Section 1: Executive Summary

Write 2-3 paragraphs explaining:

- *What system you analyzed*
- *Why threat modeling matters for this system*
- *Your top 3 findings (highest priority threats)*

Write this for Marcus Webb, the General Manager. He's not technical. Use plain language.

I reviewed the security of The Grand Marina's HYDROLOGIC water management system, which is provided and operated by our vendor Hydroficient. This system uses three connected devices (one per water service line) to monitor and control water flow throughout the hotel. Those devices send data to a cloud service, which feeds a web dashboard our engineering team uses to watch usage and, if needed, remotely shut off water or adjust gates.

Threat modeling matters here because this isn't just an IT system - it can physically open and close water valves throughout a 500-room hotel. A security weakness could let someone cut off water to guest floors, force a valve open and cause flooding, or hide a real leak from staff until it turns into major property damage. Reviewing these risks now lets us fix weak points before they're used against us or fail on their own.

Three issues stood out as the highest priority. First, the web dashboard our engineers use does not appear to require strong login protection, so a single stolen password could hand an outsider full control of the water system. Second, the connections between the devices, the cloud, and the dashboard don't clearly verify that commands are genuine, meaning someone on our network could potentially trick a device into opening or closing a valve. Third, the devices have no automatic safety behavior if they lose their connection to the cloud during a real leak, which could turn a small problem into a major one. My recommendation is to prioritize multi-factor authentication on the dashboard first since it's the fastest fix, then secure device commands, then add an automatic fail-safe shutoff.

Section 2: System Overview

Describe The Grand Marina's HYDROLOGIC system:

- *500-room luxury hotel (Hydroficient customer)*

- *3 HYDROLOGIC flow management devices (one per water service line)*
- *Cloud-based monitoring and control*
- *Web dashboard for operators and management*
- *Remote shutoff and gate control capability*

Include a simple data flow diagram (can be ASCII art or describe the flow):

```
None
[HYDROLOGIC Devices] → [Cloud API] → [Dashboard]
        ↓                    ↓
  [Gate Controls]      [Database]
  [Emergency Shutoff]  [Reports]
```

Section 3: Asset Inventory

List the key assets and their CIA priorities (use your work from Step 2):

Asset	Description	C Priority	I Priority	A Priority
HYDROLOGIC Devices	3 flow management units	Medium	Critical	Critical
Web Dashboard	Operator monitoring interface	High	Critical	High
Cloud API	Device-to-cloud communication	High	Critical	Critical
Remote Controls	Gate adjustments, emergency shutoff	Medium	Critical	Critical
Consumption Data	Savings reports, billing records	Medium	Medium	Low

Section 4: STRIDE Analysis

This is the core of your threat model. For **each major component**, analyze all six STRIDE categories:

Component 1: HYDROLOGIC Devices

Threat	Scenario	Likelihood	Impact	Risk
Spoofing	Attacker on the hotel network intercepts a device's authentication traffic and clones its credentials to register a rogue device reporting false low-flow readings, masking an actual leak.	Medium	High	High
Tampering	A contractor or intruder with physical access to a mechanical room alters a device's firmware or wiring to force a valve open while it reports normal readings to the dashboard.	Low-Medium	Critical	Critical
Repudiation	An operator (or attacker using a compromised login) remotely closes a gate	Medium	Medium	Medium

	valve during peak demand, then denies the action since the device doesn't log which account issued the command.			
Info Disclosure	Devices send flow data over the hotel network without strong encryption; an attacker on the same network sniffs traffic to learn occupancy and usage patterns across the property.	Medium	Medium	Medium
Denial of Service	An attacker floods the local network segment shared with guest Wi-Fi, knocking devices offline so they stop reporting and stop responding to shutoff commands during a real leak.	Medium	Critical	Critical
Elevation of Privilege	A flaw in a device's local management interface	Low-Medium	Critical	Critical

	(default credentials or unpatched firmware) lets an attacker gain root access, then push malicious firmware or reconfigure other devices.			

