OMV

The Good, the Bad, and the Ugly of OT Log Sources

Security Monitoring in OT – One Log at a Time

Oct 3, 2025 – Philipp Kreimel-Haindl – IT-SECX FH St. Pölten

Who am I

Philipp Kreimel-Haindl
OT Security Officer @OMV

What I Do

- Responsible for OT security strategy and implementation in Energy
- Ensure compliance with NIS directive
- Built and operate OSMS (OT ISMS) to manage OT security risks

Where I Come From

- Former OT/ICS Security Consultant @Limes Security
- Lecturer & Researcher in Industrial Security @UAS St. Pölten
- Bachelor & Master IT/Info Security @UAS St. Pölten

The Setup

OT networks are the backbone of critical infrastructure – but from a logging perspective, they often feel like a black box

- **Legacy systems** dominate, and many were designed with security in mind
- Logs, if they exist, are often **operational**, not security-focused
- **Vendors** sometimes actively **block** integration with external SIEMs

Challenge: How do we gain meaningful visibility without breaking safety rules, vendor contracts, or fragile systems?

The Cast of Characters

Let‘s meet our Logs

The Cast of Characters

Let‘s meet our Logs

- **The Good** – logs we can actually work with
- **The Bad** – partial visibility, painful integrations, or nothing at all
- **The Ugly** – borderline impossible, or absurd workarounds

The Bad – Logs That Make You Work for It

The bad is what most of us see daily

- **Vendor restrictions**: many vendors only allow integration with their own SIEM or monitoring tool – usually commercial and closed
- **No agents allowed** on OT systems – seen as too risky
  - Warranty / liability!
- **Limited log support**: CSVs, proprietary databases, undocumented formats
- Logs often **operational only** (process values etc.), not directly useful for security

Result: The data exists, but extracting useful log data is complex and resource-intensive

Vendors – No One-Size-Fits-All

Vendor-specific solutions offer value – but interoperability remains a challenge in multi-vendor OT environments

DeltaV Lifecycle Services - Security Information and Event Management (SIEM) for DeltaV™ Systems

