Corrupted GOOSE Detectors: Anomaly Detection in Power Utility Real-Time Ethernet Communications

Maëlle Kabir-Querrec[1,2], Stéphane Mocanu[1], Pascal Bellemain[1], Jean-Marc Thiriet[1], Eric Savary[2]

[1] Univ. Grenoble Alpes, GIPSA-lab, F-38000 Grenoble, France
[2] Euro-System, F-38760 Varces, France

Abstract: GOOSE protocol is used for critical protection operations in the power grid, as standardized by IEC61850. It thus has strong real-time constraints that make very hard to implement any security means for integrity and confidentiality such as encryption or signature. Our answer to this lack of dedicated cybersecurity measures is to check legitimacy of every GOOSE messages flowing over the managed network. When detectors issue an alert, the SCADA informs field devices to discard GOOSE communication and run an alternative protection strategy. This article focuses on the GOOSE attack detectors we developed: one dedicated to Ethernet storm and the other one to fraudulent GOOSE frames. The paper first introduces main GOOSE protocol mechanisms and gives a brief state of the art regarding GOOSE attack management before presenting our architecture and the detectors.

Keywords: SCADA; Cybersecurity; GOOSE; IEC 61850; Ethernet storm; Spoof attack.

1. Introduction

When speaking of power grid protection, two goals usually compete with each other. Service availability is of the highest priority so in case of an electrical flaw there must be as few devices to disconnect as possible. In parallel there is a concern for system components: damages caused by short circuits grow more serious as the current is high and the fault persists a long time. Selectivity is the fact that the protection system will minimize the effect of an electrical fault on the power system while keeping the portion of the grid to shut down to a minimum.

In an IEC 61850 substation, selectivity uses point-to-point real-time communication between the protective relays (the IED – Intelligent Electronic Devices) to fulfill protection operations. This communication runs the GOOSE protocol (Generic Object Oriented System Event).

What if these GOOSE messages, critical to the grid protection, cannot be trusted anymore? The IEC 61850 standard introducing the GOOSE protocol does not propose any cybersecurity measures. Our approach is then to design an architecture monitoring corrupted GOOSE, either accidentally or maliciously, and let the system run in a “safe” mode regarding communications. This architecture described in [1] relies on two detectors responsible for verifying the state of the real-time Ethernet-based communication network. This article focuses on these two detectors while the purpose of [1] is to present the whole architecture and the GOOSE-independent strategy for completing the protection function in a “safe” mode.

The paper is organized as follows: in section 2 is a brief presentation of the IEC 61850 GOOSE communication mechanisms, then comes section 3 about the proposed detectors - bandwidth usage checker and GOOSE frame verifier, section 5 concludes the paper.

2. GOOSE – Generic Object Oriented System Event

2.1 GOOSE frame

GOOSE protocol is mapped on the Ethernet link layer. Messages are then sent as multicast frames following a publisher-subscriber procedure: devices on the network “see” all messages but read only the ones they are interested in, meaning the GOOSE messages they are subscribed to.

GOOSE frame structure is standardized (ISO/IEC8802-3 Ethertype 0x88b8) and given in the IEC 61850. It has been the subject of some papers such as [2]. Only the GOOSE APDU (Application Protocol Data Unit) structure is relevant regarding the present work. The APDU contains the data sent by the transmitting application. It is specified in ASN.1 (Abstract Syntax Notation One) as a set of 12 items:

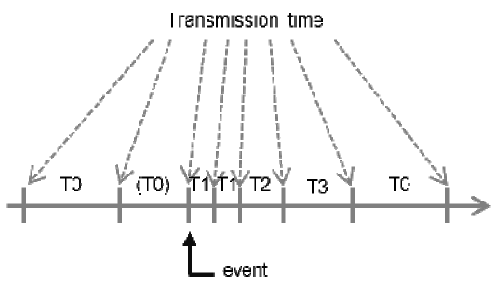
1. GoCBRef: GOOSE Control Block Reference is the name of the considered GOOSE control block, it defines transmission parameters.
2. TimeAllowedToLive: maximum time the subscriber waits before considering the connection lost.
3. DatSet: Data Set identifies the data to transfer.
4. GoID: GOOSE Identifier is the GOOSE name, unique in the whole system.
5. T: is the transmission time.
6. StNum: State Number is a counter incremented when a variable of the data set has changed and requires sending a GOOSE.
7. SqNum: Sequence Number is a counter incremented every time a GOOSE message is generated. Reset to 0 when StNum is incremented.
8. Test: boolean, true when in test phase.
9. ConfRev: Configuration Revision is the current configuration number.
10. NdsCom: Needs Commissioning indicates whether the GoCB needs an update. For instance it is true when the data set is empty.
11. NumDatSetEntries: Number of Data Set Entries.
12. AllData: the data set to transfer.

