Project CyDES Research Report

Authors: Twan Erens, Artem Tiutenko, Jordi Peeters, Faustas Litvinaitis, Chahid Yahia

DECEMBER 24, 2025

GROUP 7

Table of Contents

Introduction 2

Infrastructure Overview 3

pfSense Firewalls 3

VPN Connections (OpenVPN and IPsec) 3

Intrusion Detection (Suricata) 4

SCADA Server (Node-RED) 4

PLC Simulators 4

Data Retriever Server 5

Approach 5

Literature & Field Research 5

Client Interviews & Requirements Gathering 5

Lab Simulation Setup 6

Penetration Testing & Experimentation 6

Penetration Testing Results 6

Conclusion 9

Recommendations 9

Introduction

Project CyDES (Cybersecurity for Distributed Energy Systems) is an initiative that focuses on protecting distributed energy installations from modern cyber threats. These installations include systems such as hydrogen production powered by solar energy and seasonal storage of green gases. These new energy solutions play an important role in the transition to sustainable energy. However, because they are highly distributed and rely heavily on digital systems and network connectivity, they are more exposed to cybersecurity risks.

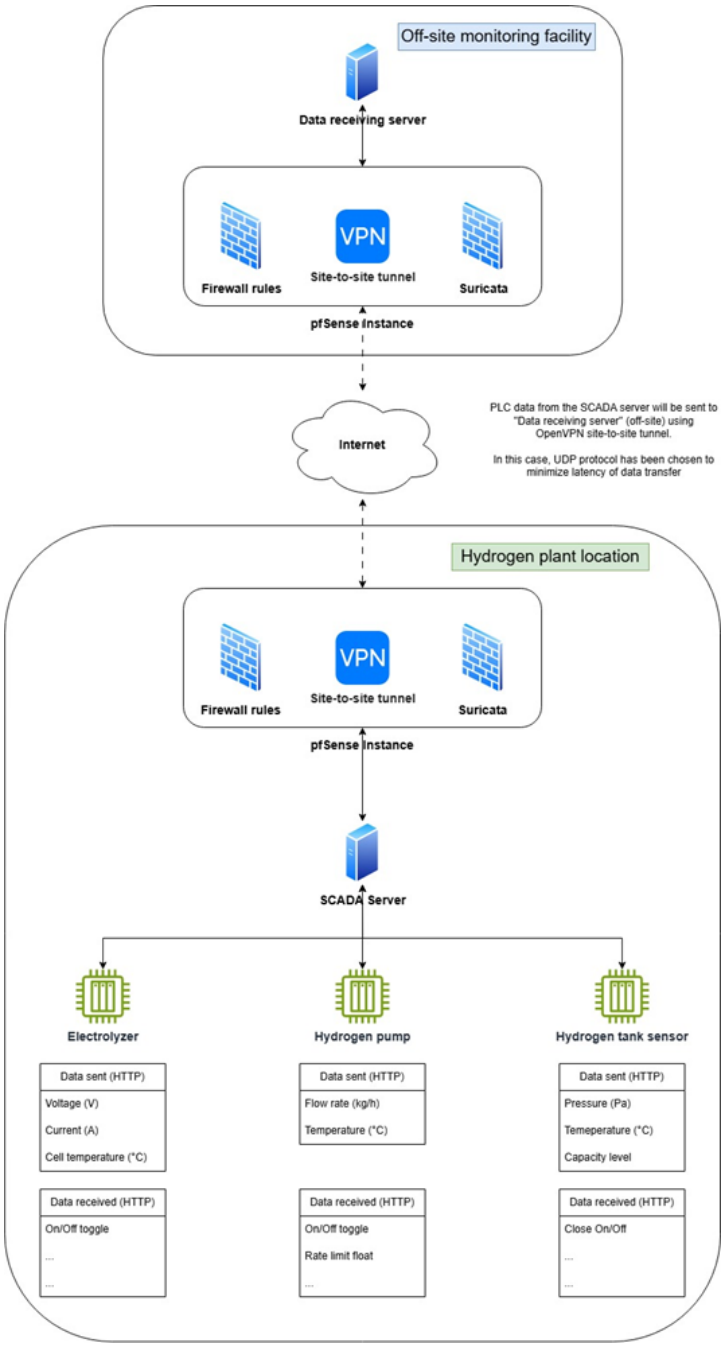
In recent years, cyberattacks on critical energy infrastructure have shown how serious the consequences can be. Such attacks can lead to systems going offline, safety risks for people and equipment, financial losses, and in some cases even environmental damage. As these energy systems move from small experimental pilots to real-world, large-scale operation, ensuring strong cyber resilience is no longer optional but absolutely necessary.

