False Data Injection Attacks Against Distribution Automation Systems

Author: Ryan McAndrews, Ryan@McAndrews.me
Advisor: *Clay Risenhoover*

Accepted: *September 28, 2024*

Abstract

Utility companies increasingly rely on automated switching to provide their customers with a reliable electric power supply. These automation systems, which offer significant operational benefits for the utility, also present a growing security risk. With adequate knowledge of the function of these automation systems and their algorithms, an adversary could implement false data injection to amplify or hide real issues that these automation systems solve. An adversary would be challenging to detect without authentication, auditing, or appropriately logged field data. Researchers have proposed these attacks theoretically, and this research intends to evaluate claims of unidentifiable false data injection attacks experimentally.

1. Introduction

In most developed countries, the tools and technologies that generate, transport, and deliver electricity are transforming quickly. The power industry, often slow to adopt technology, is effectively entering its information age (Khalid, 2024). By using automation systems, utility companies see significant benefits from reduced manual labor, increased revenue from power delivered, and decreased regulatory penalties from interrupted power service. However, the adoption of these technologies poses unexplored risks to power companies.

1.1. Distribution System Automation

The distribution system is one portion of the power grid that has recently adopted automation. The distribution system is the last mile of power lines and equipment that takes power from a nearby substation to its final destination. This section is often the least maintained and prone to damage caused by lightning strikes, tree limbs, and wildlife. Due to the high frequency of problems, Distribution Automation (DA) systems reduce mundane tasks for engineers and improve the time to restore power (Shafik, 2023).

DA systems utilize motorized equipment, sensors, and communication networks to dramatically reduce the manual work required to maintain the power grid (Leniston, 2022). These systems often have automated controls to open and close electrical switchgear to restore or disconnect electricity from sections of the power grid. Two examples of these systems are FLISR and DMS systems.

FLISR, or Fault Location, Isolation, and System Restoration, is an automated system that re-routes power during system disturbances, such as broken power lines, lightning strikes, and wildlife. These disturbances are called ‘faults’ in the industry. FLISR systems are powerful because they can restore power to customers in seconds, whereas before, it would take the utility company hours or even days to do so manually (Shafik, 2023).

DMS, which stands for Distribution Management System, is a broader technology used by utilities to help optimize the usage of their electrical system. DMS systems can perform complex calculations and adjust where power is routed on the system to optimize

power quality and maximize the use of power equipment. The DMS system performs these adjustments by closing and opening automated switchgear around the power grid. By performing these calculations and making these changes in real-time, a DMS system can drastically reduce the workload of system operators.

1.2. Historical Attacks on Power Systems

The power grid and some of its core communication protocols have been the target of attackers for decades. The BlackEnergy malware targeted synchrophaser systems, causing denial-of-service attacks on protective devices as early as 2008 (Khan, 2016). For the CRASHOVERRIED attacks in Ukraine, the malware issued commands to the switchgear directly from their exploit. In doing so, it generates loggable events that indicate their presence (Jaatun, 2018). Since then, researchers have proposed that the next evolution of these attacks may target automation systems through false data injection attacks (FDIA) (Irfan, 2023). As the use of DA systems increases, the possibility of actual incidents appears to rise.

These FDIAs on DA systems may be more subtle to detect than attacks have been historically, with researchers suggesting these attacks can be unidentifiable (Peisert, 2020). By manipulating field data fed to DMS and FLISR systems, an adversary could cause the automation systems to issue similar commands on their behalf without generating additional logging. They can evade detection and focus the utility's attention on what they believe is a malfunctioning FLISR or DMS system.

Prior researchers (Peisert,2020) have mentioned the impact of FLISR-based FDIA, but only from a theoretical perspective. Additionally, considerable research on detecting FDIA in the smart grid focuses on automation systems other than FLISR and DMS systems (Basumallik, 2020). This research details an unidentifiable FDIA in a lab setting to qualify if it is practically possible and unidentifiable. The power industry is reviewing security standards for communication protocols, and researching FDIA in these DA systems may help evaluate their effectiveness. This work may be beneficial in testing existing solutions for detecting FDIA in areas outside the bulk electric system.

2. Research Method

2.1. Test Setup

Demonstration of these attacks utilized commercial FLISR and DMS software, commercial switchgear controls, a communication network, and a 'Hardware-in-Loop' (HIL) testing system.

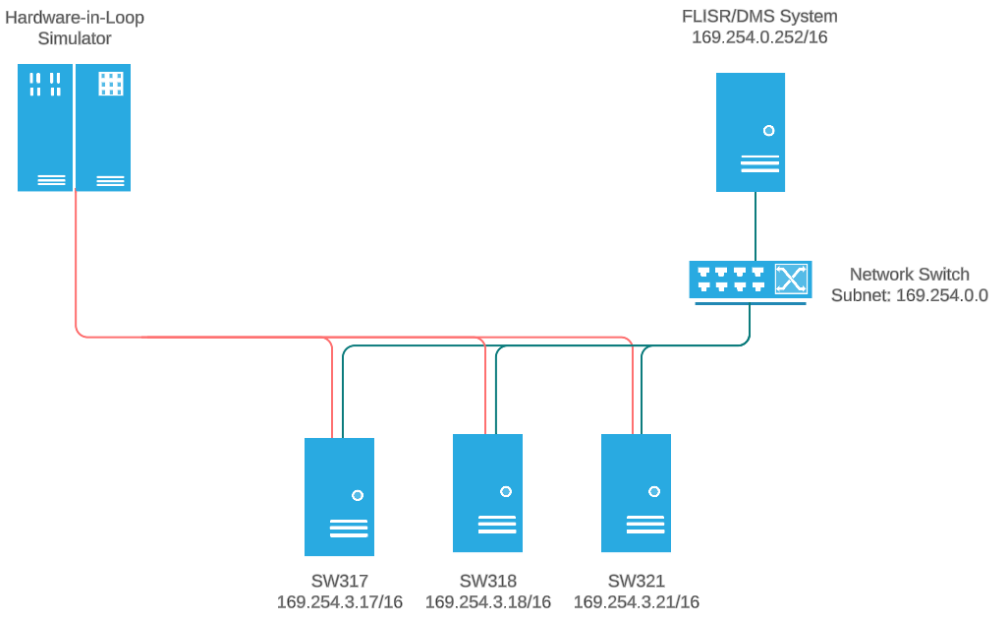

Figure 1: Lab Environment

The communication network utilized a single flat network communicating over the DNP3 protocol. The switchgear controls communicate sensor data and status information to the FLISR and DMS systems over DNP3. In response to events on the system, the FLISR and DMS systems will analyze the field data and then issue commands to the switchgear controls over DNP3.

