文章编号：2096-1472(2020)-12-27-03
DOI:10.19644/j.cnki.issn2096-1472.2020.12.008

基于模糊测试的工控网漏洞发现技术研究

刘 坤

(苏州健雄职业技术学院软件与服务外包学院，江苏 太仓 215411)
✉281589399@qq.com

摘 要：本文通过深入研究当前漏洞发现技术，分析常用漏洞发现技术的优劣势，以及应用领域。基于CVE漏洞库改进了现有NMAP源码设计，对其扫描流程进行改进，提升对已知漏洞扫描的能力。对未知漏洞挖掘，优化设计漏洞挖掘测试生成算法和漏洞挖掘算法，对目标工控测试系统进行网络漏洞挖掘，从而得到工控系统已知或未知的漏洞分析报告，形成工控系统安全评估报告及安全应对策略。

关键词：Fuzzing测试框架；漏洞扫描技术；漏洞发现技术；漏洞挖掘技术
中图分类号：TP312 **文献标识码**：A

Research on Vulnerability Detection Technology of Industrial Control Network based on Fuzzy Test

LIU Kun

(*Institute of Software and Service Outsourcing*, *Suzhou Chine-shiung Institute of Technology*, *Taicang* 215411, *China*)
✉281589399@qq.com

Abstract: This paper analyzes advantages and disadvantages of commonly used vulnerability detection technology and application areas through in-depth research on current vulnerability detection technology. The existing Nmap (Network Mapper) source code design is improved based on CVE (Common Vulnerabilities and Exposures) vulnerability library. Scanning process and ability to scan for known vulnerabilities are also improved. Known or unknown vulnerability analysis reports can be obtained in a few ways including mining unknown vulnerabilities, optimizing design of vulnerability mining test generation algorithms and vulnerability mining algorithms, and mining the target industrial control test system for network vulnerabilities. Industrial control system security assessment reports and security response strategies are also formed.

Keywords: fuzzy test framework; vulnerability scanning technology; vulnerability detection technology; vulnerability mining technology

1 引言(Introduction)

网络安全从过去的信息安全、计算机主机安全、信息系统安全延伸到基础设施安全、社会安全、城市安全甚至国家安全。大安全时代“什么都可以编程，什么都可以互相连接，”只要有人写程序，这将是有缺陷的。网络战的攻击往往是一个未知的时代，从未知的装备和地点发动，对于未知的漏洞，防御是不可阻挡的。

2014年，美国工业控制系统网络应急小组发布安全报告显示收集了167份工业脆弱性报告，涉及能源和制造等多个领域。2015年被ICS-CERT收录的攻击事件多达295件[1]，截止2016年11月份，中国国家信息安全漏洞共享平台(CNVD)最新统计数据显示，工业控制系统漏洞的类型呈现出多样化的趋势，包括信息泄漏、缓冲区溢出、拒绝服务等[2]。结合工业网络现状，本文深入研究工业控制系统网络脆弱性挖掘技术，挖掘工业控制系统可能存在的安全漏洞，找到消除安全隐患的方案，使其能够有效保障工业控制网络安全，对提高整个工业系统的安全性具有重要意义。

2 漏洞挖掘方案研究(Research on vulnerability mining scheme)

漏洞是指计算机体系中包含的软件、硬件和协议，在其设计和实现过程中可能存在的缺陷或错误[3]。工控系统使用之前或者使用过程中通过漏洞挖掘技术，挖掘工控系统中可能存在的漏洞，从而进行相应的防御操作，并进行后续的评估和修复，可以有效提高系统的安全性能。

目前虽然有一些成熟的漏洞挖掘方案可以直接利用，也有一些成熟的漏洞挖掘框架可以利用，但为工业网络开发完善的漏洞挖掘技术的方案并不多。虽然漏洞挖掘技术已经取得了长足的进步，但其在工业安全领域的应用仍存在各种问题。首先，工控系统使用的漏洞挖掘技术测试用例生成算法相对简单，无法根据网络协议中的每个协议字段生成多维测试用例来有效测试。其次，许多漏洞挖掘框架不能完全覆盖整个协议栈的运行状态，只能测试某些协议栈的运行路径。

本文深入研究当前漏洞发现技术，分析各种漏洞发现策略方案的优劣势，以及在工控领域应用的可行性。在此基础上，基于CVE漏洞库改进了现有NMAP源码设计，对其扫描流程进行改进，提升对已知漏洞扫描的功能需求。对于未知漏洞挖掘，优化设计漏洞挖掘测试生成算法和漏洞挖掘方案，对目标工业控制系统及网络进行漏洞挖掘，从而得到目标系统已知未知漏洞分析报告，实现工控系统安全评估报告及安全应对策略。

3 基于CVE漏洞库的漏洞扫描技术研究(Research on vulnerability scanning technology based on CVE vulnerability database)

