未知协议的逆向分析与自动化测试

张蔚瑶[1),3)] 张 磊[2)] 毛建瓴[1)] 许智君[4)] 张玉军[1),3)]

[1)](中国科学院计算技术研究所网络技术研究中心 北京 100190)

[2)](河北工业大学 天津 300019)

[3)](中国科学院大学 北京 100049)

[4)](北京邮电大学 北京 100876)

摘 要 在工业控制、军事通信、金融信息等创新型网络中,大量未知(私有或半私有)协议被广泛采用.对通信协议及其实现进行严格的测试是确保网络系统安全性的重要手段,现有测试手段与方法大多只能针对已知协议进行,未知协议的广泛采用对协议测试提出了挑战.本文提出了针对未知协议的逆向分析与自动化测试方法,其基本思想是基于对协议流量的逆向分析,识别出协议特征,动态生成多维测试数据,自动监控被测系统的运行状态,获得准确的测试结果,为系统安全可靠运行提供依据.具体贡献包括:(1)自动化模糊测试框架;(2)基于协议特征库的逆向分析方法;(3)基于多维变异的测试数据生成方法;(4)基于主动探测的测试执行与异常定位方法.本文设计实现了自动化测试工具UPAFuzz,试验结果表明,UPAFuzz能够基于网络流量实现协议特征的自动识别,并自动生成海量模糊测试数据,对被测系统进行测试;在生成的测试数据量达到千万级时,UPAFuzz的内存占用率为现有模糊测试工具Boofuzz的50%,且其耗时仅为Boofuzz的10%,大大提升了测试执行效率.

关键词 未知协议;逆向分析;特征识别;协议特征库;多维变异;主动探测

中图法分类号 TP393.0 **DOI号** 10.11897/SP.J.1016.2020.00653

An Automated Method of Unknown Protocol Fuzzing Test

ZHANG Wei-Yao[1),3)] ZHANG Lei[2)] MAO Jian-Ling[1)] XU Zhi-Jun[4)] ZHANG Yu-Jun[1),3)]

[1)](*Internet Technology Research Center, Institute of Computing Technology, Chinese Academy of Sciences, Beijing* 100190)

[2)](*Hebei University of Technology, Tianjin* 300019)

[3)](*University of Chinese Academy of Sciences, Beijing* 100049)

[4)](*Beijing University of Posts and Telecommunications, Beijing* 100876)

Abstract Nowadays, a large number of unknown (private or semi-private) network protocols are widely adopted in newly emerging network, such as industrial control, military communications, as well as financial information, etc. Making sure the protocol goes through a set of strict tests for both design and implement before the deployment is crucial for the usability and security of network systems. To the best of our knowledge, the majority of the existing protocol test toolkits or systems is only able to be applied to known protocols, i. e. the testers know how the examined protocol works. As a direct consequence, the prevalence of unknown protocols poses a great challenge to current protocol test systems. Therefore, before we can transplant exiting test methods for known protocols to unknown ones, there are many research problems to be noticed, and among those problems, three

收稿日期:2019-06-23;在线出版日期:2020-02-07.本课题得到国家重点研发项目(2016YFE0121500,2018YFB1800403);国家自然科学基金项目(61902382, 61972381, 61672500, 61572474);中国科学院战略性先导科技专项(XDC02030500)资助.**张蔚瑶**,博士研究生,中国计算机学会(CCF)会员,主要研究领域为网络安全与测试.E-mail:zhangweiyao17z@ict.ac.cn. **张 磊**,教授,中国计算机学会(CCF)会员,主要研究领域为智能系统的理论和工程研究. **毛建瓴**,硕士,主要研究领域为网络安全与测试. **许智君**,博士后,主要研究领域为网络安全与测试. **张玉军**(通信作者),中国计算机学会(CCF)高级会员,主要研究领域为互联网安全评估和验证,E-mail:nrcyujun@ict.ac.cn.

of them are most unignorable: First, the current test is unable to estimate the architecture and semantic characteristics for unknown protocol with the network sniffer or manual inspection, which make it difficult to obtain necessary knowledge for later tests. Second, the prevailing test data generation methods are proved to be of low-hit-rate and inefficient, and the existing single-field random filling method for generating test data lacks vulnerability mining capabilities. Furthermore, due to the unknown characteristics of the protocol, it is impossible to accurately construct the data required for testing. Last but not least, the network devices running the unknown protocols are usually strictly concealed, which means that it is impossible to install the monitor proxy programs in the devices under test, which is crucial for current test systems designed for known protocols. To address above issues, we propose a novel automated fuzzing test framework for unknown protocols. The workflow of our framework is as follows: 1. precise identification of the protocol features based on the protocol reverse analysis, 2. dynamic generation of multi-dimensionally mutated test data, 3. automatic monitor for the running state of the devices under test to make sure the accuracy of the test outcome and secure the systems. Our main contributions can be concluded as follows: First, we design an automated fuzzing test framework for unknown network protocols. Second, we propose an automated reverse analysis method for unknown protocols by virtue of the novel protocol feature database. Third, we propose an innovative method to mutate test data in a multi-dimensional way. Last but not least, we present a set of active-detection methods for the test execution, following inspection and analysis. In addition, we develop UPAFuzz, an automated fuzzing test tool, and according to the experiment outcomes, it is proved that UPAFuzz can analyze characteristics of unknown protocols based on the protocol network traces and generate massive data for later test with high hit rate and low time cost. Moreover, Compared to Boofuzz, a popular open-source fuzzing test tool, UPAFuzz's memory usage is 50% of that of Boofuzz, and the time consumption for generating tens of millions of test data is only 10% of Boofuzz, which greatly improves the test efficiency and with certain versatility.

Keywords unknown protocols; reverse analysis; features identifying; protocol feature database; multi-dimensional mutate; active-detection

1 引 言

在工业控制[1]、军事通信[2]、互联网金融[3]等创新型网络中,大量未知(私有或半私有)协议被广泛采用. 而未知协议不仅复杂多样,且大多在安全性设计上欠缺考虑,并未经历过大规模的测试和改进,存在许多未知的漏洞[4]. 另外,其未知性对现有网络安全监管亦是一大挑战. 因此,研究未知协议的安全问题,保障整个网络环境的安全刻不容缓.

协议的安全性测试通过对协议的实现组件或使用设备进行测试,发现其存在的漏洞,是确保网络系统安全性的重要手段[5]. 常用的安全性测试方法如:静态分析[6]、漏洞特征扫描[7]、模型检验[8]等,往往依赖被测设备的可执行代码或内部逻辑,而未知协议的应用设备运行在封闭环境中,无法满足以上条件.

模糊测试是一种通过向测试目标提供畸形的输入,分析程序运行的错误来挖掘被测对象漏洞的测试方法[9]. 模糊测试不依赖源代码和内部逻辑,是未知协议的安全性测试的有效方法. 协议的模糊测试重点在于充分了解的协议结构和语义特征,构造能够与被测设备交互的畸形数据包,执行并监测其运行状况. 因此,现有的网络协议模糊测试框架大多针对于已知协议,依靠人工或网络嗅探器等工具获取协议相关知识,依据协议知识逐一对单个字段随机填充生成畸形数据后发送给被测设备,并借助附加在被测设备上的调试器等监测代理程序进行监测[9],若直接应用到未知协议中,面临三个重大挑战:首先在知识层面上,现有框架无法通过人工或网络嗅探器获得未知协议的结构和语义特征,限制了

测试所需的协议知识来源;其次,在数据层面,缺乏高效的数据生成方法,现有的单字段随机填充生成测试数据的方式命中率低且漏洞挖掘能力欠缺,更进一步来讲,由于协议特征的未知亦无法准确地构造测试所需的数据;最后,在执行层面,运行未知协议的网络设备不允许加载监测代理程序,无法实现对于测试运行情况的监测.

针对目前存在的严重问题,本文设计了针对未知协议的新型模糊测试框架,在知识层面、数据层面以及执行层面均做了根本性的改变,其基本原理如下:考虑到人工获取未知协议相关知识的局限性,采用逆向分析方法实现协议知识的自动学习;考虑到单字段随机填充的测试数据生成方式效率低下,使用更为高效的多维变异方式生成测试数据,具有更好的漏洞检测能力;考虑到难以在被测设备中加载监测代理程序,使用主动探测的方式对被测设备进行监测并精确定位触发异常的数据.具体方法和贡献如下:

1) 知识层面:提出基于协议特征库的未知协议的逆向分析方法,自动学习协议结构和语义特征.进一步讲,本文给出了一套完整的协议特征推断方法,并建立了协议特征库,为协议特征定义了统一的描述规则,以及提出了新的比对算法BDTW,用于协议特征的比对,使得协议特征的推断精度趋近最优.

