2016 IEEE First International Conference on Internet-of-Things Design and Implementation

Layered Security and Ease of Installation for Devices on the Internet of Things

Roger D. Chamberlain*[†], Mike Chambers[†], Darren Greenwalt[†], Brett Steinbrueck[†], and Todd Steinbrueck[†]
*Dept. of Computer Science and Engineering
Washington University in St. Louis, St. Louis, MO, USA
[†]BECS Technology, Inc., St. Louis, MO, USA
Email: {roger,mike_c,darren,brett,todd}@becs.com

***Abstract*—One of the major issues that must be addressed in the emerging Internet of Things (IoT) is balancing the needs of security and reasonable installation and maintenance efforts. If the security infrastructure is not relatively easy to use, it will ultimately be compromised. This paper describes the industrial deployment experience of the EZConnect™ security infrastructure implemented by BECS Technology, Inc., a firm that provides water chemistry monitoring and control equipment to the aquatics market.**

I. Introduction

Firewalls are a crucial element in modern cyber-security deployments. However, they are also a substantial impediment to integrating devices in the Internet of Things (IoT). When IoT devices are attached to a local-area network, it is frequently the case that any attempt to contact these devices remotely (from outside the local-area network) is blocked by a firewall. Authorized remote access requires explicit intervention in the security infrastructure meant to protect the local-area network from attack. For field-area network installations, an even greater set of issues must be considered [12].

Many approaches to enable remote access can compromise the security of the local-area network. Yet it is important not to compromise security, rather we must enhance it, since security threats are real [9].

In 2001, an individual was convicted of hacking into a computerized waste water management system in Queensland, Australia, causing raw sewage to spill into local parks and rivers [13]. From 2003 to 2006, at least four cyber-attacks on water supply systems in the U.S. were reported to WaterISAC, an industry information sharing and analysis center. In one of these attacks, the attackers declared their presence with the message, “I enter in your server like you in Iraq.” [4].

In spite of these real threats, if a security infrastructure is overly burdensome, it will either not be used at all or will be diminished in effectiveness by lack of diligence on the part of the owners/operators. To be truly effective, any approach to security must be paired with an approach to ease the burden on the user [5]. Hertzum et al. [6] assessed the intrinsic tensions between security and ease of use in an e-banking context, and concluded that ease of use limitations can directly contribute to decreased security.

This paper describes the industrial deployment experience of our approach for providing secure connectivity to installed embedded IoT devices. BECS Technology is a manufacturer of water chemistry monitoring and control equipment for the aquatics market. BECS provides remote access capability to its devices for owners/operators from both desktop software and mobile apps for smartphones and tablets.

Marketed under the trade name EZConnect™, we describe an approach to supporting remote communication with IoT devices that satisfies the need for security yet balances that need with the equivalent need for ease of installation and maintenance. After discussing the prior security approaches that we previously suggested to our customers, we will describe EZConnect, what it is and how it works, as well as describe the security layers it embodies and properties that facilitate ease of use and maintenance. We will also comment on its acceptance in the marketplace.

II. Prior Practice

Prior to the development of EZConnect, the users of controllers manufactured by BECS were required to either use a VPN or have their IT department enable port forwarding in their firewall.

At some level, the standard-of-practice for remote secure access to a local-area network that is protected behind a firewall is the use of a VPN [11], or virtual private network. It has the advantage of common use, so that the available solutions have enough history that they can be reasonably trusted. In addition, it is a solution that IT departments are very familiar with, providing a certain level of institutional comfort.

However, truly secure VPN solutions can have significant overhead both in terms of setup, maintenance, and individual use. First, VPNs are managed and maintained by the IT department, so by definition they require the active involvement of the IT department, both for initial setup and for any changes (e.g., account changes due to personnel transitions, etc.). Second, it is difficult (often against organizational policy) in many cases to provide a VPN account to those who are not employees of the organization, since the account gives access to the local-area network as a whole, not just an individual piece of equipment on the network. However, as we will describe below in Section V, temporary remote access to the equipment for diagnostic purposes is often desired by equipment managers. Third, many VPN solutions require a fob

978-1-4673-9948-7/16 $31.00 © 2016 IEEE
DOI 10.1109/IoTDI.2015.32

IEEE computer society

or other physical authentication device that can have quantity or institutional policy limitations.

The other approach commonly used to provide remote access through a firewall into a local-area network is via port forwarding [2]. Here, an association is made within the firewall configuration that enables an attempted connection outside the network to be completed inside the network.

