第 45 卷 第 7 期 计 算 机 工 程 2019 年 7 月
Vol. 45 No. 7 Computer Engineering July 2019

·安全技术· 文章编号:1000-3428(2019)07-0164-06 文献标志码:A 中图分类号:TP393

基于循环神经网络的 Modbus/TCP 模糊测试算法

黄 河[1,2],陈 君[1],邓浩江[1]

(1. 中国科学院声学研究所 国家网络新媒体工程技术研究中心,北京 100190;
2. 中国科学院大学,北京 100190)

摘 要: Modbus/TCP 安全漏洞挖掘的相关协议包常采用随机方式生成,易产生过多无效包,降低漏洞挖掘效率。为此,基于循环神经网络(RNN)提出结构性模糊算法 Fuzzy-RNN。从 Modbus/TCP 训练集中学习协议包各部分的概率分布,并考虑极端参数条件,实现针对性的模糊生成。实验结果表明,与通用模糊测试器 GPF 相比,Fuzzy-RNN 算法在 Modbus Slave、xMasterSlave 等多种仿真软件上能以更高概率实现合法协议包的模糊生成,测试时间缩减 50% 以上,测试效率明显提高。

关键词: 工业控制协议;漏洞挖掘;模糊测试;网络安全;循环神经网络;序列到序列

中文引用格式: 黄河,陈君,邓浩江. 基于循环神经网络的 Modbus/TCP 模糊测试算法[J]. 计算机工程,2019,45(7):164-169.

英文引用格式: HUANG He,CHEN Jun,DENG Haojiang. Fuzzy testing algorithm for Modbus/TCP based on recurrent neural networks[J]. Computer Engineering,2019,45(7):164-169.

Fuzzy Testing Algorithm for Modbus/TCP Based on Recurrent Neural Networks

HUANG He[1,2],CHEN Jun[1],DENG Haojiang[1]

(1. National Network New Media Engineering Research Center,Institute of Acoustics,Chinese Academy of Sciences,
Beijing 100190,China; 2. University of Chinese Academy of Sciences,Beijing 100190,China)

【Abstract】 The related protocol packets for Modbus/TCP security vulnerability mining are often generated in a random way,which is prone to generate excessive invalid packets and reduce the efficiency of vulnerability mining. To deal with this problem,a structural fuzzy algorithm named Fuzzy-RNN is proposed based on the concept of Recurrent Neural Networks(RNN). It learns the probability distribution of each part of the proptocol packet from the Modbus-TCP training set,and takes the corner cases into account,so as to realize the targeted fuzzy generation. Experimental results show that compared with the General Protocol Fuzzer(GPF),in a variety of simulation software such as Modbus Slave and xMasterSlave,the Fuzzy-RNN algorithm can achieve the fuzzy generation of legal protocol packets with a higher probability. The test time can be reduced by more than 50% ,and its efficiency can be obviously improved.

【Key words】 Industrial Control Protocol(ICP);vulnerability mining;fuzzy testing;network security;Recurrent Neural Networks(RNN);sequence to sequence(seq2seq)

DOI: 10.19678/j.issn.1000-3428.0051190

0 概述

随着传统工业信息化程度的逐步加深,工业控制系统(Industrial Control System,ICS)得到较大发展,应用于越来越多的设备。ICS 主要包括数据采集与监视控制系统、分布式控制系统、可编程逻辑控制器、远端测控单元等,其通信模式主要包括工业设备间、工业设备与生产设备间、主机与各种服务器之间的通信。现代 ICS 不仅广泛应用于化工、电力等传统行业,在汽车控制、物联网等新兴行业也发挥重要作用。用于工业控制系统的通信协议统称为工业控制协议(Industrial Control Protocol,ICP)[1]。

ICP 所涉及的行业之广、程度之深,使得其安全问题备受瞩目。ICP 所适用的环境多元性较强,导致同一参数在不同应用中的含义千差万别,因此,安全开发者难以制定一系列通用的安全规则,渗透测试成为探查 ICP 漏洞的主要手段。渗透测试是一种用来识别未知测试系统的方法,它应用已知或详细的攻击媒介生成一系列输入来测试系统对模拟攻击的适应能力,其目的是尽可能快地发现导致目标系

