Simulating and Detecting Attacks of Untrusted Clients in OPC UA Networks

Charles Varlei Neu
University of Santa Cruz do Sul
Santa Cruz do Sul, Brazil Pontifical Catholic University of Rio Grande do Sul, Porto Alegre, Brazil
charles.neu@edu.pucrs.br

Ina Schiering
Ostfalia University of Applied Sciences, Wolfenbuttel, Germany
i.schiering@ostfalia.de

Avelino Zorzo
Pontifical Catholic University of Rio Grande do Sul, Porto Alegre, Brazil
avelino.zorzo@pucrs.br

ABSTRACT

The usage of machine to machine communication and Industrial Internet of Things is increasing nowadays, in particular in industry environments. Devices with low hardware capabilities may e.g. be used for sensing data, for example, on an industrial network. Specific protocols and frameworks were being developed for these use cases. One such framework is OPC UA, which allows signed and encrypted communication and therefore addresses already important security requirements. However, an attacker may also be able to encrypt malicious packets so that it may bypass security systems and/or empower the attack, as encrypted packets typically need more hardware consumption to be handled. In this paper the focus is on Denial of Service attacks in OPC UA networks. An analysis of possible Denial of Service attacks is presented and an approach to detect such attacks is implemented in the context of a simulation scenario. Our evaluations show how such attacks may affect server CPU consumption and could be very powerful when a large number of devices is compromised.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability;

KEYWORDS

OPC UA, Security, Industrial Internet of Things, Denial of Service, Intrusion Detection System

ACM Reference Format:
Charles Varlei Neu, Ina Schiering, and Avelino Zorzo. 2019. Simulating and Detecting Attacks of Untrusted Clients in OPC UA Networks. In *Central European Cybersecurity Conference (CECC 2019), November 14–15, 2019, Munich, Germany*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3360664.3360675

1 INTRODUCTION

The *Industrial Internet of Things (Industrial IoT)* is gaining importance in industry. One important change in this context is the increasing use of IP-based networks instead of domain specific standards as Process Field Bus (PROFIBUS) or other specific field bus technologies. This allows for an easy transfer of innovations of IP-based networks as e.g. virtualization to the industrial environment. Also IT based networks allow for remote service and easy cooperation between different production sites.

This is fostered by developments as *OPC Unified Architecture (OPC UA)*, an open source standard for machine to machine communication in the area of industrial automation developed by the OPC Foundation [8, 14]. The central idea of OPC UA is to facilitate the communication in plant automation. It allows for cross-platform communication based on an integral information model. The Arc Advisory group [11] estimates the number of globally installed OPC clients at 47 million in 2016.

Industrial IoT systems potentially constitute a considerable safety and security risk, especially if they are used in critical infrastructures as e.g. in the power industry [13]. But also in general these systems are a critical target of cyber attacks since it is possible to cause damage to production systems and even human lives [15]. Zhu et al. [18] provide a thorough investigation of security threats and cyber attacks of SCADA systems.

An advantage of standards as OPC UA is that they already incorporate measurements for authentication and encryption as central countermeasures against cyber attacks. However, security mechanisms for detecting and preventing threats of Industrial IoT systems that use OPC UA, such as *intrusion detection systems (IDS)* [16], which are well-established in IP-based networks, are not sufficiently investigated and improved for this area.

The focus in this paper is on cyber attacks in OPC UA networks, based on a thorough threat analysis of the Federal Office for Information Security [7], especially analysis for *Denial of Service (DoS) attacks* in encrypted communication, which are an important area for investigations. The contribution is the investigation of DoS attacks of untrusted clients in encrypted OPC UA networks. Different DoS attacks in OPC UA environments are described and analyzed. In a simulated OPC UA environment data corresponding to normal behavior and data concerning different DoS is generated. During these different simulation scenarios CPU consumption is measured. A data mining approach is developed to propose IDS methodologies for detecting encrypted DoS attacks in OPC UA networks.