This approach also requires the involvement of the IT department; however, it gives the misleading impression that the IT department's involvement is substantially lower than with the VPN solution. The idea is that once the port forwarding is setup, IT need no longer be involved. In fact, unless the port forwarding associations are updated (e.g., when personnel change), it is not truly secure, and when the associations are changed, both IP addresses and ports used to remotely access equipment must be updated. This approach also has the downside that users must now use a different IP address/port combination when accessing the equipment from within the local-area network vs. when accessing the equipment remotely.

In effect, both of these approaches have the significant disadvantage that they each heavily involve IT department resources (i.e., they take time to setup and maintain) and also raise legitimate security concerns on the part of IT managers. This is particularly true for the port forwarding approach, which can become a security hole if not carefully monitored (see, for example, the description of the 'port forward' exploit described by Ammann et al. [1]). From a usability point of view, the need to distribute (and maintain) IP addresses is a clear negative.

III. EZConnect

The typical configuration of an EZConnect installation is depicted in Figure 1. The controller on the left is connected via a local-area network and sits behind a firewall. Applications (either desktop programs or mobile apps) on the right wish to communicate with the controller; however, attempts to connect directly to the controller via the Internet are normally blocked by the firewall.

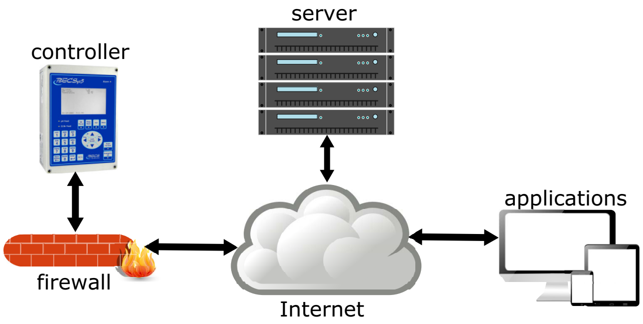

Fig. 1. EZConnect system diagram.

Rather than have any device outside the firewall attempt a direct connection, the approach used by EZConnect is to have the controller, which is inside the firewall, make an outbound connection to the EZConnect server. Whenever the EZConnect option is enabled on a controller, the controller will automatically contact the server and register the controller. The controller then maintains an open socket on the server, which will subsequently be used for authenticated remote connections. The server will then wait for an application to request a connection to the controller.

When an application wishes to connect to a controller, rather than attempting to connect to the controller directly, contact is made with the EZConnect server, eliminating the need for a dedicated access path through the site's router/firewall. The application must provide both the serial number of the controller as well as a valid authentication code (described below), upon which communication is established by the server, forwarding messages between the application and the controller.

The sending of messages back and forth between the application and the controller is still insufficient to enable modifications to the controller. Any parameter changes on the controller require an additional access code (also described below) to be provided by the user.

IV. Security Layers

Current best practices do not rely on an individual "silver bullet" to provide security, but instead rely on a layered approach, in which one component of the security infrastructure being compromised does not compromise the entire system. In the EZConnect system, there are a number of layers of security, each of which will be described below, along with some comments on the tradeoffs that they imply.

1) **TLS Encryption** – All communications between the controller(s), the server(s), and the application software are encrypted with the industry standard TLS (Transport Layer Security) cryptographic protocol [3].

   In addition to encrypting the data sent over the network, TLS also ensures that the server is the legitimate server and not an impostor with potentially malicious intent.

2) **Proprietary Communication Protocols and Applications** – The protocol used to communicate with the controllers has intentionally limited semantic capability. It only supports the download of data logs and reading/writing of parameters stored on the controller that deal with the controller's function as an aquatic controller. The protocol does not include commands to interact with the local network or embedded operating system, and cannot be used to upload software (either to the controller itself or to any other device on the network).

   In addition to the semantic limitations, the protocol itself is unpublished and only supported by applications provided by BECS. This has the effect of minimizing the attack surface exposed to potential threats [10], especially relative to systems that rely on protocols supported by any web browser (e.g., HTTP and HTML).

Maintaining this confidentiality has similar issues as maintaining any trade secret, once disclosed it is no longer a secret, and it does not encourage interoperability. The semantic constraints are definitely the strongest aspect of this security layer.

3) **Controller Access Codes** – Every controller has 3 levels of access code (password) protection: Operator, Manager, and Rep (a manufacturer's representative is typically responsible for installation and setup). Parameter changes on the controller can only be performed after the user has been granted one of these access levels, which are enforced for both local (controller front-panel user interface) and remote changes.

   Operators are allowed lowest-level access (e.g., establishing setpoints, recalibration of sensors, reset alarms, etc.). Managers have access to additional controller configuration options (e.g., default front-panel display) and establishing Operator accounts. Reps have full access to detailed control parameters (e.g., time-based proportional control setup) and various installation verification tools (e.g., relay overrides, etc.).

