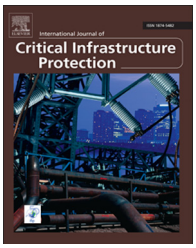
Available online at www.sciencedirect.com

ScienceDirect

www.elsevier.com/locate/ijcip



Flow whitelisting in SCADA networks

Rafael Ramos Regis Barbosa[a,*], *Ramin Sadre*[a], *Aiko Pras*[b]

[a]*Design and Analysis of Communication Systems, Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands*

[b]*Distributed and Embedded Systems, Department of Computer Science, Aalborg University, Selma Lagerlofs Vey 300, DK-9220 Aalborg, Denmark*

ARTICLE INFO

Article history:
Received 15 January 2013
Available online 20 August 2013

Keywords:
SCADA systems
Intrusion detection
Network flow whitelisting

ABSTRACT

Supervisory control and data acquisition (SCADA) networks are commonly deployed in large industrial facilities. Modern SCADA networks are becoming more vulnerable to cyber attacks due to the common use of standard communications protocols and increased interconnections with corporate networks and the Internet. This paper describes an approach for improving the security of SCADA networks using flow whitelisting. A flow whitelist describes legitimate traffic based on four properties of network packets: client address, server address, server-side port and transport protocol.

The proposed approach incorporates a learning phase in which a flow whitelist is learned by capturing network traffic over a period of time and aggregating it into flows. After the learning phase is complete, any non-whitelisted connection observed generates an alarm. The evaluation of the approach focuses on two important whitelist characteristics: size and stability. The applicability of the approach is demonstrated using real-world traffic traces captured at two water treatment plants and at an electric-gas utility.

1. Introduction

Supervisory control and data acquisition (SCADA) networks are commonly deployed to aid the operation of large industrial facilities such as water treatment plants and electric utilities. In the past, these networks were completely isolated and relied on purpose-specific hardware and software; but now they employ commodity hardware and are increasingly interconnected using standard network protocols such as TCP/IP. While this new scenario reduces costs and improves efficiency, the side effect is that the networks are exposed to a much wider range of attacks.

This paper proposes a flow whitelisting approach that seeks to reduce the number of attack vectors on SCADA networks that use TCP and UDP as their primary transport protocols. A "flow" is defined as a bidirectional sequence of packets with identical client address, server address, server-side port and transport protocol. Flow whitelists represent all the legitimate traffic based entirely on these four properties of network packets.

Flow whitelisting presents several advantages over deep packet inspection and host-level intrusion detection [7,8,10]. By not depending on the packet payload, flow whitelisting can handle proprietary protocols. Furthermore, it can operate at the network level; thus, it is not necessary to modify a host. This addresses the reluctance on the part of SCADA operators to make changes to their computing environments. Note that, although flow-level whitelists are not commonly used in traditional IP networks because the number of legitimate connections is too large to be manageable, they have been applied to specific problem domains such as reducing SPAM [5], avoiding phishing [9], guaranteeing access to important customers during DDoS attacks [15], and preventing various VoIP infrastructure attacks [6].

*Corresponding author.
E-mail address: r.barbosa@utwente.nl (R.R.R. Barbosa).

1874-5482/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ijcip.2013.08.003

The main motivation for using whitelists is that most SCADA network traffic is generated by automated processes, such as the periodic polling of field devices. Also, SCADA systems are closed with very limited external access, if any. Moreover, changes are rare, in other words, SCADA hosts and services are infrequently added to or removed from the network.

Whitelisting has been recommended by several entities as a means for implementing SCADA security. For instance, the Norwegian Oil and Gas Association [12] suggests that "all access requests shall be denied unless explicitly granted. The U.S. National Institute of Standards and Technology (NIST) [14] recommends the "[blocking of] all communications with the exception of specifically enabled communications." However, to the best of our knowledge, the viability of whitelists has never been studied in real-world SCADA environments. In the previous work [2], we have shown that connection matrices are remarkably stable in SCADA networks, suggesting that whitelists are feasible in these environments.

