Journal of Network and Computer Applications ■ (■■■■) ■■■–■■■

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Formal modelling and analysis of DNP3 secure authentication

Raphael Amoah, Seyit Camtepe, Ernest Foo

Queensland University of Technology, Brisbane, 4000 QLD, Australia

ARTICLE INFO

Keywords:
Smart grid
SCADA
DNP3
DNP3-SA
Formal methods
CPN

ABSTRACT

Supervisory Control and Data Acquisition (SCADA) systems are one of the key foundations of smart grids. The Distributed Network Protocol version 3 (DNP3) is a standard SCADA protocol designed to facilitate communications in substations and smart grid nodes. The protocol is embedded with a security mechanism called *Secure Authentication* (DNP3-SA). This mechanism ensures that end-to-end communication security is provided in substations. This paper presents a formal model for the behavioural analysis of DNP3-SA using Coloured Petri Nets (CPN). Our DNP3-SA CPN model is capable of testing and verifying various attack scenarios: modification, replay and spoofing, combined complex attack and mitigation strategies. Using the model has revealed a previously unidentified flaw in the DNP3-SA protocol that can be exploited by an attacker that has access to the network interconnecting DNP3 devices. An attacker can launch a successful attack on an outstation without possessing the pre-shared keys by replaying a previously authenticated command with arbitrary parameters. We propose an update to the DNP3-SA protocol that removes the flaw and prevents such attacks. The update is validated and verified using our CPN model proving the effectiveness of the model and importance of the formal protocol analysis.

1. Introduction

Supervisory Control and Data Acquisition (SCADA) systems are one of the key foundations of smart grids. Recent literature (Ancillotti et al., 2013; Lu et al., 2013; Yan et al., 2013; Gungor et al., 2013) shows that evolving smart grids are revolutionising the energy industry and enabling the electricity network to be more reliable and economical. A disruption, either minor or major, deliberately or mistakenly caused to these infrastructures can lead to damaging highly sophisticated devices, inflicting substantial economic loses and posing as life-threatening situations. From the security perspective, legacy SCADA systems have long-lived under obscurity; as a result, they have proven to be insecure to recent cyber attacks[1] (Nicholson et al., 2012; Miller and Rowe, 2014). For instance, had it not been for the stuxnet attack discovered in 2010 (Langner, 2011), which created awareness, SCADA security would have still lived in obscurity. Disrupting functionality in critical infrastructures is a very important issue to consider. Unfortunately, this situation has now become the target area for many malicious attackers. For example, in 2013 intruders managed to shut down a key tunnel road (Carmel Tunnels, in Haifa, Israel) for eight hours causing massive congestion.[2]

The Distributed Network Protocol version 3 (DNP3) (IEEE, 2012) is one of the standard SCADA protocols used to facilitate communications in smart grid automation. The protocol is designed such that it can allow smart grid nodes to collect, process, store and control data from DNP3-enabled IEDs (Intelligent Electronic Devices).[3] DNP3 provides a security mechanism called *Secure Authentication* (DNP3-SA), which is used to secure end-to-end communication in substations (Gilchrist, 2008).

Integrating security controls in SCADA protocols, such as authentication and encryption, are very important issues to consider in critical infrastructures since functionality, performance and behavioural correctness are crucial. This is to ensure that embedded security mechanisms fit well and do not contain errors that may weaken the security protection provided. The current state of the DNP3-SA protocol is informally described in its specifications. Informal approaches have been known to be very useful in designing

[1] Wilhoit K. The SCADA that didn't cry wolf. Technical report. Online: http://www.trendmicro.com.au/cloud-content/us/pdfs/security-intelligence/white-papers/wp-the-scada-that-didnt-cry-wolf.pdf; 2013 [accessed: 1/09/14].

[2] Peter S. Hackers target Israel's key tunnel road, cause its shut-down and severe damage. Online: http://au.ibtimes.com/articles/517710/20131029/israel-cyber-security-collective-hackers-attack-carmel.htm; 2013 [accessed: 10/01/2014].

[3] Ambient. The smart grid node in a distributed intelligence grid architecture. Online: http://cdn2.hubspot.net/hub/165743/file-56770063-pdf/docs/ambient-distributed-intelligence-white-paper.pdf, Newton, MA: Ambient Corporation; 2013 [accessed: 14/06/2014].

http://dx.doi.org/10.1016/j.jnca.2015.05.015
1084-8045/

systems. However, they have also yielded inadequate methods which have led to ambiguities and incompleteness (Bolignano et al., 2001; Hall, 2007). Incompleteness among systems may introduce disturbing flaws because they are more focused on functionality than rather behavioural correctness. Provable security (Pass, 2011), which is a common method used to prove security properties of cryptographic primitives, may be an option to deal with ambiguities in systems. However, as attested in Bodei et al. (2005) and Pointcheval (2005), the method is more effective for proving the properties of cryptographic algorithms. Moreover, the provable approach lacks the support of computer-aided tools and as a result it becomes prone to error (Ngo et al., 2010). Formal methods (Woodcock et al., 2009) (which refers to the use of rigorous mathematical techniques and tools for specification, design and verification) provide the ability to construct precise and unambiguous models. These models can be analysed to reduce errors that are often introduced in systems (Tretmans, 1999). This approach effectively helps us to reduce the efforts usually required by designers to manually investigate possible conditions that may lead to unexpected events.