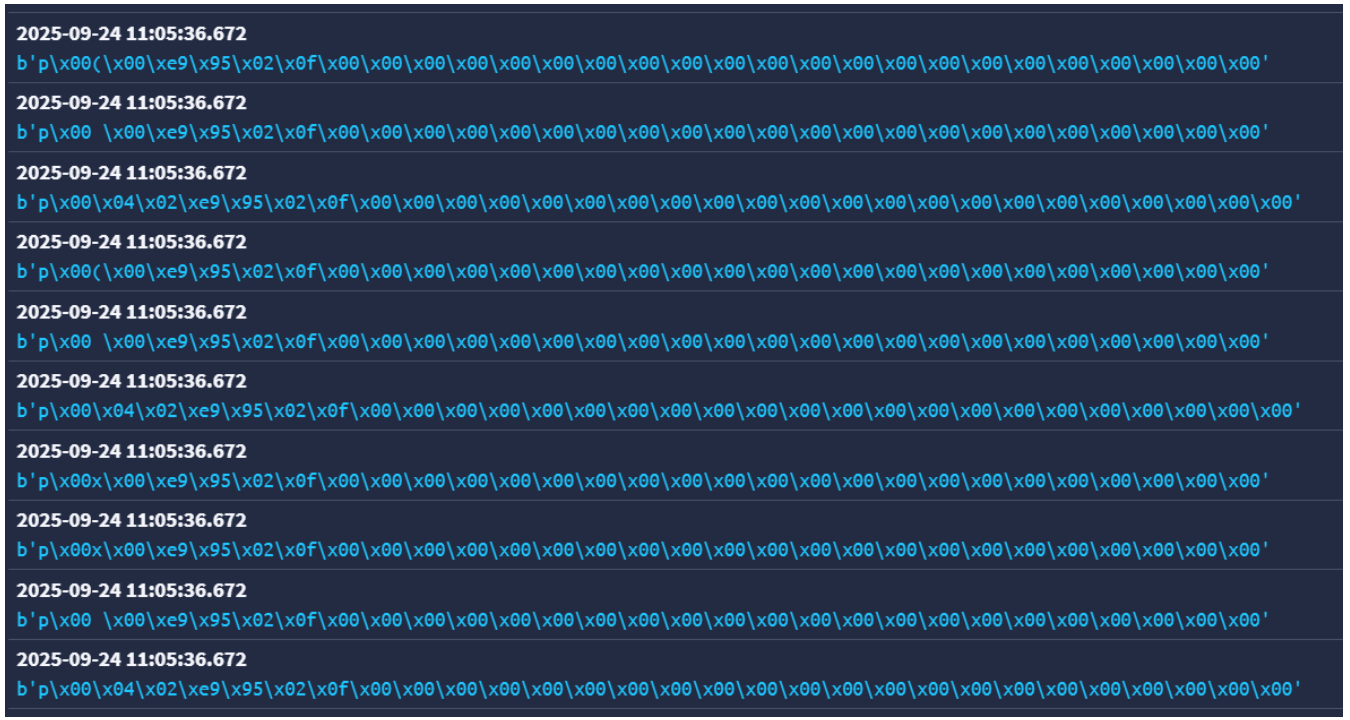
CONTACT US >

Product Description

The Bad – Unreadable by Design

You get either nothing or just a binary dump with no context

- **Legacy systems** often support no logging at all
- If you get something, it‘s often not usable
  - **Binary dumps** with no structure or documentation
  - Logs stored in **proprietary databases** with no API access

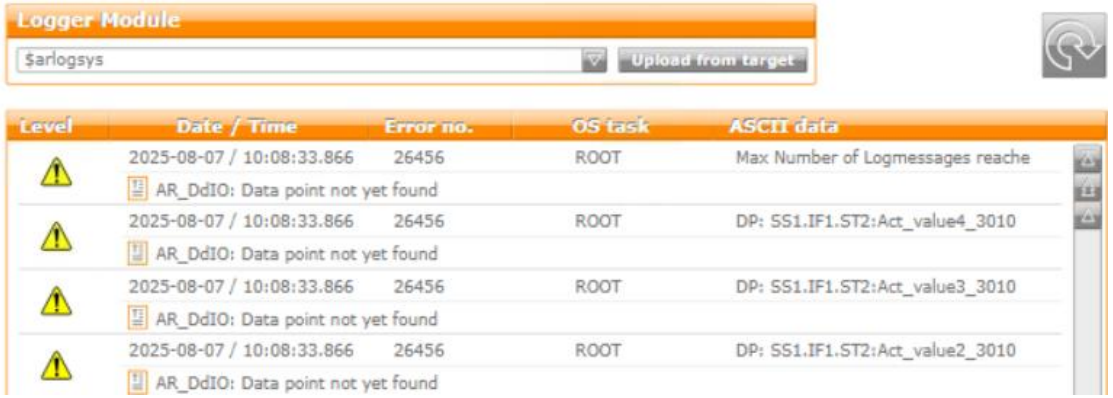


The Ugly - Logs That Barely Deserve the Name

Then there's the ugly side of OT logging

- Systems providing logs via sysdump download
- Complex setups to avoid third-party software
  - WEC / WEF across two separate domains over HTTPS
  - Nightmare to configure and maintain in a production OT network
- Logging becomes a project in itself
  - more effort than the actual monitoring