基金项目: 中科院率先行动计划“端到端关键技术研究与系统研发”(SXJH201609)。

作者简介: 黄 河(1990—),男,博士研究生,主研方向为网络安全、人工智能、数据管理;陈 君(通信作者)、邓浩江,研究员、博士。

收稿日期: 2018-04-12 **修回日期:** 2018-05-30 **E-mail:** huangh@dsp.ac.cn

统瘫痪的输入。渗透测试需要访问目标系统,但不能访问其源代码。通常由于时间和预算限制,渗透测试不能覆盖所有的输入情况,这时就需要它的分支——模糊测试来达成。模糊测试通过一定的策略来生成意外输入和监视异常,从而发现软件故障[2],它是一种高度自动化的测试技术,不需要访问源代码或系统内部。文献[3]提供了明确的执行思路并在当时的 ICP 上实现最优解。然而,一般的模糊测试使用随机方式输入,可能会运行较长时间,且只能测试一部分可能的情况。目前广泛使用的 ICP,如 Modbus 协议等,均使用了模糊测试方法。在实际应用中,同一协议对应的多种应用场景往往具有多样化的参数结构和数据规模,难以实现模糊效率和有效模糊的平衡。因此,本文针对 Modbus 协议,基于循环神经网络(Recurrent Neural Networks,RNN)提出一种新颖的结构性模糊生成算法。在给定大量训练语料的前提下,学习生成性的 Modbus/TCP 语言模型,通过 RNN 上的 seq2seq(sequence to sequence)网络模型,提供最优的预测结果。

1　相关工作

1.1　通用模糊生成

通用的模糊生成方式主要有以下 3 种:

1)基于语法的模糊生成:目前较流行的黑盒随机模糊器大多支持某种形式的语法表征,例如 SPIKE 和 Peach 等[2]。基于语法的测试输入生成工作始于 20 世纪 70 年代[4],其所生成的数据基于已知语法随机生成[5]或者具有穷举性[6]。此类方法的优势在于逻辑简单,理论难度不高,但其需要大量的先验知识和训练语料。

2)基于动态学习的模糊生成:该类算法的特点是能够用较少的训练语料生成足够多的模糊数据[7],其可在给定一组输入示例的情况下,合成上下文无关语法,并生成用于模糊的新输入。该算法使用一组泛化步骤,通过引入正则表达式的重复和交替结构,合并上下文无关语法,实现输入语言的单调泛化。该技术能够捕获输入格式的分层属性,但其不适用于 Modbus 协议等格式,因为这些格式相对比较平坦,但包含大量不同的私有协议类型和参数范围组合。本文方法使用序列式的神经网络模型来学习这种多样化平坦格式的统计生成模型,可省去一部分冗余的特征工程。

3)基于神经网络预测的模糊生成:近年来有很多关于使用神经网络进行数据包分析和合成的研究。其中一些朴素的想法是简单排列字节元素,学习该排列,然后进行微调性质的突变[8]。文献[9]使用新颖的神经架构 R3NN 对输入输出进行编码,通过逐步扩展模糊对象的特征密集区域来实现有针对性的模糊。文献[10]将上述方法运用在网络流量结构分析上,实现安全预测。基于 RNN 的 seq2seq 技术也被广泛用于修正生成性问题中的语法错误[11-12],该方法通过学习大量正确格式的数据包来实现这一功能。但此类方法尚无在 Modbus/TCP 上应用的先例。本文充分利用 seq2seq 的概率统计模型,提高生成性 Modbus 包格式的正确率。

1.2　Modbus/TCP 模糊测试

Modbus 协议是 1979 年针对 ICS 开发的串行通信协议,其已成为连接工业电子设备的通用标准。自 2004 年以来,该协议的开发和更新由 Modbus 在线组织管理。Modbus 是一种简单而强大的协议,使用主从结构的无状态通信,通信帧被分为“请求”和“响应”2 类。Modbus 帧可以通过串行链路或 TCP/IP 进行传输,实现过程较为简单。然而,Modbus 的应用大多只利用了其基本通信功能,对其异常工作状态的研究还不够深入。目前,针对 Modbus 协议的分析所涉及的领域包括协议结构探察、流程完整性和拒绝服务攻击等[13]。Modbus 的多种测试工具均可提供模糊测试[14],这些工具大多是纯粹的模糊器,可随机生成或改变 Modbus 流量并操纵协议字段,开发者可利用其开发新的安全产品。Modbus 的相关工作多集中于使用格式化的语法和限定条件产生有效模糊测试用例,文献[15]的方法精密而详细,然而其匹配规则极为复杂。文献[16]针对特定私有协议的特征空间较小的性质进行个性化覆盖,大幅缩短漏洞挖掘时间,但其在不同应用场景需要不同的设计,难以形成统一的解决方案。