2 RELATED WORK

Industrial IoT is a subject of increasing importance. Recent trends in this area and the growing significance of security are described by [4],[6],[17]. Sadeghi et al. [15] present a broad overview of security

CECC 2019, November 14–15, 2019, Munich, Germany

ACM ISBN 978-1-4503-7296-1/19/11…$15.00
https://doi.org/10.1145/3360664.3360675

and privacy risks in industrial IoT and address the need for further research. As an example for critical infrastructures security risks in the power industry are investigated by [13].

OPC UA incorporates approaches for security measurements [10]. The specific security risks in OPC UA including different types of attackers and the security modes are analyzed in [7]. There it is pointed out that by using standard authentication and encryption measurements which are integrated in OPC UA modes already a broad range of potential attacks are addressed. Remaining security risks that are possible while using authentication and encryption in OPC UA are identified in the area of DoS attacks of untrusted clients. It is important to be aware of that the use of encryption also potentially poses new risks because it is difficult to detect encrypted attacks [1], [19]. Also the overhead for this sign and encryption mode is considerable [5] under certain circumstances.

As additional security measurements Kim et al. [12] propose an unidirectional security gateway system that allows to monitor the network communication in an OPC UA system. Since DoS attacks as described in [7] use standard messages to built up a secure channel or to close a connection, additional investigations to detect such attacks would be needed. Bhardwaj et al. [2] consider *Intrusion detection systems (IDS)* for industrial protocols and describe methods to parse the specific headers of these protocols such as Modbus and Profinet. Specific aspects of DoS attacks are not addressed. An important methodology for network intrusion detection in general is data mining which was investigated by [16] in a general setting.

3 OPC UA NETWORKS

The basis for secure communication in industrial IoT in the context of OPC UA networks is to use encryption and authentication features which are already integrated in the standard. In the following an overview about OPC UA networks, existing security modes and the structure of network communication between an OPC UA server and client is presented.

Figure 1 shows an example of an OPC UA network architecture in the context of the automation pyramid. To address different levels of the automation pyramid typically the network is at least divided into the network segments plant floor network, operation network and corporate network. OPC UA servers can communicate with OPC UA clients or other OPC UA servers, and also with other systems as e.g. Enterprise Resource Planning (ERP) systems, connecting production and planning processes.

3.1 OPC UA security modes

To address a broad range of applications encompassing different security and timing requirements, OPC UA incorporates a flexible security model consisting of three security modes [14] [8]. Table 1 shows the main properties of each security mode.

To address the central threat of adding malicious clients to the network, authenticity and integrity of message content can be realized with the secure mode **Sign**. Based on a public key infrastructure and certificates, identities of servers and clients can be verified.

When also the content of network communication should be encrypted, the so-called **SignAndEncrypt** mode can be used to ensure also confidentiality. See [7] for a critical consideration of the security modes of OPC UA, which is not in the focus here.

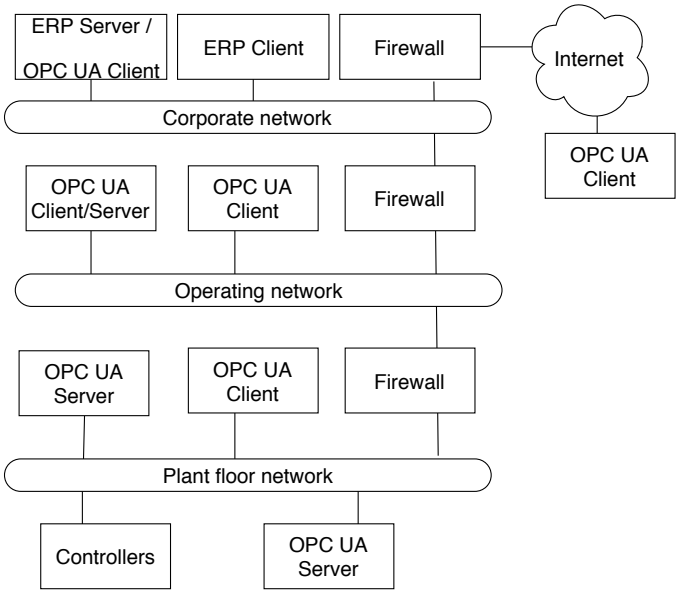

Figure 1: OPC UA network architecture

Table 1: OPC UA Security Modes

MODE	PROPERTIES
None	• No security
Sign	• Encoded with sender's private key • Only certificate owner has the private key • Anyone can verify the identity • Provides authenticity
SignAndEncrypt	• Adds encryption to sign • Encoding with receiver's public key • Anyone can encrypt • Only the certificate owner can read • Authenticity, confidentiality and integrity

