Project: Threat Modeling a Networked Insulin Pump*

Secure Systems / Empirical Security

Due date: Wednesday, March 18, 11:59pm
Format: PDF or Word report
Weight: Part of the Projects component (40% total across five projects)

Overview

You have been hired as a security consultant for **GlucoGuard**, a medical device company developing a networked insulin pump system. The product is aimed at patients with Type 1 diabetes and their care teams. Your job is to produce a threat model and a recommended security architecture that the engineering team can use to guide their design decisions.

GlucoGuard's engineering team has shared the following product description with you. It is intentionally written from a product perspective, not a security one. Part of your job is to translate it into a security artifact.

Note: Unlike a typical software product, this system has **direct physical safety implications**. A threat that compromises this device doesn't just leak data—it can harm or kill a patient. Your threat model should reflect this.

Product Description

GlucoGuard is a networked insulin pump system that continuously monitors a patient's blood glucose levels and delivers insulin automatically. The system consists of a wearable pump, a companion smartphone app, and a cloud platform. The core value proposition is accurate, automatic insulin delivery with easy monitoring for patients, their families, and their medical providers.

The Pump

The pump is a small wearable device attached to the patient's body. It contains a continuous glucose monitor (CGM) sensor and an insulin reservoir. The pump reads glucose levels every few minutes and delivers insulin based on its control algorithm.

The pump supports two operating modes. In *automatic mode*, the pump adjusts insulin delivery based on real-time glucose readings without user intervention. In *exercise mode*, the patient can temporarily adjust the target glucose range to avoid dangerous highs and lows during physical activity.

The pump communicates with the smartphone app over Bluetooth. It does not connect to the internet directly. The pump runs embedded firmware that controls the dosing algorithm, sensor readings, and Bluetooth communication.

*Thanks to Ron Thompson and Dan Votipka at Tufts for their guidance in developing this project.

The Smartphone App

The companion app (iOS and Android) is the primary interface for the patient. Through the app, the patient can view current and historical glucose readings, see insulin delivery history, activate exercise mode, and configure alerts (e.g., for high or low glucose).

The app communicates with the pump over Bluetooth and with the GlucoGuard cloud platform over the internet. It syncs glucose and dosing data to the cloud so that the data is available to others involved in the patient's care.

The Cloud Platform

GlucoGuard operates a cloud platform that stores glucose data, dosing history, and device configuration. The smartphone app uploads data to the cloud, and configuration changes flow back down through the app to the pump. The cloud platform also delivers alerts and notifications when glucose levels cross configured thresholds.

Who Uses the System

Several groups of people interact with the GlucoGuard system, and their needs differ:

- **Patients** use the system daily. They need to see their own data, control their pump, and decide who else can be involved in their care. Some patients are children or elderly adults who may depend on others to help manage the device.
- **Family members and caregivers** may need to monitor the patient's glucose levels remotely and be alerted in an emergency. In some cases—especially for a young child or an incapacitated adult—a caregiver may need to be able to do more than just observe.
- **Medical providers** (physicians, endocrinologists) need to review a patient's glucose trends and dosing history, and to adjust the pump's clinical parameters (e.g., basal rates, correction factors, target ranges). These adjustments are delivered to the pump through the cloud and the patient's app. Providers may manage many patients.

The patient should be able to decide who has access to their data and their device. The system should ensure that only authorized individuals can view patient data or affect the pump's operation. Health data must be protected—it is sensitive and subject to regulatory requirements.

Firmware Updates

The pump's firmware can be updated to deliver bug fixes, algorithm improvements, and security patches. Updates are downloaded from the cloud platform and transmitted to the pump over Bluetooth through the smartphone app. The engineering team considers firmware updates critical for long-term device safety but has acknowledged that the update process must not interrupt active insulin delivery.

Safety Constraints

The pump enforces hard safety limits on insulin delivery that are set during manufacturing and cannot be changed through software. These limits cap the maximum bolus size and maximum hourly delivery rate. If the pump's sensor fails or produces readings the algorithm considers unreliable, the pump falls back to a pre-programmed basal rate and alerts the patient.

If the Bluetooth connection between the pump and the app is lost, the pump continues operating in automatic mode using its last known configuration. The pump has a small LED and vibration motor for on-device alerts when the app is unreachable.

Your Deliverables

1. Scope and Assumptions (∼ half page)

Define the scope of your threat model. Identify what is in scope and what is out of scope, in terms of the sources of potential threats, and justify your choices briefly. State any assumptions you are making about the system that are not explicit in the product description. Note any ambiguities you encountered and how you resolved them.