The CyDES project was created in response to this growing need. It is a collaboration between an industry partner, the “Power-to-Power” hydrogen initiative, and Fontys University. The project had an exploratory goal: to better understand how current cyber threats could impact a distributed hydrogen energy facility and to identify practical and effective security measures to protect it. To achieve this, we analyzed possible vulnerabilities within a simulated hydrogen production network. We then demonstrated different cyberattacks and corresponding defenses in a controlled laboratory environment.

The main objective was to show how insecure components within the infrastructure could be better protected or their risks reduced by making changes to network architecture, configurations, and operational practices. This report describes the project background, the assumed infrastructure and its individual components, the methods used during the analysis, and the most important findings from the penetration testing activities. It also provides clear recommendations for improving the cybersecurity of distributed energy systems, especially hydrogen-based facilities. The language is intentionally kept professional but easy to understand, so it can be useful for both academic supervisors and industry stakeholders.

Infrastructure Overview

PLC = Programmable Logic Controller



As shown in the infrastructure image, the hydrogen plant site and the off-site monitoring facility are connected by secure networks. Each site has a **pfSense firewall/router** that separates its local network from the internet and hosts VPN tunnels. The pfSense devices manage routing and firewalling for each site and link the two sites via encrypted VPN connections. This makes it appear as if the two networks were on a single private network, even though the data travels over the public internet.

pfSense Firewalls

Each site's pfSense router is the main gateway for its local network. The pfSense platform is an open-source router/firewall system known for strong security features. At the hydrogen plant, the pfSense router enforces network rules and directs traffic between the plant's subnet and the wider infrastructure. At the monitoring site, another pfSense router provides the same role for the off-site network. Both routers also run the VPN software that connects the two sites (see next section). In addition, the hydrogen-site pfSense runs **Suricata** on its interfaces to watch for threats (details below).

VPN Connections (OpenVPN and IPsec)

The two sites use site-to-site VPN tunnels so that all communication is encrypted and secure. In this setup, IPsec was eventually adopted as the site-to-site VPN solution. But before that decision was made. We did some experiments with an OpenVPN solution. Which was also a valid option. Both solutions can be applied on the pfsense router. Both an OpenVPN tunnel and an IPsec tunnel Have been configured in a test environment. Each tunnel creates an encrypted "tunnel" over the Internet, protecting the data from eavesdroppers. OpenVPN is an open-source VPN 3rd party solution applicable to pfSense. It uses SSL/TLS encryption (for example AES-256) and runs over UDP (port 1194 by default) to securely link the sites. IPsec is another standard VPN method that uses pre-shared keys and uses modern encryption to secure site-to-site traffic. In practice IPsec came out on top because it was built into pfsense and was easier to

configure than OpenVPN. Connection loss did not occur on the modern versions of pfsense in our prototype environment. Which proves its reliability.

Intrusion Detection (Suricata)

On the hydrogen plant side, the pfSense firewall also runs **Suricata**, an IDS/IPS engine. Suricata inspects all incoming and outgoing packets against a set of threat rules. In this setup, Suricata is enabled on each interface of the pfSense router so that it can watch every area of traffic. If malicious or abnormal traffic is detected (for example unusual packets or known attack patterns), Suricata can flag or block it. In short, Suricata provides a layer of threat detection at the network boundary of the hydrogen plant, helping to catch attacks or intrusions in real time.

SCADA Server (Node-RED)

Within the hydrogen plant network is the **SCADA server** running **Node-RED**. Node-RED is an open-source, flow-based programming tool used here as the logic and data hub for the plant. It gathers data from the PLC simulators and handles data flows automatically. For example, one Node-RED flow periodically polls each PLC simulator for its current sensor values and writes these to a local file. A second flow waits for requests from the monitoring site: when it receives an HTTP GET request at /plcdata, it reads the stored PLC data file, converts the data into JSON, and returns it in the response. These flows effectively implement the SCADA logic – collecting sensor data from the plant and making it available over the VPN connection. In this way, Node-RED controls the data flow: it continuously polls (gets) data from the PLCs and then serves this data to the monitoring side upon request.

PLC Simulators

The hydrogen plant network includes several **PLC simulator** machines. These simulate the behavior of real programmable logic controllers (PLCs) in a plant. Each PLC simulator runs software that responds to queries (for example, HTTP GET requests on port 8001) with process data such as temperature, pressure, or other sensor readings. The Node-RED SCADA server on the plant network communicates with each PLC simulator over the local LAN. For example, a Node-RED HTTP request node sends a query to a PLC simulator's address and port, and the PLC responds with its current data. The data from all PLCs is collected and managed by the SCADA server. In our setup, there are three PLC simulators (PLC1, PLC2, PLC3), and the Node-RED flows are configured similarly for each one. This means the SCADA server regularly gathers data from each PLC and aggregates it for monitoring.

