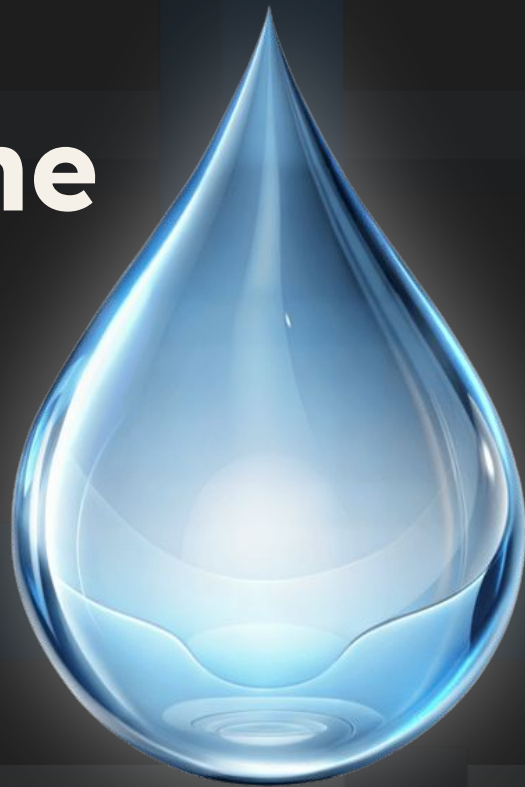


IoT Security Capstone

Experiment Results, Insights & Recommendations

Jacob Rivera | May 17, 2026



The Threat: Unprotected IoT Pipelines

What happens when MQTT messages travel without protection?

Eavesdrop

Anyone on the same network with a valid certificate can subscribe to every topic and silently read all sensor data in real time. This data can include pressure readings, flow rates, and gate commands.

Inject Fake Data

An attacker can publish messages that appear to come from a legitimate Hydrologic device. They can send fake readings to trigger false alarms or keep readings regular in times when a real problem is happening.

Replay Attack

An attacker records a real sensor message and sends it again later. The message they send is a legitimate message but it is old. With no way to tell the difference, the subscriber accepts it as if it were a fresh reading.

Summary: Encryption alone is not enough to protect an IoT pipeline. You also need to verify who is connecting, whether the message is authentic, and also if the message is new. All these gaps exist independently and there is no singular fix; each problem has their own fix.

Defense Architecture: Four Layers of Protection

Each layer catches what the previous one cannot.

1

TLS Encryption

Encrypts all traffic between devices and the broker.

Stops:

Outside eavesdroppers reading data in transit.

2

Mutual TLS

Requires every device to present a unique certificate.

Stops:

Unauthorized devices from connecting at all.

3

HMAC Signing

Attaches a tamper-evident signature to every message.

Stops:

Injected or modified messages from being accepted.

4

Timestamp + Counter

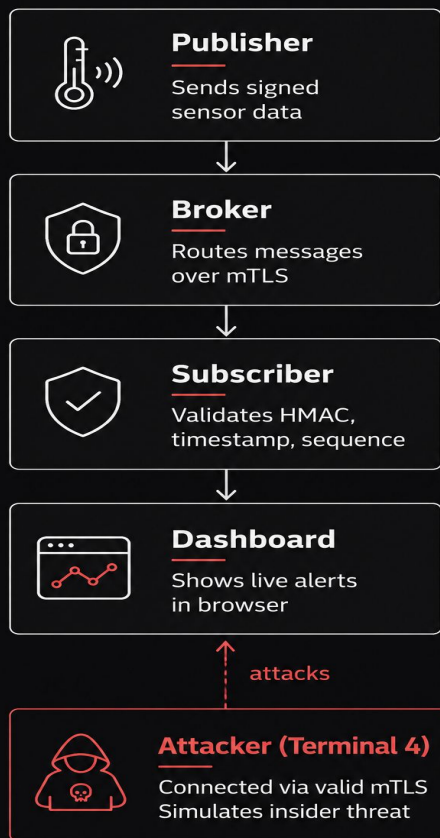
Checks message freshness and unique sequence numbers.

