DOI:10.16410/j.issn1000-8365.2020.09.013

基于工业软 PLC 源码保护的方法

杜银学,杜立强,高灵宝,杜海平,郑栋娥
(共享智能铸造产业创新中心有限公司,宁夏 银川 750021)

摘　要:针对源代码下载存在泄露,逆向工程和篡改方面风险的现状分析,以及这些风险常见的源码保护方法,研究了一种基于软 PLC 源码是否下载到工控机的检测算法。将 ST 源码转化为 XML 文本,提取 XML 文本各行标记的属性,然后根据关键字查询 XML 文本中该关键字出现的次数。当关键字出现的次数大于设定的次数后,就会出现提示,同时终止软 PLC 程序的下载,在关键字检测的基础上还增加了文件大小的检测算法;当文件大于默认的值后,也会将文件中的内容删除,保证了源码的安全性。结果表明,通过基于软 PLC 源码检测的提出,Coding-Detection 库实现了相应的功能,解决了源码下到软 PLC 程序难题。

关键词:关键字检测;XML 文本;源码:ST 语言;软 PLC

中图分类号:TG23　　**文献标识码**:A　　**文章编号**:1000-8365(2020)09-0857-04

Source Code Protection Method Based on Industrial Soft PLC

DU Yinxue, DU Liqiang, GAO Lingbao, DU Haiping, ZHENG Dong'e
(KOCEL Intelligent Casting Industry Innovation Center Co., Ltd., Yinchuan 750021, China)

Abstract: In view of the status analysis of the risk of leakage, reverse engineering and tampering in source code download, as well as the common source code protection methods for these risks, a detection algorithm based on soft PLC source code downloaded to industrial control computer were studied. Converted the ST source code to XML text, extracted the attributes of each line of XML text markup, and then queried the occurrence times of the keyword in XML text according to the keyword. When the keyword appears more than the number of times set, would appear prompt, at the same time to terminate the soft PLC program download, on the basis of keyword detection also increased the file size detection algorithm; When the file was larger than the default value, it would also delete the contents of the file, ensuring the security of the source code. The results show that through the propose software PLC source code Detection, Coding-Detection library realize the corresponding functions, and solve the source code down to the software PLC program problem.

Key words: keyword detection; XML text; source code: ST language; soft PLC

随着现代化信息技术的不断深入发展,信息化系统更新迭代与复杂程度也在不断提升;同时企业对信息化系统中的软件设计规模、软件性能、源码保护等方面的需求也越来越高,随之而来的信息安全保护问题的难度也日益突出,尤其是软件源代码的保护。因此,用来保证源码安全的工具也显得非常重要,铸造企业也不例外。

根据 NIST 的统计显示 92%的漏洞属于应用层。因此,应用系统自身源代码的安全问题是铸造企业信息安全工作中的重点,持续贯穿于企业软件开发的整个生命周期中,将源代码安全工作融入业务系统的需求分析、概要设计、编码和测试各个阶段是保证源码安全的重要一步。因此,基于 Coding-Detection 库的关键字检测和文件大小检测是值得关注的问题。

伴随着现代计算机网络技术及软件产业的迅猛发展,软件工程中涉及到的知识产权等相关安全问题日益成为行业关注的焦点。在软件开发过程中,存在着核心技术源代码会被内部或外部恶意用户窃取的危险。在软件发布和使用过程中,恶意用户通过对软件进行逆向分析,可以获取软件实现技术细节,从而达到窃取数据、仿制软件或非法攻击的目的。由此可以看出从软件设计、发布到用户使用的各个环节均存在着安全隐患。

1　软件安全数据的威胁问题

内网通常是指一个企业内部使用的网络,内网用户通过共享文件的方式来传递和使用数据,这样虽然可以提高了一定的便捷性,但这样的方式同时带来了一定的数据安全隐患。首先,这样的数据传递

收稿日期:2020-04-29

作者简介:杜银学(1986-),宁夏同欣人,专科,高级工程师.主要从事 3D 打印设备软件研发.电话:15378963271,
E-mail:yinxue.du@kocel.com

方式,内网用户可能会有获取及泄漏重要信息数据的机会;其次,这样的安全漏洞会被恶意的外部用户利用。比如:外部恶意用户采用一些木马、病毒等攻击手段,通过侵入网络,对特定主机上存储的重要数据和软件源代码等企业核心重要资源文件进行窃取和破坏,这种方式已经严重的损害了企业的核心利益。权威机构对数百多家公司调查,发现超过85%的安全威胁来源于公司内部。

当前,对于软 PLC (Programmable Logic Controllers 可编程逻辑控制器) 程序最大威胁来自于源代码遭到篡改及软件逆向工程。伴随着现代计算机网络技术的发展,软件运行环境变得更加复杂,甚至会在恶意的网络环境下运行。同时,随着软件逆向工程技术的发展,软件中的核心算法、秘密数据、业务逻辑的安全问题已经变的越来越严重,成为重点关注的问题。