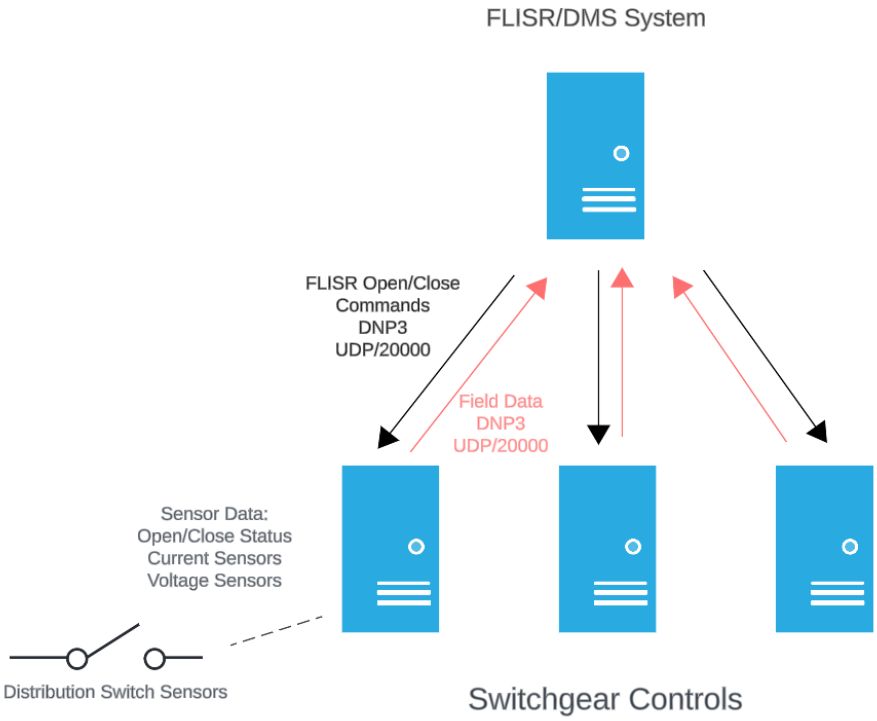

Figure 2: Data Flow Diagram

The HIL system feeds sensor data into the switchgear controls as if the controls were reading from sensors on the power grid. Under normal operating conditions, the signals from the HIL system to the switchgear controls will appear normal. During events, the HIL system will adjust those signals to match the expected behavior of the events.

Figure 3 shows a simulated loss of voltage event for the left-side source. The HIL system adjusts the signals for voltage and current to zero to simulate a loss of power across the distribution system. However, it will continue to provide other signals like switch position status since those remain unchanged during a power outage.

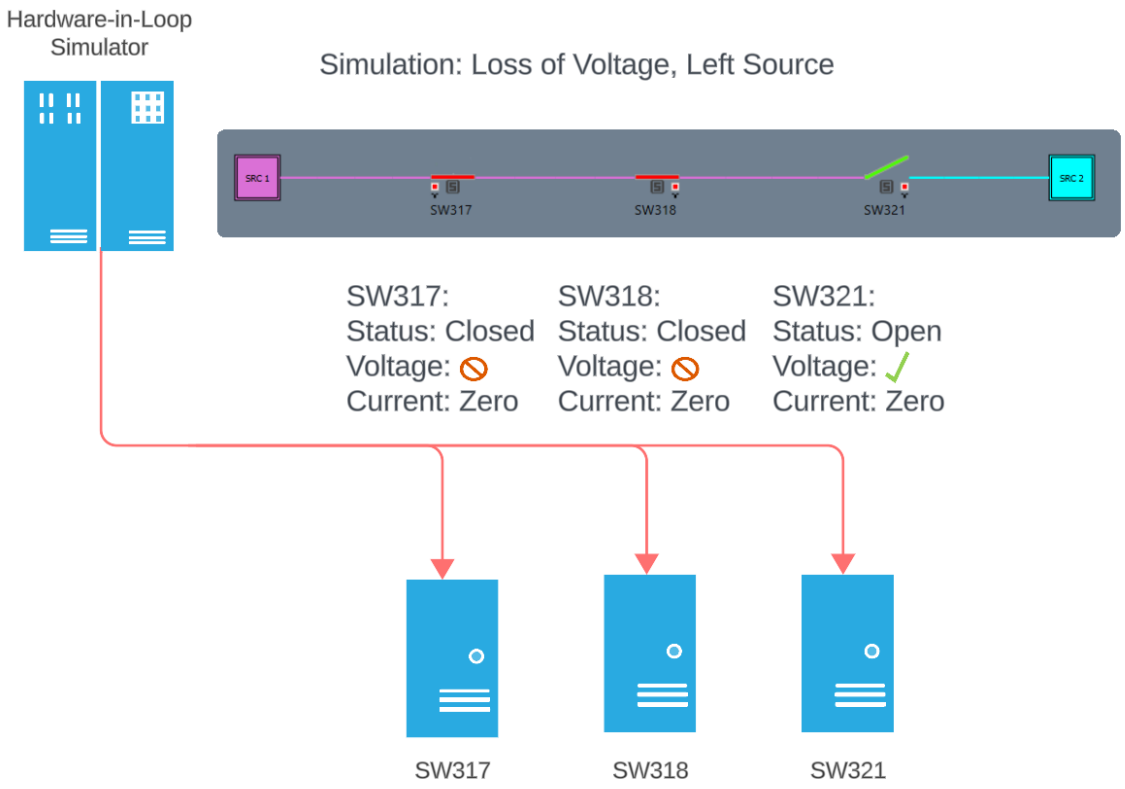

Figure 3: Hardware-in-the-Loop Simulation Example. The HIL simulator sends a bad voltage signal to the voltage sensors of the devices connected to the left source.