2) 数据层面:提出多维变异的测试数据自动化生成方法,参考常见漏洞的攻击模式,为不同的协议字段类型定义不同的数据生成规则,并以多维变异的方式生成测试数据,在提高测试数据的命中率的同时,提升数据的漏洞挖掘能力;

3) 执行层面:提出基于主动探测的测试执行与异常定位方法,不同于传统测试在被测设备上加载监测代理程序,本文在被测对象运行时主动探测其运行情况,并提出循环执行法用于异常发生后精确定位触发异常的数据.

基于上述方法,本文设计实现了自动化模糊测试工具UPAFuzz.实验证明,UPAFuzz可以在没有任何先验知识的情况下,基于网络流量准确地逆向分析出协议的特征,自动生成海量的测试数据,完成对多种不同类型协议的自动化模糊测试.此外,与现有的广泛应用的模糊测试工具Boofuzz[10]相比,UPAFuzz在生成等量的千万级测试数据时,UPAFuzz的内存占用率为Boofuzz的50%,其耗时仅为Boofuzz的10%,大大提升了测试的效率.

本文的第二节介绍了未知协议及其模糊测试的背景知识和相关工作,第三节针对现有框架的存在问题,详细介绍解决问题的方案,包括未知协议模糊测试的框架和关键方法.第四节对本文系统UPAFuzz进行实验测评;最后在第五节对本文的工作进行总结.

2 相关工作

本章首先分析了未知协议及其安全测试的难点.随后对现有模糊测试框架做简单介绍,列出现有框架应用于未知协议的测试所面临的主要问题,总结了模糊测试改进工作的不足之处,并对新框架设计的提出了要求.最后详细介绍了协议逆向工程及其研究工作.

2.1 未知协议应用与测试需求

互联网与传统行业的融合创新,促进了各行各业的革命与升级,同时也带来众多安全问题.随着各类创新型的网络应用的涌现,网络协议结构也在不断变化,为满足不同的需求,针对不同应用场景的自定义网络协议出现.为保证私密性和个性化,大多数自定义协议相关知识不公开,成为未知协议.目前,未知协议应用广泛,尤其是肩负一线测量与控制的工业控制网络领域[1],信息化对抗的军事网络领域[2],以及涉及国计民生的金融信息领域[3]等等.据全球安全研究中心统计,骨干网流量中未知协议流量已经达到总流量的45%[11].未知协议在最初时被应用于物理隔离或私有的网络中,因此在设计时往往聚焦于实际应用的功能和性能,而在安全性设计上有所欠缺,且没有经历过大规模的测试和改进,导致协议本身存在诸多安全隐患.更进一步说明,应用在大规模分布式控制系统中的工控协议如OPC、ProfiNet等,在设计时更多地注重其控制功能而放弃了其安全特性.更有甚者,未知协议的“未知性”给网络安全监管带来很大的挑战[12],协议资料不公开、协议形式复杂多样、协议运行设备内部结构私密等特点导致对未知协议的安全测试工作难以展开.因此,保证未知协议的安全性是各个应用领域亟需解决的问题,也是当下研究的热点[4].

对协议及其实现进行严格的测试是确保其安全性的重要手段.而现有测试手段与方法大多针对已知协议进行,如:静态分析[6]、漏洞特征扫描[7]、模型检验[8]等.结合未知协议的特点,并不能适用于未知协议的测试.而模糊测试能够不依赖源代码和内部

逻辑,通过向被测设备直接发送畸形数据,挖掘其内部的漏洞[9]. 模糊测试的适用对象众多,包括web应用程序、文件、网络协议、内存数据等等,其中网络协议的模糊测试被广泛应用. 因此,对于未知协议而言,模糊测试是其安全性测试的有效方法.

2.2 模糊测试

针对网络协议的模糊测试框架,需要解决三个层次的问题:一是知识层面,获取被测协议的结构和语义特征,在本文中简称为协议特征,这是测试数据生成的根本条件;二是数据层面,根据知识层面获取的协议特征,生成用于测试的畸形数据;三是执行层面,将数据层生成的数据发送给被测设备,并监测被测设备的运行情况.

图1体现了现有的网络协议的模糊测试框架的完整结构,其依赖人工或网络嗅探器等工具对协议进行解析,基于解析结果将单字段随机填充成畸形数据发送给被测设备,并在被测设备上附加监测代理程序监测其运行情况.

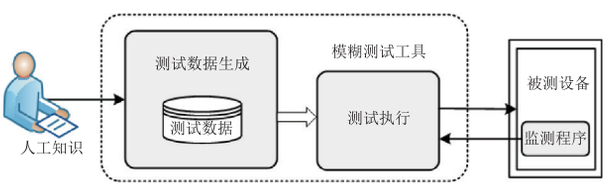

图1 现有的模糊测试框架图

总的来说,网络协议的模糊测试建立在充分了解协议知识的基础上. 但是对于未知协议来说,协议知识不公开,这意味着人工无法获得未知协议的特征;再者,现有单字段随机填充的测试数据生成方式不能够保证数据可以正确地与被测设备交互,更不能保证数据能够高效地挖掘出设备的漏洞;除此之外,运行未知协议的网络设备基本不支持监测代理程序的额外附加,无法对测试状态进行监测,进而无法对触发异常的测试数据进行定位. 以上三点导致现有框架无法对未知协议进行有效的模糊测试.

为提高网络协议模糊测试的效率和效果,许多研究人员致力于改进现有的模糊测试. 框架的改进工作大致分为三个方面:一是更高效的协议知识的获取方式:AspFuzz[13]直接更改协议数据的发送顺序和格式,但仅限于应用层的协议;Yamel等人[14]研究的成果可以对较短的文本协议进行逆向,但对如何进行模糊测试以及监测未做介绍;二是提升测试数据生成效率:Aitel[15]、Vuagnoux[16]、Eddington等人[17]将协议扁平化,减少数据生成空间的复杂性. IOTFUZZER[18]使用应用程序的逻辑生成有意义的测试用例. 三是提升测试数据漏洞挖掘能力,SPFuzz[19]使用不同的变异策略来模糊协议数流量的不同层次,但无法对未知协议的层次进行划分;SaGE[20]、VUzzer[21]等借助程序分析保证数据的代码覆盖率,但是这不适用于程序不可得的网络设备.

目前已有许多针对协议的模糊测试工具如:SPIKE[15]、Peach[17]、Sulley[22]等,但其运行的前提是协议特征的描述,并借助监测代理程序实现监测功能;Zhang等人[23]改进了Peach使其可以模糊基于MAC层的协议,但仍未解决上述问题. 通用模糊测试器GPF[24]直接将正确的数据样本进行变异,导致测试数据空间巨大且测试数据的命中率低;在GPF基础上,EFS[25]结合了逆向工程和调试框架PaiMei[26]实现协议的模糊测试,但是EFS只适用于文本协议,无法进行更复杂的二进制协议的测试. 除此之外,还有许多专用的协议模糊工具,如:ProFuzz[27]可以对工控协议profinet协议族进行测试. T-Fuzz[28]可以为电信协议进行测试,但专用的工具不具备通用性.

目前的模糊测试框架和工具的改进工作,均无法对未知协议进行有效的测试,因此本文对未知协议的模糊测试框架的构建提出了以下要求:1)能够在没有协议先验知识的前提下,自主学习协议特征,建立协议模型;2)能够在1)所建立的协议模型的基础上高效地生成高命中率的测试数据;3)能够在不附加调试器等监测代理程序的情况下监测被测设备的运行情况,并且能够精确定位触发异常的数据. 除此之外,还进一步要求在提高测试效率基础上具备良好的通用性.