2　Modbus/TCP 协议结构

Modbus/TCP 应用数据单元(Application Data Unit,ADU)的结构如图 1 所示。其中,Trans 指的是 transaction,即连接序号数,Protocol 恒定为 0x0000,Length 表示从 Unit 开始的消息长度,即字节数,Unit 恒定为 0xff,F-Code 表示 Function Code,即指示该消息性质的识别码,Parameters 是该条消息的具体参数($1 \leqslant x \leqslant 252$)。整个消息的最大字节数是 260,从 Trans 到 Unit 的 7 Byte 被称为 Modbus 消息头(Modbus Application Header,MBAP),余下的最多 253 Byte 被称为 Modbus 数据单元(Protocol Data Unit,PDU)。Modbus/TCP 的通信结构由主站和从站构成,前者可类比于发出指令的客户端,后者可类比于服务端,主站可能与从站有多个待处理事务,而一个从站可能与多个主站进行通信,如图 2 所示。

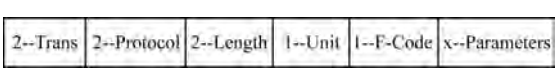

图 1　Modbus/TCP 消息结构

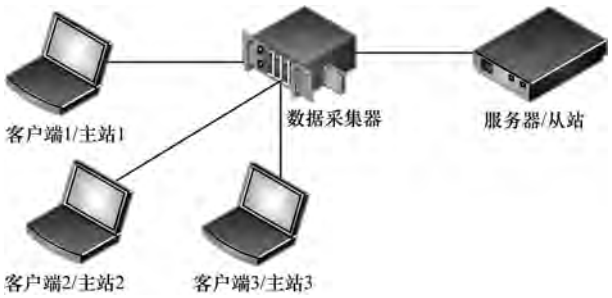

图 2 Modbus/TCP 通信拓扑结构

Modbus Function Code 的正向指令范围是从 0 到 127,根据 Modbus 通用标准的定义,其只取以下范围值:1 ~ 64,73 ~ 99,111 ~ 127,其他值用作私有协议备用,负向响应范围从 128 ~ 255。Function Code 后面跟随 Parameters,将请求参数传递给从站。例如,名为 *read holding register* 的 Function Code 之后要跟随特定保持寄存器的地址(编号)。图 3 给出该类 Function Code 的一个响应,表示读取特定地址的保持寄存器的返回结果。

```
00 06 00 00 00 3B 01 03 28 55 6E 69
74 32 33 2D 41 FF FF 80 00 FF FF FF
FA 80 00 00 00 43 7E E2 C6 42 0A C3
26 42 7D 7A EB 41 07 0E 38 00 00 00
```

图 3 Modbus/TCP 消息示例

本文研究如何利用神经网络来学习 Modbus/TCP Parameters 的语法。这些数据对象是格式化文本,如图 3 和表 1 所示。用于定义和组成这些数据对象的规则占据 Modbus 格式规范的大部分[17]。这些规则数量较多,难以记忆,但其重复和结构化的特点使其非常适合用神经网络进行学习。

表 1 Modbus/TCP 示例消息字段释义

类型	寄存器	字节	含义	备注
64b string8	40001	556E 6974	Unit23-A	文本标签
16bit UINT	40005	FFFF	65535	无符号整数
16bit INT	40006	8000	-32678	带符号整数
32bit UINT	40007	FFFF FFFA	4294967290	无符号整数
32bit INT	40009	8000 0000	-2147483648	带符号整数
32bit Float	40011	437E E2C6	254.88583	浮点参数 1
32bit Float	40013	420A C326	34.69057	浮点参数 2
32bit Float	40015	427D 7AEB	63.37004	浮点参数 3
32bit Float	40017	4107 0E38	8.440971	浮点参数 4
8bit UNIT	40019	00	0	无符号整数
8bit UNIT	40019	00	0	无符号整数
8 bit	40020	00	0000 0000	状态码 1 ~ 8
8 bit	40020	07	0000 0111	状态码 9 ~ 16

