Indirect Synchronisation Vulnerabilities in the IEC 60870-5-104 Standard

Alessio Baiocco
Department of Information Security and
Communication Technology
Norwegian University of Science and Technology
N-2815 Gjøvik, Norway
Email:alessio.bcc@gmail.com
Email: alessio.baiocco@ntnu.no

Stephen D. Wolthusen
Department of Information Security and Communication Technology
Norwegian University of Science and Technology
N-2815 Gjøvik, Norway
and
School of Mathematics and Information Security
Royal Holloway, University of London
Egham, Surrey TW20 0EX, UK
Email: stephen.wolthusen@rhul.ac.uk

***Abstract*—Control systems rely on correct causal ordering and typically also on exact times and time relationships between events. For non-trivial systems, this implies synchronisation between distributed components, potentially from sensors and actuators to SCADA hierarchies. Whilst this can be accomplished by point-to-point synchronisation against a common reference such as GNSS (global navigation satellite) signals, common practice and codification in the ISO/IEC 60870-5-104 protocol widely used in the power control domain calls for the Network Time Protocol (NTP).**

In this paper we therefore describe attack patterns allowing the undetected partial re-play of legitimate messages and injection of messages even in the presence of ISO/IEC 62351 protective measures in a multi-staged attack targeting time synchronisation protocols and specifically the NTP protocol, and resulting in a de-synchronisation between a PLC/RTU and higher-level SCADA components. We demonstrate the feasibility of such attacks in a co-emulation environment.

I. Introduction

The flexibility and cost advantages of structured networks based on IEEE 802.3 Ethernet and the TCP/IP protocol suite have driven the developments in process control systems away from dedicated field bus systems for well over a decade. This, however, has also led to the notion of air gaps as a security provision [4] long becoming obsolete except in highly specialised domains [12]. Instead, process control equipment vendors and standardisation bodies have embraced control systems with higher degrees of integration also with wide area networks.

At the same time as increasing the reachability of process control networks, however, the reliance on IETF standards for the communication infrastructure has also opened process control protocols such as the ISO/IEC 60870-5-104 standard widely used in the power sector to direct and indirect attacks not only against the protocols or the endpoints, but also the communication protocols. Moreover, as this infrastructure is more widely shared and understood, it can be safely assumed that potential adversaries face fewer obstacles compared to attacks against more specialised process control networks.

Beyond attacks targeting endpoint semantics or implementation vulnerabilities, distributed control systems by definition at least partially expose their communication channels, creating an attack surface that the ISO/IEC 60870-5 family of standards did not foresee explicitly, and where protection primitives defined in the ISO/IEC 62351 standard are not offering full protection. One such property that is implicitly guaranteed in point-to-point links is ordering and hence causality of messages, which has to be reproduced through multiple protocols in an IEEE 802.3 environment and ultimately by transport layer protocols.

A sophisticated attacker may therefore perform *indirect* attacks against the communication links of a SCADA system, thereby not only achieving the main objective but also allowing to reduce the effectiveness of detection and mitigation measures such as intrusion detection sensors and logging, which *similarly depend on sequential patterns and causal orderings as well as anomalies*. This may allow adversaries to design attack patterns that are not only effective but also largely undetectable [8], particularly where attacks on a distributed or hierarchical control system are themselves co-ordinated and distributed [3].

We argue that the security guarantees currently required particularly from time synchronisation protocols referenced in the ISO/IEC 60870-5 standard are insufficient, and that mitigation and recovery mechanisms are required. These issues are, moreover, not fully addressed by the ISO/IEC 62351 standard. The contribution of this paper is therefore to demonstrate *conditions under which conforming use of manipulated time synchronisation may result in desynchronisation up to command rejection by RTU/PLC.*

A. Paper Structure

The remainder of this paper is structured as follows: Section II reviews the state of the art for direct and indirect attacks against the ISO/IEC 60870-5-104 protocol and its dependencies; we then describe a reference framework for complex attacks in section III and offer simulation and experimental results for a thus derived scenario. Section IV offers an analysis and mitigation mechanisms for the attack and its effect on detection mechanisms followed by our conclusions and outlook on future and ongoing work in section V.

II. Related Work

Both the network time synchronisation protocol (NTP, RFC 5905) protocol and the ISO/IEC 60870-5-104 protocols have been studied for vulnerabilities and attacks by a number of authors; however, whilst the ISO/IEC 60870-5-104 standard makes explicit reference to the NTP protocol for time synchronisation, the interplay between the two protocols has not been studied in depth.