   The access codes are controlled and managed by the owner/operator of the controller. As such, they embody all the pros and cons of user control (e.g., users generally pick codes based on ease of remembering rather than difficulty in breaking). Currently, access codes only include numeric characters (which are unfortunately vulnerable to brute force attack), since the front panel of the controller does not support alphanumeric input; however, this is slated to be expanded to full alphanumeric support in the next generation of the controller line.

4) **EZConnect Authentication Codes** – In addition to the access codes described above, each controller has a list of authentication codes, which identify users that have been authorized to access that controller remotely. Authentication codes are 8-character alphanumeric values generated at random by the controller. Generating, changing, or deleting them requires either Manager or Rep access, persons with Operator access are not able to view or modify them. They are entered into the application that is requesting remote connectivity; however, they are never visible in the application (i.e., after validation, the applications display "validated" rather than show the authentication code).

   Since the controller creates randomly generated authentication codes, they are less susceptible to list-based attacks than user-generated passwords.

5) **Physical ROM Program Store** – The controllers are multi-processor systems, and the processor core that performs the actual control functions is a dedicated chip that reads its program only from a physical read-only memory (ROM). The only way to alter the code executed by this processor is to physically replace the program memory chip.

   While this design has clear implications for supporting the controllers (e.g., requiring physical access to provide software updates), it makes it that much harder for those with malicious intent to subvert the fundamental control function by replacing the controller code.

   The processor cores executing other functions are using an embedded Linux kernel that has state-of-the-art defenses integrated (e.g., see [7] for many of the techniques utilized). For example, the only open port is the one that is accessed by the proprietary protocol, and messages sent to that port that are malformed in any way do not receive any response.

One way in which the limited semantics of the proprietary interface supports secure operations is that the controller does not support software alterations from the network. We have already mentioned that the processor responsible for control operations requires a chip change to alter the software. In addition, we only support software updates for the remaining processor cores through a USB port on the controller.

Clearly, the overall security is also a function of the servers that provide connectivity to the controllers. The security of these servers is beyond the scope of this paper; however, we note that the above implies that even a server breach does not allow access to the local-area network on site.

The above security layers support the ease of installation goals as described next.

V. Ease of Installation and Maintenance

One of the primary motivations for the development of the EZConnect system was to ease the installation effort required on the part of both equipment managers and their colleagues in the IT department. Rather than asking the IT department to alter firewall settings (to support port forwarding) or setup a VPN, all that is required is that the firewall support outbound connections, which are commonly supported in modern deployments. In the event that outbound connections are disabled by default, it is sufficient for the IT department to enable outbound connections on only one port, a request that has only been denied by by one customer's IT department to date.

The actual installation is almost completely plug and play. At startup time, the EZConnect option must be enabled (by default it is disabled) on the controller itself. The controller then initiates the connection to the server as described in Section III. Once this connection is established, users can then connect (virtually) to the controller through the server.

This ease of installation should be contrasted with the steps necessary to enable remote access using either of the previously described methods. Installers needed to interact with their IT departments attempting to communicate using terms that are completely unfamiliar to them (e.g., IP address, subnet mask, etc.). Even simple transcription mistakes (which are made more likely by the unfamiliarity of the terminology) cause the system to fail, triggering diagnostic effort on the part of both the IT and equipment installation teams. The EZConnect installation approach has already shown itself to be vastly superior in the field.

One feature of the EZConnect system that has proven to be very popular is the ability for equipment managers to temporarily allow remote access to service personnel. When the water chemistry is an undesirable state (e.g., some alarm condition), it is not unusual for equipment owners to contact service personnel from either the manufacturer (BECS) or the manufacturer's representative to help diagnose and correct the issue.

Under normal circumstances, these service personnel do not (and should not) have access to the equipment. However, it is straightforward for a manager to create a new remote authentication code and provide it (along with an access code if desired) to the service personnel. Once the service is complete, the access and authentication codes can be readily disabled.

In this way, service personnel can be granted remote access to the controller for diagnostic purposes and potentially for corrective action, without requiring the intervention of the IT department to either enable or subsequently disable the access.

Contrast this with the approach required when secure access is being provided by a VPN. First, the IT department must be involved in both creating a new account and disabling it when no longer needed. Second, the access provided to the service personnel is not only to the controller, but is instead access to the internal network of the organization. This requires a higher level of trust than that needed with EZConnect. Third, remote connectivity requires that the IP address of the controller be known outside the organization, a circumstance that is avoided entirely with EZConnect.