2.3 协议逆向工程

协议逆向工程能够以网络流量或指令执行轨迹为分析对象,推断协议的特征[29],对于入侵检测系统[30]、僵尸网络对抗[31]等众多领域具有重要意义. 应用在模糊测试中,完美地解决了人工无法获取未知协议相关知识的难题. 传统的协议逆向工程为人工操作,耗时长且效率低下. 而自动协议逆向工程可以自动提取协议相关知识,提高协议分析的效率. 基于网络流量的自动协议逆向工程能够在没有先验知识的情况下,由数据序列得到协议的相关特征. 使用方法大致分为序列对比、数据挖掘、加入程序分析三大类.

序列对比算法在协议逆向中的应用是

Beddoe[32]的PI项目提出的,PI采用序列比对算法提取协议的结构特征.但该种方法适用于结构简单的协议,为解决此问题,Cui等人[33]提出Discover,利用递归聚类和基于类型的序列对比来逆向出协议特征,但该种方法无法避免先验知识的引入,且准确率有待提高.为此,Netzob[34]使用渐进多序列比对算法提取协议的特征,但Netzob需要观察应用程序的执行.

ProDecoder[35]将数据挖掘方法用于协议特征的提取,利用n-gram之间的潜在关系来推断协议的格式.Luo等人[36]采用Apriori算法建立关键词序列来推断协议的结构特征,但该种方法仅限于应用层协议.另外,众多数据挖掘方法[37]LDA、KS、t-test、Viterbi、AprioriTID都可以对协议特征进行提取.但数据挖掘方法一次使用所有消息作为输入,在选择待处理流量和生成测试数据方面计算成本很高.

Lim等人[38]从应用程序的输出函数中提取协议格式,Wondracek[39]、Comparetti[40]、cho等人[41]通过观察程序处理协议的方式来逆向协议特征;NDroid[42]可以捕获数据流进行程序的动态污染分析.但是以上方法均存在程序不可用的风险.

总的来说,协议逆向虽方法众多,但应用在未知协议上存在诸多问题,且大多注重于协议格式的划分,对于协议的语义识别很少提及,更重要的是,目前并未形成一套完整的针对未知协议的逆向分析方法.

3　未知协议的分析与测试

结合第二节中提出的问题,本文舍弃了现有的测试框架,设计了一种新型的针对未知协议的自动化测试框架,并在框架的指导下提出了实现未知协议自动化测试的一系列方法,接下来,本文将对框架及方法进行详细地介绍.

3.1　测试框架

本文提出的未知协议的自动化模糊测试框架在三个层次上均做了根本性的改变.

如图2所示,在知识层面,鉴于现有框架中人工或借助网络嗅探器等工具无法获得未知协议的相关知识,本文提出使用自动化的方法对协议流量进行逆向分析,自动学习出协议的结构和语义特征;在数据层面,考虑到单字段随机填充的测试数据生成方式限制了数据的命中率以及漏洞挖掘能力,本框架根据逆向得到的协议特征动态生成多维变异的测试数据,提高测试数据的命中率和漏洞挖掘能力.在执行层面,为了克服私密的网络设备无法附加监测代理程序的问题,框架采用主动探测的方式,对被测设备进行自动监测.

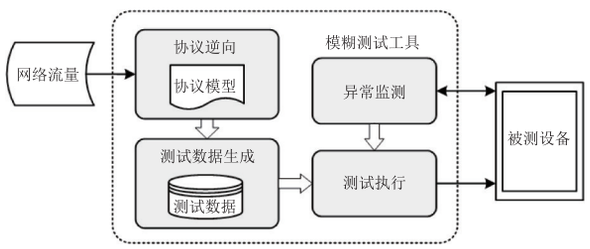

图2　未知协议的模糊测试框架图

本文使用一个有限状态机来描述整个框架,状态机如图3所示:

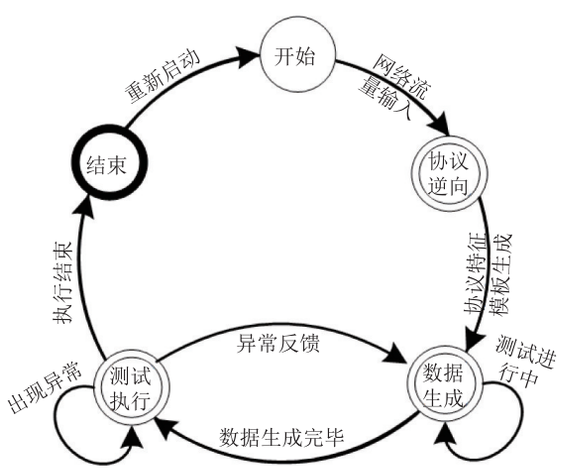

图3　未知协议自动化测试状态机

首先,模糊测试的执行需要以足够详细的协议知识为基础,即使用本文提出的基于协议特征库的逆向分析方法,基于协议流量自动学习出协议的特征,构建协议的模型.与传统方法相比,我们不仅进一步完善了协议特征的推断方法,而且创建了已知协议特征库,为协议特征定义统一的描述规则,使用本文提出的比对算法BDTW,提升被测协议特征的推断精度.

在协议特征模型构建完成后,测试进入数据生成状态.在该状态下,使用更为高效的测试数据生成方法生成多维变异的测试数据.不同于现有的单维度随机填充的数据生成方式,本文不仅为不同字段定义不同的数据生成规则,而且在数据生成时参考常见漏洞的攻击模式,选取最容易触发漏洞的多个字段进行组合后同时变异,生成多维变异的测试数据,使得测试数据的命中率和漏洞挖掘能力趋近最优.

数据生成完毕后,测试进入执行状态.执行状态下,框架一方面执行测试数据,另一方面定时发送探测包用来主动探测被测设备的运行情况.若无异常情况发生,则测试一直保持执行直到数据执行完毕后结束,若出现异常,使用本文提出的循环执行法对触发异常的数据进行精确定位,并根据异常的反馈结果调整数据的生成规则,保证模糊测试的有序执行.

接下来,本文将对提出的方法进行详细地阐述.

3.2 协议逆向分析

协议的逆向分析是基于协议流量自动学习出协议特征,生成协议的模型作为模糊测试数据生成的根本条件和重要依据.然而目前尚未形成一套适用于未知协议的协议逆向方法.因此,在本章节中,我们首先进一步完善了协议特征的推断方法,在有效识别字段格式的基础上,保证了协议的语义正确性.与此同时,为了使得协议特征推断的精度最大化,本文创新性地构建已知协议特征库,为协议特征定义了统一的描述规则,并设计了新型比对算法BDTW,用于比对出协议特征库中与被测协议相似度最高的已知协议,以指导被测协议的描述.如图4所示,协议特征的推断与协议特征库的比对形成一套完整的未知协议逆向分析方法.其中,协议特征的推断过程可以完全实现自动化,在人工收集已知协议完成构建之后,协议特征库的比对也可以实现高度的自动化.下面本文将分别对二者进行详细地介绍.

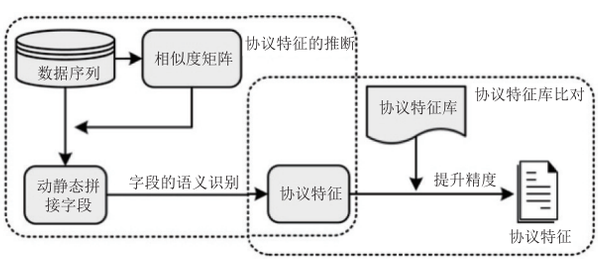

图4 基于协议特征库的协议逆向分析

3.2.1 协议特征推断

目前的协议特征推断更多地聚焦于协议的格式,并未考虑协议字段的语义特征,依据此种推断结果不能保证协议数据在真实应用场景中的有效性.因此,本文将语义识别引入协议特征的推断中,进一步完善了协议特征的推断方法.如图5所示,协议特征的推断是在数据序列的对齐、协议字段的划分的基础上,加入了语义识别的过程.更进一步讲,我们在大量调研工作基础上明确了需要识别的重要字段,并针对不同类型的字段设计了不同的识别方法.协议特征推断的过程如下:

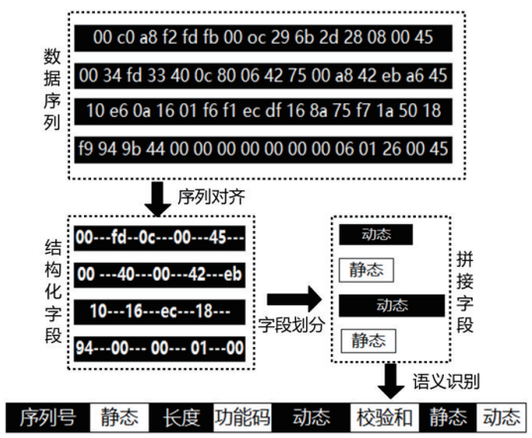

图5 协议特征的推断

数据序列的对齐.采用经典的序列对比算法[32]:首先使用Smith Waterman算法对数据序列两两进行比对,得到对齐的最长公共子序列以及相似度矩阵.在此基础上,依据相似度矩阵,采用UPGMA算法对数据序列聚类,得到协议树.从协议树的叶节点开始进行渐进序列比对,直到根节点结束.比对过程中标识每个序列的空位,得到标准对齐的带空位的数据序列.