3.2 Structure of Network Communication

This section describes the structure of network communication in OPC UA networks when the secure mode SignAndEncrypt is set. As the other security modes (None and Sign) are not in the focus of the analysis in this paper, they are not considered here.

When SignAndEncrypt is used as security mode, as a first step the client has to discover the configuration options to connect to the server, corresponding to the security mode used. This process is realized by GetEndpoints request and response messages (when the client application is preconfigured and already knows how to connect to the server, this step can be skipped). Then a secure channel has to be established, by using OPN messages. After, an OPC UA session has to be established. During this Session, client and server can exchange OPC UA data by performing read and write requests through MSG messages. Figure 2 shows the communication structure when secure mode SignAndEncrypt is used.

By analyzing the generated packets, the OPC UA session establishment is realized by the following OPCA UA message exchange:

- Client sends an OPC UA *HEL message* to the server
- Server responds with an OPC UA *ACK message*
- Client sends an OPC UA *OPN request message*, including his certificate for requesting a secure channel.
- Server responds with an OPC UA *OPN response message*. In the case of successful authentication, it includes its certificate. Otherwise, it rejects the secure channel establishment by an *OpenSecureChannelResponse* indicating that the trusted connection could not be established. This process consumes processing power for the encryption process and certificate validation.
- In the case of successful authentication, the client sends an OPC UA *MSG CreateSessionRequest message* to the server.
- The server responds with an OPC UA *MSG CreateSessionResponse message*.
- Client sends an OPC UA *MSG ActivateSessionResquest message*.
- Server responds with an OPC UA *MSG ActivateSessionResponse message*.

After that, OPC UA messages may be used for data exchange (read and write). When the client wants to close the session, it sends a *MSG CloseSessionRequest message* and the server replies with a *MSG CloseSessionResponse*, both via OPC UA MSG packets. If the client agrees to close the secure channel, it sends an OPC UA *CLO message* indicating a Secure Channel Close request.

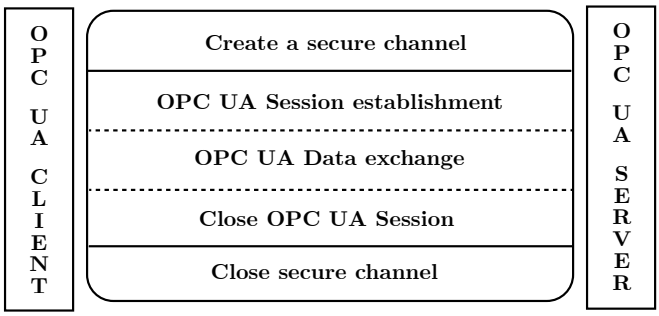

Figure 2: OPC UA Sign and Encrypted communication

Via the secure channel that is established in secure mode, authenticity of the communication partners, integrity and confidentiality of the message content is ensured. But during the setup phase of the secure channel, some attacks, mainly DoS, are still possible.

4 ANALYSIS OF DOS ATTACKS OF UNTRUSTED CLIENTS

In a detailed study of the German Federal Office for Information Security (BSI) [7] about OPC UA security, the main vulnerabilities and possible threats are analyzed, based on message type and security mode chosen. The major amount of attacks are only possible in the security modes None or Sign. Since the focus of this paper is on the security mode SignAndEncrypt, these attacks are not considered here.