Data Retriever Server

On the off-site monitoring side is the **Data Retriever Server** (an Ubuntu machine) that collects plant data. Over the VPN, this server reaches across to the hydrogen plant network. It periodically sends requests to the Node-RED SCADA server to get the latest data. Specifically, it issues an HTTP GET to the /plcdata endpoint exposed by Node-RED. When this request arrives, Node-RED's flow reads the stored PLC data and sends it back in JSON format. In effect, the Data Retriever acts as a client: it asks for data and receives the plant status. The retriever then displays this data for analysis in a dashboard so that operators at the monitoring site can view the hydrogen plant's real-time data. Because this happens over the secure VPN, the data travels safely between the sites.

Overall, the components work together so that the hydrogen plant's PLC data is collected, secured, and available to remote users. The pfSense routers and VPNs form a secure network backbone, Suricata watches for threats on the hydrogen site, the Node-RED server runs the SCADA logic collecting PLC data, and the Data Retriever server pulls that data from across the VPN. This setup ensures that the monitoring site has real-time access to plant data, while all communication remains protected and monitored.

Approach

To achieve the project goals, our team followed a structured approach combining research, stakeholder input, lab simulation, and security testing. The methodology can be summarized in four main phases:

Literature & Field Research

We began with an in-depth study of existing literature on industrial control systems (ICS) security and known cyber threats to energy systems. This included reviewing recent incident reports, academic papers, and security advisories relevant to distributed energy and hydrogen production. The team mapped out potential attack vectors on a hydrogen facility's network (e.g,. common ICS vulnerabilities, network intrusion techniques) and surveyed best-practice defenses. This groundwork provided a threat overview and informed the design of our simulated environment with realistic considerations.

Client Interviews & Requirements Gathering

Alongside the literature study, we gathered practical insights from project stakeholders. Because we had little direct contact with the company, most communication was done with the project coach, who acted as a stakeholder. The coach helped us understand a typical hydrogen facility setup and common security concerns.

Since the project is part of a collaboration, it was important to understand the company's needs. However, because we did not receive details about the existing network, the system and network

design were based on assumptions and best practices, supported by the coach's input. These discussions helped define the project scope and goals and ensured the simulation included relevant elements such as VPN usage and protected data. The stakeholder input also helped shape the research questions and success criteria.

Lab Simulation Setup

We built a laboratory environment to simulate a hydrogen production site and a remote monitoring center. This section describes how the lab environment was implemented.

The setup was built using virtual machines and simulated network devices. Firewalls, servers, and industrial control components (PLC and SCADA) were emulated to produce realistic data exchange and basic control behavior. Some parts of the system were later in testing intentionally configured with weaker security settings, such as outdated software or legacy VPN options, to allow the study of security vulnerabilities. The lab environment was designed so that security features could be easily enabled or disabled, making it possible to compare secure and insecure configurations. Before starting the experiments, the setup was checked to ensure it reflected key characteristics of a real distributed energy system, including network separation, remote access, real security, and realistic data traffic.

Penetration Testing & Experimentation

After setting up the lab environment, we performed penetration testing in a controlled setting. The testing focused on identifying weaknesses in the network components and PfSense firewall.

The testing focused on identifying weaknesses in the network, servers, and Programmable Logic Controller components. Different attack scenarios were carried out to assess how misconfigurations and outdated components could be abused. We also compared situations with and without key security measures, such as the use of a VPN, to understand their impact on overall security.
All tests were carefully documented, including the actions taken and their results. After identifying weaknesses, security measures were applied and tested again to observe improvements.

Penetration Testing Results

The penetration testing phase yielded valuable insights into the vulnerabilities of the infrastructure and the effectiveness of certain defenses. Below is a summary of the key findings from our tests (each corresponds to a specific scenario we examined in the lab)

External and Internal Network Enumeration

The penetration test started with network enumeration to understand which systems and services were exposed. External port scans were performed from a DHCP network targeting the WAN interfaces of the pfSense firewalls and other exposed servers. These scans did not reveal any services that could be reachable from the internet

Internal port scans were also conducted from within the network, targeting the public network, PLC network, and the SCADA server. These scans showed that once an attacker gains internal access, many critical systems become reachable. This emphasizes that perimeter security alone is not enough and that internal segmentation and secure tunneling are required to limit attack impact.

Importance of a VPN for Secure Communication

One of the most important test scenarios focused on what happens when no VPN is used between the hydrogen plant and the monitoring site. In this scenario, the site-to-site VPN was intentionally disabled and replaced with port forwarding rules combined with weak firewall configurations. This exposed the SCADA server and PLC simulators directly to the public internet.