协议字段的划分.该阶段实现标准对齐的数据序列划分为字段块.按照协议的内容是否为连续可打印的ASCII字符分为文本协议和二进制协议.对于文本协议,按照分隔符进行字段的划分,得到若干文本字符段,字段长度为两个分隔符之间数据长度;对于二进制协议依据变化率进行划分,取上一步所得的标准对齐的数据序列,逐字节计算变化率,变化率近乎为0的字节组成静态字段,以静态字段为界划分数据序列,得到一系列静态与动态字段的拼接序列,字段的长度为分割后对应数据块的长度.

字段的语义识别.字段的语义识别包括分析哪些重要字段的语义需要识别,以及为不同字段的设计不同的识别方法.由大量的调研工作[35,40,43-45]可得,协议有着通用的格式框架,即协议号、序列号、长度、校验和、文本字段等一种或几种重要字段组成了协议的通用框架,其语义特征如表1中所示.若测试可以准确分析出数据包中存在的以上字段,即可重组和填充形成规范的数据实现测试.当然,对于未知协议的其他自定义字段,其携带的控制信息将会

表1　字段的语义识别

字段名称	字段特征	识别方法
序列号字段	靠近段首,取值随时间递增	取值随时间递增
长度字段	描述数据包某一部分的长度	循环比对法
文本字段	文本字段较少,一般蕴含着丰富的信息	连续可打印字符
校验和字段	验证输入的完整性,取值有限	循环比对法
功能码字段	指示应用对象的操作,取值较固定	循环比对法

在3.3.1节提出的数据生成规则下同样得到保持,从而保证协议信息的完整性.

明确需要识别的字段后,我们根据字段的特征定义两类识别方法.首先,对待识别的字段进行分类:一类为取值无限的字段:①保证数据包的传送顺序的序列号字段,通过判断其取值与序列的时间顺序是否成正相关进行识别;②蕴含着重要的信息的文本字段,通过是否是连续可打印的ASCII字符进行识别;另一类是取值为有限集合的字段:如长度字段、校验和字段、功能码字段等取值为有限集合的字段,使用本文提出的循环比对法进行识别.

循环比对法的定义如下:

1) 根据待识别字段F_i(i∈[0,n])特征计算该字段所有可能取值,保存为可能取值集合P_i.

2) 遍历被测协议推断出的所有字段,记录每个字段的取值,并检验该取值是否与P_i中的可能取值匹配.

3) 若匹配,当前字段设为候选待识别字段.

如图6,以校验和字段为例:首先,根据常用的校验和算法CRC校验、异或校验、奇偶校验等,计算并保存校验和可能取值.接下来,逐一遍历上一步所得的未知字段并与可能取值比对,若匹配,则该字段为校验和候选字段.

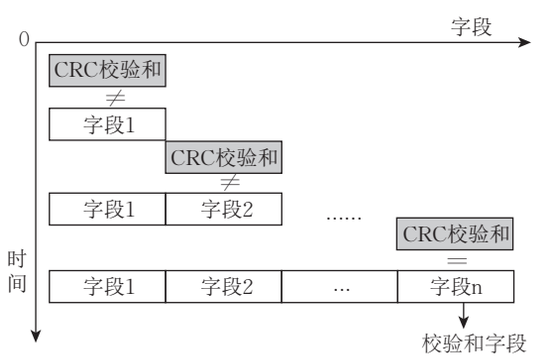

图6　校验和字段的识别

3.2.2　协议特征库的比对

为了使协议特征的推断精度趋近于最优,本文创新性地构建已知协议的协议特征库,为协议特征定义了统一的描述规则,并提出了新型比对算法BDTW用于比对出协议特征库中与被测协议最相似的已知协议特征,利用二者之间的相似性来提升未知协议特征的推断精度.协议特征库采用如下方式构建:

1) 收集不同协议簇(如:TCP/IP协议簇,Modbus协议簇)中已知协议的真实协议特征描述T,以及相应的网络流量;

2) 以已知协议的网络流量为输入,使用本文的协议特征推断方法,得到推断的协议特征L.

3) 将真实协议特征T与推断的协议特征L,以及二者的映射关系保存为协议特征库.

在协议特征库中,本文为协议特征定义了统一的的描述规则:协议特征以字符串的形式存在,不同字符代表协议的不同字段,字符个数为字段所占字节数.在本文中,这样的字符串称为特征序列.如图7所示,工控领域标准通信协议Modbus/TCP协议字段为:序列号,长度,地址,功能码和变长的数据字段,其特征序列为*SSCCLLVF**.其中,S定义为序列号,C定义为静态字段,L为长度字段,V为动态字段,F为功能码字段,*为变长字段.相同字符的个数表示相应字段所占字节数,字符的顺序与协议中字段顺序相匹配.

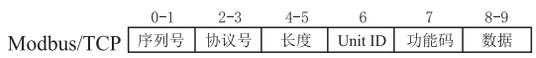

图7　Modbus/TCP协议特征

为了找出与被测协议最相似的已知协议,需要比对二者相应的特征序列.如图8所示,比对时,本文使用推断出的已知协议特征序列L与被测协议特征序列U进行比对.若二者之间的相似度在合理的范围之内,则用最相似已知协议对应的真实协议特征T,精确地描述被测协议;若相似度超出合理范围,则不对协议特征的推断结果做出改变.

衡量特征序列之间的相似度的方法,是本文提出的基于布尔值的DTW算法BDTW.BDTW算法继承了DTW算法的基本思想:把未知量伸长或缩

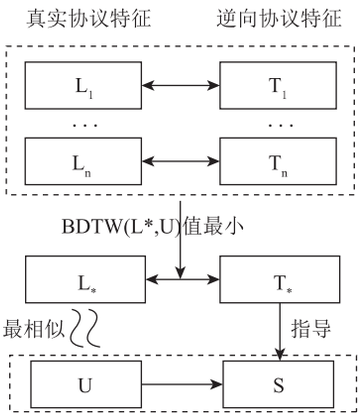

图8 协议特征库的比对

短,直到其特征量与标准模式对应.即未知量长度为I,参考模板长度为J,且I和J不等,寻找一个规整函数$j=w(i)$,它将未知量的元素i非线性地映射到参考模板J的元素j上,并使该函数$w(i)$满足:

$$D=\min_{w(i)}\sum_{i=1}^{I}d\left[T(i),R\left(w(i)\right)\right] \tag{1}$$

其中,D称为代价函数,得到的是最优调整下所有元素的总累计距离.算法的目的是使匹配样本之间的距离最小,因此该算法保证了两组特征量之间有着最大相似性.$d\left[T(i),R\left(w(i)\right)\right]$是未知量的第$i$个元素$T(i)$与参考模板第$j$个元素$R(j)$之间的距离,对于特征序列来说,序列的取值不具备顺序性,因此本文使用布尔值计算两组特征量之间的距离,即将序列的匹配得分定义相同为1,不同为0.具体的距离计算函数,即代价函数$d\left(T(i),R(j)\right)=g(i,j)$如下:

$$g(i,j)=min\begin{cases}g(i-1,j)+2d(i,j-1)+d(i,j)\\ g(i-1,j-1)+2d(i,j)\\ g(i-2,j-1)+2d(i-1,j)+d(i,j)\end{cases} \tag{2}$$

3.2.3 基于BDTW的协议特征库对比算法

用于提升协议特征推断精度的基于BDTW的协议特征库对比算法如算法1.其中,算法的输入为已知协议特征序列库:*FeatureBase*,未知协议特征序列:U,相似度阈值:*threshold*;其中,特征库中包含已知协议真实特征序列:T,已知协议推断后的特征序列:L;算法的输出为BDTW距离集合D_{List},未知序列的最终逆向结果特征序列S.