In the following we concentrate on DoS, which are the most important attacks that are possible in security mode SignAndEncrypt. Since these DoS can be also conducted by clients that are introduced into the network by an attacker, these attacks are summarized as Denial of Service attacks of untrusted clients. In the following these attacks are described, analyzed and countermeasures are proposed.

4.1 Denial of Service Attacks of Untrusted Client

An attacker may be able to insert untrusted clients on the network. Thus, it is able to perform a DoS by flooding the network and the OPC UA server by continuously sending specific OPC UA messages. Such DoS scenarios are described in the following and summarized in Table 2. Beside the impact of network flooding, it is important to note that there are also compute intensive attacks possible when the server needs to evaluate certificates to answer requests [5].

Table 2: OPC UA DoS Attacks

THREAT	MESSAGE	IMPACT
HEL Flooding	HEL	network
ACK/ERR Flooding	ACK	network
GetEndPoints and FindServers	MSG	network
CLO Flooding	CLO	network
Incorrect messages Flooding	Any	network
HEL/OPN Flooding	**HEL/OPN**	**network/CPU**

- *HEL Flooding*: The client floods the server by sending HEL messages continuously. In this case, the server replies each HEL message with an ACK. This process may overload the network with HEL and ACK messages, but will not significantly impact server processing power consumption.
- *ACK or ERR Flooding*: The client floods the server by sending ACK and/or ERR messages. In this case, the server replies with an ERR message. This overloads the network with ACK and ERR messages, but will not significantly impact server processing power consumption.
- *Incorrect messages flooding*: The attacker may also perform a denial of service attack by continuously sending incorrect messages. The server then replies with ERR messages, so overloading the network, but with low impact on server processing power.
- *CLO Flooding*: Another denial of service attack that an attacker may launch through an untrusted client, is sending continuously channel close request messages, that the server answers with ERR messages. It overloads the network with CLO and ERR messages.
- *Find Server or Get Endpoints flooding*: A client can establish a channel by using the secure mode none, and then continuously send FindServers() or GetEndpoints() messages to the server. As the client has no established secure channel and is using a secure mode that not match the secure mode of the server, the server replies with a FindServers() and GetEndpoints() through OPC UA MSG messages. This overloads the network, with low impact on server CPU.
- *HEL plus OPN request flooding*: An untrusted client may also perform a denial of service attack by sending continuously

HEL and OPN requests to the server, which replies with ACK and ERR messages. As this OPN channel request message sent by the client is encrypted and security mode sign and encrypt is used by the server, it validates the certificate on the request message and replies with another encrypted message, consuming processing power on the certificate validation process and also on the message encryption process. Thus, it overloads the server CPU and the network by using HEL and OPN messages. This attack becomes even more powerful when the Certificate Authority is located on a different system, i.e. outside the OPC UA Server. In this case, the certificate validation time is increased more than 45 times as when located on the OPC UA Server itself [5].

4.2 Detection Approach based on Data Mining

In order to detect such DoS attacks, it is important to note that the message types used for DoS attacks are only present in a low proportion in normal OPC UA network communication compared to messages to read and write data. Hence to detect DoS attacks in OPC UA networks a first step is to determine the proportion of such messages in a normal network communication.

To determine such proportions, respectively derive an appropriate threshold, an data mining approach based on the J48 algorithm [3] in WEKA was used in this simulated scenario. This classifier takes advantage of the fact that the tree can be split into smaller subtrees with the information obtained from the attribute values. Whenever the algorithm finds a set of items that can clearly be distinguished from the other class by a specific attribute, it branches out a new leaf according to the value of the attribute [3].

The pre-processing that is needed for the training process is described in detail in Section 6. In a real environment more features and different data mining algorithms need to be considered.