A manager's ability to disable remote authentication codes also supports the effective management of access in the case of departing employees as well. If each authorized employee has a unique authentication code, one individual's departure need not impact the other employees' credentials.

VI. Conclusions and Future Work

At the time of writing, controllers that support EZConnect have been through beta testing in between 50 and 100 installations, and they are currently being manufactured and installed as a released product.

So far, only one IT department has denied the request to support outbound connections through the firewall. This is a significant change from earlier circumstances, in which a typical installation required substantial meetings (and negotiation) with the IT department before connectivity might be supported, and it was all too frequently ultimately denied.

The EZConnect system has the following benefits:

- Highly secure with multiple layers of security measures.
- Hassle-free setup and operation.
- IT department does not need to establish VPN or forward ports on router.
- No need to know or distribute the controller IP address.
- User of remote application need only know controller serial number and authentication code.
- Equipment manager has full control over remote access.

There are, however, a number of limitations to be acknowledged. The use of a proprietary language is only secure while the trade secret is maintained, and a more robust character set for access codes would be less vulnerable to brute force attack. The use of proprietary mechanisms also limits interoperability. All in all, it is a nice compromise between the requirements for security and the reality that if it is not easy to use it will not get used.

There are a number of directions we are investigating that will further improve the system. Probably the most impactful is the desire to incorporate active intrusion detection mechanisms into the existing layered security approach. Within the controller itself, we can investigate the ideas presented by Li et al. [8], in which a separate, trusted OS runs concurrently with the general purpose Linux kernel and is charged with monitoring for integrity violations. More generally, intrusion detection mechanisms that are deployed on the server(s) in the cloud have the unique benefit of knowing what controllers have connected to them and are therefore potential targets for attack. They are therefore subject to intrusion detection monitoring.

References

[1] P. Ammann, D. Wijesekera, and S. Kaushik, "Scalable, graph-based network vulnerability analysis," in *Proc. of 9th ACM Conf. on Computer and Communications Security*. ACM, 2002, pp. 217–224.

[2] A. Apvrille and M. Pourzandi, "Secure software development by example," *IEEE Security & Privacy*, vol. 3, no. 4, pp. 10–17, 2005.

[3] T. Dierks, "The transport layer security (TLS) protocol version 1.2," 2008, rfc5246. [Online]. Available: http://tools.ietf.org/pdf/rfc5246.pdf

[4] R. Esposito, "Hackers penetrate water system computers," *ABC News*, Nov. 1, 2006. [Online]. Available: http://blogs.abcnews.com/theblotter/2006/10/hackers_penetra.html

[5] D. Gefen and D. W. Straub, "The relative importance of perceived ease of use in IS adoption: a study of e-commerce adoption," *Journal of the Association for Information Systems*, vol. 1, no. 1, p. 8, 2000.

[6] M. Hertzum, N. Jørgensen, and M. Nørgaard, "Usable security and e-banking: Ease of use vis-a-vis security," *Australasian Journal of Information Systems*, vol. 11, no. 2, 2004.

[7] D. Kleidermacher and M. Kleidermacher, *Embedded Systems Security: Practical Methods for Safe and Secure Software and Systems Development*. Waltham, MA, USA: Elsevier, 2012.

[8] N. Li, Y. Kinebuchi, and T. Nakajima, "Enhancing security of embedded Linux on a multi-core processor," in *IEEE 17th International Conference on Embedded and Real-Time Computing Systems and Applications*, vol. 2, Aug. 2011, pp. 117–121.

[9] E. Luiijf, "SCADA Security Good Practices for the Drinking Water Sector," *TNO, The Hague, TNO-DV*, p. C096, 2008. [Online]. Available: http://m.tno.nl/media/1538/tno-dv-2008-c096_web.pdf

[10] P. Manadhata and J. Wing, "An attack surface metric," *IEEE Transactions on Software Engineering*, vol. 37, no. 3, pp. 371–386, May 2011.

[11] C. Medaglia and A. Serbanati, "An Overview of Privacy and Security Issues in the Internet of Things," in *The Internet of Things*, D. Giusto, A. Iera, G. Morabito, and L. Atzori, Eds. Springer New York, 2010, pp. 389–395. [Online]. Available: http://dx.doi.org/10.1007/978-1-4419-1674-7_38

[12] P. Palensky and T. Sauter, "Security considerations for FAN-Internet connections," in *Proc. of IEEE Int'l Workshop on Factory Communication Systems*. IEEE, 2000, pp. 27–35.

[13] T. Smith, "Hacker jailed for revenge sewage attacks," *The Register*, Oct. 31, 2001. [Online]. Available: http://www.theregister.co.uk/2001/10/31/hacker_jailed_for_revenge_sewage