算法1. 基于BDTW的协议特征库对比算法

Require: *FeatureBase*, T, L, U, *threshold*

Ensure: D_{List}: S;

1. function COMPARISON(*FeatureBase*, U, *threshold*)
2. *FeatureBase*←{(T_i,L_i)|i∈(0,n)}
3. for each $L_i\in L$ do
4. D_i=BDTW(U,L_i)
5. write D_i to D_{List}
6. end for
7. D_{min} is the min of D_{List}
8. *minIndex*=min_Arg(D_{List})
9. if D_{min}<*threshold* then
10. $S=T_{minIndex}$
11. else
12. $S=U$;
13. end if
14. end function

计算每一个已知协议推断出的特征序列L_i,与被测未知协议的特征序列U之间的BDTW距离$BDTW\left(U,L_i\right)$,并记录在距离数组D_{List}中.取D_{List}最小值D_{min},若:

1) D_{min}小于距离阈值,则按照已知协议的真实特征描述该未知协议.

2) D_{min}大于阈值,不做处理,保持现有协议特征的推断结果.

其中,受到Halfaker[46]的启发,阈值的求解方式为:取所有已知协议推断出的特征序列,以所属的协议簇为边界,计算两两序列间的BDTW距离.将所有计算出的BDTW距离值拟合成一个有两个分量的混合高斯模型,两个分量的交点即为阈值.

网络协议的三要素是语法、语义与时序.至此,语法和语义的逆向本文均已实现.但对于应用于军事、工业控制等功能性领域的未知协议来说,其具有会话时间短、时间敏感性强的特性,协议中不会出现复杂的时序关系,例如基于离散信令交互的工控协议.相比之下,协议数据和控制信息即协议语法语义信息,对于功能各异的自动化系统来说更为重要.因此本文出于实际应用的考虑,将研究重点更多的放在语法和语义分析上,不再对协议的时序关系进行过多的研究.

3.3 多维数据的生成

有效地生成测试数据是模糊测试成功执行的基础.在本章节,我们依据3.2节中逆向方法得到的协议特征构造测试数据.现有的模糊测试的效率以及测试的效果在很大程度上受限于单个字段随机填充的数据生成方式.因此,本文提出多维变异的测试数据生成方法,在本方法中,我们不仅为不同的字段定义了不同的数据生成规则,而且在数据生成时参考常见漏洞(如:超长数据、SQL注入等)的攻击

模式,生成多维变异的测试数据,保证测试数据的命中率及漏洞挖掘能力.

3.3.1　数据生成规则

数据生成规则的制定以字段的特征为依据.对于语义明确的字段,如长度字段、校验和字段等,按其语义进行计算后填充.对于语义未知的字段,有图9所示的不同数据生成规则.

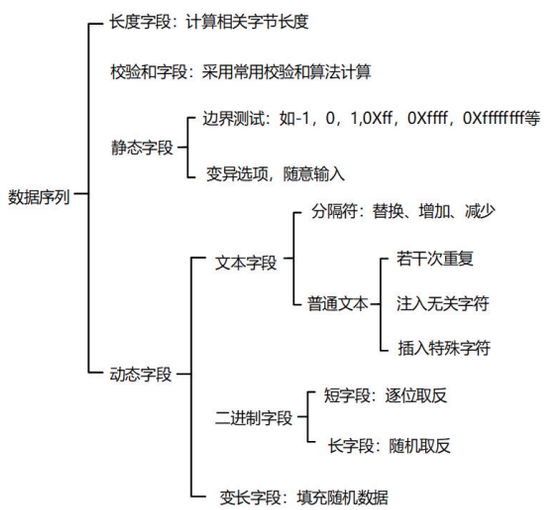

图9　测试数据生成规则

静态字段,即变化率几乎为0的字段(如:保留字段),构造少量变化的输入值来生成测试数据.为保证字段取值为临界值时的安全性,还需使用-1,0,1,0Xff,0Xffff等临界值作为测试数据.

动态字段类型丰富:1)对于文本数据,生成变长的任意字符串.值得注意的是,为了扩大测试数据的覆盖范围,用于生成上述字符串的字符集中应该包括特殊符号(如:%、#等);2)对于二进制数据来说,短二进制数据逐位取反,长二进制数据随机取反,取反位数随字段长度动态调整;3)对于变长字段,使用随机数据来填充该字段.

3.3.2　测试数据的多维变异

区别于现有的随机填充来生成测试数据,本文根据3.3.1节中的数据生成规则,参考常见的漏洞攻击模式,生成具备触发"多点漏洞"能力的测试数据,即数据可以触发至少两个字段同时畸形才能够被触发的漏洞类型.但是在生成多维变异的数据时,容易出现组合爆炸的问题.为了避免这种情况出现,本文在每次生成数据时选择最有可能触发漏洞的若干字段同时进行变异.其中,字段触发漏洞的可能性由字段触发漏洞的频率来衡量,该频率是前期工作中,随机生成测试数据作为样本数据统计所得.在本文中,触发漏洞的频率的大小随着测试的执行反馈不断调整.在生成每一批多维变异的测试数据时,选取从未操作过的频率最大的两个字段,按各自的生成规则,以最容易触发漏洞的攻击方式进行变异.图10展示了Modbus/TCP协议测试数据生成方式.

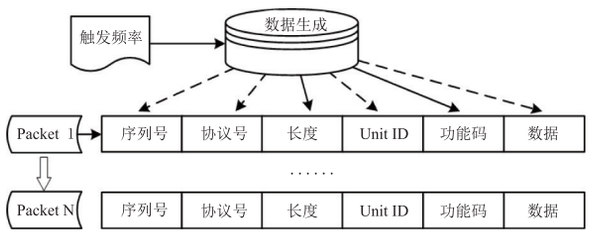

图10　Modbus/TCP协议数据的多维变异

3.4　主动状态检测与异常定位

在数据执行时,对被测设备的监测是判断测试状态的依据,且模糊测试需要在异常发生时做出正确的响应.目前的模糊测试在监测网络设备运行状况方面受限于网络设备是否支持内置的监测代理程序.因此,本文提出基于主动探测的测试执行与异常定位方法,即采用主动探测的方式,周期性的发送探测包监测被测设备的运行情况,并根据反馈结果,更新测试数据生成策略以及决定测试数据是否继续执行.除此之外,本文还提出了循环执行法,用于异常发生时对触发异常的测试数据进行精确定位,保证测试的有序执行.

由于被测的网络设备复杂多样,因此多种方式相结合的主动探测覆盖率更高.在本文中,使用表2中常见的探测方式,包括探测目标主机的PING探测,ARP探测、ICMP探测,探测端口情况的TCP_SYN包探测或UDP Scan,以及链路层探测等等,监测被测设备的运行情况.不论何种方式,均记录、分析测试系统与被测设备交互所产生的数据包,并根据被测设备的反馈结果调整测试数据的生成与执行.

值得注意的是,为保证测试的效率,在测试过程中不能对每一个测试数据都构造一个相应的探测

表2　主动探测方式

类型	主机	链路层	端口
PING	√		
ARP	√		
TCP_SYN			√
UDP Scan			√
链路探测		√	

包,这就意味着探测包的发送周期远大于测试数据的发送周期,导致在两个探测的反馈结果之间已经有众多测试数据被执行.因此,在探测到异常发生后,应立即中断测试,对触发异常的测试数据进行精确定位,分析并记录被测设备的异常情况.

本文提出循环执行法,用于精确定位触发异常的测试数据.其定义如下:算法的输入包括已发送的全部数据*dataSet*,以及数据的数量X;前一次正确反馈的时间点与当前异常反馈的时间点之间的全部测试数据,记为*errorSet*,以及数据的数量N,并定义$m=X/N$.

具体算法如算法2所示:以*errorSet*的第N/m个数据为分割点,将*errorSet*分为$[0, N/m)$和$[N/m, N]$两部分,执行*errorSet*的前N/m个数据,执行结束后发送探测包查看被测设备是否异常.