Clear-text Traffic Risks

Without a VPN, SCADA data was transmitted over the internet using unencrypted HTTP. Data sent over the public internet passes through many independent networks and routers, often owned by different organizations. If the traffic is not encrypted, any compromised or malicious device along the route can inspect, log, or modify the data. During testing, unencrypted SCADA traffic was successfully captured, and the contents were fully readable, including process data and responses from PLC simulators.

When the same communication was routed through the VPN tunnel, packet captures still showed traffic, but the payload was encrypted and unreadable. This clearly demonstrated that a VPN is the only effective way to protect data confidentiality and integrity during inter-site communication.

Active Attacks Without a VPN

The absence of a VPN also enabled active attacks, not just passive monitoring. In one test, a machine placed outside the network was able to send HTTP POST requests directly to a PLC simulator. This resulted in the PLC simulator being turned off remotely. Such an attack could have serious consequences in a real industrial environment, potentially causing process disruptions, safety risks, or physical damage.

This attack was only possible because internal systems were exposed directly to the internet. With a proper site-to-site VPN in place, PLCs and SCADA systems would only be reachable from trusted internal networks, not from the public internet.

Man-in-the-Middle and Denial-of-Service Effects

A man-in-the-middle style test was also attempted by targeting WAN endpoints from the DHCP network. Although traffic interception was not successful in this scenario, the attack caused packets to be dropped, resulting in a denial-of-service condition. The data retrieval server was no longer able to receive data from the PLC site, demonstrating how insecure network designs can also impact availability, not just confidentiality.

Risks of Using Outdated Systems

Another major part of the penetration test focused on outdated software. For this test, the pfSense firewall was downgraded to version 2.5.2, which is known to contain multiple publicly documented vulnerabilities.

Exploitation of Known Vulnerabilities

One critical vulnerability tested was CVE-2021-41282. This vulnerability allows command injection through a diagnostic page in pfSense. Using an exploit framework, root access to the firewall was successfully obtained. Once root access was gained, the firewall was fully compromised, allowing complete control over routing, firewall rules, and traffic inspection.

This result demonstrates that running outdated firewall software can completely undermine the security of the entire network. Even if VPNs and firewall rules are configured correctly, a single unpatched vulnerability can allow attackers to bypass all protections.

Impact After Firewall Compromise

After gaining root access to the firewall, further actions were possible, such as capturing traffic from both network interfaces. This means that all data passing through the firewall, including internal communications, could potentially be monitored or manipulated. In a real-world scenario, this would give an attacker full visibility and control over the industrial network.

Risks of Outdated VPN Configurations

The penetration test also evaluated older IPsec configurations available in outdated pfSense versions. Older configurations allow the use of aggressive mode, which uses a shorter and less secure handshake process. If an attacker is able to capture this handshake, the pre-shared key can potentially be brute-forced offline.

If successful, an attacker could establish their own VPN tunnel into the internal network, gaining the same access as a trusted site. This shows that even VPNs can become insecure if outdated protocols or weak configurations are used.

Conclusion

In this project, we studied the cybersecurity of a distributed hydrogen energy system by building a representative lab environment and testing its security. The results showed that modern distributed energy systems face real cyber threats, but also that practical security measures can greatly improve their protection.

The penetration tests demonstrated that outdated software and unencrypted connections can lead to serious risks. These include unauthorized access to sensitive sensor data, manipulation of equipment, and even full control over critical network devices. When systems were updated and all communication between sites was secured using a VPN, these attack paths were effectively blocked.

Recommendations

Based on our findings, we recommend a defense-in-depth approach for securing distributed energy infrastructures. First, all devices such as firewalls, SCADA systems, and PLC controllers should be kept up to date, and vendor security patches should be applied regularly. Second, secure site-to-site VPN connections (or equivalent encryption methods) should be used for all communication between field sites and central systems. Sensitive data should never be transmitted over public networks without encryption.

In addition, strong network segmentation should be maintained to separate industrial control systems from corporate IT networks and direct internet access. Intrusion detection and prevention systems should be used to monitor unusual activity, and strict access controls should be enforced so that only authorized users and systems can access critical components. Regular security assessments and penetration tests should also be performed to identify and fix weaknesses before they can be exploited.

In summary, the Project CyDES initiative shows that cybersecurity is a key requirement for distributed energy systems. By using updated systems, encrypted communication, and layered security controls, organizations can significantly reduce the risk of cyber incidents. As the energy sector becomes more digital and distributed, these measures will be essential to ensure safe, reliable, and resilient operations. The results of this project provide guidance not only for the simulated hydrogen facility, but also for similar distributed energy systems facing modern cyber threats.