21st International Congress on Modelling and Simulation, Gold Coast, Australia, 29 Nov to 4 Dec 2015
www.mssanz.org.au/modsim2015

Uncovering Industrial Control Systems Vulnerabilities by Examining SCADA Virtual Packages and Their Communication Protocols

W. J. Seo [a] and E. Sitnikova [b]

[a] Division of Information Technology, Engineering and the Environment, School of Information Technology and Mathematical Sciences, Mawson Lakes Campus, University of South Australia, South Australia
[b] Australian Centre for Cyber Security, School of Engineering and Information Technology, University of New South Wales Canberra at ADFA

Email: seowy002@mymail.unisa.edu.au

Abstract: Supervisory Control and Data Acquisition (SCADA) is the centralized computer system that controls and monitors the Industrial Control Systems (ICS) that are connected to it. SCADA is used in various industries and the military, and it is essential to many critical infrastructures. However, in most nations SCADA was introduced a few decades ago, as one of their security measures. With rapid technology advancement over the decades, this out-dated security measure could never be on par with the security risks that SCADA systems are facing. It is necessary to investigate the potential vulnerabilities on the structural level and mitigate them. Focusing on the software vulnerabilities, this paper aims to investigate the potential security issues that could cause malfunction of SCADA systems that control critical infrastructures, including electricity, water, gas, traffic, military, and others which causes impact to the survival of a nation. With various SCADA packages available in the market, Citect SCADA from Schneider Electric Corporation has been selected for this research due to their popularity in Australian industry. Several versions of the Citect SCADA are being used in the investigation due to the fact previous versions are still widely used in industry at the time of writing. This paper investigates the structure and functionality of the SCADA packages to uncover their vulnerabilities and proposes recommended countermeasures.

Keywords: *SCADA, cyber security, critical infrastructure*

1. INTRODUCTION

Supervisory Control and Data Acquisition (SCADA) systems are computerised systems that are responsible for supervising the overall operations in critical infrastructures. These critical infrastructures include the production and distribution of telecommunications, water supply, electric power, oil & gas, defense systems, satellites, and other essential frameworks that sustain the general wellness of the society (Moteff 2004). SCADA not only controls and monitors the functionality of the Industrial Control System (ICS) that are connected to the network, it also collects and analyse the data from the ICS network to ensure timely reports are generated to support decision making of the operator. In another words, SCADA is the centralized system that supervises the entire automated operation in a site or even complexes of systems across different sites.

Traditionally, these critical infrastructures were secured due to the isolation of these facilities from the outside world. Even the corporation administration and system control are built separately; until nowadays, when critical infrastructure providers' growing demands for robust monitoring and real-time data acquisition, to increase the performance, flexibility and efficiency of SCADA system. Since then, SCADA systems are being connected to corporate intra networks and some are even connected to the internet directly or indirectly (Amanullah et. al. 2005). Such exposure of SCADA systems has opened up a series of methods for cyber attackers to exploit the vulnerability of SCADA systems without physical contact. Although many claimed to be protected, SCADA systems that are exposed are vulnerable to malicious attacks and might cause catastrophic disaster to those who depends on them.

In this paper, our main focus is to summarize the results of investigating Citect SCADA packages that have widely been used in critical infrastructure industry as well as proposing some methods to counter those issues. The investigation started with examining the overall file structure and file systems of the SCADA packages to form a series of test plans, test strategies, and test cases that are relevant. Based on the testing that has been conducted, we propose some methods that could mitigate the vulnerabilities or reduce the risk to the minimum level. The following section is an overview of the generic prevention and protection security measures that are commonly being practiced in the industry. This is followed with the description of Citect SCADA packages as well as its components and vulnerabilities. Based on the vulnerabilities that have been uncovered, the next section proposes several methods that could help to resolve these vulnerabilities. This paper is concluded with future work and acknowledgements.