5 SIMULATION SCENARIOS

To analyze the impact of the different attacks in an OPC UA environment, as a first step a simulation scenario is employed. The network architecture is composed by virtual hosts, using Virtualbox, based on the following configuration:

A computer with an Intel core i7 CPU, 16GB of RAM and a 256GB SSD disk was used to host the virtual machines that are representing three trusted and three untrusted clients. The other two trusted and untrusted clients, and also the server, were running on another hardware, based on an Intel i7 CPU, 8GB of RAM and 256GB SSD disk. Each Virtualbox client is configured with an UBUNTU 16.4 Operating System, on a virtual machine with 1 GB of RAM, 1 CPU core and network bridged networking mode. The hardware of the OPC UA server consists of 2GB RAM and 1 CPU core. The standard platform and toolkit provided by Unified Automation [9] is used to implement the OPC UA Server and clients. The data generated was collected by WIRESHARK/LIBCAP and stored as PCAPNG files. Figure 3 shows the implemented network topology.

Based on this general scenario three specific scenarios to simulate network behavior are used to create datasets for further investigation of DoS attacks. As a baseline a dataset representing normal behavior is important. Afterwards the two different general DoS attack scenarios are simulated, i.e. DoS attacks with impact mainly on the network and DoS attacks with impact on network and CPU of the OPC UA server.

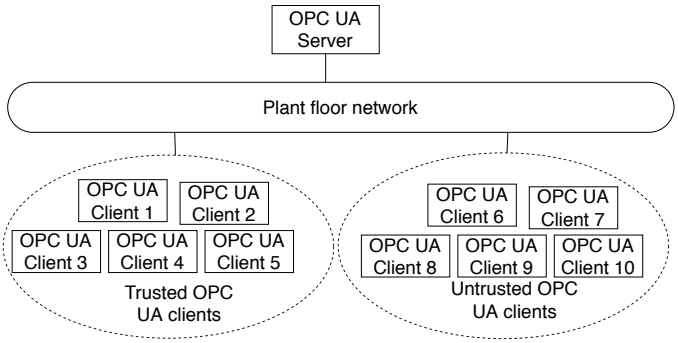

Figure 3: OPC UA Simulation Scenario

Dataset Simulating Normal Behavior. In order to investigate the impact of DoS attacks and for a test of the detection mechanism presented above, normal network behavior is needed as a baseline. Thus, the environment with OPC UA server and clients according to Figure 3 was used without untrusted clients. Each trusted client device runs a script that first establishes a secure channel and then a secure session. Afterwards, it performs data read and write requests and responses, through the message type MSG, representing a normal OPC UA data exchange. On this experiment, each client was running for 60 minutes.

The number of generated packets on this experiment is shown on the second column of Table 3. In this simulation, the client only needs one HEL message to find the server alive and get endpoints(). The server replies such HEL message with an ACK, including get endpoints() data, because of the secure mode that this server is using. Then a channel request is made via an OPN request, which is replied with OPN response. The client then sends two more messages in order to establish a secure session and the server replies with HEL messages. After that, clients are connected to the server and are finally able for exchanging data through MSG messages, until a session close is requested by client using a CLO message.

Dataset Simulating DoS Network Impact. As a next step, a simulated dataset was built, containing malicious packets with impact on the network, i.e. network flooding. Therefore, the environment as shown in Figure 3 with trusted and untrusted clients was used. Normal data on trusted clients as described in the previous experiment was created, i.e., five clients were connected to the server, with valid signatures, exchanging OPC UA data. In addition five untrusted clients were considered. Those clients ran a script that continuously sends malicious messages, performing DoS attacks on the network and the server by sending messages as follows:

- HEL: HEL messages were sent continuously (40 messages per second), so flooding the server, that replies with ACK and/or ERR messages, overloading the network.
- ACK and ERR: Those messages were also sent by the untrusted clients continuously (40 messages each per second). The server replied with ERR messages. This also overloaded the network.

- CLO: Untrusted clients were also sending CLO messages continuously (40 messages per second), so flooding the server that has to reply with ERR messages and overloading the network.

The number of OPC UA packets generated during this experiment is shown on the third column in Table 3.