These sources technically exist, but they're so impractical that most teams give up




```
▼<Section title="Logger">
  ▼<Logger>
    ▼<Module Id="0" Name="ApLogAnsl" Version="1.02.0" Address="0x0234A184">
      ▼<Module>
        <Entry Id="0" Severity="1" Timestamp="2025-03-11_08-26-42" Info="0x00010000" Error_no.="0"
        Entered_by="TC#8" Description="" ASCII_data="StartModule (ApCnfAnslS) activated"
        Binary_data="00 00 00 00"/>
        <Entry Id="1" Severity="1" Timestamp="2025-03-11_08-26-42" Info="0x00010000" Error_no.="0"
        Entered_by="TC#8" Description="" ASCII_data="Found modified configuration (ApCnfAnslS)"
        Binary_data="00 00 00 00"/>
        <Entry Id="2" Severity="1" Timestamp="2025-03-06_07-13-32" Info="0x00010000" Error_no.="0"
        Entered_by="TC#8" Description="" ASCII_data="StartModule (ApCnfAnslS) activated"
        Binary_data="00 00 00 00"/>
        <Entry Id="3" Severity="1" Timestamp="2025-03-06_07-13-32" Info="0x00010000" Error_no.="0"
        Entered_by="TC#8" Description="" ASCII_data="Found modified configuration (ApCnfAnslS)"
        Binary_data="00 00 00 00"/>
```

WEC (Windows Event Collector), WEF (Windows Event Forwarding)

The Ugly – Readable but Usable?

Looks like text, behaves like chaos

- Ugly logs often demand **significant integration** effort
  - Building custom parsers for proprietary formats
  - Filtering out noisy or irrelevant traffic
  - Handling inconsistent timestamps or missing metadata
- Ongoing **maintenance** can be time-consuming
  - E.g. system upgrades break your parsers
  - New software packages / new log formats
- **Effort vs. value** is questionable

```
xx.xx.2024--xx:xx:42.472--runtime--
32618--AnslDriver_003--ctrl_lbk_info-
-REDACTED--1717675542.472000000--
REDACTED--1--$arlogconn--4128--31839-
-anslSvTcpsRecv_0xa4ebde8---1---1--
rrConnection created:
REDACTEDREDACTED_s--4128--8542--
196608--0--0--0--f6-54-24-41-84--
None--3--61--00 00 00 00 0a 00 00 00
00 00 00 00--00 00 00 00 0a 00 00 00
00 00 00 00
```

Coping with the Bad & Ugly - Techniques That Work Anyway

How to survive the bad and ugly:

- **Passive collection**
  - SPAN ports, TAPs, or IDS sensors
  - **BUT:** Encryption in OT networks is coming (although slowly)
- **Protocol converters / relays**
  - turn vendor-specific logs into syslog or JSON
- **ETL pipelines**
  - scripts or tools (e.g., Logstash, NXLog, custom parsers) to normalize CSVs, XML, or other awkward formats

