Report on Real-Time System Attack, Testing IDS Algorithm

IDS Background : (Rule + ML) based

1. Attack Overview

The purpose of this report is to document the execution and impact of a simulated attack on a real-time system to test the effectiveness of an Intrusion Detection System (IDS) algorithm. The attack targets the GOOSE (Generic Object Oriented Substation Events) protocol in the IEC 61850 standard, which is widely used in electrical substations for protection and control. The objective of the attack is to manipulate GOOSE messages to inject false data and disrupt the normal operation of the system.

2. Attack Script Operation

The attack script performs the following operations:

- Captures live GOOSE traffic or processes a provided pcap file.
- Filters and selects a specific source-destination GOOSE packets based on user input.
- Modifies the selected GOOSE packets by altering critical fields such as `stNum`, `sqNum`, `bool`, `goose_length` and the payload data.
- Sends the modified GOOSE packets back into the network to simulate an attack.

3. Prerequisites

Before running the attack script, ensure the following prerequisites are met:

- Python 3.x
- Necessary Python libraries: `os`, `sys`, `datetime`, `inspect`, `struct`, `time`, `argparse`, `pyshark`, `numpy`, `random`, `pyasn1`, `scapy`

- Access to the target system or network with administrative privileges
- Network interface configured and operational (e.g., "eth1")

Prerequisites

> ⚠ First, ensure you have the attack script and the `goose` directory in the same directory. This setup is required for the script to locate and import the necessary modules from the `goose` directory.

4. Running the Script

To execute the script for the given `pcapfile`, use the following command in your terminal:

```
python3 script_name.py --pcapfile example.pcap --output outpu
t.pcap
```

Replace `script_name.py` with the actual name of your script, `example.pcap` with your input pcap file, and `output.pcap` with your desired output file name.

For live capture and attacking, use the following commands in your terminal:

```
touch output.pcapng
```

```
python3 script_name.py --livecapture --output output.pcapng
```

Replace `script_name.py` with the actual name of your script

5. Script Execution

The script execution involves the following steps:

1. **Initializing the attack**: The script starts by either reading a provided pcap file or capturing live network traffic.
2. **Filtering and selecting packets**: The script identifies GOOSE packets and filters them based on source and destination MAC addresses.
3. **Modifying packets**: The selected GOOSE packets are modified by changing critical fields and payload data.
4. **Sending malicious packets**: The modified packets are injected back into the network, simulating the attack.
5. **Saving modified packets**: Optionally, the modified packets can be saved to a new pcap file for further analysis.

6. Impact of this Attack:

During testing on the live testbed, the attack script successfully manipulated GOOSE protocol packets by altering critical parameters (`sqNum` and `stNum`, `bool`, `goose_length`). This manipulation aimed to disrupt data synchronization and control logic among IEDs.

Despite attempts to obfuscate these modifications, the Intrusion Detection System (IDS) deployed within the network environment detected the attack.

7. Conclusion

IDS algorithm Successfully Detected all the changes performed in the Attack.