2. GENERIC PREVENTION AND PROTECTION SECURITY MEASURES

It is important to detect the vulnerabilities of a system before they are discovered by people who have evil intentions that may result in disastrous incidents. Several approaches have been developed to discover the vulnerabilities which are mainly categorized as Penetration Testing, War Games, Anti-virus and Firewalls. Reconnaissance and exploitation tools, such as Nessus, Nmap, Metasploit, CORE Impact, and others were designed to perform penetration testing on computer networks (Northcutt 2015). Realizing the importance of the Industrial Control System (ICS) industry, these tools also include SCADA modules that are specifically used to test the SCADA system with known security issues (Nicholson et. al. 2012). These tools were designed to explore and identify the potential vulnerabilities before the attackers do. Different to other vulnerability assessments, penetration testing will go as far as the tools can within the scope of the contract signed by the SCADA system operator with the risk of compromising the system (Burns et. al. 2007). Penetration testing exposes the vulnerabilities of the network, and also verifies and provides valuable information to further refine the security configuration of the SCADA system. However, some of these tools are publicly available to download from the Internet, and if used by the adversaries, the unpatched SCADA system might be exposed to the attackers. These tools are developed for the actual penetration of a network, and with sufficient knowledge and experience, one could penetrate the security of a network and gain access or even full control of the SCADA system. Due to the fact that the knowledge and information to understand the threats posed by cyber intrusion in the scale of a war are very limited, war games have been held to gain an in-depth learning experience for both military and critical infrastructures operators. Information that incites even the interest of the national security planners is difficult to obtain by in-house drill or training (Geers 2010). Similar to a physical military exercise, the war games are simulated attacks in mock intrusion scenarios, where there are teams of hackers and defenders of the SCADA system. War games are performed even with by governments and critical infrastructures, to test the security of their systems (Global Security 2011). As one of the most basic security measures in all IT related industries, anti-virus and firewall has always been relied as the front line defence against malware and illegal access. On the contrary, Cai et. al. mentioned in their study that the responsiveness of the SCADA systems was being reduced by these protective measures. Above all, SCADA systems require communication in a timely manner; slothful response is not acceptable in regards of SCADA system functionality. As a result, these security measures

are being neglected or ignored by the operators of the SCADA system, to reduce latency while enhance the responsiveness of SCADA systems (Cai et. al. 2008). In other words, the overall IT infrastructure and the SCADA system could be defenseless and exposed to malicious attacks.

3. SCHNEIDER ELECTRIC CORPORATE – CITECT SCADA

In this research, we focus on Citect SCADA. Citect was created by Control Instrumentation in Australia during 1973 but was acquired by the French multinational corporation, Schneider Electric group in 2006. It was reported in 2008 that Citect sold more than 150,000 licenses of its software worldwide. Citect covers industries from Aerospace and defense to manufacturing and others, and it is still expanding and growing. Citect has been in Australia for over 40 years and has always been the most popular SCADA package regardless of the different type of industries in Australia (Yang et. al. 2003). Citect has an extraordinary coverage of bundled protocols and drivers which made it compatible with almost all the ICS devices in the industry (Schneider Electric 2009). Also well known for its reliability and flexibility, Citect is widely deployed in other countries (Shi et. al. 2005). Citect is a package of five applications, including Citect Explorer, Citect Project Editor, Graphics Editor, Cicode Editor, and Citect Runtime.

A few vulnerabilities have already been exposed by other scholars in regarding Citect. For example in 2008, the director of Netragard Network Penetrating Testing Company, Kevin Finisterre, published a stack-based buffer overflow code that could gain complete control of the software. It was shown that, whoever has access to the TCP connection of the Open Database Connectivity (ODBC) port 20222, is capable of launching an attack on the SCADA system. In other words, if the SCADA system is connected to the Internet, the attacker only needs to have an Internet to execute the attack. The code has been published as a module of the Metasploit penetration testing tool kit. The vulnerability was made known to Schneider five months before the vulnerability was being published for research purposes. Consequently, a patch was released by Citect. However, it is questionable if all the vulnerable Citects integrated in the industry have been patched against this vulnerability.

3.1 Cicode and Kernel

Cicode is the structured programming language developed by Citect SCADA, capable of multi-tasking, multi-threads, and remote procedure. Most of the structures and syntax used in Cicode are similar with Visual Basic or Visual C programming language with the support of automation libraries and drivers (Zou & Gao 2008). Implemented as part of the developer tools, Cicode were designed for the accessibility of sophisticated commands that are not illustrated on the default graphical user interface. Cicode is accessible at any time during SCADA operation.As mentioned in the Cicode Reference Guide page 411, there is a command in Cicode named “DspKernel” which displays the Kernel window of the Citect SCADA system. All Cicode functions could be executed without privilege restrictions, which also means that the total control of the SCADA system could be seized and overwritten from the Kernel (Schneider Electric 2010). The Kernel has direct access to all real-time data and all facilities (including tags, alarms, trends, reports, and etc) that is not even presented in the default Citect Human Machine Interface (HMI). Although the Cicode Reference Guide was written for Citect version 7.20, the access to the Kernel has been implemented into Cicode ever since the introduction of Cicode. If the security features are not properly configured, and the remote access feature is activated, the Kernel is accessible anywhere by anyone who has basic knowledge of Citect.