We assume the reader to be familiar with both protocols and only illustrate selected aspects of both protocols and their structure below in order to describe the attack framework linking the two protocols.

NTP is an IETF standard protocol first proposed in 1988 and now in version 4 for time synchronisation in geographically distributed systems and is used universally for Internet-connected systems which do not have direct access to a precision time reference. For scalability, it is formed into a hierarchical structure of *strata* in which an entity with access to a higher precision time reference (which may be a legal time reference or higher stratum) will provide a time reference to lower strata. This implies that the accuracy and precision of time signals will degrade for lower strata as errors accumulate.

NTP has been extended multiple times to provide security functionality and supports both symmetric and asymmetric cryptographic primitives for authentication. However, support for these mechanisms is highly variable as implementations do not consistency provide these extensions, or operators of NTP servers do not offer the facility. Moreover, the symmetric authentication primitive relies on the long-obsolete MD5 cryptographic keyed hash function, and relies on manual configuration of the symmetric key for each server, casting doubt on scalability and effectiveness of the mechanism. The asymmetric authentication, on the other hand, is based on the Autokey extension introduced in NTPv4, and is not universally supported in part because of performance concerns.

The security of NTP has been the subject of scrutiny since its earliest implementations or at least version 2, when Bishop described in detail a set of attacks that can be perpetrated against the NTP protocol in said version, namely attacks allowing to *masquerade* , *delay*, perform *denial-of-service*, *modification* and *replay* operations. Bishop also proposed a number of mitigation mechanisms that have in part been incorporated into later versions of NTP [2], however.

In recent work, Malhotra [5] explored the attacks that an adversary can perform against NTP without the need to authenticate. In particular, the author reports and analyzes various attack methods both *on-path* (i.e. where an attacker has obtained privileges or may intercept communication) or *off-path*, where the attacker may interfere without any privileges or stolen credentials. We concur with the author in observing that despite a number of attacks having been documented, there appears to be still a basic assumption of trustworthiness in the NTP ecosystem and the time signals transmitted.

In particular, Malhotra focuses on the security mechanisms of the NTP protocol: Older, but still partially supported versions (versions 2 and 3) relied on the long-obsolete DES block cipher primitive, whereas version 4 stipulates the use of TLS, typically implemented in the form of the OpenSSL open source library. NTP version 4 also introduced a number of other reliability and security enhancements, most notably the known attack pattern of the so-called *Kiss-of-Death* as exploited as a source of attacks against NTP servers, which is an example of an *off-path denial of service attack* noted above.

Moreover, Malhotra [5] also described possible consequences of compromising the NTP ecosystem; this is inter alia a prerequisite for the correct functioning of a number of security primitives and infrastructure elements that rely on accurate time stamps and the ordering of messages and events according to said time stamps; the same is also true for application level and communication protocols [5]. Examples of such attacks include manipulation of the TLS authentication where both certificate validity and freshness may be affected, and similar patterns also for the DNSSEC mechanism used to authenticate domain name service (DNS) resource records as well as indirectly border gateway protocol (BGP) prefix announcement authorisations as well as attack types such as the *cache-flushing-attack* perpetrated against caches, the *interdomain-routing-attack* resulting from misusing RPKI security mechanism of the BGP protocol and to authentication protocols whose execution requires timestamps. Whilst on-path attacks may be more difficult to achieve and can be detected more easily, this is not a necessary assumption for a number of attacks where off-path access is sufficient.

As with the original NTP, SCADA communication protocols had originally been conceived and developed without protection mechanisms as authentication was implicit in point-to-point serial links such as the ISO/IEC 60870-5-101 variant, further supported by the principle of *safety through isolation* and use of segregated, air-gapped field buses. The progressive migration from a monolithic structure to a more open and standardised system based on IETF standards, however, has exposed the same SCADA systems to an increasing number of threats.

While the security threats to SCADA are widely recognised, no systematic modeling and analytical technique exists for the evaluation of critical assets in the critical infrastructures and their respective dependencies internally on communication infrastructures noted here. This is also the case for power systems at this point [10].

Ten [10] documents a detailed analysis from the point of view of the IT infrastructures that make up the control system of a (power) cyber system stating that the main defense mechanism consists of firewalls and that the most vulnerable network segments are the substation networks as, unlike these control center networks are better isolated and with limited monitoring facilities. The greatest vulnerabilities are assessed to arise from the geographical distribution of HMI workstations and the reliance on remote connections for operation, maintenance, and operator access. Moreover, maintenance and policy violations by operators or maintenance staff may result in temporary breaches of network segregation with the consequence of malicious network or file-based attacks transitioning to process control networks.