Stops:

Replayed messages even when HMAC is valid.

As stated in the previous slide, there is no single solution to fix all problems. You need all four layers of protection to close most, if not, all gaps

Experiment Setup: Three-Phase Attack Simulator



Setup Process

1. I started the mTLS broker in my device's terminal.
2. In a another terminal, I started the defense subscriber, which also opens the live security dashboard in the browser
3. In a third terminal, the defended publisher was started, which began sending signed and timestamped sensor readings every 5 seconds.
4. In a fourth terminal, I launched the attack simulator while monitoring both subscriber and terminal outputs

The attack simulator connected to the broker using valid mTLS credentials to simulate an insider threat scenario. It then ran three attack phases. They were listening to traffic, injecting fake sensor data, and replaying a previously captured message.

Attack Results: Every Attack Blocked

Which defense caught each attack phase and why it works.

Attack Phase	What the Attacker Tried	Defense That Caught It	Why It Works
Phase 1: Eavesdrop	Subscribed to all topics and read live sensor readings	mTLS	mTLS encrypts communication and accepts only trusted devices. Outside attackers won't be able to read the data.
Phase 2: Inject	Published fake sensor data with a fake HMAC signature	HMAC verification	The subscriber compares the correct HMAC to the fake one from the attacker. It gets rejected when messages don't match
Phase 3: Replay	Reused a recorded message that had a valid HMAC signature	Timestamp counter	Seeing the same sequence/timestamp again triggers a rejection from the subscriber.

Result: 0 attacks succeeded = 100% blocked

Dashboard: Attack Detected

Red alerts, shaking zone cards — every attack caught in real time.

⚠️ SECURITY ALERT

Attack detected and blocked by HMAC + sequence + timestamp verification!

Last security check: 11:57:30

⚠️ ATTACK DETECTED

Type:

Message Tampering

Source:

HYDROLOGIC-Device-001

Target:

Water System

✓ **BLOCKED BY SECURITY SYSTEM**

📅 SECURITY EVENT LOG

- ✓ 11:53:50 Main Building: pressure=58.21 PSI, flow=46.3 LPM
- ✓ 11:53:45 Main Building: pressure=60.44 PSI, flow=48.31 LPM
- ✓ 11:53:40 Main Building: pressure=60.75 PSI, flow=49.34 LPM
- ✓ 11:53:35 Main Building: pressure=60.03 PSI, flow=48.28 LPM
- 🚨 11:53:33 **BLOCKED: Replay Attack from HYDROLOGIC-Device-001**

Stats: Attacks Blocked

The numbers tell the story — every attack rejected, every valid message accepted.



SECURITY STATUS

All systems operating normally. HMAC + timestamp + sequence verification active.