13. Security: a field that can be used for security purpose. Its use is not explained in the standard. Security measures are indeed recommended by the IEC 61850 standard but their implementation is up to the IED vendors.

Some of these fields can be exploited to check the message conformance with the system configuration.

2.2 Transmission mechanism

In a publisher-subscriber messaging procedure, there is no acknowledgment for received messages. To ensure reliability, GOOSE protocol has a specific transmission scheme as shown in figure 1. When an event occurs resulting in some change in one or more variables whose values are transmitted by the GOOSE message, a message is generated while StNum is incremented and SqNum is reset to 0. This GOOSE message is sent periodically at a high frequency T1 first and then at slower frequencies T2 and T3 until the frequency T0 for stable conditions.



T0	retransmission in stable conditions (no event for a long time)
(T0)	retransmission in stable conditions may be shortened by an event
T1	shortest retransmission time after an event
T2, T3	longer retransmission times until achieving stable conditions

Fig. 1. GOOSE transmission

2.3 Attacks

We consider two types of Ethernet attacks which are mentioned in the literature [3]. The first type of incident we consider is Ethernet storm. It was because of such an incident that an American nuclear plant was shut down in an emergency state [4]. The second attack is the publication of fraudulent GOOSE messages, mistakenly interpreted as valid by subscribers [5]. As shown in [5], an attacker just needs an access point to the network to launch a GOOSE message spoof attack. Attacks described in [6], [7] and [5] capture GOOSE messages flowing on the network, modify them before re-injecting them into the network to gain access and control of the IEDs. In the second paper, attack is automated using a script and code is available online to use as a basis for our experiments [8].

The main weakness of GOOSE communication as implemented nowadays is that there is absolutely no security mean: no encryption, no digital signature because today technologies still cannot perform such security computing while respecting the strong real-time requirements (4ms end-to-end transfer time).

3. Corrupted GOOSE detectors

3.1 Related works

The literature gives many interesting articles about anomaly and intrusion detection in SCADA communication systems [9, 10, 11]. Regarding IEC 61850 environments, there are only a few publications among which it is worth mentioning [12] and [7].

The IDS (Intrusion Detection System) proposed in [12] uses an open-source detection software, Snort, specifying rules enumerating badness. It does not deal with GOOSE communication though, but rules mostly check signatures of known attacks on ARP and ICMP traffic. A stand-alone implementation was chosen because authors consider that IEDs do not have computational capabilities to host an IDS.

The work proposed in [7] is more closely related to our approach. The specification-based IDS presented in [7] is dedicated to IEC 61850 real-time communications, meaning GOOSE and SV (Sampled Values, protocol used by field merging units to send process measures to IEDs). However learning phase from trustable collected data is required to train part of the rules while we want to avoid such a phase and generate our rules using only IEC 61850 standard specifications and configuration files. For their testbed, authors made the choice of a unique network while recommendation is to have physically separated networks dedicated to their own purpose, especially a network dedicated to horizontal real-time connections between IEDs. Moreover Hong et al. have not made their C code available.

The two cyber incidents we consider in the present work are fraudulent GOOSE messages and Ethernet storm. We found no mention of detection of Ethernet storm in IEC 61850 real-time environment in the literature.

3.2 Communication architecture

The proposed communication architecture is presented in figure 2. It is composed of three separated communication networks:

- A network for vertical flows between SCADA (Supervisory Control And Data Acquisition) and IEDs,
- A real-time network dedicated to GOOSE messages between IEDs,
- A Modbus/TCP network for the supervision to get reports and alarms from the detectors.

For this work, vertical communication between supervision and IEDs is supposed reliable. Indeed, this part of the architecture is out of the scope of this work which focuses on detecting fraudulent GOOSE communications.

The detectors send their analysis results to the SCADA over Modbus/TCP using two different mechanisms. Supervision periodically gets analysis results of the bandwidth checker through polling. To reduce propagation delay of an alert in case of a false GOOSE detection, the GOOSE frame verifier is associated to a Modbus/TCP client while the supervision is configured as a server.