More specifically, common attack vectors as identified in [12] are:

- Backdoors and breaches of network perimeter
- Vunerabilities in common protocols (e.g. NTP and IEC 60870-5-104)
- Direct cyber attacks on field devices
- Database (e.g. historian) attacks
- Communication hijacking and *man-in-the-middle* attacks
- Cinderella attacks on time provision and synchronisation (i.e. the NTP or precision time (PTP) protocols)

while attacks to the control side, as reported by Zhu [12], are focused on malicious data introduced by faulty or manipulated sensors, manipulated and misleading output data to the actuators and reactors, or compromised network links, controller historians, and *denial-of-service* attacks.

Attacks against SCADA systems can target arbitrary levels of protocol stacks [12], but thus far limited attacks against the IEC 60870-5-104 protocol have been reported in the literature, mainly a class of *man-in-the-middle* attacks [8] described in depth by McLaughlin in [7]. As noted, the baseline ISO/IEC 60870-5-104 protocol is vulnerable to this attack because of the lack of any support of packets authentication and verification mechanism.

The attack to ISO/IEC 60870-5-104 described by [7] is classifiable as a multi-step attack, that is, capturing packets, manipulating these, and ultimately and injecting malicious TCP packets into the network under attack. Furthermore, depending on the network topology, it is necessary to manipulate the ARP tables of the routers in order to redirect the network traffic in malicious routers in order to capture the data traffic that flows in the same network to be attacked (ARP spoofing attacks).

III. Attack Framework and Experiments

A. Assumptions and Hypothetical Scenarios

The initial assumptions we made in for testing the our attack methodology against the IEC 60870-5-104 standard are summarized in the following points:

1) the system under attack is geographically distant from the central control infrastructures.
2) The substation communication network attacked has perimeter control systems, the firewall, that processes all the incoming and outgoing traffic of the substation network itself.
3) Network separation, physical or virtual, might not to be implemented: control network data flow and operative data flow run through the same network.
4) Network intrusion detection (N-IDS), or any system behavior analyzer, are used in some points of the entire SCADA infrastructure and not in any substation. The positioning strategies of such intrusion detector placement are not research argument of this paper.
5) NTP timing signal comes from an external timing server to the substation network and the routers represent the stratum higher than RTU and other local timing devices connect.
6) The substation network topology is already known to attacker.
7) Any hacker has penetrated the substation network by gaining a privileged position (superuser privileges) of a unit used for external control or that has been able to connect to the same substation network.
8) The data flow sent through the substation network is not encrypted.
9) The attacker is able to learn how the RTU unit query the designed NTP servers.
10) The attacks are performed against the RTU and not to the NTP servers.

B. Attack Construction

We are now introducing a compounded and multi-stage attack which is structured in five different stages whom are enlightened as follow:

1) Network stealth breaching
2) Network data traffic acquisition
3) System behaviour analysis
4) packet crafting
5) packet injection

We defined our proposes attack as multi-stage one because of the assumptions we made and reported in section III-A on which we assumed that an attacker already knowns the substation network topology. How the network topology has been discovered or investigated is not dealt in detail on this paper. We consider that the attacker physically introduced itself in the unmanned power plant facility and got physically connected to any of the switch/hub in place. The attacker, in a totally different scenario, might have gained a remote access to the substation network from outer connection: this scenario however requires to be carefully evaluated because, according the assumptions reported III-A the attacker have to be able to elude both the N-IDS and any firewall which interfaces the substation internal network with the WAN connection. Also, the N-IDS might not be placed in every segment of the entire SCADA network providing some unsecured blind spots to the attacker for stealthy performing any malicious operation against the SCADA infrastructure. However, due to space limitation wee will not go into details about how an attacker can stealthy introduce itself in the substation network.

After having gained access to the substation network (attack stage nr. 1) the attacker performs a *silent man-in-the-middle-attack* using a traffic sniffer software (e.g. Wireshark) in order to capture the network data traffic. The attacker can performs an *educated* data traffic sniffing procedure by selecting the protocol packets for the IEC104 (TCP packets) and the NTP ones (UDP). We defined the man-in-the-middle-attack *silent* because we want achieve un undetectable attack to SCADA de-sync it or to disconnect it from the main timing signal source.