工业控制系统漏洞是工业控制系统攻防双方关注的焦点，掌握工业控制系统漏洞扫描与挖掘分析技术，是做好工业控制安全防护的前提[4]。针对工控系统采用的漏洞扫描技术，主要是对工控系统进行信息收集和分析，形成专业工控系统漏洞指纹数据库，数据库中包括硬件设备版本、通信协议、漏洞特征信息等漏洞指纹信息。工控漏洞扫描技术是根据被测工控系统设备的类型，版本信息，通信协议等指纹信息进行检测规则的自动匹配，从而检测是否存在已知或未知漏洞。漏洞扫描技术除了进行已知漏洞特征匹配功能还包括工控通信协议支持、主机系统存活判断、主机端口扫描、服务识别、操作系统类型判断等，同时具备PLC、DSC、SCADA等工业控制系统和软件识别功能[5]。

3.1 漏洞扫描基本功能设计

漏洞扫描模块主要通过TCP SYS、TCP ACK、UDP等数据包任意组合起来发送到被探测目标主机，探测目标主机是否存活，以及目标主机活动端口的状态。漏洞扫描模块通过对远程机器端口探测，使用NMAP自建的服务组成NMAP-services数据库进行端口对比，探测目标系统的操作系统、开启端口服务情况。通过使用TCP和UDP报文，将返回结果与NMAP-OS-fingerprints[6]进行比较，查找匹配操作系统。

3.2 基于CVE漏洞库的漏洞扫描功能设计

通过对国内外著名工控系统厂商产品分析研究，我们了解到当前漏洞扫描模块公开的漏洞主要涉及的是国际著名的工业控制系统厂商。国内也有两家工业控制系统厂商进入到了前十的行列，其中北京亚控科技发展有限公司和北京三维力控科技有限公司分别公布有17个漏洞和11漏洞，其中分别被CVE收录14个和2个[7]。这里我们将CVE漏洞库加入NMAP扫描器进行漏洞扫描，通过目标工控系统与漏洞库中漏洞进行比对，预测发现目标系统中可能存在的CVE漏洞和风险。Nmap是一款非常强大的开源扫描工具，Nmap扫描器不仅可以针对操作系统进行扫描还可以针对工控系统上位机进行扫描。Nmap扫描器主要工作由文件nmap.cc完成。它的main.cc文件主要工作是负责包装nmap_main()函数，而nmap_main()函数是扫描器执行流程的核心函数。改进前NMAP扫描器主要工作流程为：

(1)准备阶段：进行参数解析、资源分配、基本扫描信息输出、端口和地址列表初始化、NSE环境准备、前置脚本操作等基本准备操作。

(2)工作阶段：然后进入主循环，每个主循环会对一组目标地址执行相同操作如主机发现、端口扫描、服务扫描、版本探测、操作系统探测和漏洞扫描等，直到所有设置目标地址都被扫描。

(3)恢复阶段：在完成所有扫描探测工作后，脚本调用相应的处理程序，打印出扫描报告，并释放分配的资源。在NAMP扫描器基本工作流程基础上，我们通过编写脚本实现基于CVE漏洞库扫描流程，设计专门针对工控系统扫描流程，改进设计后NMAP扫描器扫描流程如图1所示。

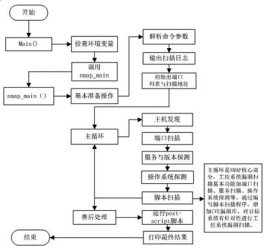

图1 NMAP扫描器扫描流程

Fig.1 Nmap scanner scanning process

4 基于Fuzzing测试框架的漏洞挖掘技术研究(Research on vulnerability mining technology based on fuzzy testing framework)

4.1 漏洞挖掘算法优化设计

目前针对工控系统漏洞挖掘主要集中在操作系统类型、工控协议、文件类型、Web界面、后台数据库、网络协议、移动应用程序等方面。模糊测试技术主要应用于操作系统、工控协议、主动控制、文件格式、数据库这些方面的漏洞挖掘。模糊测试技术针对工控系统通信协议漏洞挖掘方法是先分析协议特点，根据协议特点来构造特定的数据包，将数据包发送到工控上位机服务器或者下位机，监控被测目标响应情况，最后根据响应异常进行漏洞挖掘。

针对工控协议的模糊测试，需要深入理解每个工控协议的规定特征，生成输入数据和测试用例去遍历协议，根据数据内容、结构、消息、序列等出现各种异常进行漏洞挖掘。同时，漏洞挖掘过程中需要引入大数据分析和人工智能算法，将初始数据主要集中在某类工控设备最容易发生故障的范围进行密集测试，测试中需要动态跟踪设备的异常反应，