Another notable potential vulnerability of Cicode is the accessibility of the code. Cicode is typically stored as a “*.ci” extension in the project directory. To access and modify Cicode, one just locates the file and opens it using simple text editor such as Notepad. Designed to provide customization for specific needs of corporation, however Cicode is stored as raw code without compilation that could be jeopardized by anyone who has physical access to the SCADA system. The following screenshot shows that the Cicode from the “EXAMPLE.CI” with the codes that execute external application in MS Excel. In another words, just by adding a few more lines of code, Cicode could also be used to execute other scripts or even malwares. There may have been cases where usernames and hard coded passwords for accessing the database have been exposed as text in Cicode files. For example, in order to access a SQL server constantly, the login details are usually hard coded in the codes, without going through compilation this information is openly available when Cicode files are accessed with text editor as shown in Figure 1. Similar events have been stated in the Citect SCADA 7.20 Cicode Reference Guide as well, listed on page 567 that illustrates the syntax to open a FTP connection (Schneider Electric 2010). To contrast, all the information stored in Cicode is open text, including credential information.

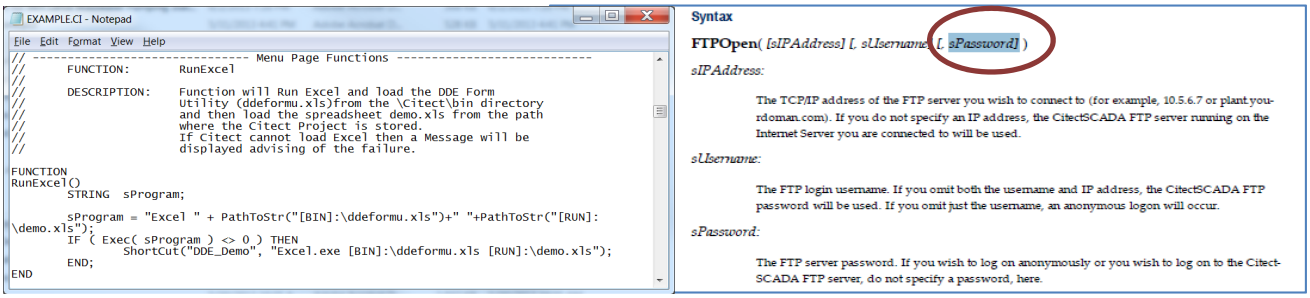

Figure 1. Codes in Example.ci and Screenshot from Cicode Reference Guide.

3.2 DBF File Storage

The data storage systems used by the Citect is a dBase database which is one of the earliest database management systems on the market since the 1970s. Supported with Clipper file extension of ".ndx", the dBase files used in Citect have the file extension of ".dbf". More than 50% of the whole Citect SCADA software is made out of dBase and its Clipper files. The DBF file is used to store drivers and protocols information, project file location, and other information. There is however no encryption or authentication of accessing the dBase files; the files could be open with freely distributed dBase file readers including credential information stored in the dBase files which are pure ASCII that does not require any interpreter to comprehend the data stored in the dBase file. Even the sensitive information such as username and password are unencrypted and stored as ASCII text in the "users.DBF" file under the project folder. Figure 2 below is the screenshots taken from Citect.