Dataset Simulating DoS Network and CPU Impact. The third dataset that was created, is representing data on a scenario under attack that has impact on Network and CPU. Therefore, we also used the entire environment shown in Figure 3 and repeat the experiment as described for the dataset for Normal Behavior. Besides, the five untrusted clients ran a script that floods the network and server CPU, by continuously sending packets as follow:

- HEL: HEL messages were sent continuously (40 messages per second), so flooding the server, that replies with ACK messages, overloading the network.
- OPN: Those untrustworthy clients also sent OPN messages that normally are used to establish a secure connection with the server, which analyzes the certificate and then responds with an OPN response to deny the connection, ever when it identifies a not authorized client. This OPN flooding was simulated by sending 40 OPN request messages per second, from each untrusted client. The server has to react to this secure channel request validation by analyzing senders certificate and perform encryption and decryption on the requested message and its reply.

The number of OPC UA packets generated on this experiment is shown on the last column in Table 3.

Table 3: Generated OPC UA packets of Scenarios

Message Type	Normal	Network Impact	Network/CPU Impact
HEL	15	432,015	432,015
ACK	15	432,030	432,018
ERR	0	432,013	0
OPN request	15	15	432,015
OPN response	15	15	432,015
CLO	10	432,010	10
MSG	1,062,313	1,062,313	1,062,313

6 SIMULATION RESULTS

6.1 DoS Impact on CPU utilization

In order to analyze CPU usage on each scenario, CPU utilization of the server was monitored every ten milliseconds. From this data an average of a 5 minutes interval is considered. That experiment was repeated 4 times and a mean value was calculated. On the normal behavior scenario, the CPU utilization is around 3 percent. It is increased to more than 5 percent on the next scenario, when untrusted clients were added and generating DoS attacks with network impact and more than 10 percent when the untrusted clients are performing DoS attacks based OPN packets which has an impact on network and CPU. Figure 4 shows the CPU utilization for each experiment.

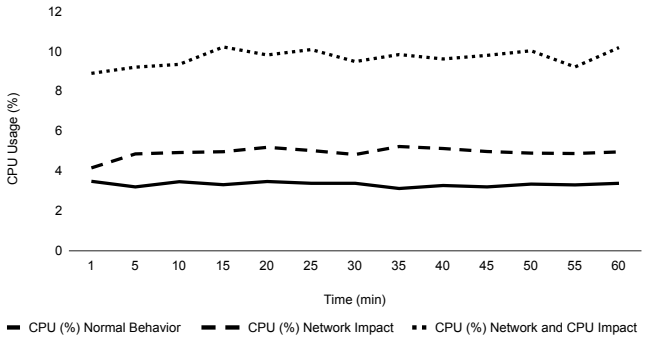

Figure 4: CPU usage on our experiments

As our scenario is composed of a few devices only, including trusted and untrusted clients, the CPU consumption may not have a high impact on the server. However, in a real environment hundreds and even thousands of devices may be used, and more messages may also be generated during the flooding process, so hampering the server and the network in such a way that it may deny service to legitimate OPC UA clients.

6.2 DoS Detection Approach

By analyzing packets of the simulation datasets, features were derived from OPC UA network communication which may be used for detecting a denial of service attack performed by untrusted clients:

- Messages Type (HEL, OPN, ERR, CLO)
- OPN responses that indicate that the certificate is missing or invalid on field ReceiverCertificate. This indicates an invalid certificate (out of date or injected by an attacker)
- Number of messages
- CPU usage.

Thus, our approach is based on defining a threshold for each message type sent by a client on a given time frame. To determine this threshold the J48 classifier was used on the dataset.

In order to perform packet classification and define a pattern on a normal behavior of OPC UA packets on the network, firstly we perform a pre-processing step on the data generated in the simulation scenarios. Therefore, the two PCAPNG files were concatenated to build our dataset, such that more data could be used for the pattern definition process. This dataset was labeled, based on the message type and time stamp, to allow for packet classification. This process was done by first exporting the PCAPNG data to a CSV file, that was imported on a SQL database, adding another column with a label, based on IP source, packet time and OPC UA message type. A filter was applied to exclude all non OPC UA packets from the database. Then, all the remaining packets received a label as *NORMAL*, *HEL FLOOD*, *ERR FLOOD*, *ACK FLOOD*, *CLO FLOOD* and *OPN FLOOD*. After that, the data was exported to a CSV file again, so that it may be used as an input on a data mining or classification process, concluding the pre-processing step.