2. Data Flow Diagrams (2 diagrams)

Produce Level 0 and Level 1 DFDs. A Level 0 DFD (sometimes called a *context diagram*) shows the entire system as a single process (a single bubble or box labeled something like "GlucoGuard System"), with external entities around it with data flows to/from. A Level 1 DFD expands this to capture the major components of the system (the pump, smartphone app, cloud platform, and so on), the data flows between them, and the trust boundaries. Your diagram should be clear enough that a new engineer on the team could use it to orient themselves. You may use any tool you like (OWASP Threat Dragon, draw.io, or even a neatly drawn diagram scanned or photographed).

3. Threat Enumeration (main body)

Use your Level 1 DFD as the basis for a systematic threat analysis. For each component, data flow, and trust boundary in your diagram, apply the STRIDE categories to identify relevant threats. You may also identify threats through other means (e.g., by reasoning about the system's safety properties or by considering the needs of specific user groups), but your analysis should be grounded in the DFD—a reader should be able to trace each threat back to a specific element in your diagram.

For each threat, ordered by risk/priority, provide:

- A descriptive name
- The component or data flow it targets
- The threat category (use STRIDE)
- A description of the attack scenario
- The likelihood that the threat can be realized, and its impact if it is (low/medium/high for both, with justification)
- A proposed response. For mitigations or eliminations, there are two parts:
  - Control: What must be done? (technology-agnostic)
  - Implementation Guidance: How do we build it (additions, removals, changes, ...)?

  For transfers or acceptance or risk, say why this is the best choice

Because this is a medical device, you should consider both **security** threats (e.g., unauthorized access, data breaches) and **safety** threats (e.g., scenarios that could result in incorrect insulin delivery). Where a threat has both security and safety dimensions, note this explicitly.

You are not expected to find every possible threat. You will be assessed on the quality of your reasoning, the coverage of meaningful threats, and the clarity of your descriptions. Aim for depth over breadth—a well-reasoned threat with a clear scenario is worth more than five vague entries.

4. Security Architecture Summary (∼ 1 page)

Based on your threat analysis, describe how the system should work with your proposed mitigations in place. This is not a repeat of your threat list—it is a coherent description of the system's security architecture as you would recommend it to the engineering team. Where individual threats led you to propose similar or overlapping mitigations, consolidate them here into a unified design.

Your summary should be written so that an engineer who reads only the original product description and this section comes away with a clear picture of the security mechanisms the system needs and why. You may organize it however you think is clearest (by component, by security property, or otherwise).

5. Reflection (∼ half page)

Briefly answer the following: What was the hardest scoping decision you made, and why? In what ways does threat modeling a medical device differ from threat modeling a conventional software application? Are there any threats you identified that you believe the engineering team would be most likely to overlook?

Practical Notes

- You may use OWASP Threat Dragon, Microsoft Threat Modeling Tool, draw.io, or any other tool for your DFD (I recommend the last). Submit the diagram as part of your final report; do not submit a raw tool file unless also submitting an export. This Microsoft Learn training module might be useful.
- There is no single correct threat model. Reasonable people will scope this system differently and identify different threats. What matters is that your choices are justified and your reasoning is clear.
- The product description contains at least one deliberate ambiguity that the engineering team has not yet resolved. You are expected to identify it, state how you are handling it, and explain the security and safety implications of each option.
- You do not need to be a medical device expert to complete this assignment. However, you should recognize that the stakes are different from a typical software product—the consequences of a successful attack can include physical harm, not just data loss or inconvenience.

Use of Generative AI Tools

Students are permitted to use generative AI tools (e.g., large language models) to assist with aspects of this project, including:

- Clarifying threat modeling concepts or STRIDE categories,
- Brainstorming potential threats or attack scenarios,
- Understanding medical device terminology or regulatory context,
- Improving the clarity or structure of written descriptions.

However, all submitted work must reflect **your own understanding and reasoning**. In particular:

- You may not outsource the core threat analysis or security architecture to an AI system,

- You should be able to explain every threat you include—why it matters, why you scoped it the way you did, and why your proposed mitigation is appropriate,
- You should be able to defend your DFD and architectural choices if asked.

You are responsible for the correctness and coherence of your submission, regardless of whether an AI tool was used to help produce it. Be aware that we may ask questions about your threat model (e.g., on an exam or in discussion) to assess understanding.

This project emphasizes **reasoned analysis, security judgment, and clarity**. Over-reliance on generative tools without understanding may negatively impact grading, particularly in the threat enumeration quality and security architecture components.