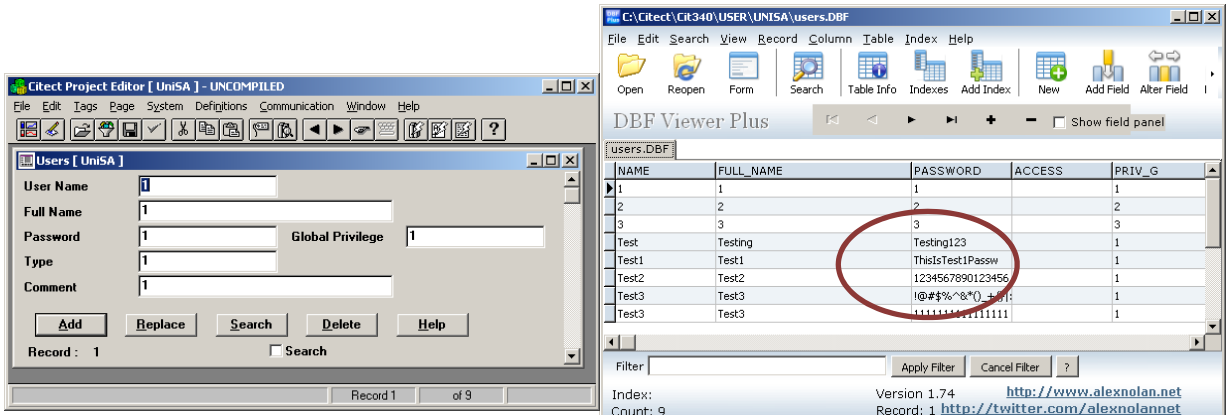

Figure 2. Screenshot of Users Creation from Citect 3.40 and Viewed with DBF Viewer Plus.

The following screenshots were taken with Citect version 3.40, with several entries of sample user creation. Shown on the left is the user creation dialog box, and DBF Viewer Plus on the right, showing the entries that have been created. A few sample users have been created with password of just one character; some have been made to test if the password or other data field has been encrypted; as well as different users with the same user name is also included in the test.

As illustrated, the name, full name, password level of privilege of the registered users is easily accessed and visible using a DBF file reader. The first three users are created with a single character of a username and password, and the system did not prompt any warning or checking. It is then followed by test sample passwords with the combination of different characters and lengths. However the system did not check the redundancy of the same username, which means that the system could accept different users with the same username or users with the same username could login with different passwords. The passwords are stored and could be easily accessed using any dBase file reader.

It has been discovered that only after version 7.30 that such security issue has been counteracted with the implementation of encryption of the password field. It appears that upgrading to the version 7.30 would have solved the issue of encrypting the password saved in the dBase files, but nothing has been done in regards to disabling the accessibility and manipulation without authentication. Thus it is still possible for attackers who have physical access to the SCADA system to change the privilege and access control easily.

3.3 INI Configuration Files

The INI files which are heavily used in every version of Citect contain the primary information of Citect file structure but are not properly protected or encrypted. The INI file is easily accessible using simple text editor like notepad, and the information stored are ASCII code which is usually human readable in English. Originated from the word initialize, the INI file is designed to store the informal configuration setting commonly used by previous version of MS Windows operating system since Windows 95.

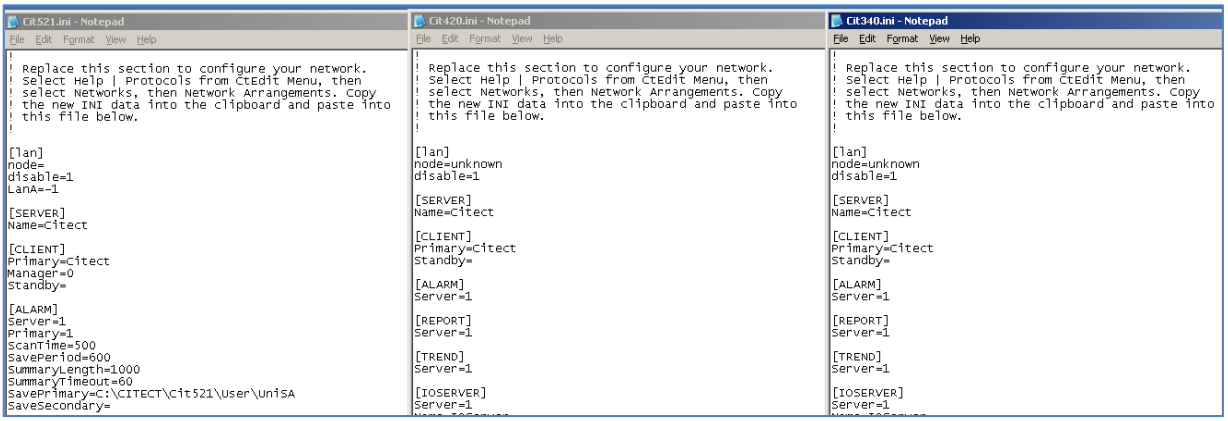

Figure 3. Screenshots of INI Configuration files from Citect 5.21, 4.20, 3.40, respectively.