Then, J48 in WEKA was used to perform classification for each attack, considering the following attributes: IP Source, Message Type, Packet Time and Label. Then this process was applied to find patterns. The option *training set* was chosen on WEKA and the classification algorithm J48 was executed, generating patterns for a normal amount of HEL, ACK, ERR, CLO and OPN packets for non malicious behavior. This classifier was chosen due to the results generated by it being a decision tree, which is useful for defining detection rules.

This process generates the following patterns for normal behavior of OPC UA clients, differentiated according to message types, on a 10 seconds time frame. This established a messages threshold that clients can send to be considered as normal behavior:

- HEL Flood: 3 HEL messages from same source
- ACK Flood: 3 ACK messages from same source
- ERR Flood: 3 ERR messages from same source
- CLO Flood: 3 CLO messages from same source
- OPN flood: 5 OPN messages from same source or 5 OPN response messages with invalid authentication reply from the server to the same destination.

Those patterns are considered as a baseline to define the concrete detection rules. As explained before, during normal behavior each client needs only a few control messages, like HEL, ACK,CLO and OPN, to connect to the server, establish a secure channel and a secure session and close the session and the channel, when sign and encryption mode is set. Moreover, even when a secure channel is established, it is not necessary to close it and re-establish the connection frequently for normal client behavior when there are no network related problems in the environment. Thus, the computed thresholds may be considered appropriate. It is important to mention, that messages originated from the server are not considered by this detection method, as the focus is mainly on untrusted clients. Therefore, the IP address of the server is not considered as a source to be analyzed.

7 CONCLUSION AND FUTURE WORK

In this work we presented an analysis of denial of service attacks that untrusted clients may perform against OPC UA servers. A virtual environment with a number of client devices and an OPC UA server was implemented, with normal OPC UA data exchange and additional untrusted clients, that were flooding the network and server, performing denial of service attacks. Although our experimental environment comprises only few devices, and simulated data was used, the evaluation shows that such attacks may be very powerful in resource consumption if a large number of devices is compromised, especially when SignAndEncrypt mode is used, due to certificate validation and encryption process. We derived a detection method for such attacks based on the number of packets of each message type on the generated dataset. Therefore, we used the J48 classifier to define a normal behavior in OPC UA communication, i.e., a threshold of OPN, CLO, ERR and HEL messages that may be considered as normal behavior. As future work, this detection methods should be evaluated on real OPC UA data (not yet available, but we are working in collecting such data), on a real industrial IoT environment with a large number of real devices.

ACKNOWLEDGMENT

The Authors would like to thank CAPES-Brazil, Deutscher Akademischer Austauschdienst - DAAD (Grant 57417991) and the Brazilian National Institute of Science and Technology in Forensic Sciences (INCT forense) project (Grant 465450/2014-8) for funding this work. Also, many thanks to Tobias Bolze for helping in this work.

REFERENCES

[1] B. Anderson, S. Paul, and McGrew D. 2017. Deciphering malware's use of TLS (without decryption). *Journal of Computer Virology and Hacking Techniques* (2017), 1–17. https://doi.org/10.1007/s11416-017-0306-6

[2] S. Bhardwaj, P. Larbig, R. Khondoker, and K. Bayarou. 2017. Survey of domain specific languages to build packet parsers for industrial protocols. In *2017 20th International Conference of Computer and Information Technology (ICCIT)*. https://doi.org/10.1109/ICCITECHN.2017.8281842

[3] L. Bilge, S. Sen, D. Balzarotti, E. Kirda, and C. Kruegel. 2014. EXPOSURE: a passive DNS analysis service to detect and report malicious domains. In *ACM Transactions on Information and System Security (TISSEC), vol. 16, no. 4, p. 14*. https://doi.org/10.1145/2584679