算法2. 异常数据定位算法

Require: *errorSet*, N, X, m

Ensure: *errorSet*

1. function LOCATE(*errorSet*, N, m)
2. $m \leftarrow \frac{X}{N}$
3. while $N>10$ do
4. send $errorSet[0:\frac{N}{m}]$
5. and send a detect packet
6. if error is happend then
7. $errorSet \leftarrow errorSet[0:\frac{N}{m}]$
8. $N \leftarrow \frac{N}{m}$
9. LOCATE(*errorSet*, N, m)
10. else
11. $errorSet \leftarrow errorSet[\frac{N}{m}:N]$
12. $N \leftarrow N-\frac{N}{m}$
13. LOCATE(*errorSet*, N, m)
14. end if
15. end while

return *errorSet*;

16. end function

若有异常发生,则递归执行*errorSet*的前N/m^2个数据,若没有异常则递归执行*errorSet*的后N-N/m个数据.直到执行数据的长度小于10,完成异常数据包的定位,并记录下异常的类型和造成异常的数据,以及更新字段触发漏洞的频率.定位结束后,从测试数据中断处继续测试,直到全部测试数据执行结束,完成整个测试过程.

4 测试评估

基于上文提出的未知协议的模糊测试框架与测试方法,本文开发了相应的模糊测试系统UPAFuzz. UPAFuzz分为三个模块,包括协议逆向模块:通过对输入的网络数据的特征分析主动学习出协议的特征,形成描述协议特征的脚本;测试数据生成模块:基于协议特征脚本,以及常见漏洞的攻击模式,生成多维变异的测试数据;数据执行模块:向被测设备发送大量的测试数据,并定时发送探测数据包监测被测设备的运行情况.UPAFuzz的各个模块相互独立,具备很强的可移植性和扩展性.在本章节中,首先介绍系统的测试环境,之后从准确性、高效性、通用性三个方面进行分析,对UPAFuzz系统进行评估.

4.1 系统测试环境

系统运行环境的硬件环境包括一个搭载通用Linux操作系统的64-bit PC作为测试机,另有一台运行Windows操作系统的64-bit PC作为被测机器,具体配置如表3所示.在测试机上部署本文提出的测试系统,在被测机上部署被测协议相应的协议模拟器.

表3 系统的测试环境

测试机	Intel(R) Core(TM) i5-4590 CPU@3.3GHz 4096MB 1600MHz Ubuntu 16.04.5 LTS
被测机	Intel(R) Core(TM) i5-3371 CPU@1.7GHz 8192MB 1600MHz Windows10 professional

4.2 准确性评价

本节验证测试系统UPAFuzz在没有先验知识的情况下,完成协议结构和语义特征的逆向分析并创建描述协议特征的脚本,根据协议脚本生成畸形的测试数据,发送给被测设设备的同时,主动探测被测设备的运行情况.为了便于验证结果,本文选取工业控制领域的Modbus/TCP协议作为被测对象,工控领域是未知协议广泛应用的代表领域,且Modbus/TCP协议为工控领域的标准通信协议.

在接下来的测试中,系统的输入为Modbus/TCP协议的网络流,该网络流是真实应用场景下的服务产生的数据集.测试使用MODBUS-SLAVE

(modbustools.com)作为协议模拟器.涉及到的网络抓包操作,通过Wireshark完成.

图11展示了系统自动化学习出的Modbus协议的结构和语义特征结果.图中数字表示字段所占字节,文字为字段含义.由图可以发现,协议真实结构与系统自动学习的特征基本一致.其中序列号字段、长度字段以及功能码字段的长度和语义识别结果均与真实协议特征吻合;协议号字段被识别为静态字段,这是因为该字段固定使用0x00表示Modbus协议;另外,最后的不定长数据字段,被识别为一个动态字段和随机变长字段,符合协议的实际表现.以上协议特征的推断结果证明,本系统能够基于网络流量对协议进行逆向分析且分析结果准确.

图11　Modbus协议逆向分析结果

根据协议特征的逆向分析结果,系统自动生成了Modbus/TCP协议的测试数据,发送给MODBUS-SLAVE客户端,本文使用Wireshark捕获UPAFuzz与被测协议模拟器的交互数据包.如图12所示,系统正在对序列号字段进行模糊,被测设备能够响应且长度字段校验正确.实验证明本文的系统能够精准地逆向出协议的结构和语义特征,成功完成协议的模糊测试,并支持对被测设备的主动探测.

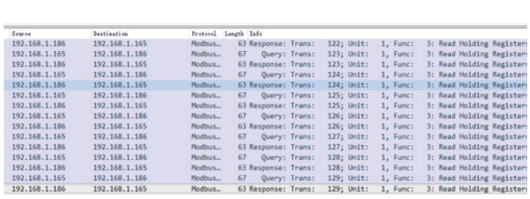

图12　Modbus协议测试过程生成数据包

4.3　效率评价

在实际的应用中,测试系统需要具备高效的数据生成能力,这是测试效果的保证.因此,在本章节中,我们测评在生成海量数据时UPAFuzz的表现,并与现有模糊测试工具Boofuzz[10]进行比较.Boofuzz[10]是经典模糊测试框架Sulley[22]的继承者,对协议的模糊测试有着良好的支持,且其代码开源,目前被广泛使用,但Boofuzz无法直接获取协议相关知识,需人工定义协议模型.因此,本章节比较UPAFuzz与Boofuzz,在输入相同协议知识,生成相同数量的测试数据的情况下,程序运行时间以及内存占用情况.

本文赋予了UPAFuzz自定义功能,使其具备更强的灵活性,即使用者可以自定义描述协议特征的脚本,以及自定义测试数据的生成方式.为了更加迅速有效地对两种工具进行比较,本文自定义协议为动态的二进制字段,数据生成方式为逐位翻转.在实验过程中,通过调整二进制字段的位数来控制数据的生成数量.例如,10位的二进制字段可以生成1MB的数据量.如图13所示,在测试数据的数量为100万以下时,二者的表现并无差别,但是当数据的数量继续增加时,Boofuzz与在运行时间和内存占用情况方面,逐渐超过本文系统UPAFuzz,且差距越发明显.在生成16,000,000个测试数据时,UPAFuzz的用时为Boofuzz的10%,且内存消耗仅为Boofuzz的50%.因此,在生成的数据量较大时,本文的系统消耗的系统资源更少,具备更高效的测数据生成能力.

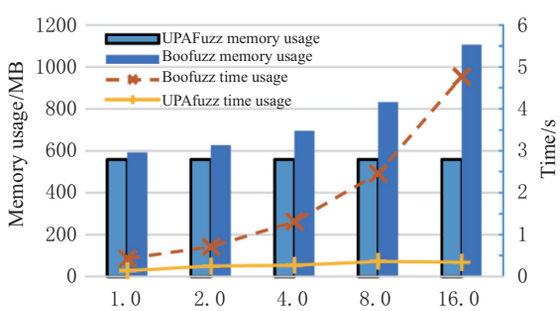

图13　UPAFuzz高效性测试

4.4　通用性评价

未知协议复杂多样,系统的通用性是识别千差万别的未知协议的基础,测试系统通用性越强,代表其有能力对更大范围的未知协议进行测试.因此,在本章节中,我们使用类型截然不同的协议ICMP、S7COMM协议来评估UPAFuzz的通用性.

4.4.1　ICMP协议

在已经确认UPAFuzz已经能够正确识别Modbus/TCP协议的基础上,本文继续选取与Modbus/TCP协议分属不同领域的ICMP协议作为测试对象,重复对Modbus/TCP协议进行的测试流程并观察测试结果.

测试结果如图14所示,图中数字表示字段所占字节,文字为字段含义.学习后的协议特征与ICMP协议真实特征相比,校验和字段、序列号字段具有一

	0	1	2-3	4	5	6-7	8-
逆向协议特征	动态字段	静态字段	校验和	静态字段	动态字段	序列号	静态字段
	0	1	2-3	4	5	6-7	8-
真实协议特征	类型	代码	校验和	标识符		序列号	选项字段

图14 ICMP协议逆向分析结果

致性. 代码字段、选项字段被识别为静态字段,标志符字段被识别为一个字节的动态字段和一个字节的静态字段,这是因为这些字段在ICMP协议中取值有限,在协议识别过程中,由于供测试的数据包数量有限,使得以上字段在有限的数据包中取值变化较小或无变化,导致系统会将这些字段识别成静态字段或分割成部分静态字段和部分动态字段. 对于这种情况,可以增加数据采集量来突显出这些字段间的特征差距,从而更准确地划分出这些字段.