人工智能算法选择有效地输入属性构造新样本进行测试。

当前常用的模糊测试技术一般只能针对通信协议的某个字段进行变异的缺陷，设计测试用例算法，这类设计的局限性是协议如果其他字段出现变异缺陷算法并不能发现。针对这个问题我们设计可以针对数据内容、结构、消息等多字段变异的测试用例生成算法，从而大大提高漏洞挖掘的效率和正确率。对测试用例生成算法进行优化改进后具体算法描述如下：

(1)根据需要测试的工控协议，进行协议分析，确定协议字段集合P，特殊数据集合M，多个字段测试用例生成向量G和对应的协议字段特殊数据生成矩阵S。

(2)根据G选取协议测试字段集合。

(3)针对选定协议字段集合中的协议字段P_i，分别按顺序选取特殊数据生成矩阵S_i，得到对应的特殊数据矩阵。

(4)整理特殊数据，生成可以用的测试用例q，并加入测试用例集合Q。

(5)同样的方法去遍历选定协议所有字段集合生成测试用例。

(6)遍历结束后，输出测试用例集合，完成测试用例生成。

根据改进算法的测试用例，在Fuzzing框架对工控系统进行漏洞挖掘，可以有效避免遗漏随机变异所触发的安全漏洞。

4.2　基于Fuzzing测试框架的漏洞挖掘技术研究

漏洞挖掘技术是软件安全领域的一个重要研究方向。通常漏洞挖掘技术分为动态分析、静态分析、模糊测试等。随着软件规模的爆炸式增长，软件的漏洞挖掘实现越来越困难。Fuzzing技术作为一种具有自动化程度高、误报率低、且不依赖于目标程序源代码等优点的挖掘技术，成为当今漏洞挖掘领域的主要技术。

模糊测试技术是一种常用的自动化漏洞挖掘技术，它起源于软件测试中的黑盒测试技术。通过编写模糊工具为目标程序提供某种形式的输入并在程序运行过程中监视异常，通过记录导致异常的输入数据来进一步定位软件中的缺陷。Fuzzing测试技术通常使用边界值附近的值对目标进行测试，通过对现有Fuzzing测试框架成果研究，总结出目前测试方法存在的主要问题：

(1)Fuzzing测试具有一定的盲目性。针对常用工控协议的测试，仍然没有解决测试用例路径重复的问题，导致测试效率较低。

(2)Fuzzing测试用例冗余度较大。由于目前采用测试策略多是随机策略，导致有一定概率产生重复或相似的测试用例。

(3)Fuzzing测试关联字段的针对性不强。目前多数Fuzzing测试方法，只能针对多个协议字段进行数据的随机生成或变异，缺乏对协议中各关联字段的针对性。

根据Fuzzing测试存在的问题，通过对漏洞扫描流程改进以及漏洞挖掘算法进行优化，设计改进的Fuzzing测试框架：

(1)基于工控系统常见漏洞数据生成测试用例。通过分析工控系统常见漏洞，针对不同工控协议的各字段，有针对性的进行数据改变，缩小测试范围。由于测试用例直接与漏洞相关，因此更加具有针对性。

(2)生成的测试用例样本以其文件内容的MD5值命名，保证测试用例的唯一性，尽量避免重复或相似的测试用例，通过减少测试时间提高测试效率。

(3)将改进优化后的生成用例算法运用到Fuzzing框架，根据不同的协议，针对多个字段进行同时变异，提高代码的覆盖率和测试样本的有效性。

目前常用的模糊测试生成算法一般只能针对某个字段进行测试，存在严重的缺陷。本文通过深入研究协议中各字段关系，设计针对数据内容、消息、结构等多字段变异测试用例生成算法，这样可以大大提高工控系统漏洞挖掘效率。测试用例生成算法进行优化改进后，具体算法设计流程图如图2所示。

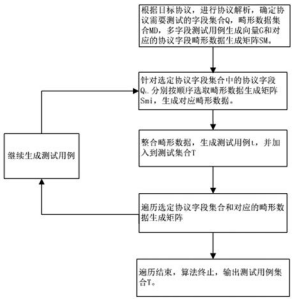

图2 测试用例优化算法

Fig.2 Test case optimization algorithm

由于Fuzzing在测试过程存在的缺陷，采用优化后的多字段变异测试用例生成算法对目标程序进行测试，记录异常数据时采用MD5记录样本，避免样本重复，改进后Fuzzing测试框架如图3所示。

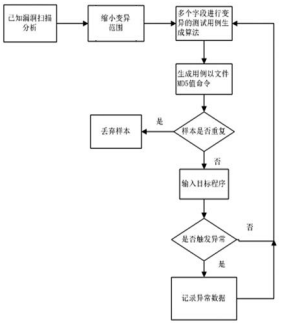

图3 改进后Fuzzing测试框架

Fig.3 Improved fuzzy testing framework

(下转第23页)