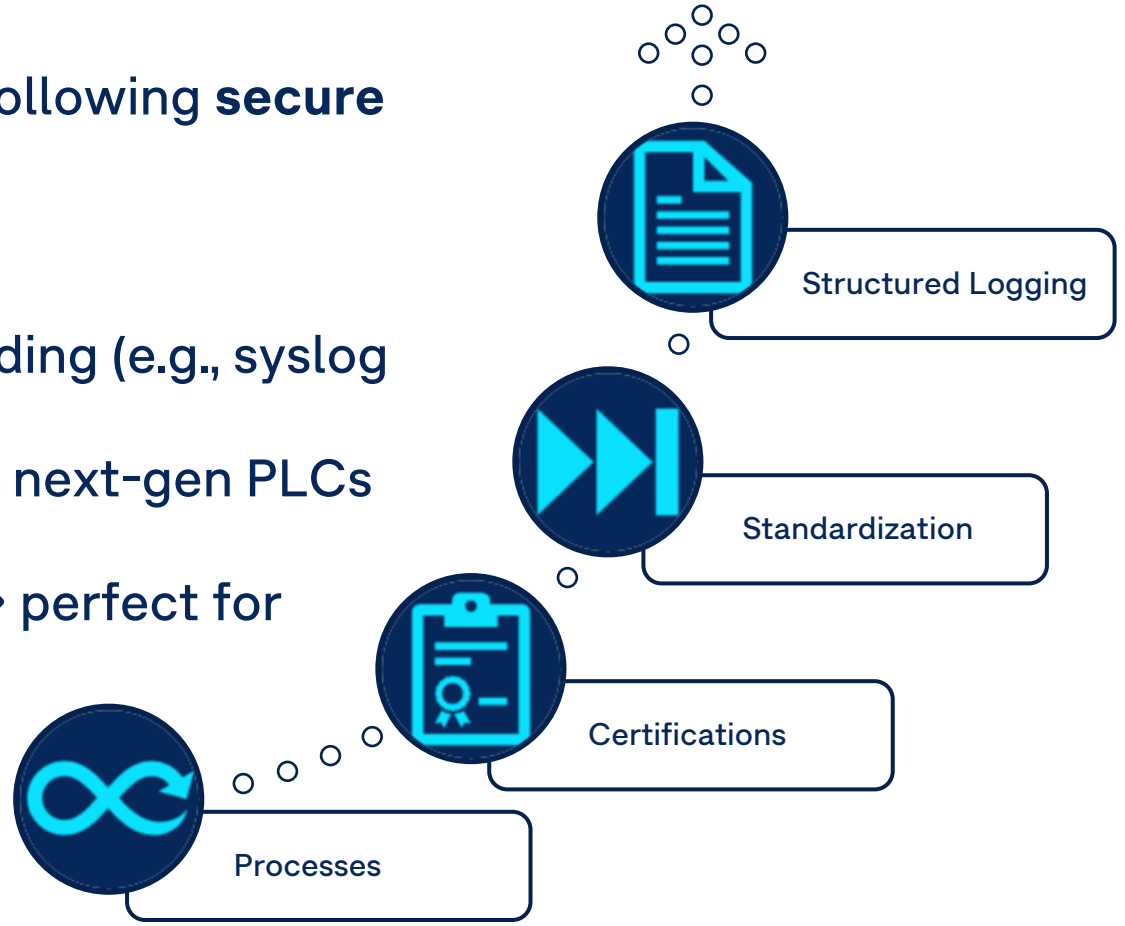
Creativity and patience are often the only way to make OT logs usable

The Good – Yes, It Exists!

The good news: things are improving

- Positive trend that OT manufacturers/vendors start following **secure development lifecycles** and processes
  - OT devices feature security capabilities
  - **IEC 62443** certifications
- Modern OT devices now support standard log forwarding (e.g., syslog over UDP/TCP, JSON)
  - Example: modern firewalls, industrial switches, and next-gen PLCs can directly forward to a SIEM
- Sources provide structured, machine-readable data → perfect for correlation and alerting

Good sources are not yet common in OT, but the trend is positive



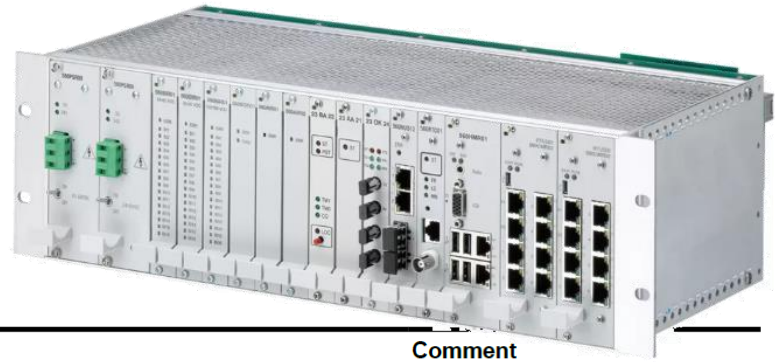
Logging Capabilities of State-of-the-art RTUs

Modern Remote Terminal Units (RTUs) offer robust logging capabilities, including:

- **User access** (logins, logouts)
- **User management** (creation, deletion, roles)
- **Audit trails** (log file integrity)
- **System diagnostics** (config / firmware changes)
- **Control operations** (influence physical process)

Review and prioritize – create OT-relevant alerts

- Config change out of business hours?



