analysis of PASTA. Through this the critical assessment of PASTA can be seen such that it does not focus on privacy trees and taxonomy but instead focuses on security risk management leading to privacy being a secondary analysis after the system has already been created. This lack of privacy by design is the problem that we attempt to solve through the inclusion of LINDDUN and PETs.

While the analysis is comprehensive, its empirical boundaries are evident. Behavior and threat likelihood are inferred through secondhand laboratory data and research leading to key restraints within the system.

1. Human-Factor Uncertainty: Our model's mitigations rely solely on human implementation and thorough upkeep. User studies and field simulations regarding a normal variety of trained individuals would be crucial in validation of our assumptions of human action. This includes red team exercises and privacy impact measurements within a live environment to measure empirical responses from the human operators.

2. Lack of Risk Data: The inference of MQTT threat data within this system requires real field risks to become more technical and accurate regarding the specific drone deployment. Risk scores rely on assumed attacker effort and impact scales; without empirical breach data there must be continuous changes implemented within the system based on the reactivity of the deployment. MQTT gaps and misconfigurations are likely in real deployment [10].

In summary, although the detailed privacy threat modeling allows for granular insights into prevention of privacy and regulatory disasters, there are methodological tradeoffs involved. LINDDUN provides a focus on linkability of data, identification, and non-repudiation of inside involvement that a conventional STRIDE analysis will often miss [13]. The lack of quantitative data, however, may overinflate the threat list without inclusion of critical operational objectives scaling with it. A hybrid approach of including steps from each of these methods as well as coupling LINDDUN for threat discovery with quantitative techniques such as FAIR or Bayesian attack graphs regarding risk scoring metrics may create a more balanced defensive understanding of the system and include reactive data. LINDDUN's emphasis for homomorphic encryption and management interfaces as primary PETs are highly tested and documented in literature to provide an exact implementation of real time viability.

This work offers a realistic and actionable privacy roadmap for the secured disaster response systems. It blends strong technical defenses with practical steps for improving system resilience over time. While there are still challenges to overcome, the framework we've outlined shows that by continuously adapting to new threats, it's possible to maintain secure, reliable communication.

VI. REFERENCES

[1] OCHA, "Relief Web Topics," ReliefWeb, 2024. [Online]. Available: https://reliefweb.int/.

[2] AP News, "Floods devastate Nigeria, Niger, Chad, Mali amid intense rains and climate concerns," AP News, Mar. 2024. [Online]. Available: https://apnews.com/article/floods-nigeria-niger-chad-mali-rains-climate-change-397b303cdae17ef2a0076192c8c908ac.

[3] The Guardian, "Much of West without internet after undersea cable failures," The Guardian, Mar. 14, 2024. [Online]. Available: https://www.theguardian.com/technology/2024/mar/14/much-of-west-and-central-africa-without-internet-after-undersea-cable-failures.

[4] "Tonga volcano: Internet restored five weeks after eruption," BBC News, Feb. 22, 2022. [Online]. Available: https://www.bbc.com/news/world-asia-60474362.

[5] A. Aldashev and B. Batkeyev, "Broadband Infrastructure and Economic Growth in Rural Areas," Information Economics and Policy, vol. 57, p. 100936, 2021, doi: 10.1016/j.infoecopol.2021.100936.

[6] ITU-T Recommendation Y.1540, "Internet protocol data communication service – IP packet transfer and availability performance parameters," International Telecommunication Union, Dec. 2019. [Online]. Available: https://nisp.nw3.dk/standard/itu-t-y.1540.html

[7] "LINDDUN Threat Categories," LINDDUN.org. Accessed: Apr. 29, 2025. [Online]. Available: https://linddun.org/threats/

[8] K. Wuyts, "LINDDUN GO: Privacy Threat Modeling for Everyone," presented at the CIF Seminars, Sept. 29, 2020. [Online]. Available: https://cif-seminars.github.io/slides/20200929-kwuyts-linddun-go.pdf

[9] Rochester Institute of Technology, "Cybersecurity of MQTT-based drone communications in disaster scenarios," Thesis, Rochester Institute of Technology, Rochester, NY, USA, 2023. [Online]. Available: https://repository.rit.edu/cgi/viewcontent.cgi?article=12719&context=theses.

[10] Bevywise Networks, "MQTT Security Best Practices," Bevywise Networks, 2023. [Online]. Available: https://www.bevywise.com/blog/mqtt-security-best-practices/.

[11] SecureFlag, "Threat modeling for Privacy: What is it and how can you use it?" SecureFlag Blog, Feb. 14, 2024. [Online]. Available: https://blog.secureflag.com/2024/02/14/threat- modeling-for-privacy-what-is-it-and-how-can-you-use-it/

[12] N. Kirtley, "LINDDUN Threat Modeling," Threat-Modeling.com, Sep. 21, 2023. [Online]. Available: https://threat-modeling.com/linddun-threat-modeling/

[13] DistriNet Research Unit, KU Leuven, "LINDDUN: Privacy Threat Modeling," linddun.org, [Online]. Available: https://linddun.org/