2.2. Adversary Simulation

A machine-in-the-middle (MitM) device with packet sniffing and manipulation capabilities is added to the communication network to demonstrate an adversary's presence on the system. All software used by this MitM device is open-source.

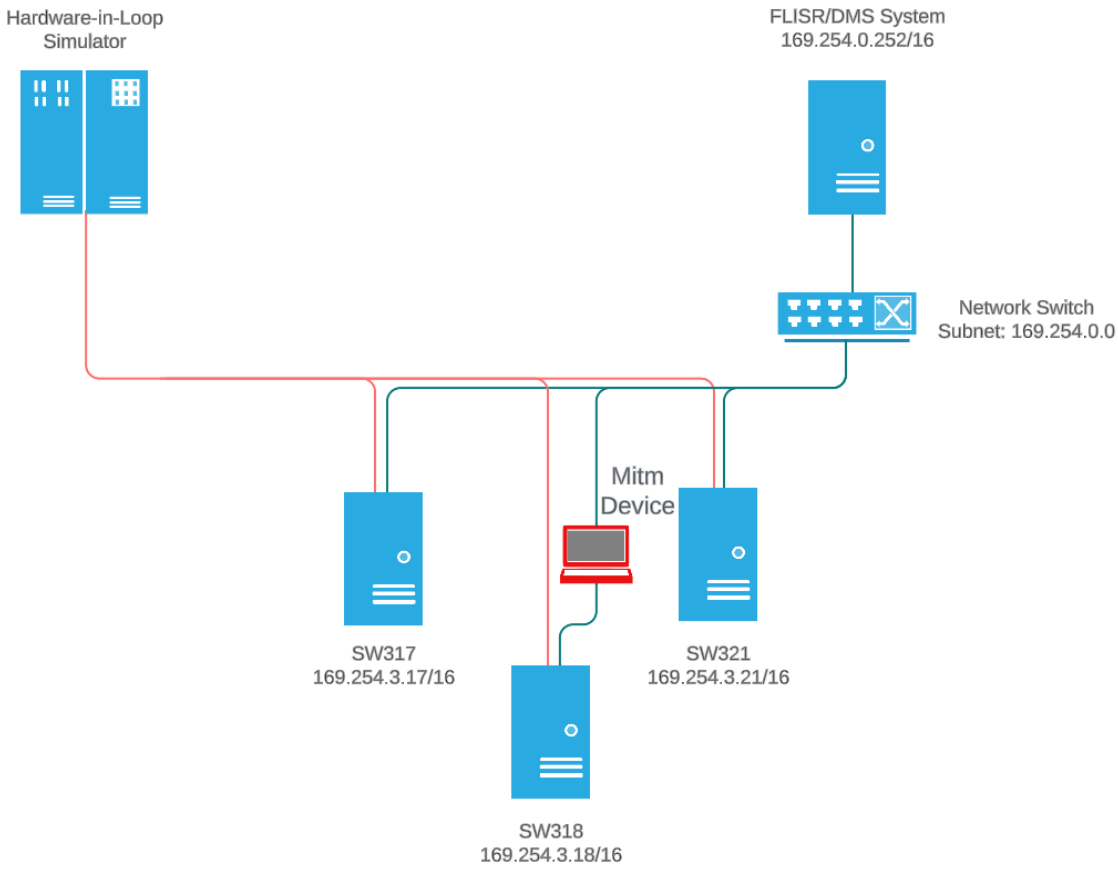

Figure 4: Machine-in-the-Middle (MitM) Device added to test environment. The device can see any network traffic between SW318 and the network.

2.3. Open-Source Tools

2.3.1. Scapy

Scapy is a packet manipulation library developed by Philippe Biondi in 2003. This Python extension library can sniff and dissect incoming packets and construct and send packets. Using Scapy's automated packet inspection and construction library on the Mitm device, the adversary can monitor traffic between the switchgear controls and the FLISR or DMS system. Depending on the type of attack, inspection of the field data may trigger the MitM device to begin dropping, manipulating, or forging new packets from the switchgear controls.

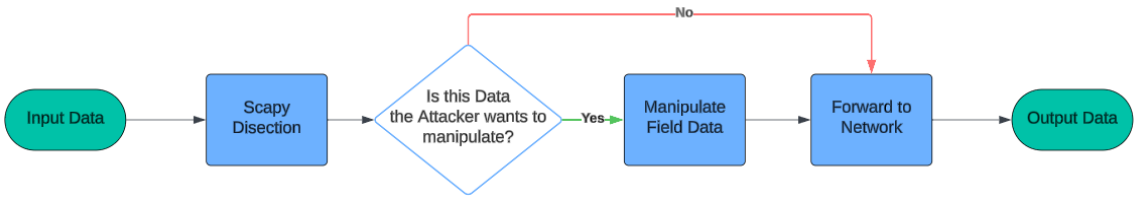

Figure 5: Scapy Logical Diagram

2.3.2. Iptables and NetfilterQueue

Iptables is a commonly used IP packet filter for Linux. The Mitm device utilizes Iptables to send all DNP3 traffic on UDP port 20000 to a queue for Scapy to evaluate.

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
NFQUEUE    udp  --  anywhere             anywhere             PHYSDEV match --physdev-in enp3s0 udp spt:20000 NFQUEUE num 0

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Figure 6: Iptables forward rule to direct DNP3 traffic to a queue.

NetfilterQueue is a Python library that provides software tools access to packets filtered within Iptables. The MitM device utilizes NetfilterQueue to give Scapy access to the queue set in Iptables.

```
def main():

    queue = nfq()
    queue.bind(0, packet_listener)
    try:
        print("Running Queue")
        queue.run()
    except KeyboardInterrupt:
        print("Removing iptables rule")
        os.system('iptables -D FORWARD 1')
```

Figure 7: Python code to feed the DNP3 traffic queue to packet_listener. packet_listener is the Scapy dissection function.