This paper presents a Coloured Petri Nets (CPN) (Jensen et al., 2007) based approach that is used to create a parameterised model for DNP3-SA. CPN is a formal and discrete-event modelling language for system design, specification, simulation, validation and verification. Its graphical and programming interfaces provide the ability to express concurrency in communication protocols, data networks, and creating concepts at different levels of abstraction. Parameterisation in CPN is a technique used to create a single model in order to prevent the possibility of having separate models for different behaviours. Applications of CPN have been beneficial in modelling and analysing various industrial processes; ranging from protocols and networks to military systems (Tritilanunt et al., 2006; Floreani et al., 1996). DNP3-SA operates in two modes: Non-aggressive *Challenge-Response* (NACR) and *Aggressive Mode* (AGM). In our previous work, we provided a security analysis of the NACR, with a focus on packet inspection at the reception level of an outstation (Amoah et al., 2014). This paper extends our previous work by adding the AGM mode. Specifically, our contribution is three fold. Firstly, we use the concept of parameterisation to create a CPN model that covers the two communication modes of the DNP3-SA; *NACR* and *AGM*. The model is based on the specification and the experimental observations of real device behaviour (Substation Modernisation Platform/Distribution Processor Gateway (SMP4/DP)). We used the CPN state space analysis tool (Jensen et al., 2006) to validate the correctness of the model and check the authentication property. Secondly, we identified a violation of the authentication property in the aggressive mode through extensive state space analysis and simulation using the parameterised model. The violation is revealed by a previously unidentified security flaw in the NACR. The flaw allows an attack to manipulation certain sequence of messages to execute commands. Thirdly, by using the parameterised model, we present two different approaches that are used to counter the identified flaw. We analyse the two proposed approaches to show that the flaw has been resolved.

The paper is organised as follows. Section 2 gives an introduction to DNP3. Sections 3 and 4 respectively describe the approach used in modelling and model description as well as colour set declarations. Section 5 presents the validation and verification analysis of our DNP3-SA CPN model. Our proposed solution is presented in Section 6. Finally, Section 7 presents our discussion, conclusion and future work of the paper.

2. Overview of the Distributed Network Protocol Version 3 (DNP3)

DNP3 is the defacto communication protocol for master stations and outstations in power grids. Exchange of messages in the protocol is in the form of requests and responses. Each of these messages (requests and responses) contains an *application control field* (AC), *function code* (FC) and *object header* (OH). AC is used to determine whether a given fragment has been received in the correct order. FC specifies the action of the request or response sent. OH is supplementary information, usually associated with DNP3 objects that *may be* required to create a complete DNP3 message. DNP3 objects are index points within the protocol database software that store data such as binary input/output, analog data, and counters. OHs *may* sometimes be required to accompany function codes in messages to specify what format, type or group of data a station must process and return as response. For example, a master station may use the FC `0x01` and OH `g12v1` to *read* the current analog input type value from the outstation. Furthermore, a response fragment contains an additional field called the Internal Indicator (IIN). IINs are found in responses from outstations. They indicate certain states and error conditions within outstations (IEEE, 2012, p. 13–23).

DNP3-SA is the security mechanism, which provides authentication in the application layer of the DNP3 protocol. The mechanism ensures that stations are protected against "rogue applications" that may want to manipulate the protocol. The authentication mechanism is unilateral but it operates in two ways; *one-pass* and *two-pass* authentication through a Keyed-Hash Message Authentication Code (HMAC). Two-pass authentication is known as the *challenge-response* or *non-aggressive challenge response* (NACR) while the one-pass is the *aggressive mode* (AGM). It is to be noted that before the AGM (one-pass) operation can be carried out, there should be at least *one or more* occurrences of the NACR operations. This enables the AGM to make use of certain crucial components from NACR operation (this behaviour is elaborated and illustrated in Fig. 1). In terms of operation, DNP3-SA strictly ensures that certain requests, particularly '*critical requests*', invoked by either master stations or outstations, are challenged and authenticated correctly for every session before they are further processed. A request or unsolicited response is considered critical, if the message contains a *mandatory code*. A *mandatory code* is any code that can potentially control a given station, by performing set-point adjustments or setting certain parameters. Any station that makes use of mandatory codes in a given message shall be challenged by the receiving device's security mechanism to prove its identity.

Table 1 depicts a message sequence chart (MSC) that presents the behaviour of DNP3-SA. For simplicity, we have omitted the AC fields in all DNP3 packets because they do not contribute to the result of this paper. Master station and Outstation represent the communicating entities. Csk_{mo} is a controlling session key obtained from a *long-term secret key*, L_k, which is manually distributed among the entities. The session key is used to authenticate data transmitted in the control direction by the master station. *FC*, *OH* and *IIN* respectively represent the function code, object header and internal indicator data that may be contained in a request or response (standard and error). Standard responses are expected responses for a particular request sent, while an error response could be a failure in authentication. *Chlg* represents a challenge message. It contains a Challenge Sequence Number (CSQ), *Sn*, that increases by i (where $i \leftarrow 1$) each time a challenge is issued, a Message Authentication Code (MAC) algorithm, *H* and a nonce *N*. $HMAC_1$ and $HMAC_2$ represent HMAC tags generated by the master station and outstation respectively. Finally, U_{ID} represents a user identification number, which is associated with the communicating parties.

In Table 1, **NACR** presents the non-aggressive challenge-response operation. The master station sends a request that requires a critical service to the outstation. On receipt of the request, the outstation issues *Chlg* accordingly. The master station

Please cite this article as: Amoah R, et al. Formal modelling and analysis of DNP3 secure authentication. Journal of Network and Computer Applications (2015), http://dx.doi.org/10.1016/j.jnca.2015.05.015