This paper presents an approach for flow whitelisting in SCADA networks that assist network administrators in detecting illegitimate network traffic. In order to be viable, a whitelist must possess two characteristics. First, its size must be manageable. A very large list with millions of entries, as encountered in traditional IP networks, would make the approach infeasible to implement and manage. Second, the whitelist must be stable. If the list is unstable (i.e., changes frequently), it would require continuous updating by the network administrator or it would generate a large number of false alerts. The feasibility of the whitelisting approach is demonstrated using real-world traffic from two water treatment facilities and an electric-gas utility.

2. Flow whitelisting approach

Fig. 1 presents an overview of the flow whitelisting approach. The traffic in the SCADA network is captured, aggregated to connections and then aggregated to flows. A "connection" is defined as all packets with the same source/destination IP address, source/destination port and IP protocol, regardless of the direction in which the packets are sent. In the subsequent learning phase, the flows are observed over a certain period of time in order to create an initial flow whitelist. A flow whitelist contains entries of the form: client IP address, server IP address, server port, IP protocol. After the whitelist is generated, the connections are analyzed in the detection phase. All network traffic matching a whitelist entry is considered to be legitimate. Every connection that does not match a whitelist entry generates an alarm.

2.1. Connection and flow creation

Since the approach relies on IP packet header information, the packet headers have to be captured in the (sub)networks that are to be monitored by the whitelist. This paper only considers TCP and UDP packets. Connection creation involves the aggregation of the captured packets to connections. As mentioned before, a connection is defined as all packets with the same source/destination IP address, source/destination port and IP protocol, regardless of the direction of the packets. The end of a connection is determined either by using a TCP state machine or an inactivity timeout of 300 s. Our experiments used the `argus` open source tool to perform this task.

The flow creation step identifies the client and server sides of the connections and further aggregates the connections according to the four-tuple flow definition. Four rules are applied in sequence to identify the server side:

- Rule 1 applies to all TCP connections for which a three-way handshake is observed. The server is set to be the host that received the SYN packet or sent the SYN/ACK packet.
- Rule 2 is applied if a well-known port (below 1024) is observed: the host using such a port is set to be the server. Note that in the case of an active FTP session, where the originator of a data connection is the server, the source port 20 is set as a service port. In the case of protocols that use the same (well-known) port on both hosts (e.g., NTP), Rule 1 or Rule 4 is used for classification.
- Rule 3 is a heuristic. If the same protocol and port are reused by a host in multiple connections, then the host is set as the server and the protocol–port combination is used to identify the service. The heuristic relies on the fact that client ports normally vary with each connection and are less likely to be repeated. Rule 3 makes it necessary to keep every connection that is not classified by Rules 1 and 2 in memory until a second connection with a repeated host address, protocol and port is observed; this can potentially delay the analysis indefinitely. A timeout could be used in an online implementation, after which time the connection is classified using Rule 4. The offline implementation described in this paper employs an infinite timeout.

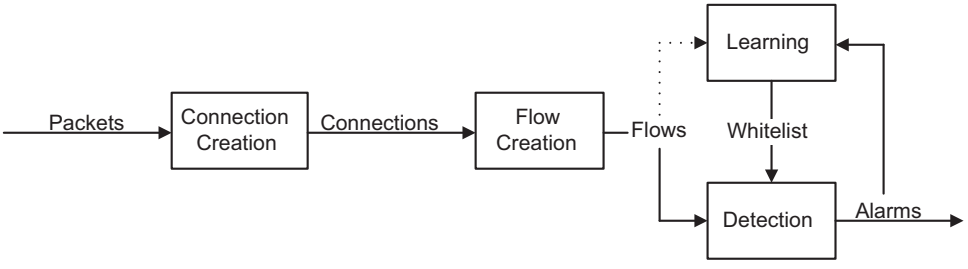


Fig. 1 – Flow whitelisting approach.