The use of any ARP spoofing technique performed to the any specific router for diverting the data traffic to a malicious router or, according to the networking devices used, exploiting the vulnerabilities os some proprietaries self-scanning protocols used by some manufacturers (e.g. Cisco CDP protocol) has to be carefully evaluated because these type of attacks or fingerprint operations can be detected without a big effort by any local N-IDS agent. Hence, the network topology knowledge becomes crucial for stealthy performing any hostile action against the RTU device.

With the *data traffic acquisition* stage we have collected sensitive data which allows the attacker to study the NTP-RTU packet exchange behaviour and then to perform the *packet crafting* stage. The attacker needs to select the NTP packets carried by the UDP packets which are exchanged between the RTU and the NTP servers configured as its timing signal sources. In particular, we target the RTU's NTP inner client to adjust its clock time using a fake new time introduced by the malicious NTP packet tricking the NTP *peer-poll* and*system* processes (to see figure 1).

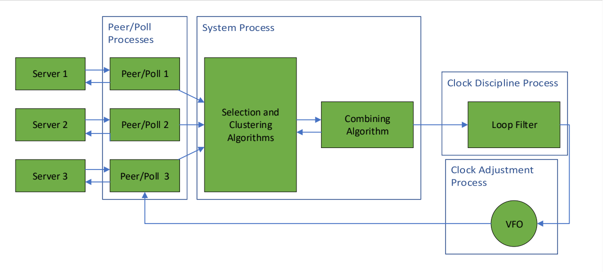

Fig. 1. NTP Process Overview

To craft NTP packets we use Ostinato [9], a powerful packet crafter software that allows to recreate whole TCP and UDP packets also providing the possibility to add customised payload.

During the packet craft process we replaced the UDP packet fields *IPID*, *UDP Checksum*, we have adjusted the UDP packet length field *Length* in order to keep consistent the whole packet length with respect to the new adds performed. We also modified the some of the NTP packet fields such as *Mode* field and part of the NTP payload fields, *Reference Timestamp*, *Origin Timestamp* and *Transmit Timestamp* so as to make the malicious UDP packet seem like genuine.

The crafted malicious NTP packet is then injected in the substation

ON the packet injection stage the attacker inject in the substation network the malicious NTP packets. there is no mechanism for authentication and verification of NTP packets - besides the NTP protocol uses UDP with transport protocol - the RTU client updates its time status with what has been received and will continue to operate as if the responses received to its ping messages were authentic and not manipulated. This will lead to the de-synchronization between RTU and the rest of the SCADA systems, including any connected HMI , with serious consequences including the RTU's refusal to execute commands sent by the operator because the timestamps violate the timing restrictions on timing, their own operating restrictions to avoid replay attacks.

C. Simulation Scenario

In order to test and our propose attack we used our research framework CHAOS [1] which contains SCADA IEC 60870-5-104 clients and one IEC 60870-5-104 server, the NTP servers and a storage server as showed in figure 2:

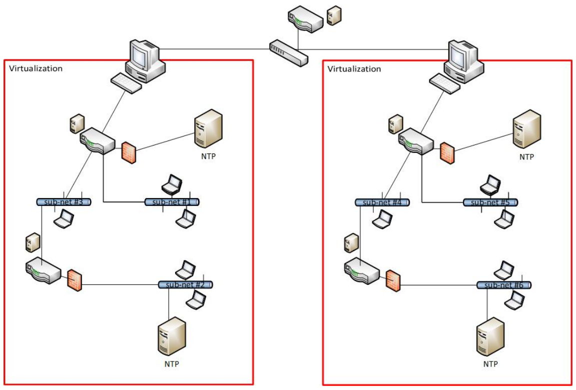

Fig. 2. CHAOS framework general scheme

Three bare metal PC are used as hosts using linux Ubuntu 16.04 LTS and Virtualbox is the chosen hypervisor. The networking service is performed by PFsense, an open source BSD based firewall-router software which is hosted by some virtual machine spreader inside the whole framework. A PFsense bare metal installation PC works as both peripheral firewall and central router for interconnecting the whole host PCs. However, the PFsense firewall instances of all the virtualised routers inside the CHAOS framework sub-networks are deactivated. Only the firewall instance of PFsense unit that interfaces the entire framework network with the WAN is kept active.

Other virtualized PFsense based routers operate in the network providing network stratification and further data traffic security by implementing PFsense routers at each sub-network branch. CHAOS framework can both works as a standalone platform and even connected to a real SCADA RTU simply by reconfiguring the border router: when the framework works in standalone mode the main peripheral router/firewall isolates all sub-networks blocking the IN/OUT communications, especially the NTP UDP port 123 (proper of NTP service). We choice to run the framework in the latter modality, using a real RTU unit for our experiments, an ABB RTU560 of the 500series and using our own time source provided by one of the PC. Even if using the local PC time as time source, we wanted our system isolated from the real stratus 1 (e.g. google time servers) in order to have full control over the timing infrastructure.