The MitM device is placed directly between the FLISR/DMS system and one of the switchgear controls. Traffic is routed through the MitM device as if it were a layer two network switch and is vulnerable to manipulation by Scapy during the simulated attack.

2.4. Commercial Software

The FLISR, DMS, Switchgear control, and HIL software are commercial products. Where necessary in this paper, information like communication data structures and specific data register locations have been obfuscated to protect against unintentional disclosures.

2.5. DNP3 Protocol

DNP3 is a commonly used protocol in both water and electrical utility industries. DNP3 is a part of the IEEE Std 1815-2012. All devices in this lab communicate using DNP3. DNP3 uses UDP port 20000.

IEEE Std 1815-2012 includes features like secure authentication extensions to the DNP3 protocol, but its usage has not matured (Rosborough, 2019). The devices in this test setup do not implement those features, including pre-share keys or a Public Key Infrastructure. Further discussion on these security features' use, or lack of use, can be found in the Recommendations section.

2.6. Running Experiments

The HIL system provides the FLISR and DMS systems with realistic steady-state data from the switchgear controls. The field data from the controls in this steady-state setup will be the baseline from which the FLISR and DMS systems decide to take autonomous action. This paper refers to these baselines as the 'control experiments.'

Depending on the type of attack, the adversary may manipulate traffic during steady-state conditions. During steady-state attacks, the tests initiate by manipulating packets within the MitM device. Some attacks may rely on external events, like power outages, before triggering. As the DNP3 communications go through the MitM device, it detects changes in the state of the systems based on the field data it sees. For these opportunistic attacks, the tests initiate by triggering a simulated fault event within the HIL system.

Each scenario will run without any data manipulation to set a baseline of behavior. These baselines will be the control experiments. Experiments are compared against these control experiments to determine the impact of the attacks and if there were any indications of compromise.

2.7. Data Collection

There are three primary data sources during an experiment:

1. Switchgear Control Logs
2. FLISR or DMS System Logs
3. Packet Captures from Network switches

These key details are pulled from these data sources to evaluate the differences between the control and the experiment:

- Open or Close Operations of Switchgear
- Warnings/Errors/Logs indicating system intrusion
- Network logs indicating the presence of the adversary

2.8. Experiments

2.8.1. Moving Faulted Segments

This experiment involves the attacker manipulating data fed to a FLISR system during a fault event on the distribution system. This is an opportunistic attack, where the device waits for a signal and then manipulates data.

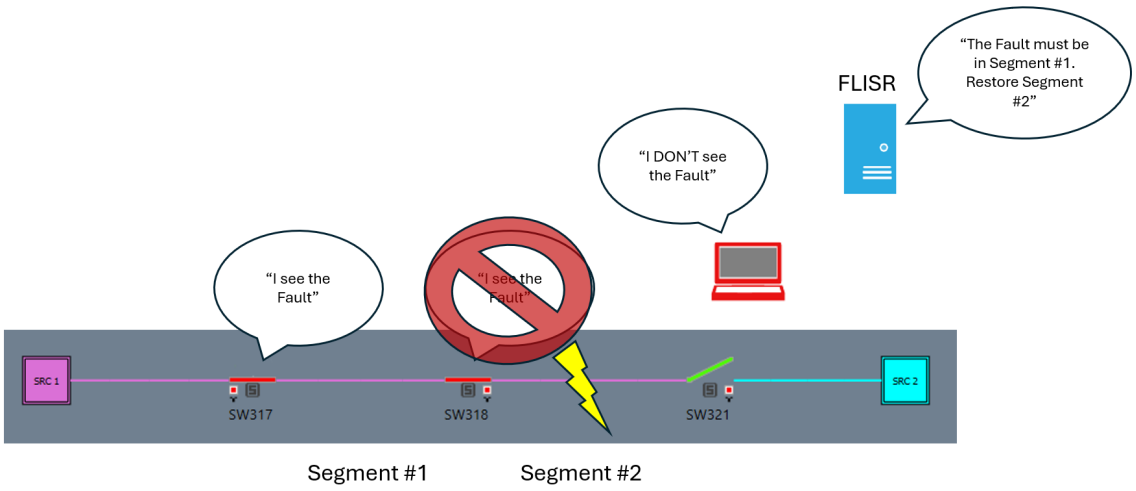

Figure 8: Moving Faulted Segments Example. SW318 cannot tell the FLISR system the fault is in Segment #2 because the MitM device feeds the FLISR system malicious data.

The FLISR system aims to locate the fault, isolate it, and restore power to unfaulted sections. The attacker's goal during this experiment is to manipulate field data and cause the FLISR system to detect the fault in the wrong segment.

2.8.2. Falsely Shed Load

This experiment involves the attacker manipulating data fed to a DMS system during steady-state. This steady-state attack involves transmitting falsified field data back to the DMS system, indicating that the power lines are overloaded in this system section.

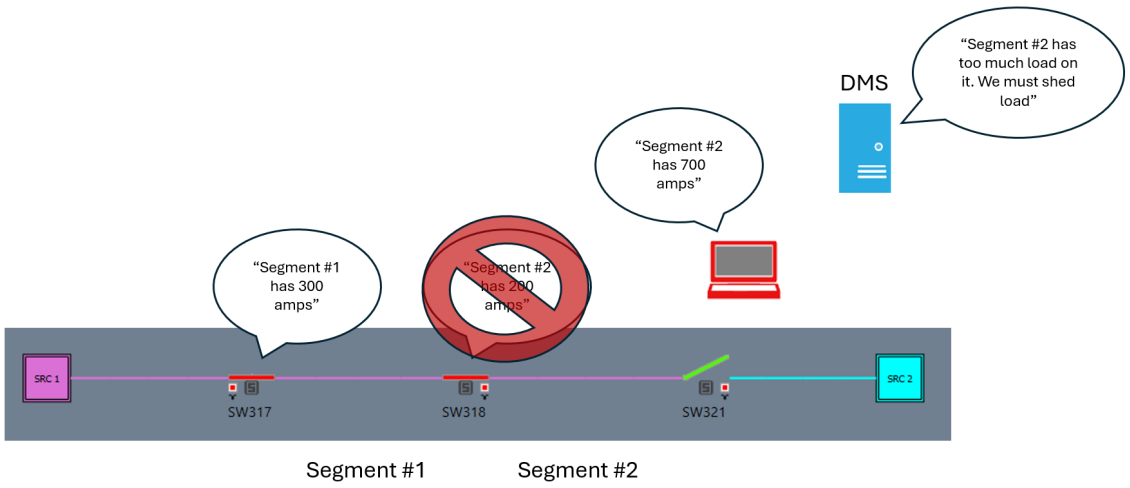

Figure 9: Falsely Shed Load Example. The MitM device tells the DMS system that SW318 is overloaded, causing the DMS system to cut off power to Segment #2.