其他 ICP 如 Ethernet/IP、S7、IEC61850 等,也是由相类似的语法结构构成的,虽然具体规则有较大差异,但结构化思想是一致的。因此,选择对 Modbus 协议进行研究,有助于将模糊测试方法推广到其他协议中。

3 Fuzzy-RNN 模糊算法设计

本文提出一种机器学习统计方法,用于学习 Modbus/TCP 消息的概率模型。其主要思想是在给定大量训练语料的前提下,学习生成性的 Modbus/TCP 语言模型。本文使用 RNN 上的 seq2seq 网络模型,其已被证明可为不同的学习任务提供最优的预测结果。传统的基于格式化语法的方法受上下文长度的限制,而 seq2seq 模型允许学习任意长度的上下文来预测下一个字符序列。这种 RNN 模型的输入序列为协议消息中的字符序列,其通过将输入移动一个位置(固定长度的 1 个或若干个字符)来获得相应的输出序列,字符之间的概率分布通过位移前后的2 个序列的差异来获得。训练好的模型可直接用于生成新的 Modbus/TCP 消息。

3.1 RNN 学习模型适用性解析

RNN 是一种在输入序列 $X=\{X^{(1)},X^{(2)},\cdots,X^{(T)}\}$ 上工作,拥有隐藏状态 $H=\{H^{(1)},H^{(2)},\cdots,H^{(T)}\}$ 和输出 $Y=\{Y^{(1)},Y^{(2)},\cdots,Y^{(T)}\}$ 的神经网络,其能将从序列 X到序列 Y 的映射储存在网络中,如图 4 所示。换言之,该模型学习到的是条件概率 $\Pr((Y^{(1)},Y^{(2)},\cdots,Y^{(T)})|(X^{(1)},X^{(2)},\cdots,X^{(T)}))$,当输入测试值 x'时,可预测出对应的输出 y'。在图 4 中,U 和 W 为映射参数,L 表示期望输出,E 表示实际输出 Y 与 L 的误差,每一轮训练的目标是降低 E 值。

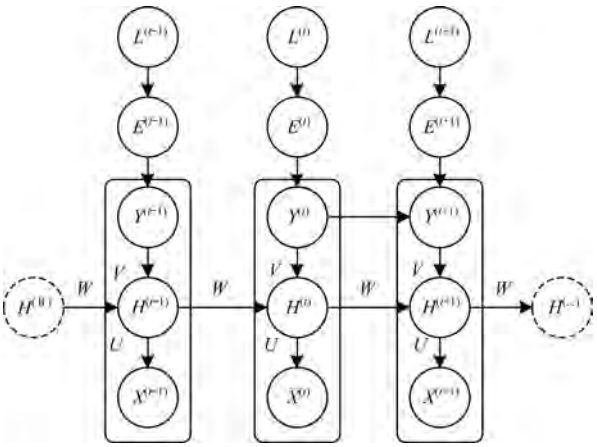

图 4 本文 RNN 模型示意图

将 t 时刻的梯度设定为 $\delta(t)$,则

$$\delta(t)=\frac{\partial E}{\partial U}+\frac{\partial E}{\partial W}+\frac{\partial E}{\partial V}=\frac{\partial E}{\partial H^{(t)}}+\frac{\partial E}{\partial V} \tag{1}$$

V 不涉及数据序列不同时间戳之间的关系,可以略去。计算 U、W 从时刻 $t+1$ 向 t 的倒推,有下式:

$$\frac{\partial E}{\partial H^{(t)}}=\frac{\partial E}{\partial Y^{(t)}}\frac{\partial Y^{(t)}}{\partial H^{(t)}}+\frac{\partial E^{(t)}}{\partial H^{(t+1)}}\frac{\partial H^{(t+1)}}{\partial H^{(t)}}=V^{(T)}(L^{(t)}-Y^{(t)})+W^{(T)}\frac{\partial E}{\partial H^{(t+1)}}\mathrm{diag}(1-(H^{(t+1)})^2) \tag{2}$$