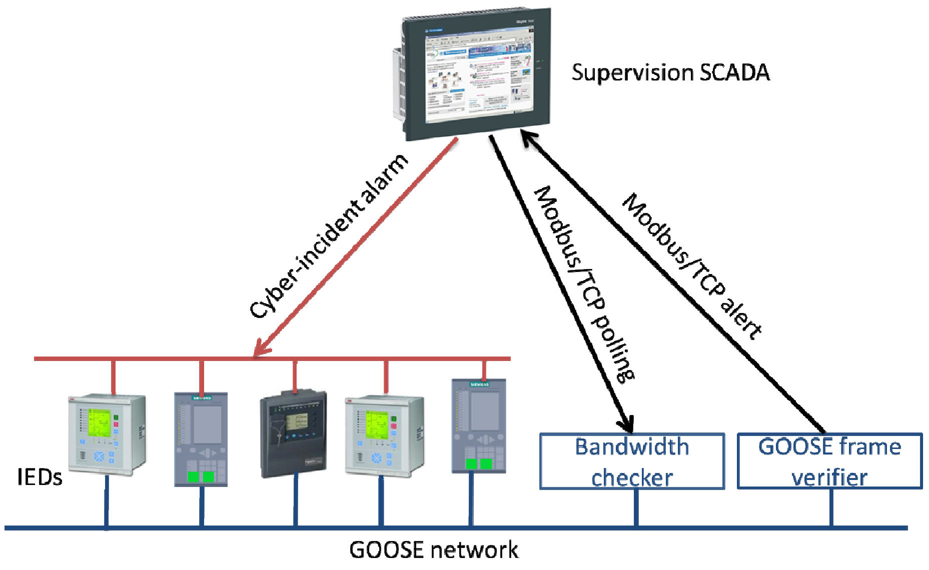


Fig. 2. Communication architecture

The supervision forwards alarms to the IEDs when a threat is identified. In normal mode, when an IED received a GOOSE frame, it waits for an alert from SCADA. If it has not come after detection time has expired then it takes into account the transferred information and launches the required actions. This is possible because electrical protection functions operate in 100ms to 1s while GOOSE end-to-end transfer time must be of less than 4ms.

When IEDs are informed of a cyber-threat they enter a safe mode: an alternative strategy takes over the real-time communication relying on specific programs and the communication with the SCADA. When alarm is deactivated, IEDs are back to normal mode. This alternative strategy is detailed in [1].

3.3 Bandwidth usage checker

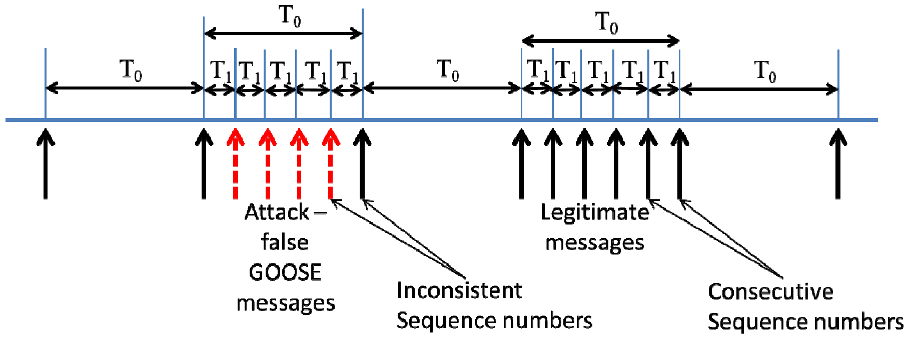
From a Linux bandwidth monitor, ifstat, we created a sensor to measure instantaneous and average bandwidth use; duration can be configured by the user. Here the algorithm of the ifstat-based Ethernet bandwidth checker:

Algorithm of the Ethernet storm detector

```
Start ifstat in Modbus server mode
Initialize Modbus server
Wait for client connections
While (ifstat runs)
    While (Client_Connection_Counter < Configured_Window)
        Mean_Bandwidth += Number_of_IN_Frames_Since_Last_Connection /
        Configured_Window
    Reset Client_Connection_Counter
```

3.4 GOOSE frame verifier

Detecting false GOOSE messages is more challenging. As presented in [5], an attacker sends a quick sequence of GOOSE messages (with a false state change on the publisher side) which have correct sequence and state numbers and timestamps regarding the previous legitimate frames. Figure 3 is a timeline of such an attack.