The DMS system's goal is to prevent equipment damage from overloading, and it will disconnect customers to reduce load. The attacker's goal during this experiment is to manipulate field data to make the system load appear larger than it is. In this experiment, the electrical load is small enough not to overload the equipment, but the DMS system will observe that the equipment is overloaded.

3. Findings and Discussion

Both experiments ran successfully in the test lab. Data were collected from the data sources identified and reviewed. Per the research method, differences between these experiments and the controls were noted as findings and later analyzed for impact.

3.1. Validity of a Finding

Results are validated based on two categories:

1) Whether or not field data manipulation resulted in the FLISR or DMS system taking additional action.
2) Whether or not field data manipulation left evidence in any of the data log sources.

3.2. Impact of a Finding

This paper categorizes impact into one of these four categories:

1) Extended outages – whether data manipulation resulted in more prolonged power outages.
2) Momentary outages – whether data manipulation resulted in momentary power outages.
3) Cascading outages – whether data manipulation resulted in power outages over a larger area of the power grid.

Potential damage to equipment or people – whether data manipulation results in potentially hazardous real scenarios.

3.3. Results

3.3.1. Moving Faulted Segments

Logs from the switchgear control devices indicate that the field devices correctly identified the fault location in Segment #2. The controls communicated this to the FLISR system, and the MitM device changed the DNP3 communication from SW318 to indicate that the fault was in Segment #1 instead.

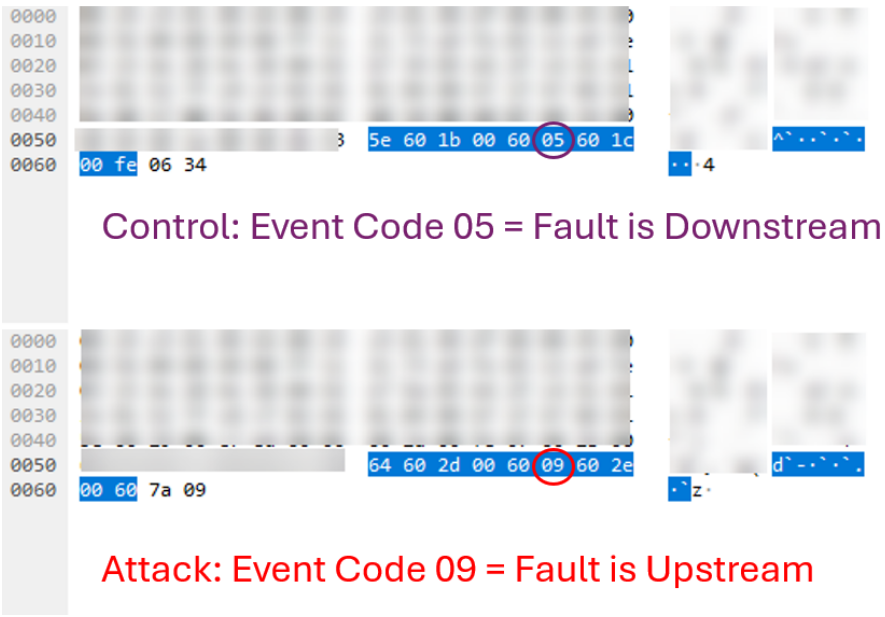

Figure 10: Packet Capture Comparison. The top packet shows the DNP data chunk sent by the switch controller at SW318. The bottom shows the modified data chunk sent by the MitM device. The circled byte contains the data the FLISR system uses to obtain fault position.

The switchgear control logs showed that the FLISR system sent an open command to SW318 and a close command to SW321. When SW321 closed, there was a cascading power outage on the alternate source.

In the control experiment, the FLISR system sent an open command to SW318 to isolate the fault and did not send the close command to SW321. This finding indicates a successful attack on the FLISR system.

Without an FLISR system, the outage would have stayed in the system fed by the primary source. In this experiment, the attacker leveraged the FLISR system to cause the same fault to affect the alternate source. The main impact of the attack was the cascading power outage that extended to the alternate source.

The FLISR system logs from the attack and control experiments were identical until the cascading outage occurred. The FLISR system indicated its behavior was consistent with a fault found in Segment #1, which was the objective of the attack.

Author Name, email@address

Inspection of packet captures at network devices did not provide additional indicators of compromise. Due to the MitM device being between network switches and the switchgear control, the network switch cannot determine field data manipulation.

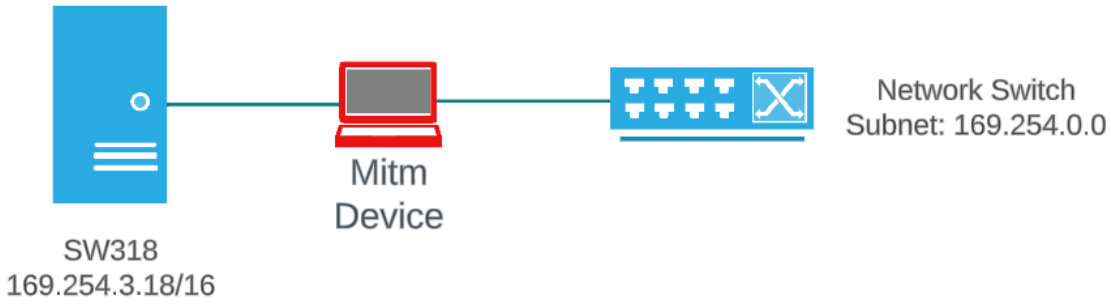

Figure 11: Network Diagram showing MitM device positioned between SW318 and the Network Switch.