Component 2: Web Dashboard (Same table structure)

Threat	Scenario	Likelihood	Impact	Risk
Spoofing	An attacker obtains a hotel engineering staff member's reused password (via credential stuffing, since a breach involving that password already exists publicly) and logs into the dashboard as that operator.	High	High	Critical
Tampering	After gaining access, an attacker raises flow-alert thresholds so real leaks stop triggering alerts, or edits historical logs to hide the unauthorized change.	Medium	High	High

Repudiation	A compromised operator account issues a remote shutoff affecting guest rooms; without per-user action logging, management can't tell who made the change or when.	Medium	Medium	Medium
Info Disclosure	A broken access control flaw (e.g. an insecure direct object reference in the dashboard's API) lets a lower-privileged user view billing and consumption reports for the entire hotel.	Medium	Medium	Medium
Denial of Service	An attacker overwhelms the public-facing dashboard login page with traffic, preventing engineering staff from monitoring flow status or issuing an emergency shutoff during a leak.	Medium	Critical	High
Elevation of Privilege	A flaw in the dashboard's role-based access control lets a low-privilege 'viewer' account escalate to an 'administrator' role by	Low-Medium	Critical	High

	modifying a hidden form parameter, granting shutoff/gate command access.			

Component 3: Cloud API (Same table structure)

Threat	Scenario	Likelihood	Impact	Risk
Spoofing	Through DNS manipulation or a compromised device configuration, an attacker sets up a rogue endpoint mimicking Hydroficient's cloud API, tricking a device into sending data to (and accepting commands from) the fake server.	Low-Medium	Critical	High
Tampering	Without end-to-end TLS/certificate validation, an attacker performs a man-in-the-middle attack on hotel network traffic to alter a flow-rate command sent to a device, opening a valve further than intended.	Medium	Critical	Critical

Repudiation	An automated billing integration sends an incorrect command through the API; because calls aren't individually authenticated and logged, Hydroficient can't tell whether it was the integration, a stolen credential, or a bug.	Medium	Medium	Medium
Info Disclosure	An overly permissive API key, shared across all Grand Marina devices, is exposed in a misconfigured mobile app or leaked in a public code repository, letting anyone with the key pull consumption data for the whole property.	Medium	High	High
Denial of Service	A volumetric DDoS attack against Hydroficient's shared cloud API knocks it offline, simultaneously disconnecting all three Grand Marina devices from monitoring and remote control at once.	Low-Medium	Critical	Critical

Elevation of Privilege	A flaw in the API's authorization checks lets a device-level API key (meant only to submit readings) also call administrative endpoints, such as the one that issues gate/shutoff commands, letting a compromised device attack the rest of the system.	Low-Medium	Critical	High

Component 4: Remote Controls (Gate/Shutoff) (Same table structure)

Threat	Scenario	Likelihood	Impact	Risk
Spoofing	Using stolen dashboard or API credentials, an attacker issues a remote 'close gate' command to a specific water line, appearing to come from a legitimate operator session.	Medium	Critical	Critical
Tampering	An attacker on the local network intercepts a routine 'reduce flow by 10%' command and modifies it to	Low-Medium	Critical	High

	‘fully close,’ cutting off water to an entire wing before any operator notices.			
Repudiation	During an active emergency, staff can’t verify whether an emergency shutoff command was actually received and executed by the physical actuator versus lost in transit, delaying confirmation that a leak was stopped.	Medium	High	High
Info Disclosure	Exposed technical documentation or a debug endpoint reveals the exact command syntax used to trigger gate controls, giving an attacker a direct recipe to command actuators without going through the dashboard.	Low-Medium	High	Medium
Denial of Service	An attacker times a targeted network disruption to coincide with a real leak, preventing the emergency shutoff	Low	Critical	High

	command from ever reaching the affected device and maximizing water damage.			
Elevation of Privilege	A third-party smart-building integration meant only to read status is found to also have write access to the shutoff API due to overly broad permission scoping, letting a compromise of that unrelated system shut off the hotel's water supply.	Low-Medium	Critical	High

For each scenario, be specific. "Attacker does bad thing" is not acceptable. Describe HOW they would execute the attack.