The screenshot in Figure 3 shows the INI files from three different versions of Citect, which are Citect 5.21, Citect 4.20, and Citect 3.40. The location of the INI file is located in the installation directory, which is by default "C:\Citect\" and without any physical restriction or authentication of the computer, the INI file is exposed to anyone who has access to the computer. The INI file of Citect 5.21 contains details of the internet connection configuration, such as the Internet Server, Primary, and Standby DNS server as shown in the next figure. If being accessed, an attacker could reroute all the data transferring between the SCADA system and the ICT devices via the attacker's DNS server. Doing so, the attacker could manipulate any of the genuine data and forge legitimate data while during transmission, creating a man-in-the-middle attack.

In Citect version 7.30 the INI configuration files could still be located in a different location than previous versions, however it no longer contains server information, instead it only contain basic display configuration of Citect. Thus, it has reduced the risk of being compromised by a man-in-the-middle attack.

4. PROPOSED METHODS

Based on the three major Citect security issues, the following recommendations are suggested:

4.1 Cicode and Kernel

Cicode by default has no authentication that could restrict the Cicode user from accessing the features embedded in the kernel. Due to the fact that it is part of the developer tools, it is reasonable not to apply security in the coding environment. Conversely it is crucial to safeguard Cicode from illegal access. Very often in order to save time and trouble of having to go through multiple layers of authentication, many SCADA operators have disabled the screensaver and the logon screen for convenience. As a result, it exposed the SCADA system to anyone who can get physical access to it. Implementing security policies to restrict physical access to the SCADA system is one of the most basic mitigations that should be implemented by the SCADA operators. Identified as a human mishandling, the best possible solution for such is to enforce the security policies to ensure the proper managing of the SCADA system.

Furthermore, regarding the issues of raw and not compiled Cicode, Schneider Electric should release patches or applications that provide an Integrated Development Environment (IDE) for Cicode programming. The application should be able to compile all the Cicode files into encrypted files that should not be accessible by any third party applications. As well as removing all the previous "*.ci" files, to avoid any information leakage that might be a result of credential information disclosed to adversaries. The consequences of Cicode

vulnerabilities could easily cause the SCADA system to fail, therefore mitigation to resolve these issues should be implemented as soon as possible.

4.2 DBF File Encryption

All the DBF files found in Citect are not encrypted and could be easily access by any dBase file viewers. Sensitive information such as usernames and passwords have been freely accessible by anyone who has physical access to the SCADA system. Without prior knowledge or experience, other information such as privilege level and access level could be easily manipulated. This vulnerability existed only on the previous version of Citect, according to the investigation, all other version before version 7.30 have been found vulnerable with password exposed as pure text. Moreover, knowing that it will be impossible to physically help all their clients to comply with the FDA 21 CFR Part 11 of US, Schneider Electric published a white paper in 2008 to urge SCADA operators to implement changes to comply with the requirement by themselves. But Sands of AutomateNow, argues that such a publication reaching all the SCADA operators is doubtful. As mentioned earlier, even with version 7.30, it is still possible to manipulate the access level of the users in Citect. The version 7.30 only solves the security problem partially. It is recommended that Schneider Electric should release a patch for all vulnerable versions that could encrypt some of the dBase file which contain sensitive information, such as the illustrated “users.dbf” file. As files like such should not even exist in simple accessible form which could attract attention of the adversaries and expose credential information of the SCADA system.

4.3 INI Configuration Files

INI files were widely used in many configurations of software in the last few decades, such as the start-up configuration “Boot.ini” used by the operating system like Windows 95, 98, ME, 2000, and XP. However it is no longer used for significant purposes such as start-up or initialisation in most of the applications now-a-days. Instead, it is being replaced with more complicated file structures that are not easily accessible by normal users. Currently INI files could only be found in some simple message or insignificant configuration storage, such as “desktop.ini” and “folder.ini”. Similarly, Citect 7.30 is also following the footstep of such evolution. The vulnerability that has been found in this section is in Citect version 5.21, which store configuration information of servers. It is however no longer appearing in the latest version. Therefore it is suggested that Schneider Electric should release patches for relevant version to mitigate the risk of exposing Citect to potential man-in-the-middle attack.