2 常见的源码保护方法

2.1 文件加壳

软件加壳技术是采用特殊算法,对软件中可执行文件里的资源进行加密压缩,并生成一个新的可执行文件,可以独立运行,解压流程完全隐蔽,均在计算机内存中完成,不会被泄漏。附加在原始程序上的"壳"通过加载器载入至内存后,优先于原始程序执行,并得到软件控制权,执行过程中对原始程序进行解密和还原,还原完成后再把控制权交还给原始程序,执行原来的代码部分。在软件加壳技术中,原始程序加上"外壳"后,在磁盘存储时,通常原始程序的可执行文件是以加密后的文件形式存在的,只在软件执行时会在内存中解密和还原。这种方法比较有效地防止了破解者对软件文件进行非法修改,以及静态程序反编译。

2.2 代码混淆

代码混淆是指将源代码中各种元素的名称,如局部变量名,全局变量名、函数名,类名等在编写代码时赋予其无意义的名字。比如,命名为单个字母,或者简短且无任何意义的字母组合,甚至直接命名为含有"__"这样符号的名字,使刚开始接触阅读源代码的人员无法根据名称直接猜测其含义及用途。更有甚者会修改代码中的部分逻辑,将其变为执行结果完全相同,但是形式上更难理解。比如,精简函数的中间变量,将循环逻辑修改为递归逻辑。将容易理解的 for 循环改写为较为难理解的 while 循环等。同时,也可以通过将多行代码合并为一行,将一行代码拆分成多行代码逻辑,或删除空格等,以打乱代码的格式等。

虽然代码混淆在一定程度上可以保护程序源代码,但同时也带来一些问题,包括:混淆后的代码难于理解,给调试与除错工作带来了一些困难;团队中互相交流代码比较困难,不适合团队或跨团队的开发任务;对于像 java 这样支持反射属性的开发语言,代码混淆有可能与反射发生冲突。事实上,代码混淆并不能完全阻止软件反向工程。因此,对于软件安全性要求很高的场合,仅仅采用代码混淆的方式来保护源代码安全仍是远远不够的。

3 基于软 PLC 源码保护的方法

随着工业自动化的发展,工业控制使的 PLC 种类繁多,但这些 PLC 只能进行简单的逻辑运算,无法进行大数据处理等高数运算,人机交互困难。因此,选用工业软 PLC,可以实现与上位机通信、数据处理、模拟量采集与处理、信号控制、复杂算法等控制,实现高性能工艺要求,如:单晶硅片切割机、贴片机、纺织机等柔性要求高的行业;设备 OEM 制造厂商实现基于工业软 PLC 进行软件算法及源码控制等。

3.1 Coding-Detection(编码检测)设计目的

工业软 PLC 是基于 Windows 平台开发的,可以同时将上位机和 PLC 运行在同一台设备上,这样可以保证程序之间通讯的可靠性。与上位机交互,并可以进行二次开发自己的 HMI(调制解调器主机接口)界面。同时,可以通过后台在线监控变量的实时状态。然而,下载时如果不更改设置,会自动将源码下载到软 PLC 中,这样可能会造成源码的泄露。为了避免源代码泄漏问题,设计了 Coding-Detection 库检测算法。

3.2 设计步骤

首先,下载 PLC 程序时,先将运行工程下面的文件进行全部清除,并通过边缘检测工具检测文件是否完全删除;然后执行程序下载,下载时会调用 Coding-Detection 库,实时检测指定文件中的关键字,当文件中出现关键字时,停止程序下载,同时将下载的文件删除;最后返回设置界面对程序下载进行检查并重新设置。

3.3 实现算法和具体代码

(1)实现算法　Coding-Detection 库的实现算法如下:①将关键字保存到 ReferenceStr 字符串中;②实时读取文件中的内容,每次读取的字符串数量是固定的,并且大于关键字的字符串长度;③比较关键字中的第一个字符 ReferenceStr[0]和文件中的第一个字符 buffRead[0]是否相同,然后比较第二个是否

相同；当第一个和第二个字符相同时，后面依次循环比较；当找到关键字和文件中内容相同时则报警，同时将下载文件删除；④当比较到第三个字符时发现不同，则关键字的第一个字符和文件中的第二个字符比较，后面依次循环比较，直到关键字的长度大于文件最后一个字符减去关键字的长度加一的位置，如果在文件中没有检测到关键字，则继续执行程序下载。

(2)具体代码　如图 1 所示，为 Coding-Detection 算法核心代码块

```
ReferenceStr:='MAIN.TcPOU';
ReferenceStrLen:=LEN(ReferenceStr);
StrLenUDINT:=DINT_TO_UDINT (ReferenceStrLen)-1;
FOR Addition:= 0 TO cbReadLength BY 1 DO
    ddddddddddd:=buffRead[Addition];
IF buffRead[Addition] = ReferenceStr[0] THEN
   Pass_M:=TRUE;
   IF buffRead [Addition+1] = ReferenceStr [1] AND Pass_M THEN
      Pass_A:= TRUE;
   END_IF
  END_IF

   IF buffRead [Addition+2] = ReferenceStr [2] AND Pass_M AND Pass_A THEN
      Cunt_M:= Cunt_M + 1;
      Pass_A:= FALSE;
      Pass_M: FALSE;
      IF (Cunt_M<=1) AND ((Addition+StrLenUDINT+2)) >Numlength) THEN
        RecCraw1:= TRUE;
        Step:=1;
        RETURN;
      ELSE
        getValue:=TRUE;
        FOR index:=1 TO StrLenUDINT BY 1 DO
          IF buffRead [Addition+index]<>ReferenceStr[index] THEN
            getValue:= FALSE;
            RecCraw1:= FALSE;
            Step:= 1;
            RETURN;
        END_IF
        END_FOR
    END_IF
END_FOR
```