根据学习后的结果生成测试数据发送给被测设备,并捕获与被测设备交互的数据包. 实验结果如图15所示,被测设备能够响应且校验和字段校验正确.

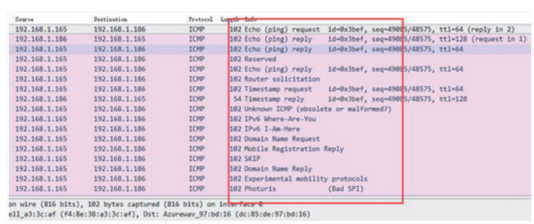

图15 ICMP协议测试过程生成数据包

4.4.2 S7COMM协议

在ICMP协议作为测试对象的基础上,我们进行S7COMM协议的测试. S7COMM协议属于西门子S7通讯协议簇,广泛应用于西门子的可编程逻辑控制器(PLC)中. S7COMM协议是复杂的三层组合协议,图16展示了S7COMM的协议模型.

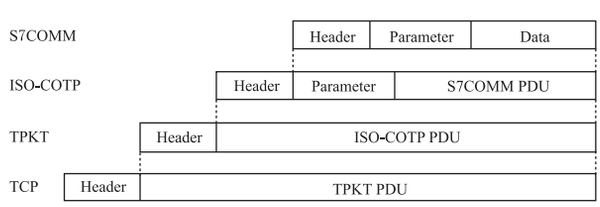

图16 S7COMM协议层模型

本实验的环境为河北工业大学的工控网络安全实验室,该实验室实现了工业控制系统主要应用场景的再现,包括三相异步电机控制系统、供热温度控制系统、生产线运动控制系统、风力发电半物理仿真系统等. 如图17所示,本实验应用于使用S7-1200西门子PLC的化工多液体混合的自动加工系统. 上位机与PLC之间运行S7COMM协议进行PLC(6ES7214-1AG40-0xB0)寄存器读写操作,从而改变其输出电信号.

图17 应用S7COMM协议的多液体混合自动加工系统

实验过程中,使用Wireshark捕获上位机与PLC之间交互的数据包,形成S7COMM协议的网络流文件作为系统输入,图18展示了UPAFuzz的逆向分析结果. 图18(a)为执行写寄存器操作的S7COMM协议功能包的真实特征,图18(b)为逆向后的协议特征. 协议通过指定变量的存储及大小或类型来执行写操作. 因此三层协议的header中,诸如版本、协议ID、PDU类型、Unit、操作类型、项目、规格、区域等均为固定值,被识别为静态字段;2—3字节的长度字段表示三层协议的总长度,被准确识别出;而对于协议包头以及23—24字节的长度字段,由于实验过程中COTP包类型、PLC类型及数据存储容量为固定值,被识别为静态字段;另外,DB号及地址因为使用的数据模块及地址不断改变而识别为动态字段;最后的不定长数据字段被识别为动态字段和随机长度字段的组合,符合协议的实际表现.

(a) S7COMM功能包真实协议特征

(b) S7COMM功能包逆向协议特征

图18 S7COMM协议的真实特征与逆向特征

由实验结果可得,我们复现了S7COMM协议的结构和语义特征,且生成的数据能够正确的和S7-1200交互并实现输出电平的变化.

5 总 结

未知协议的安全作为近年来安全研究的新方向,受到了广泛的关注.保障未知协议的安全成为各应用领域需要解决的的重要问题.基于此,有必要对各种未知协议进行安全性测试,提前发现存在的安全风险或漏洞,进一步确保使用设备的安全稳定运行.

本文分析了未知协议的特点和存在的安全问题,指出现有模糊测试框架应用于未知协议时,在协议知识获取、测试数据生成以及异常监测三个方面存在的重大问题,并为了解决这些问题设计了适用于未知协议的自动化模糊测试框架,其基本思路为,摆脱人工参与,自动学习出协议特征,生成多维变异的测试数据,并支持被测设备的主动探测.基于此框架,我们提出基于协议特征库的协议逆向分析方法,用于自动学习协议特征;提出基于多维变异的测试数据生成方法,生成命中率高且漏洞挖掘能力强的测试数据;采用主动探测的测试执行与异常定位方法,以及提出循环执行法用于精确定位触发异常的数据.基于以上方法,我们开发了未知协议的自动化模糊测试系统UPAFuzz.经实验结果证明,UPAFuzz能够在没有任何先验知识的情况下,逆向出多种不同类型协议的特征,基于逆向结果生成的测试数据能够与被测设备进行有效的交互,且支持被测设备运行情况的主动探测.对比现有的模糊测试工具Boofuzz,本文的UPAFuzz的效率更高,性能更优.

参 考 文 献

[1] StoufferKeith, PillitteriVictoria, LightmanSuzanne, Abrams-Marshall, and HahnAdam. Guide to industrial control systems security. National Institute of Standards and Technology, 2015, 800(82):16-48

[2] Winkler M, Tuchs K D, Hughes K, et al. Theoretical and practical aspects of military wireless sensor networks [J]. Journal of Telecommunications and Information Technology, 2008, nr 2: 37-45

[3] Ping Wang, Lei Wu, AslamBaber, and Cliff C Zou. A systematic study on peer-to-peer botnets//Proceedings of 18th International Conference on Computer Communications and Networks, San Francisco, USA, 2009: 1-8

[4] GooYoung-Hoon, ShimKyu-Seok, LeeMin-Seob, and Myung-Sup Kim. Protocol specification extraction based on contiguous sequential pattern algorithm. IEEE Access, 2019, 7: 36057-36074

[5] Ziegler S, Crettaz C, Kim E, et al. Privacy and security threats on the internet of things. Internet of Things Security and Data Protection. Cham, Switzerland: Springer, 2019: 9-43

[6] Sangwho Kim and JaecheolRyou. Source code analysis for static prediction of dynamic memory usage//Proceedings of International Conference on Platform Technology and Service, Jeju, Korea, 2019: 1-4

[7] MaedeZolanvari, TeixeiraMarcio A, GuptaLav, KhanKhaled M, and JainRaj. Machine learning based network vulnerability analysis of industrial internet of things. IEEE Internet of Things Journal, 2019, 6(4): 6822-6834

[8] Dias-Neto A C, Travassos G H. A picture from the model-based testing area: concepts, techniques, and challenges. Advances in Computers. 2010, 80: 45-120

[9] Sutton M, Greene A, Amini P. Fuzzing: brute force vulnerability discovery. Massachusetts: Addison-Wesley Professional, 2007

[10] KunzStefan. Penetration Testing Framework forOCSP-Responders [D]. University of Passau, Passau, Germany, 2018

[11] Yun X, Wang Y, Zhang Y, et al. A semantics-aware approach to the automated network protocol identification. IEEE/ACM transactions on networking, 2015, 24 (1): 583-595

[12] Chabinsky S R. Cybersecurity strategy: A primer for policy makers and those on the front line. journal of national security law &. policy, 2010, 4: 27-40

[13] Kitagawa T, Hanaoka M, Kono K. AspFuzz: A state-aware protocol fuzzer based on application-layer protocols// Proceedings of the IEEE symposium on Computers and Communications. Riccione, Italy, 2010: 202-208

[14] Perez-Guadarrama Y, Simón-Cuevas A, Hojas-Mazo W, et al. A Fuzzy Approach to Improve an Unsupervised Automatic Keyphrase Extraction Process//Proceedings of 2018 IEEE International Conference on Fuzzy Systems. Miyazaki, Japan, 2018: 1-6

[15] Aitel D. The advantages of block-based protocol analysis for security testing. Immunity Inc., 2002, 105: 106-114

[16] Vuagnoux M. Autodafe: An act of software torture// Proceedings of the 22th Chaos Communication Congress. Berlin, Germany, 2005: 47-58

[17] Eddington M. Peach fuzzing platform. Peach Fuzzer, 2011, 34: 32-43

[18] Chen J, Diao W, Zhao Q, et al. IoTFuzzer: Discovering Memory Corruptions in IoT Through App-based Fuzzing// Proceedings of the Network and Distributed System Security Symposium. San Diego, California, USA, 2018: 1-15

[19] Song C, Yu B, Zhou X, et al. SPFuzz: A Hierarchical Scheduling Framework for Stateful Network Protocol Fuzzing.