Section 5: Risk Summary

List your findings by risk level:

Critical Risks:

1. Physical tampering with HYDROLOGIC devices (Tampering – HYDROLOGIC Devices): someone with facility access could force a valve open while masking it from operators, directly risking flooding and guest safety.
2. Network-level command tampering enabling unauthorized valve manipulation (Tampering – Cloud API; Spoofing – Remote Controls): commands between the cloud and devices can be altered or spoofed to force gates open or closed with no warning.
3. Denial of service against devices or the shared cloud API during an active leak (DoS – HYDROLOGIC Devices & Cloud API): losing monitoring and remote shutoff exactly when it's needed most turns a contained leak into major property damage.

4. Weak dashboard authentication enabling account takeover (Spoofing – Web Dashboard): a single stolen or reused operator password could hand an attacker full visibility and control over the entire water management system.

High Risks:

1. API key leakage or over-permissioning (Information Disclosure / Elevation of Privilege – Cloud API): exposes property-wide consumption data and can be abused to reach administrative functions.
2. Dashboard privilege escalation (Elevation of Privilege – Web Dashboard): a low-privilege viewer account could be manipulated into gaining administrator-level control.
3. Man-in-the-middle command interception at the remote control layer (Tampering – Remote Controls): legitimate commands can be silently altered in transit, disrupting service to specific building areas.
4. Repudiation of emergency shutoff commands (Repudiation – Remote Controls): uncertainty about whether a safety-critical action actually executed slows incident response.

Medium Risks:

1. Unencrypted or poorly segmented device telemetry (Information Disclosure – HYDROLOGIC Devices): reveals usage patterns but is not directly safety-critical.
2. Missing per-user action logs on the dashboard and API (Repudiation – Web Dashboard & Cloud API): complicates investigations after an incident but doesn't cause the incident itself.
3. Disclosure of control endpoint details (Information Disclosure – Remote Controls): could be a stepping stone to a more serious attack but requires further exploitation to cause harm.

Section 6: Recommended Mitigations

For each Critical and high-risk, propose a defense:

Risk	Proposed Mitigation	Implementation Complexity
Example: Weak dashboard authentication	Enable multi-factor authentication	Low — configuration change
Physical/firmware tampering with HYDROLOGIC devices	Add tamper-evident enclosures and require	Medium – requires hardware/firmware

	signed/verified firmware updates (reject unsigned firmware); alert locally if reported flow deviates from expected flow.	changes and vendor coordination
Command tampering/spoofing between cloud, dashboard, and devices	Enforce mutual TLS (mTLS) or signed command payloads between the cloud API and devices so a device only accepts commands it can cryptographically verify.	Medium – software/configuration change plus certificate management
Denial of service against devices or shared cloud API during emergencies	Add a local fail-safe mode so devices default to a safe state (e.g. automatic shutoff on sustained anomalous flow) even when disconnected from the cloud; add DDoS protection/rate limiting in front of the API.	Medium-High – fail-safe logic requires firmware changes and testing
API key leakage or over-permissioning	Issue scoped, per-device API keys (least privilege) instead of shared keys, rotate keys regularly, and add automated scanning to detect leaked keys in code repositories.	Medium – requires reworking API key management
Dashboard privilege escalation	Enforce server-side role checks on every dashboard action (not just UI-level restrictions) and conduct an access-control audit/penetration test before relying on the dashboard for production use.	Medium – access-control rework and testing
Man-in-the-middle command interception at the remote control layer	Add per-command integrity checks (HMAC signatures) between the API and remote control actuators so any tampered or replayed	Medium – builds on the mTLS/signing work above

	command is automatically rejected.	
Repudiation of emergency shutoff commands	Add end-to-end command acknowledgment logging showing who issued a command, when the device executed it, and confirmation the actuator physically moved, visible in the dashboard.	Low-Medium – mostly logging and UI work