Last security check: 11:56:25



SECURITY EVENT LOG



11:53:33

BLOCKED: Replay Attack from
HYDROLOGIC-Device-001



11:53:30

Main Building: pressure=60.23 PSI,
flow=51.18 LPM



11:53:30

BLOCKED: Message Tampering from
HYDROLOGIC-Device-001



11:53:25

Main Building: pressure=60.68 PSI,
flow=51.23 LPM



11:53:20

Main Building: pressure=61.52 PSI,
flow=48.12 LPM

110

Valid Messages

2

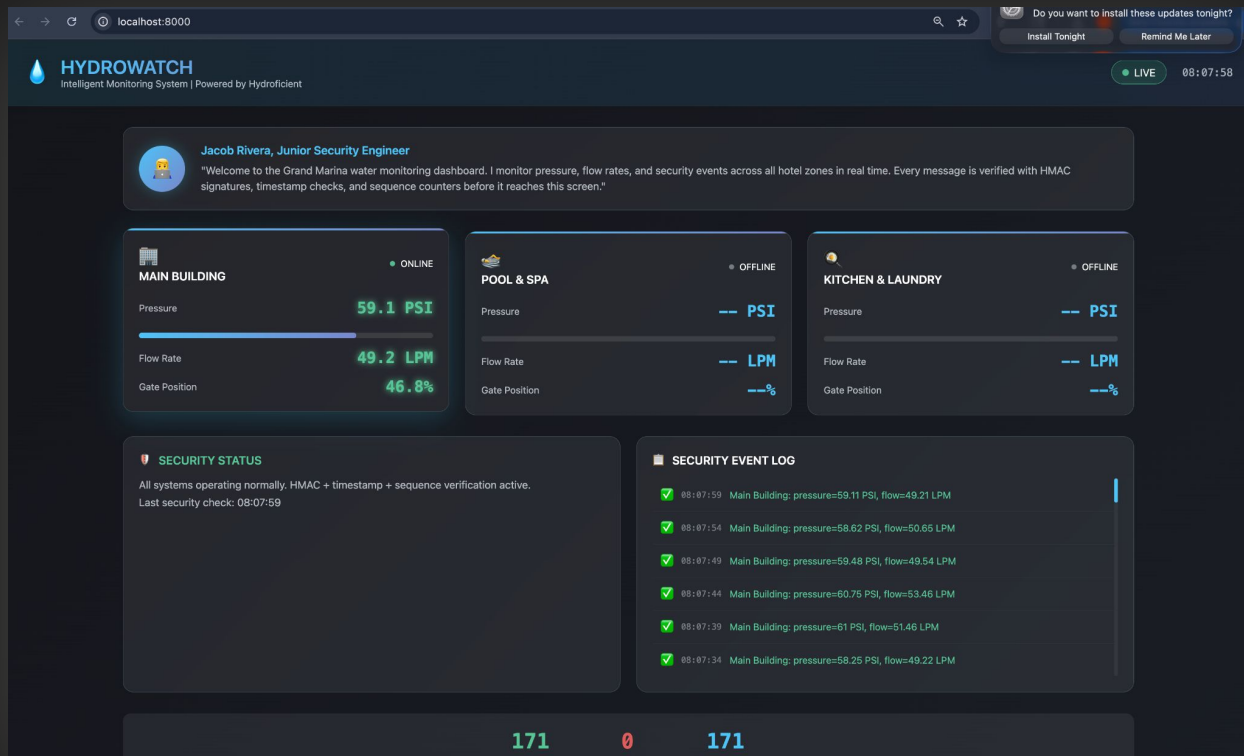
Attacks Blocked

112

Total Processed

Custom Dashboard: Making It Mine

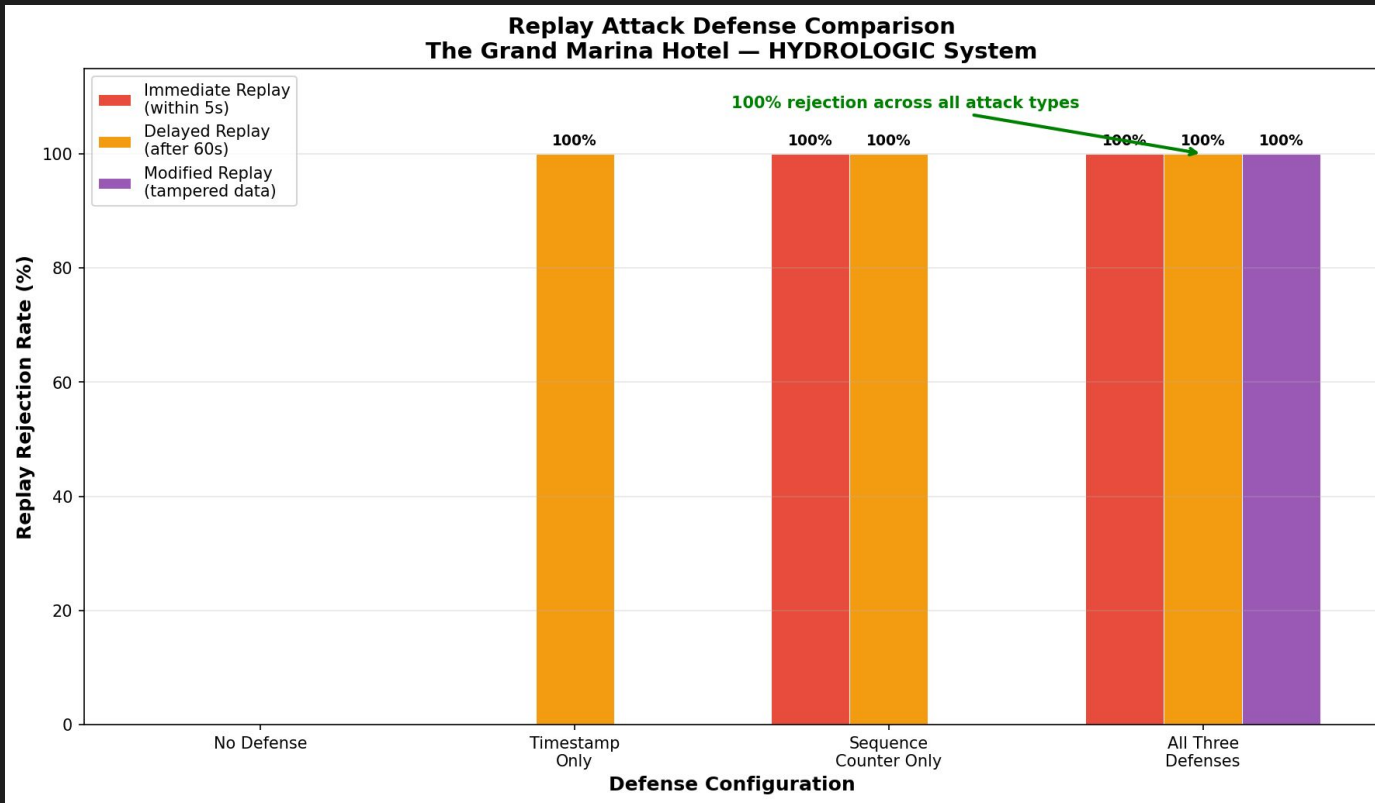
Personalized branding, custom color theme, and live sensor data.



Bonus

Security Analysis Chart

Defense effectiveness: rejection rates across 4 configurations and 3 attack types.



Key Insights

What this experiment revealed about IoT security.

1 No single defense is enough

mTLS stops outsiders but not compromised insiders. HMAC stops forgeries but not exact replays. Timestamps stop delayed replays but miss fast ones. All layers need to work together to cover every attack type.

2

The sequence counter is the most reliable replay defense

It caught every replay attempt regardless of timing, while the timestamp check only caught replays after the 30-second window expired. In a real attack where someone acts within seconds, the counter is what actually blocks it.

3

Using multiple security checks makes the system stronger

This experiment showed that one security features alone can't stop every attacker. Even though the attacker had a valid access to the broker, the HMAC verification and timestamp checks were still able to detect fake and replayed messages.

4

The effectiveness of a dashboard

The subscriber terminal showed the same rejection messages, but the browser dashboard made the attack visible in under a second without reading through log output.

Recommendations for Grand Marina

What should happen next to secure the IoT infrastructure at scale.

1

Deploy the defended scripts to all three devices

The defended publisher and subscriber scripts are already made, tested, and confirmed working against the different attack types. Deployment requires no new hardware.

2

Limit what devices can publish or subscribe to

The attack simulator showed that an authorized device can still read all sensor data. Topic-level access control would limit what each client can subscribe to.

3

Run regular security tests

Regular testing can catch security or configuration problems early before they affect the system.

4

Create a certificate rotation schedule

With the new resort wing adding more devices, a documented process for issuing, renewing, and revoking certificates will keep provisioning organized at scale.