The whole framework NTP infrastructure is represented in figure 3 where a physical PC provides the time to the other levels (stratum 2 and stratum 3) up to stratus 4 which represents the NTP client of the IEC104 client/server units. The NTP infrastructure (figure 2) also has some peer-to-peer link in order to guarantee a certain degree of redundancy. Also, in order guarantee the max flexibility we used Ubuntu server OS based virtual machine as NTP client/server for both the stratum 2 and stratum 3. Also PFsense offers an NTP client/server daemon which we used for providing time reference to the sub-networks we connected to.

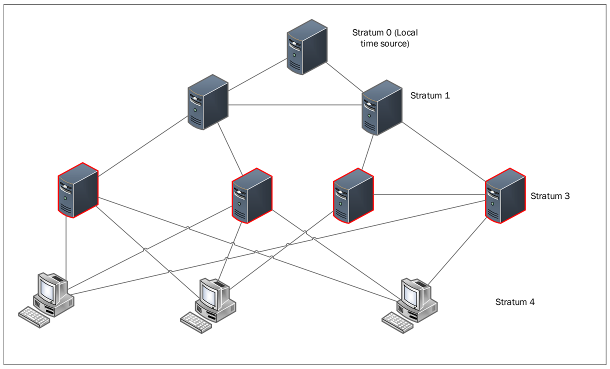

Fig. 3. CHAOS framework NTP hierarchical scheme. In red the NTP stratum servers spoofed.

We decided to tamper - or properly said to make the attack figure as launched from the NTP stratum which the RTU and the other SCADA devices of the attacked substation are connected at (figure 3 server with red lines). This is due to the fact that we don not need to see how the NTP hierarchy poisoning effects propagates through the different stratum as we specified by assumption number 5 in subsection III-A).

IV. Analysis

In section III-A we have submitted an indirect attack to the SCADA system who relies on IEC 60870-5-104 as communication procol by spoofing the NTP protocol, ie the timing protocol external to the IEC TC57 standards as the native sub-protocol of IEC104 (the IEC 60870-5-5) has been declared deprecated after the introduction of ethernet networks. The NTP protocol does not provide any mechanism to control the packets exchanged between the various units for which we are not able to verify the authenticity and the integrability of the exchanged messages.

Also the IEC 60870-5-104 communication protocol does not have any authentication mechanism and, about timing status it only has a primitive time sync mechanism, a flag called *Invalid Time Flag* (ITF) send with the ADSU by the TCP packet that tells us when the timestamp is different from that of the destination - ie the RTU is not synchronized with any NTP server and it consequently uses its internal time as a reference time - it is not activated when the reference signal time is degraded and therefore both RTU and NTP to which the latter is connected are out of sync with the rest of the network: this means that just compromise the NTP server to which a RTU is connected to ensure that even the only system control of the time synchronism foreseen by the IEC 60870-5-104 is bypassed without being able to recognize the actual timing status.

The fact that IEC 104 communications provide for PIDs that largely rely on timestamps requires an accurate and secure timing synchronization mechanism and NTP in terms of security is not to be considered robust. The same can be said of IEC104 and other communication protocols of the IEC TC57 group (excluding the new versions of IEC 61850 and the IEC 62351 protocol series) which has not been natively implemented (except for the application of IEC 62351 standards) no control over integrity and authenticity of data. The consequences of an attack on NTP have repercussions on IEC 104 in a more or less serious way depending on the extent of the attack and the out of service status: as known and described in [1] timestamp re-branding procedure can lead to potentially disruptive confusion in the analysis time of the SCADA commands received and sent.

The time control mechanism adopted by IEC 104 provides that a command sent by an operator is performed within a predetermined time interval (generally 10-30 seconds) based on the type of PID sent. This security mechanism has been designed to prevent old commands from being executed in the event of RTU desync. By extension, this mechanism puts the system away from reply attacks.