5. CONCLUSIONS

The several versions of Citect have been selected as the research targets, including Citect versions 3.40, 4.20, 5.21, and 7.30. Investigations have been conducted towards the software packages, a few vulnerabilities have been identified, which could be categorized into three main categories. Namely Cicode and Kernel, DBF file storage, and INI configuration files. The significant and consequences of each of these vulnerabilities have been reviewed and elaborated accordingly. To summarize, the Cicode and Kernel has exposed the risk of allowing remote access to anyone without legitimate authentication. In the same category, Cicode files that might have contained credential information are being easily accessible with text editor. DBF files storage has the vulnerability of exposing sensitive information; especially the “users.dbf” which stores the usernames and passwords of all SCADA users could be easily accessed and manipulated using any dBase file view. Lastly, the INI configuration files which were used to store initializing the start-up of Citect version 5.21 has been found containing information that could be used for man-in-the-middle attacks.

6. FUTURE WORK

This research has revealed only a few SCADA security vulnerabilities that are known by the researcher. It is believed that there are more hidden vulnerabilities that need to be uncovered before they are found by the adversaries. More research is needed to be conducted on other SCADA packages and its various versions, to ensure other the SCADA packages that are currently used in the industries are properly secured.

Acknowledgement: This research project was supported by AutomateNow, a South Australian automation integration company providing intelligent control system solution for domestic and industrial applications. The author knowledge the efforts and support from Cameron Sands of AutomateNow who shared his experience and considerable time to this work.

REFERENCES

Amanullah, M.T.O., Kalam, A., Zayegh, A. 2005, Network Security Vulnerabilities in SCADA and EMS, 2005 IEEE/PES Transmission and Distribution Conference & Exhibition: Asia and Pacific, Dalian, China.

Burns, B., Granick, J.S., Manzuik, S., Guersch, P., Killion, D., Beauchesne, N., Moret, E., Sobrier, J., Lynn, M., Markham, E., Lezzoni, C. & Biondi, P. 2008, Security power tools. O'Reilly Media.

Cai, N., Wang, J., Yu, X. 2008, SCADA system security: Complexity, history and new developments, 2008 6th IEEE International Conference on Industrial Informatics, page 569-574, DCC, Daejeon, Korea July 13-16.

Geers, K. 2010, Live Fire Exercise: Preparing for Cyber War, Journal of Homeland Security and Emergency Management: Vol. 7: Iss. 1, Article 74.

Global Security 2011, Eligible Receiver, URL: (http://www.globalsecurity.org/military/ops/eligible-receiver.htm) (last accessed 16 May 2015)

Moteff, J. & Parfomak, P. 2004, Critical Infrastructure and Key Assets : Congressional Research Service - The Library of Congress.

Nicholson, A., Webber, S., Dyer, S., Patel, T., Janicke, H. 2012, SCADA security in the light of Cyber-Warfare, Computers & Security, Volume 31, Issue 4, June 2012, Pages 418-436.

Northcutt, S., Shenk, J., Shackleford, D., Rosenberg, T., Siles, R., & Mancini, S. 2015, Penetration Testing: Assessing your overall security before attackers do. SANS Institute, URL: http://www.sans.org/reading-room/whitepapers/analyst/penetration-testing-assessing-security-attackers-34635 (last accessed on 27 May 2015)

Schneider Electric 2009, History – The history of Citect, URL: http://www.schneider-electric.com.au/sites/australia/en/company/about-us/global-history/citect-history.page (last accessed 27 May 2015).

Schneider Electric 2010, Citect SCADA v7.20 – Cicode Reference Guide, Schneider Electric (Australia) Pty. Ltd. Octorber 2010.

Shi, S., Zhang, S.M., Huang, L.P. 2005, A Survey of the Dynamic Data Exchange Technology of Citect, Journal of Kunming Metallurgy College, Vol 21. No. 1, Jan 2005.

Yang, X.Z., Cheng, G.G., Zhu, X.B. 2003, Application of the Citect Configuration Software in Industrial Control System, Journal of Wuhan University of Science and Technology (Natural Science Edition), Vol 26. No.3, Sep 2003.

Zou, J. H. & Gao, L. 2008, Programming Technique for Supervising System Based on CITECT, Journal of Kunming University of Science and Technology (Science and Technology), Vol. 33 No.5, Oct 2008.