Secunia
Stay Secure

The (In)Security State of SCADA Software Systems

Author: Dmitriy Pletnev, Secunia Research Security Specialist

Date: 11th April, 2011

Recently, there has been an increase of publicly disclosed vulnerabilities in Supervisory Control and Data Acquisition (SCADA) software systems. This wave of reports underlines the ever increasing interest in analysing the security of such systems. In this blog entry we are going to discuss the vulnerable components of SCADA software systems based on analysis performed by Secunia Research while verifying the reported vulnerabilities. Additionally, we will provide some recommendations to mitigate exploitation and reduce the attack surface.

As previously discussed **[1],** Secunia Research spends considerable effort to verify all reports to confirm their validity and appropriately rate the severity and impact of a given vulnerability. As a result, we have analysed numerous vulnerabilities that affect several software components of SCADA solutions offered by various vendors.

There's a common misconception that SCADA systems are secure and more difficult to attack due to their segregated implementation and required specialized knowledge to operate. However, from the perspective of a security researcher or an exploit developer a SCADA system is just another software application, which accepts external input, parses it, and carries out a predetermined action. Thus, an attacker may only need to analyse a smaller component of a system in order to compromise its security.

A typical SCADA system consists of several software and hardware components, often dispersed across multiple networks within an organization and a single (or multiple in case of redundancy) operating and monitoring location. A simplified data flow of any given process (e.g. water supply station) starts with sensors in the Remote Terminal Units (RTU) collecting certain measurements (e.g. water level). The measured data is then submitted to a supervisory system using a vendor specific communication protocol via an enterprise's TCP/IP network. The supervisory system acquires the process data and may additionally respond with commands to the RTU. The entire process' control, configuration, and monitoring is performed via a Human Machine Interface (HMI) application, installed on an operator's workstation. Due to multiple different possibilities in the implementation of a SCADA system dictated by an organization's needs e.g. network design, administrative flexibility, or budget restrictions a given configuration may inadvertently increase the attack surface.

Specifically, multiple SCADA software vendors provide web-based HMI applications to reduce operating complexity as well as provide administrative flexibility. This allows an organization to provide remote monitoring and control capabilities, which may simply require a web browser and a secure network connection (e.g. HTTPS or VPN). However, this also creates a new attack vector in the form of an ActiveX control installed by the HMI application, which may allow compromising a workstation when not connected to a secure network. An attacker may exploit vulnerabilities in an ActiveX control **[2]** (SA42650) and install malicious software, which may continue execution after the operator connects to a secure network with access to a SCADA system. Potential mitigation against this type of an attack may include restricting the use of a particular ActiveX control to a trusted zone when an operator is connected to a secure network only.

Additionally, supervisory systems are normally located within a protected network, but may not always be segregated from the enterprise WAN due to various reasons e.g. budget restrictions, administrative flexibility. These systems typically are not reachable from the Internet, thus greatly reducing the risk of exploitation from untrusted networks. However, they are still vulnerable to attack from within the enterprise, which may span multiple geographic locations. This reduces the level of trust due to increased enterprise complexity. Moreover, recent evidence shows that many supervisory systems do not securely implement the required SCADA communication protocols leaving them open to a multitude of attack vectors.

Most of the vulnerabilities associated with supervisory systems analysed by Secunia Research do not require authentication in order to successfully exploit **[3]** (SA43849). The analysed vulnerabilities vary in impact anywhere from “Less critical” information disclosure issues **[4]** (SA42730) to the more severe allowing arbitrary code execution **[5]** (SA43851). It should be noted that based on our analysis the vulnerabilities are straight-forward to exploit due to apparent insecure coding practices by the vendors’ developers and simple designs of the communication protocol, which do not require intimate knowledge of a SCADA system nor its implementation details. Some systems implement more complex forms of communication such as Remote Procedure Calls (RPC), but that does not appear to significantly reduce the attack surface **[6]** (SA43877). A potential mitigation strategy in such cases, which significantly reduces the risk of exploitation, is to restrict access to a supervisory system to trusted hosts only.

In conclusion, based on analysis performed by Secunia Research of publicly disclosed vulnerabilities in SCADA software systems by external researchers, the evidence clearly shows the apparent insecurity of such systems when left unprotected or insecurely implemented. Moreover, recent security incidents resulting from the infamous Stuxnet worm **[7]** highlight the fact of increased interest by attackers to manipulate and compromise various industrial control systems. The common myth that specialized knowledge is required to successfully exploit these systems is exactly that, a myth.

Concerned parties wishing to significantly reduce the risk of successful exploitation should follow security best practices when designing and implementing the supporting infrastructure for a SCADA system, such as secure network design and more stringent software controls for HMI-based workstations. Vendors of SCADA software should incorporate secure coding practices into the software design lifecycle and potentially enhance the Quality Assurance (QA) process. Lastly, some organizations may be in a position to exert certain pressure on vendors of SCADA software in order to provide more security features, which will benefit the entire SCADA community. The provided examples underline the interest within the security community to analyse SCADA software systems and based on this trend, Secunia Research expects to see an increased focus and number of vulnerability reports being disclosed in the near future.

[1] http://secunia.com/blog/128/

[2] http://secunia.com/advisories/42650/

[3] http://secunia.com/advisories/43849/

[4] http://secunia.com/advisories/42730/

[5] http://secunia.com/advisories/43851/

[6] http://secunia.com/advisories/43877/

[7] http://en.wikipedia.org/wiki/Stuxnet