IEEE Access, 2019, 7: 18490-18499

[20] Godefroid P, de Halleux P, Nori A V, et al. Automating software testing using program analysis. IEEE software, 2008, 25(5): 30-37

[21] Rawat S, Jain V, Kumar A, et al. VUzzer: Application-aware Evolutionary Fuzzing//Proceedings of the Network and Distributed System Security Symposium. San Diego, California, USA, 2017: 1-14

[22] Devarajan G. Unraveling SCADA protocols: Using sulleyfuzzer//Proceedings of the Defon 15 Hacking Conference. Las Vegas, USA, 2007: 27-39

[23] Zhang D, Wang J, Zhang H. Peach improvement on profinet-DCP for industrial control system vulnerability detection//Proceedings of the 2015 2nd International Conference on Electrical, Computer Engineering and Electronics. Jinan, China, 2015: 1622-1627

[24] Bratus S, Hansen A, Shubina A. LZfuzz: a fast compression-based fuzzer for poorly documented protocols[D]. Darmouth College, Hanover, USA, 2008

[25] DeMott J, Enbody R, Punch W F. Revolutionizing the field of grey-box attack surface testing with evolutionary fuzzing. BlackHat and Defcon, 2006, 26(5): 1097-1101

[26] Amini P. Paimei-reverse engineering framework//Proceedings of the Reverse Engineering Conference. Benevento, Italy, 2006: 21-49

[27] Patel S C, Graham J H, Ralston P A S. Quantitatively assessing the vulnerability of critical information systems: A new method for evaluating security enhancements. International Journal of Information Management, 2008, 28(6): 483-491

[28] Johansson W, Svensson M, Larson U E, et al. T-Fuzz: Model-based fuzzing for robustness testing of telecommunication protocols//Proceedings of the2014 IEEE Seventh International Conference on Software Testing, Verification and Validation. Cleveland, Ohio, USA, 2014: 323-332

[29] Sija B D, Goo Y H, Kim S, et al. Survey on network protocol reverse engineering approaches, methods and tools//Proceedings of the 2017 19th Asia-Pacific Network Operations and Management Symposium.Seoul, Korea, 2017: 271-274

[30] Dreger H, Feldmann A, Mai M, et al. Dynamic application-layer protocol analysis for network intrusion detection//Proceedings of the 15th USENIX security symposium. Vancouver, B.C., Canada, 2006: 257-272

[31] Shuai W, Xiang C, Peng L, et al. S-URL flux: A novel c&c protocol for mobile botnets//Proceedings of the International Conference on Trustworthy Computing and Services. Beijing, China, 2012: 412-419

[32] Beddoe M A. Network protocol analysis using bioinformatics algorithms. Toorcon, 2004, 26(6): 1095-1098

[33] Cui W, Kannan J, Wang H J. Discoverer: Automatic Protocol Reverse Engineering from Network Traces//Proceedings of the USENIX Security Symposium. Boston, USA, 2007: 1-14

[34] Bossert G, Guihéry F, Hiet G. Netzob: un outil pour la rétro-conception de protocoles de communication//Proceedings of the Actes du Symposium sur la sécurité des technologies de l'information et des communications. Rennes, France, 2012: 43-77

[35] Wang Y, Yun X, Shafiq M Z, et al. A semantics aware approach to automated reverse engineering unknown protocols//Proceedings of the 2012 20th IEEE International Conference on Network Protocols. Austin, USA, 2012: 1-10

[36] Luo J Z, Yu S Z. Position-based automatic reverse engineering of network protocols. Journal of Network and Computer Applications, 2013, 36(3): 1070-1077

[37] Kleber S, Maile L, Kargl F. Survey of Protocol Reverse Engineering Algorithms: Decomposition of Tools for Static Traffic Analysis. IEEE Communications Surveys & Tutorials, 2018, 21(1): 526-561

[38] Lim J, Reps T, Liblit B. Extracting output formats from executables//Proceedings of the 2006 13th Working Conference on Reverse Engineering. Benevento, Italy, 2006: 167-178

[39] Wondracek G, Comparetti P M, Kruegel C, et al. Automatic Network Protocol Analysis//Proceedings of the Network and Distributed System Security Symposium. San Diego, California, USA, 2008: 1-14

[40] Comparetti P M, Wondracek G, Kruegel C, et al. Prospex: Protocol specification extraction//Proceedings of the 2009 30th IEEE Symposium on Security and Privacy. Oakland, USA, 2009: 110-125

[41] Cho C Y, Babi ć D, Shin E C R, et al. Inference and analysis of formal models of botnet command and control protocols//Proceedings of the 17th ACM conference on Computer and communications security. Chicago, USA, 2010: 426-439

[42] Xue L, Qian C, Zhou H, et al. NDroid: Toward tracking information flows across multiple Android contexts. IEEE Transactions on Information Forensics and Security, 2018, 14(3): 814-828

[43] Leita C, Mermoud K, Dacier M. Scriptgen: an automated script generation tool for honeyd//Proceedings of the 21st Annual Computer Security Applications Conference. Tucson, AZ, USA, 2005: 203-214

[44] Cui W, Paxson V, Weaver N, et al. Protocol-Independent Adaptive Replay of Application Dialog//Proceedings of theNetwork and Distributed System Security Symposium. SanDiego, USA, 2006: 1-15

[45] Bossert G, Guihéry F, Hiet G. Towards automated protocol reverse engineering using semantic information//Proceedings of the 9th ACM symposium on Information, computer and communications security. Kyoto, Japan ,2014: 51-62

[46] Halfaker A, Keyes O, Kluver D, et al. User session identification based on strong regularities in inter-activity time//Proceedings of the 24th International Conference on World Wide Web.Florence, Italy, 2015: 410-418

Zhang Wei-Yao, Ph. D. Her main research interests focus on security and testing of the network.

Zhang Lei, Ph. D. Professor. His research interests focus on theoretical and engineering research of intelligent systems.

Mao Jian-Ling, M. S. His research interests focus on security and testingof the network.

Xu Zhi-Jun, postdoc. His research interests focus on security and testing of the network.

Zhang Yu-Jun, Ph. D. Professor. His research interests focus on internet security assessment and verification.

Background

In innovative networks such as industrial control, military communications, and financial information, a large number of unknown (private or semi-private) protocols are widely adopted. Strict testing for protocols is an important method to ensure the security of network systems. Most of the existing testing methods can only be applied to known protocols. However, before we can transplant these methods to unkonwn protocols, there are majorly three defects to be noticed: First, current fuzzing frameworks depend on manual participation to acquire protocol-related knowledge. Second, the test data is generated in a random and inefficient way. Third, current frameworks rely on embeded monitor to inspect the device under test, which is infeasible in the case of unknown protocols running on private devices.

To address above issues, we propose both a novel automated fuzzing framework and a series of automated fuzzing method for unknown protocols.Our main contributions can be concluded as follows: First, we propose an automated reverse analysis method for unknown protocol where the protocol characteristics are automatically learned with increased inference accuracy and no manual efforts in virtue of the novel protocol feature database we bring up.Second, we propose an innovative way to generate test data by mutating date in a multi-dimensional way and referring to attack modes of common protocol vulnerabilities. The generated test data is proved to be with high hit rate and desirable vulnerability-mining ability. Third, we present a set of active-detection methods to replace the embedded monitor for test execution and following inspection and analysis, where the anomaly data will be located by the loop execution method we proposed with high accuracy. Last but not least, under the guidance of above framework, we developed UPAFuzz, an automated fuzzing system for unknown protocols. Experiments outcomes show that UPAFuzz can carry out protocol fuzzing test without prior knowledge and the active detection module worked well as expected. Moreover, UPAFuzz is able to generate test data more efficiently and with certain versatility.

This work was supported by the National Key Research and Development Program of China(2016YFE0121500, 2018YFB1800403), the National Science Foundation of China (61902382, 61972381, 61672500 and 61572474), and the Strategic Priority Research Program of Chinese Academy of Sciences (XDC02030500).