Inspection of the switchgear control logs indicates that SW318 correctly identified the location of the fault in Segment #2. The switchgear control operated as intended, opening to isolate the fault. The logs do not detail the message sent to the FLISR system, meaning there is no logging to help indicate the MitM attack. While the switchgear control logs indicate the FLISR system did not operate as intended, these logs do not provide indicators of compromise. It is indiscernible if it is due to an attacker or a defect in the FLISR programming.

3.3.2. Falsely Shed Load

Logs from the switchgear controls indicate they correctly measured the system's load. The controls communicated segment loading to the DMS system, and the Mitm device changed the DNP3 communication from SW318 to report that the load in Segment #2 was higher than it was. The MitM device raised the apparent load value above the system's capacity, and the DMS system rejected the load from the system. This resulted

in a sustained power outage for the miscalculated segment of the system.

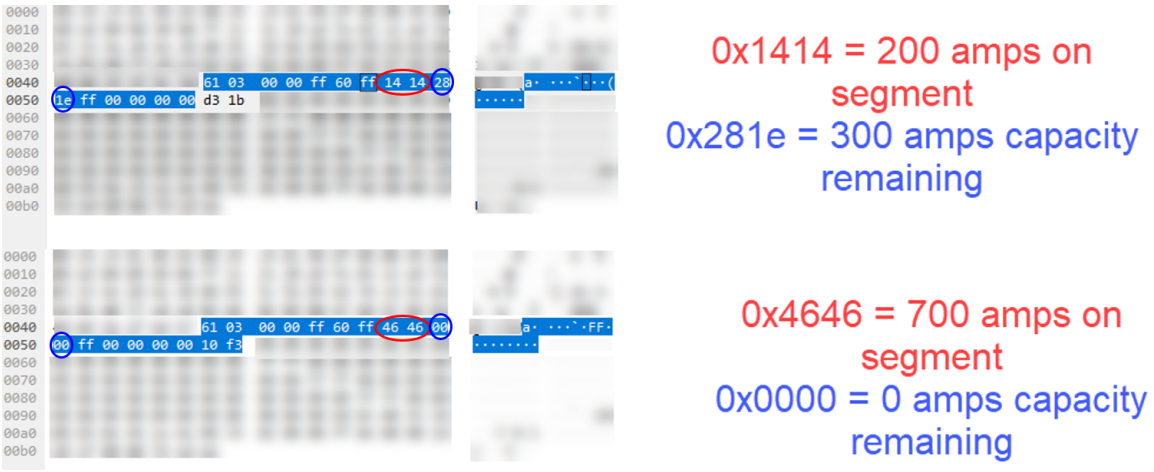

Figure 12: Packet Capture Comparison #2. The top packet shows the DNP data chunk sent by the switch controller for SW318. The bottom shows the modified data chunk sent by the MitM device. The red and blue circled bytes tell the DMS system Segment #2's loading and capacity.

The switchgear control logs showed that the DMS system calculated the available capacity of the system as 600 amps. The actual load on the segment was 200 amps, and the manipulated data indicated it was 700 amps. When the primary source rejected the load on Segment #2, the DMS system never allowed SW321 to close to pick up Segment #2.

In the control experiment, the DMS system identified that the alternate source had enough capacity to pick up Segment #2. This finding indicates that the attacker successfully caused an extended outage, which was the objective.

The FLISR system logs did not indicate compromise when compared to the control. Similarly, without any packet-capturing functionality inside the switchgear control, the network packet captures provided no indications of data manipulation. The switchgear controls do not have historical logs for load data.

3.4. Impact of MitM Device on DNP3 Message Formatting

Initially, the MitM device manipulated only DNP3 data chunks at the core of the DNP3 messages. When tested, the FLISR system dropped DNP3 packets due to incorrect checksums in the IP and UDP headers.

07/25/2024 08:17:17.895	DNP Message Rejected
07/25/2024 08:17:19.918	DNP Message Rejected
07/25/2024 08:17:21.941	DNP Message Rejected
07/25/2024 08:17:23.964	DNP Message Rejected
07/25/2024 08:17:25.988	DNP Message Rejected
07/25/2024 08:17:28.008	DNP Message Rejected
07/25/2024 08:17:30.029	DNP Message Rejected
07/25/2024 08:17:32.069	DNP Message Rejected
07/25/2024 08:17:34.091	DNP Message Rejected
07/25/2024 08:17:36.113	DNP Message Rejected

Figure 13: Rejected DNP Message Logs

The tests also included recalculating IP and UDP checksums to address this issue when data modification occurred.

```
84        #delete checksums so scapy recalculates
85        del scapy_packet[UDP].chksum
86        del scapy_packet[IP].chksum
87        packet.set_payload(bytes(scapy_packet))
88        packet.accept()
```

Figure 14: Checksum Recalculation in Scapy

After that, the switchgear controls accepted the modified DNP3 packets. Besides these two checksums, all other intentionally modified data was inside the DNP3 frame.

The MitM device used a network bridge to passively forward traffic between SW318 and the rest of the network. Except for the DNP3 traffic, which would route to the NetFilterQueue. Due to the nature of a network bridge, the Ethernet header of the original DNP3 transmission is not modified by the MitM device when it forwards the message. This is significant because a packet capture at the network switch will not see indicators of compromise within the Ethernet, IP, or UDP headers in addition to the DNP3 frame. Since the switchgear controller is considered the system's source of truth for the DNP3 data, the MitM device can go undetected.