[4] Hugh Boyes, Bil Hallaq, Joe Cunningham, and Tim Watson. 2018. The industrial internet of things (IIoT): An analysis framework. *Computers in Industry* 101 (2018), 1–12. https://doi.org/10.1016/j.compind.2018.04.015

[5] S. Cavalieri, G. Cutuli, and S. Monteleone. 2010. Evaluating Impact of Security on OPC UA Performance. In *3rd International Conference on Human System Interaction*. 604 – 609. https://doi.org/10.1109/HSI.2010.5514495

[6] Li Da Xu, Wu He, and Shancang Li. 2014. Internet of things in industries: A survey. *IEEE Transactions on industrial informatics* 10, 4 (2014), 2233–2243.

[7] German Federal Office for Information Security (BSI). 2017. OPC UA Security Analysis. [Online]. https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/Studies/OPCUA/OPCUA.pdf?__blob=publicationFile&v=2.

[8] OPC Unified Architecture Foundation. 2017. OPC Unified Architecture Pioneer of the 4th industrial (r)evolution. [Online]. https://opcfoundation.org/wp-content/uploads/2014/03/OPC_UA_I_4.0_Pioneer_US_v2.pdf.

[9] Unified Automation GmbH. 2018. Unified Automation. [Online]. https://www.unified-automation.com.

[10] R. Huang, F. Liu, and P. Dongbo. 2010. Research on OPC UA security. In *2010 5th IEEE Conference on Industrial Electronics and Applications*. https://doi.org/10.1109/ICIEA.2010.5514836

[11] Arc Insight. 2018. OPC Technology Well-positioned for Further Growth in Tomorrow's Connected World. [Online]. https://opcfoundation.org/wp-content/uploads/2018/02/ARC-Report-OPC-Installed-Base-Insights.pdf.

[12] B. Kim, Y Heo, and J. Na. 2017. Design of unidirectional security gateway system for secure monitoring of OPC-UA data. In *2017 International Conference on Information and Communication Technology Convergence (ICTC)*. https://doi.org/10.1109/ICTC.2017.8190923

[13] Maria B. Line, Ali Zand, Gianluca Stringhini, and Richard Kemmerer. 2014. Targeted Attacks Against Industrial Control Systems: Is the Power Industry Prepared?. In *Proceedings of the 2Nd Workshop on Smart Energy Grid Security (SEGS '14)*. ACM, New York, NY, USA, 13–22. https://doi.org/10.1145/2667190.2667192

[14] W. Mahnke, S. Leitner, and M. Damm. 2009. *OPC Unified Architecture*. Springer-Verlag. 339 pages.

[15] Ahmad-Reza Sadeghi, Christian Wachsmann, and Michael Waidner. 2015. Security and Privacy Challenges in Industrial Internet of Things. In *Proceedings of the 52Nd Annual Design Automation Conference (DAC '15)*. ACM, New York, NY, USA, Article 54, 6 pages. https://doi.org/10.1145/2744769.2747942

[16] C. So-In, N. Mongkonchai, P. Aimtongkham, K. Wijitsopon, and K. Rujirakul. 2014. An evaluation of data mining classification models for network intrusion detection. In *2014 Fourth International Conference on Digital Information and Communication Technology and its Applications (DICTAP)*. 90–94. https://doi.org/10.1109/DICTAP.2014.6821663

[17] John A Stankovic. 2014. Research directions for the internet of things. *IEEE Internet of Things Journal* 1, 1 (2014), 3–9. https://doi.org/10.1109/JIOT.2014.2312291

[18] Bonnie Zhu, Anthony Joseph, and Shankar Sastry. 2011. 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*. IEEE, 380–388. https://doi.org/10.1109/iThings/CPSCom.2011.34

[19] M. Zolotukhin, T. Hamalainen, T. Kokkonen, and J. Siltanen. 2016. Increasing Web Service Availability by Detecting Application-Layer DDoS Attacks in Encrypted Traffic. *23rd International Conference on Telecommunications (ICT)* (2016). https://doi.org/10.1109/ICT.2016.7500408