Event id	Event name	Comment
1120	Log-in failed - Unknown user	Reason logged but not shown as error message
1130	Log-in failed - Wrong password	Reason logged but not shown as error message
1150	Log-in failed - Password expired	Logged and shown as error message
1170	Log-in failed 3 times	
1210	Log-out (user logged out)	
1220	Log-out by user inactivity (timeout)	Timeout configurable
1370	Viewed Security Event logs successfully	
1670	Viewed security event list failed	
1720	User Accounts reset to factory default	
2110	User account created successfully	
2120	User account deleted successfully	
2130	User account creation failed	
2140	User account deletion failed	
2160	New role assigned to user successfully	
2162	Permission added successfully	Permission assigned to role successfully
2170	User role assignment removed successfully	Role withdrawn from user successfully

RTU Logs – Parsing and Normalization

Transforming raw RTU log messages into structured data for analysis

```
rule "Extract RTU Event ID and Host Name"
when
    // Define the condition for when this rule should be applied.
    has_field("message") && to_string($message.application_name) == "rtu-app"
then
    // Split message on pipe char and set new fields
    let message_split = split("[|]", to_string($message.message));
    set_field("rtu_event_name", to_string(message_split[0]));
    set_field("rtu_event_id", to_string(message_split[2]));
    set_field("rtu_host_name", to_string(message_split[4]));
    set_field("rtu_type", to_string(message_split[5]));
    set_field("rtu_user", to_string(message_split[6]));

end
```