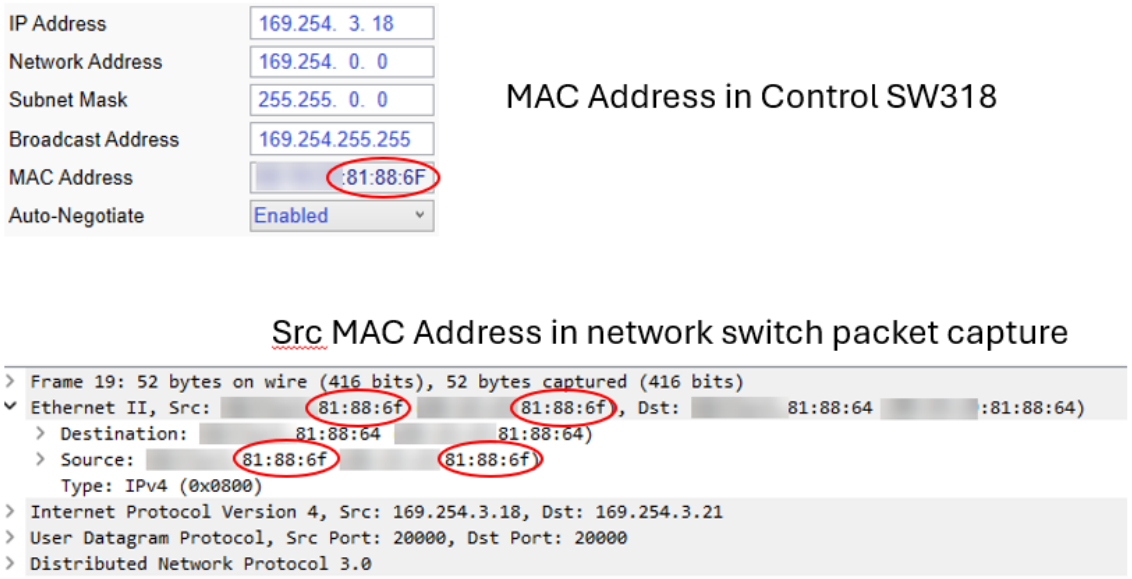

Figure 15: Comparison Showing the Mitm Device does not change the Ethernet Frame

3.5. Impact of MitM Device on DNP3 Message Timing

Timing analysis through packet captures is challenging. In a controlled environment like the test environment, network conditions are ideal, and distances to travel are short. The network likely consists of many networking devices in a live environment, whereas the test environment uses a single switch. The performance of the lab's network should not correspond to expected performance in a live environment.

There were no timing issues in the load shed experiment. The polling rate for load data was around 30 seconds, meaning the MitM device had up to 30 seconds to manipulate the packets and send them along before packet ordering issues might occur. The relocated fault test used a faster polling rate, and the MitM device sometimes struggled to manipulate packets quickly enough.

During a fault event, polling rates drop from 30 seconds to approximately a quarter of a second. The MitM device's hardware, an Intel N100 Mini PC running Ubuntu, did not have the processing power to run its script and keep packets in order. Without further experimentation, it is inconclusive if packet ordering indicates compromise or an artifact of the test setup.

3.6. Evaluation of the Test Environment

Evaluating the likelihood an attacker could execute an FDIA in a live environment is important. A comparison of the testing environment to a live environment is necessary. The two aspects to focus on are:

1) How comparable are the tools used in the test environment to those used in a live environment?
2) How feasible can an attacker get a MitM device into the live environment?

3.6.1. #1: Comparing the Test Environment to a Live Environment

The test environment used commercial switchgear controls and commercial FLISR/DMS software, which are identical to live environments today. The simulation components used in the Hardware-in-the-Loop system acted no differently than ideal sensors in a live environment. The network architecture for communications was a flat OT network without TLS/SSL encryption. Even if the OT network in the live environment used network segmentation and encryption, the end device – the commercial switchgear controller – does not support encryption. Since the controller only supports unauthenticated DNP3 communication, the physical medium between any security appliance and the switchgear controller would still be vulnerable to this Mitm attack device.

3.6.2. #2: Feasibility of the Attacker Gaining Physical Access

The distribution power system looks vastly different than substations and power plants generally associated with the power grid. The poles, wires, and boxes outside neighborhoods and workplaces make up the physical pieces of the distribution grid. On this system, often miles from physical supervision by utility personnel, these intelligent switchgear controllers can be found mounted to the side of the power poles.

To achieve the results from the experiments, an attacker needs to place a single MitM device between one switchgear controller and the communications network. The attacker would need physical access to one of these control boxes in a live environment. Once inside, the attacker would install a similar MitM device, and they could gain network access, as demonstrated in this testing.

If the attacker were to pick the lock, instead of cutting it, they could lock the cabinet again to make it appear as if they had not been there. Some SCADA-connected devices like these come with door indicator switches, which relay a signal back to a central control room that the cabinet has opened. This door signal, if monitored, could give the utility an indication of compromise. From there, the utility would require a physical cabinet inspection to identify the malicious device.

Prior research has already indicated that this kind of physical access by an attacker remains an issue (Jaatun, 2018). Solutions to this problem will require protocols with end-to-end authentication inside the field devices, hopefully alleviating some physical access concerns.

4. Recommendations and Implications

The industry is aware of the vulnerabilities present in the current popular communication protocols. IEEE PES PSCCC Subcommittees are actively developing guides for security OT systems utilizing similar protocols and developing updated standards for more secure versions, such as DNP3-SA (ieee.org, 2023). While their work focuses on solutions to these vulnerabilities, this research strengthens the argument that manufacturers and operators must move on from the status quo.

Much of the existing body of research on detecting FDIAs focuses on the mathematics of determining anomalies (Irfan, 2024) rather than on the impact of successful attacks. Research in the future that extends this topic of impact may help better identify where FDIA detection is necessary.

4.1. Recommendations for Practice

From the findings of this research, security practitioners overseeing Distribution Automation Systems should look to implement the following, where possible:

- Utilize TLS encryption and maintain a private key infrastructure across OT systems, where applicable (Holguin, 2023).
- Directly monitor and log physical access alarms from field control boxes and emphasize physical security controls for these devices.

- Communicate with equipment vendors when anomalous behaviors occur on these systems.
- Implement end-to-end authentication features in remote field devices.

4.2. Implications for Future Research

The research demonstrates that a knowledgeable attacker could manipulate Distribution Automation Systems to operate maliciously. Future research into this area could include some portions of the following:

- Reproduction of this testing using other Industrial Protocols, such as Telnet, Modbus, GOOSE, or IEC-104.
- Experiment with adding multiple synchronized Mitm devices to demonstrate further impact.
- Compare different FLISR and DMS system architectures or vendors to susceptibility to FDIA.
- Replicate these experiments with devices that support the DNP3 protocol's authentication features.
- Implement different security controls to indicate these attacks, for example, applying guidelines within IEC 62351-5.
- Test existing state estimation techniques for FDI detection focusing on FLISR and DMS systems.