其中,diag 表示矩阵取对角线。

BP 算法[18]每次训练的修正过程如下:

$$\Delta w_{ij} = \alpha\delta(t)X(t) \tag{3}$$

其中,α 为迭代步长。

由以上公式可知,随着训练迭代的进行,U、W 所反映的序列关系将越来越接近真实值,同时,输出层变换矩阵 $\boldsymbol{V}$ 使用 Softmax 概率密度函数,可使输出不固定在一个确定值,因此可生成合乎语法规则但又形态各异的 Modbus/TCP 消息。

基于 RNN 提出的 seq2seq 模型使用了 2 个 RNN,一个用于编码,另一个用于解码,如图 5 所示。其中,编码 RNN 将多维输入序列映射成固定维数的隐藏表达式,对应图 4 的 H。然后解码 RNN 把从时刻 t 学习到的条件概率分布用于预测 $t+1$ 时刻的输出,如此可在概率空间穷尽之前预测 $t+2, t+3, \cdots, t+T$ 时刻的输出。将输入序列视为整体 X_0,T 值不固定,则该模型可以学习到条件概率 $\Pr((Y^{(1)}, Y^{(2)}, \cdots, Y^{(T)}) \mid X_0)$。因此,该模型通过特定的消息头可学习到 Modbus/TCP 消息的多种形式。

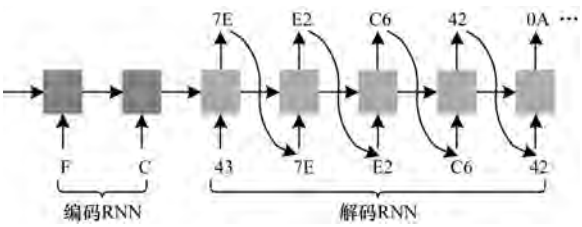

图 5　基于 RNN 的 seq2seq 模型

解决了训练过程中的概率理论问题,要预测序列还需在生成序列的输出过程确定采样策略。通常从消息头的若干个字符开始查询已训练的模型,生成一系列输出字符,直到触碰到该消息的规定长度,即包含在消息头中且不超过 260 Byte。设定每个输出节点的候选输出为 $\{y_1, y_2, \cdots, y_n\}$,最终输出为 y,本文采用 2 种不同的采样策略:

1)贪心采样:其逻辑较简单,直接采用 Softmax 指示的最大概率选项作为输出。其优点是最大程度保证了生成消息的合法性,但输出结果过于固定,缺少多样性。其具体过程如下:

$$y = \underset{y_i}{\operatorname{argmax}}\{p(y_i) \mid i \in (1,n)\} \tag{4}$$

2)随机采样:基于输出的概率来随机选择对应项作为输出,其保留了生成的 Modbus 消息的多样性,但易出现不合法的消息。其具体过程如下:

$$\Pr(y = y_i) \propto p(y_i) \tag{5}$$

3.2　模型训练

由于 Modbus 消息可多达 260 Byte,难以逐个标记每个生成消息的状态,因此本文选择在无监督学习环境中训练上述 seq2seq 模型,即没有测试标签来评估学习模型的生成效果。采用生成消息的合法率,以及挖掘出对应仿真系统漏洞所用的时间作为最终的评估指标。把训练过程中遍历一次训练集的训练称作 1“代”,本文针对 5 代、10 代、15 代、20 代和 25 代训练 seq2seq 模型。在本文实验环境下,seq2seq 模型训练 1 代大约需要 70 s,训练 25 代约需要 30 min。本文使用了 LSTM 模型[19]和 2 个隐藏层,每个隐藏层包含 64 个隐藏状态。