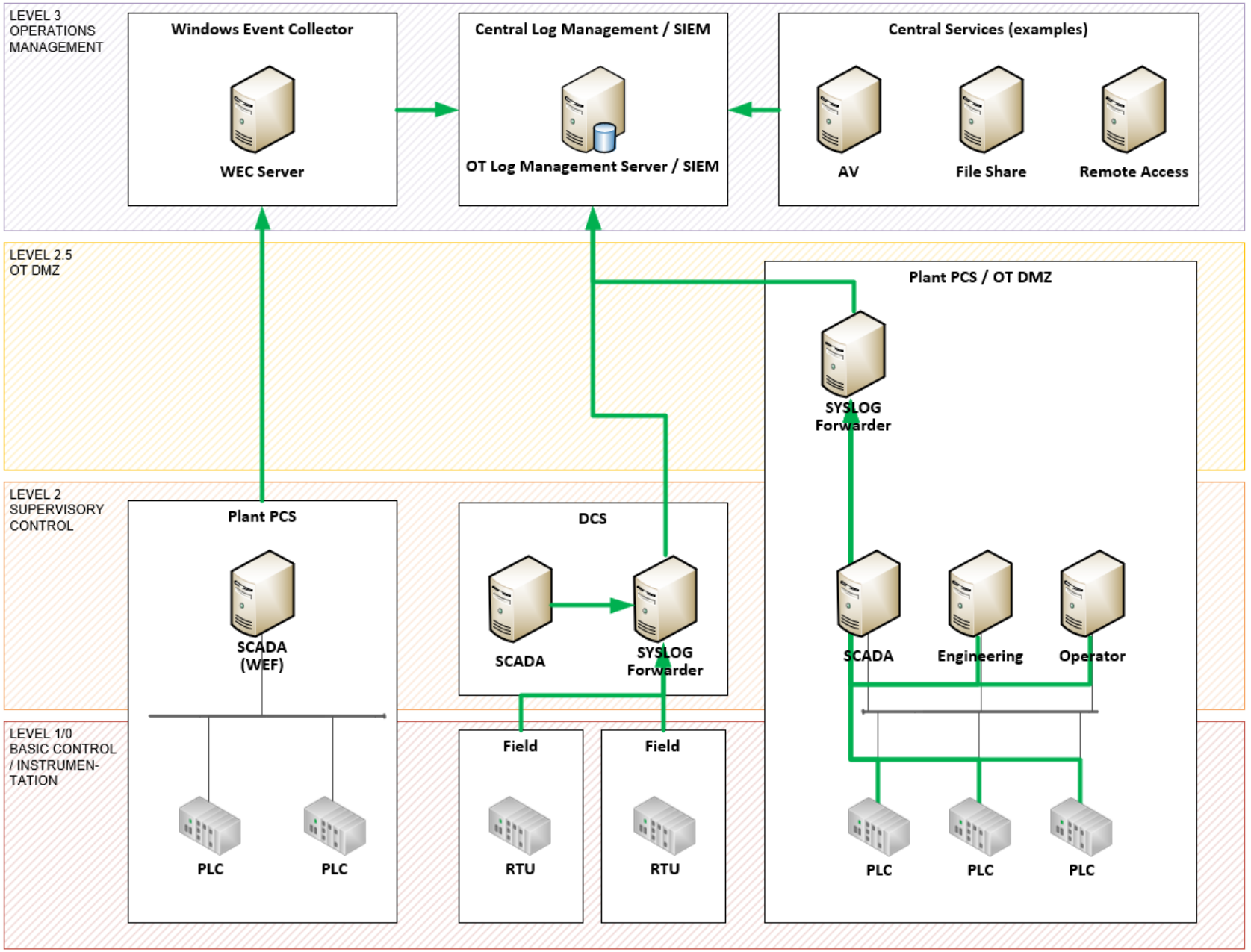
```
message
01:Benutzeranmeldung fehlgeschlagen - Falsche
Anmeldedaten||1130|00110|<hostname>|RTU500|<name>
```


```
{
    "application_name": "rtu-app",
    "facility": "user-level",
    "facility_num": 1,
    "level": 5,
    "message": "01:Benutzeranmeldung fehlgeschlagen -
    Falsche Anmeldedaten||1130|00110|<host>|RTU500|<name>",
    "rtu_event_id": "1130",
    "rtu_event_name": "01:Benutzeranmeldung fehlgeschlagen
    - Falsche Anmeldedaten",
    "rtu_host_name": "<host>",
    "rtu_type": "RTU500",
    "rtu_user": "<name>",
    "source": "x.x.x.x",
    "timestamp": "2025-01-01 10:00:00.000"
}
```

Architecture Example

A hybrid approach

- **Collection layer**: syslog relays, WEC collectors, parsers
- **Log Management / SIEM**: normalization pipelines, dashboards, alert rules adapted for OT
- **OT zone**: agentless, safety-first



Human Factor & Politics

Technical barriers are only half the problem

- Operations teams prioritize uptime and safety → logging is seen as “nice to have”
- Vendors may resist integration, citing warranty or compliance
- Organizational silos: OT and IT often don’t share priorities

Success requires

- Involving security early in design
- Building trust with OT engineers & showing quick wins
- Respect for safety-critical processes

Lessons Learned – Plan Early or Pay Later

- (OT) Security must be integrated in the of **design phase**
  - The sooner, the better – retrofitting is painful
- Example: RTU syslog server IP is embedded in firmware
  - Must be set during initial setup
  - Next downtime might be years away
- Visibility is not just technical – it's strategic
- **Avoid SIEM sprawl** – centralize where possible

Lessons Learned – Best Practices

Making the most of what you've got

- Agentless ≠ data-less – there are still ways to collect
  - Use passive methods where agents aren't allowed
- Normalize and filter – don't collect everything, just the logs that matter
- **Alert logic** must be contextualized for OT
  - Emphasis is on safety, reliability, and availability
  - E.g. "cryptographic certificate expired" use case might not be critical in OT
  - A **configuration change** of a PLC outside of business hours might be

Got ugly logs, let‘s talk!