Evaluation of these vulnerabilities helps define the security controls necessary for these systems. Further reproducing these results may help test secure protocols actively in development.

5. Conclusion

An attacker with sufficient knowledge of FLISR and DMS system functions could leverage them to cause additional harm to the distribution power system. Due to limited logging, visibility, and security controls within commercial OT devices in this space, indications of compromise would be difficult to find. Within this paper's findings, the most apparent indicator of compromise is a physical access alarm since no indicators exist in any of the log sources. Utilities can mitigate these risks with security controls, but ultimately, equipment vendors must implement data authentication features to address FDIA vulnerability. Security practitioners should emphasize the risks, support the industry's migration to more secure protocols, and adopt technologies that emphasize authentication and end-to-end encryption at the device level, not just within network security appliances.

Author Name, email@address

References

Jaatun, M. G., Moe, M. E. G., & Nordbø, P. E. (2018, June). Cyber security considerations for self-healing smart grid networks. In 2018 International Conference on Cyber Security and Protection of Digital Services (Cyber Security) (pp. 1-7). IEEE.

Leniston, D., Ryan, D., Power, C., Hayes, P., & Davy, S. (2022). Implementation of a software defined FLISR solution on an active distribution grid. *Open research Europe, 1*, pp. 142. https://doi.org/10.12688/openreseurope.14115.2

Khalid, M. (2024). Smart grids and renewable energy systems: Perspectives and grid integration challenges. *Energy Strategy Reviews, 51*, 101299.

Khan, R., Maynard, P., McLaughlin, K., Laverty, D., & Sezer, S. (2016, August). Threat analysis of blackenergy malware for synchrophasor based real-time control and monitoring in smart grid. In *4th International Symposium for ICS & SCADA Cyber Security Research 2016* (pp. 53–63). BCS.

Irfan, M., Sadighian, A., Tanveer, A., Al-Naimi, S. J., & Oligeri, G. (2023). False data injection attacks in smart grids: State of the art and way forward. arXiv preprint arXiv:2308.10268.

Chen, P. Y., Yang, S., McCann, J. A., Lin, J., & Yang, X. (2015). *Detection of false data injection attacks in smart-grid systems. IEEE Communications Magazine, 53(2)*, 206–213.

Basumallik, S. (2020). A taxonomy of data attacks in power systems. arXiv preprint arXiv:2002.11011.

Author Name, email@address

Peisert, S., Roberts, C., & Scaglione, A. (2020). Supporting Cyber Security of Power Distribution Systems by Detecting Differences Between.

Holguin, I., & Errapotu, S. M. (2023, October). Mitigating Common Cyber Vulnerabilities in DNP3 with Transport Layer Security. In 2023 North American Power Symposium (NAPS) (pp. 1-6). IEEE.

Rosborough, C., Gordon, C., & Waldron, B. (2019, March). All about eve: comparing DNP3 secure authentication with standard security technologies for SCADA communications. In 13th Australasian Information Security Conference (Vol. 161).

Irfan, M., Sadighian, A., Tanveer, A., Al-Naimi, S. J., & Oligeri, G. (2024). A survey on detection and localisation of false data injection attacks in smart grids. IET Cyber-Physical Systems: Theory & Applications.

Shafik, M. B., Elbarbary, Z. M., Siqi, B., Azmy, A. M., & Hussien, M. G. (2023). Distribution networks reliability assessment considering distributed automation system with penetration of DG units and SOP devices. Energy Reports, 9, 6199-6210.

Cybersecurity subcommittee (S0). IEEEPES Technical Committee on Power System Communications and Cybersecurity PSCC. (2023, March). https://site.ieee.org/pes-pscc/cybersecurity-subcommittee-s0/#1484883560308-6c12c146-3ae4

Author Name, email@address

Appendix A
Instructions for Duplicating the MitM Device

Hardware Shopping list:

1. Ethernet Cable
2. Intel N100 Mini PC device, minimum of two ethernet ports
3. Power Supply (if not provided with the Mini PC)

Instructions:

1. Remove the OS from the Mini PC and install Ubuntu (ubuntu.com/download)
2. Install Scapy (instructions at https://scapy.readthedocs.io/en/latest/installation.html)
3. Install Pip (https://pip.pypa.io/en/stable/installation/)
4. Install NetfilterQueue (https://pypi.org/project/NetfilterQueue/)
    a. This May require setting up a Python virtual environment (https://packaging.python.org/en/latest/guides/installing-using-pip-and-virtual-environments/)
5. Configure a network bridge between two ethernet ports on the Mini PC using Netplan.
6. Create an Iptables rule to filter traffic to an NFQueue. Refer to Figure 6 for an example.
7. Create a Scapy script to bind to the NFQueue from #6. See Appendix B for a sample code to start with.

Appendix B
Scapy Template

```
import os
from netfilterqueue import NetfilterQueue as nfq
from scapy.all import *

#INSERT AN IPTABLES FORWARD RULE THAT FILTERS SPECIFIC TRAFFIC
#EXAMPLE: iptablesRule="iptables -A FORWARD -m physdev --physdev-in enp1s0 -
p udp --sport 20000 -j NFQUEUE"
iptablesRule=

print("Iptable rule:")
print(iptablesRule)
os.system(iptablesRule)

def packet_listener(packet):
   scapy_packet = IP(packet.get_payload())

   ##INSERT PACKET MANIPULATION HERE

   ##CURRENTLY, THE SCRIPT WILL PRINT OUT PACKETS SEEN BY THE FILTER
AND THEN PASS THEM ALONG
   print(scapy_packet.show())
   packet.accept()


def main():

        queue = nfq()
        queue.bind(0, packet_listener)
        try:
                print("Running Queue")
                queue.run()
        except KeyboardInterrupt:
                print("Removing iptables rule")
                os.system('iptables -D FORWARD 1')




if __name__ == "__main__":
        main()
```

Author Name, email@address