T_0
T_0
T_0
T_1 T_1 T_1 T_1 T_1
T_0
T_1 T_1 T_1 T_1 T_1
T_0
Attack – false GOOSE messages
Legitimate messages
Inconsistent Sequence numbers
Consecutive Sequence numbers

Fig. 3. Legitimate (full line) and false (dotted line) events in GOOSE sequence

A GOOSE attack can be detected by comparison of the sequence and state numbers from two consecutive messages. Even a perfect attack is detected after T0 at the latest, when the first true GOOSE message highlights incoherent counters or clock.

Our GOOSE frame verifier is based on tcpdump, an open-source packet analyzer. From all the frames captured by libpcap functions, only GOOSE messages are kept (Ethertype 0x88b8). For every GOOSE, the analyzer checks:

- that the intercepted GOOSE is defined in the system configuration;
- that SqNum and StNum counters are relevant compared to previous messages of same GOOSE Identifier (see 2.1);
- that timestamp is coherent regarding the previous messages of same GoID and the counters (SqNum, StNum).

Algorithm of the GOOSE frame verifier

```
Start tcpdump in Modbus server mode
Initialize Modbus server
While (tcpdump runs)
        Get captured GOOSE message
        Get RxTime
        Get GOOSE PDU fields and store them
        Check Source_Address
        Check GoID
        Check StNum and SqNum compared to previous same-GoID GOOSE
        Check RxTime compared to previous same-GoID GOOSE
```

4. Conclusion

In an IEC 61850 power grid, critical protection and control functions rely on real-time GOOSE messages. Today gears are not able to support any security measures for real-time communication and GOOSE protocol may be considered vulnerable to cyber incidents and attacks. To leverage the trust in the automation system, we developed two detectors in charge of checking the Ethernet bandwidth keeps an acceptable value and of verifying GOOSE messages not corrupted. In this paper we presented these detectors. The whole architecture we proposed has not been detailed here but in [1].

Experiments are ongoing work. We expect to be able to add some results to the final version of this article.

References

1. Kabir-Querrec M., Mocanu S., Bellemain P., Thiriet J.-M., Savary E. (2015), Architecture des systèmes d'automatisation des postes résiliente aux attaques des trames GOOSE, submitted to C&ESAR 2015
2. Kriger C., Behardien S., Retonda-Modiya J. (2013), A Detailed Analysis of the GOOSE Message Structure in an IEC 61850 Standard-Based Substation Automation System, INT J COMPUT COMMUN 8(5):708-721
3. Talha M., Salman Y., Yunus Y., Roslan I. (2014), A Review of Security Attacks on IEC61850 Substation Automation System Network, International Conference on Information Technology and Multimedia
4. NRC, Effects of Ethernet-based, Non-safety Related Controls on the Safe and Continued Operation of Nuclear Power Stations, NRC Information notice 2007-15
5. Hoyos J., Dehus M., Brown T. (2012), Exploiting the GOOSE protocol: A practical attack on cyber-infrastructure, IEEE Globecom Workshops

6. Kush,N., Ahmed, E., Branagan,M. and Foo, E. (2014), Poisoned GOOSE: Exploiting the GOOSE Protocol, In Proc. Twelfth Australasian Information Security Conference (AISC 2014)
7. Hong J., Liu, Chen C., Govindarasu M. (2014), Detection of cyber intrusions using network-based multicast messages for substation automation, IEEE PES Innovative Smart Grid Technologies Conference
8. mdehus / goose-IEC61850-scapy, `https://github.com/mdehus/goose-IEC61850-scapy`
9. Cheung S., Dutertre B., Fong M., Lindqvist U., Skinner K., Valdes A. (2007), Using model-based intrusion detection for SCADA networks, Intern. Scientific Symp. on SCADA Security
10. Barbosa R., Sadre R., Pras A. (2012), Towards periodicity based anomaly detection in SCADA networks, IEEE International Conference on Emerging Technologies and Factory Automation
11. Do V., Fillatre L., Nikiforov; I (2014), A simple algorithm for detecting cyber / physical attacks on SCADA systems, IEEE Conference on Control Applications
12. Premaratne U., Samarabandu J., Sidhu T., Beresh R., Tan J.-C. (2010), An intrusion detection system for IEC 61850 automated substations, IEEE Transactions on Power Delivery 25: 2376–2383