However, this security mechanism can be used (which we introduced in III-B) as a tool to perpetrate attacks that exploit the desynchronization between SCADA drives and lead to targeted denial of services. The attack we introduced in **??** section aims to desynchronize RTU and the tampered NTP server by the rest of the system: doing so the RTU gets the sync signal as correct even if it is not the *invalid time flag* is not activated. The latter is displayed by the operator in the HMI, causing it to check the status of the synchronism between NTP and RTU. When the validity window of execution of the commands is exceeded, the security mechanism that blocks the execution of IEC 104 commands sent to the RTU are rejected because they are considered obsolete: the operator can continue to view the status of the RTU and the other SCADA components but will not be able to implement any direct command to the RTU. The command time control mechanism of the RTU has amplified the effect of our proposed attack (section III-B) which evolved to something classifiable as a *denial-of-service* attack because, even if the RTU is still capable to answer to query legitimate query requests sent by the operator through the HMI interface, we are not able to make the RTU executing any control commands. This situation of DoS can continue as long as the condition of desync is detected, at least not through the IVT flag.

It is evident that the system behavior knowledge and the a deep knowledge of the protocols involved in the attacks are required in order to guarantee any attack successful outcome.

A. Mitigating Effects Attacks

Hereafter some solutions which helps to mitigate the attack effects:

- **Cryptographyc Authentication:** all the NTP clients and servers should adopt the cryptographic authentication mechanism. It is well known about the vulnerabilities of the NTP symmetric authentication mechanism and all the issues led by the asymmetric authecation mechanism adopted by the NTP but, even these countermeasure can be strength the whole control system security and led the attacker to perform any other fingerprint attack or spoofing to the control system NTP timing infrastructure.
- **Networking Separation/Segmentation and segregation:** A local power control subnetwork should be *segmented* in many different smaller subnetworks on which N-IDS and firewalls are alway operative. Also, the possibility to jump from a network segment to the other ones without any restriction and without any access control policy must be avoided.
- **Local IDS deployment:** the implementation of local N-IDS is hardly recommended in order to prevent any fingerprint attack to the local control system infrastructure and for detecting replay and spoofing attack. When N-IDSs are deployed in a network the topology knowledge becomes crucial in order to stealthy perform the attack (section III-B) to the SCADA infrastructure.

V. Conclusions

A new attack scenario to IEC 60870-104 standards has been introduced: we have demonstrated that the IEC 60870-5-104 can be indirectly attacked by using one of his auxiliary service (NTP time synchronization protocol). In addition, some restrictions adopted to limit other types of attack (eg replay attack), to avoid confusion or potentially dangerous situations, such as the imposition of commands now obsolete, can be exploited to amplify the effects of another attack specifically designed for exploit these vulnerabilities.

References

[1] Author 1, A. . Causality re-ordering attacks on the iec 60870-5-104 protocol.

[2] Bishop, M. A security analysis of the ntp protocol version 2. In *Computer Security Applications Conference, 1990., Proceedings of the Sixth Annual* (1990), IEEE, pp. 20–29.

[3] Gregg, M. *Certified Ethical Hacker Exam Prep (Exam Prep 2 (Que Publishing))*. Que Corp., 2006.

[4] Krutz, R. L. *Securing SCADA systems*. John Wiley & Sons, 2005.

[5] Malhotra, A., Cohen, I. E., Brakke, E., and Goldberg, S. Attacking the network time protocol. In *NDSS* (2016).

[6] Maynard, P., McLaughlin, K., and Haberler, B. Towards understanding man-in-the-middle attacks on iec 60870-5-104 scada networks. In *ICS-CSR* (2014).

[7] McClure, S., Scambray, J., Kurtz, G., and Kurtz. Hacking exposed: network security secrets and solutions.

[8] Pidikiti, D. S., Kalluri, R., Kumar, R. S., and Bindhumadhava, B. Scada communication protocols: vulnerabilities, attacks and possible mitigations. *CSI transactions on ICT 1*, 2 (2013), 135–141.

[9] Srivats, P. Ostinato, network traffic generator and analyzer.

[10] Ten, C.-W., Liu, C.-C., and Manimaran, G. Vulnerability assessment of cybersecurity for scada systems. *IEEE Transactions on Power Systems 23*, 4 (2008), 1836–1846.

[11] Ullmann, M., and Vögeler, M. Delay attacksimplication on ntp and ptp time synchronization. In *Precision Clock Synchronization for Measurement, Control and Communication, 2009. ISPCS 2009. International Symposium on* (2009), IEEE, pp. 1–6.

[12] Zhu, B., Joseph, A., and Sastry, S. A taxonomy of cyber attacks on scada systems. In *Internet of things (iThings/CPSCom), 2011 international conference on and 4th international conference on cyber, physical and social computing* (2011), IEEE, pp. 380–388.