图 1 Coding-Detection 算法核心代码块

Fig. 1 Coding-Detection algorithm core code block

3.4　文件数据检测步骤

文件数据检测算法主要是检测目标文件的大小，在关键字检测的基础上增加了数据检测，从而增加了对软 PLC 源码保护的有效性。该算法实现的步骤是：①没有源码的文件大小是固定的，将这个值作为一个常量；②在基于关键字检测的基础上，检查文件中的数据量是否大于固定值；③如果小于等于固定值，则表示正常下载程序，未下如源码；④如果大于固定值，则非法下载，需要删除下载的文件同时报警提示。

基于关键字和文件数据检测的详细流程如图 2 所示：

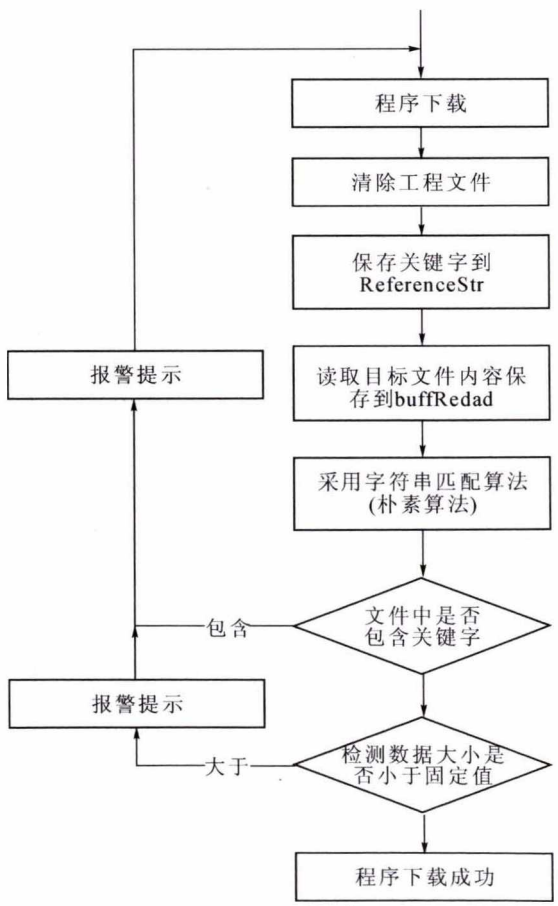

图 2 详细流程图

Fig.2 Detailed flowchart

3.5　安全分析

对于软 PLC 程序源码下载到设备中，主要风险为恶意用户可以使用软件逆向工程分析工具对软件进行逆向分析，获取软件实现技术细节，从而达到窃取和攻击的目的，以上 Coding-Detection 库检测算法实现源代码及时监测和删除，有效控制了组织源

代码的泄露。

4 结论

本文主要是基于PLC源码保护，针对以上源码安全问题进行了深入研究，针对PLC源代码下载过程中的安全问题，提出了针对PLC软件下载过程防止源代码泄露给出了详细的设计过程，并对安全性做了分析。

将ST(Structure Text)源码转化为XML(eXtensible Markup Language 可扩展标记语言)文本，提取XML文本各行标记的属性，然后根据关键字查询XML文本中该关键字出现的次数。当关键字出现的次数大于设定的次数后，就会出现提示，同时终止软件PLC程序的下载，在关键字检测的基础上还增加了文件大小的检测算法，当文件大于默认值后，也会将文件中的内容删除，保证了源码的安全性。

参考文献：

[1] 郭栋，孙峰，唐植明. 加密与解密实战攻略[M]. 北京：清华大学出版社，2013.
[2] 陈昊鹏. 软件逆向工程技术研究[D]. 西安:西北工业大学，2001.
[3] 飞天诚信. 软件加密原理与应用 [M]. 北京：电子工业出版社，2004.
[4] 看雪. 加密与解密—软件保护技术及完整解决方案 [M]. 北京：电子工业出版社，2001.
[5] 武新华. 加密解密全方位学习[M]. 北京:中力铁道出版社，2006.
[6] 陈勤，贾琳飞，张蔚. 基于代码与壳互动技术的软件保护方法研究[J]. 计算机工程与科学，2006, 28(12):36-37.
[7] 刘宗田. 软件维护与逆向工程总数[J]. 计算机工程与软件，1995(1):1-8.
[8] 李益. 基于Linux环境的源代码保护系统的研究与实现 [D]. 西安:西安电子科技大学, 2009.
[9] 刘晓冬. 软件加壳技术的研究与实现[D]. 沈阳:沈阳工业大学. 2006.