3.3　Fuzzy-RNN 模糊测试

学习 Modbus/TCP 协议的生成模型的最终目标是执行模糊测试。从渗透测试的角度来说,一种完美的机器学习算法生成的协议消息序列不会造成任何正常抛出,如参数越界、无效格式等,同时又能尽可能快地找到内存越界、超限访问等异常抛出的点,从而发现漏洞。较差的机器学习算法会生成大量格式错误的协议消息序列,造成正常抛出,从而被服务器迅速拒绝,无法进行测试。为了探索正常、异常抛出的平衡点,本文提出一种 Fuzzy-RNN 模糊生成算法,RNN 的学习能力使得该算法能够避免正常抛出。在采样过程中,本文引入一个反随机策略,提升异常抛出的概率。

训练完毕之后,Modbus 消息序列的生成过程如算法 1 所示。

算法 1　Fuzzy-RNN 算法

输入　$D(x,\theta)$, $\Pr_f$, $\Pr_t$, “Modbus_head”

输出　*RES*

```
1. RES = "Modbus_head"
2. get LEN from "Modbus_head"
3. while RES.length < = LEN do
4. if RES.length > = 260 then
5. RES = "Modbus_head" //重置序列
6. end if
7. y,p(y) = sample(D(RES,θ))//从学习到的分布中采
//样 y
8. if random(0,1) > Pr_f and p(y) > Pr_t then //关键的反
//随机模糊策略
9. y = argmin_y' {p(y') ~ D(Seq,θ)}//y'代替 y
10. end if
11. RES = RES + y
12. end while
13. return RES
```

该算法需要输入学习到的条件分布 $D(x,\theta)$、模糊概率 $\Pr_f$、采样阈值概率 $\Pr_t$ 和初始化的 Modbus 消息头“Modbus_head”,最后输出 Modbus 结果序列 *RES*。该算法保证 *RES* 单元长度不会超出 Modbus 的上限 260 Byte 以及“Modbus_head”规定的长度 *LEN*。在每个循环中(当前时间戳为 t),算法利用条件分布来采样获得 $t+1$ 的字符单元 y,若概率 $p(y)$ 比采样阈值 $\Pr_t$ 高,则算法会选择最小概率的字符单元 y' 来代替 y 出现在该位置,从而生成许多临界参数的状况,增大漏洞挖掘的成功概率。这个输出字符单元还要根据随机函数 random(0,1) 和 $\Pr_f$ 的大小才能最终输出,通过设定随机函数的参数能更好地实现对模糊速率的控制。

4　实验结果与分析

本文主要实验设备的参数如下:使用 2017 年

11 月发布的 TensorFlow1.4(GPU 版本)作为本文实验的运行框架,Python 版本为 3.6.3,采用 Nvidia GTX1070 的 GPU 来加速训练,显存达 2 GB。该框架在 64 位 Windows 10 操作系统上运行,对应中央处理器为 4 核的 Intel-CORE i7-7700HQ,内存 8 GB。

4.1 数据来源与评价指标

本文的训练数据来源于实验室核心交换机采集的 Modbus/TCP 数据,包括已授权工程项目和实验室内部 ICP 仿真数据。挑选长度在 14 Byte ~260 Byte 的 50 000 条协议消息作为训练输入,每个训练批次为 200 条消息。测试设备接线情况如图 2,串接一个数据采集器来分析通信日志,该采集器可按协议对日志进行分类,以评估实验效果。

选择广泛应用的 GPF(General Purpose Fuzzer)[20] 作为对照组的模糊测试器。待检测的系统为 4 种 modbus 仿真软件,分别是 Modbus Slave、xMasterSlave、pymodbus 和 MOD_RSSIM。实验组和对照组的模糊测试器在充分训练之后,通过 TCP/502 端口发送模糊生成的协议消息实现渗透测试。

选用以下 2 个评价指标:

1)消息合法率(η_1):对于每次的模糊测试,以编程方式检查数据采集器中 Modbus/TCP 协议操作日志是否存在正常错误码,比如请求参数错误、不支持的功能、不支持的地址、不支持的数据类型等。如果没有,则将该次测试视为合法,否则视为不合法。该指标对于漏洞挖掘来说,其价值不在于直接增益,而是会反映模糊生成的质量,从而间接提高漏洞挖掘的效率。记每次测试发出的所有消息个数为 N_{all},所有合法信息个数为 N_{legal},则合法率计算如下:

$$\eta_1 = \frac{N_{\text{legal}}}{N_{\text{all}}} \times 100\% \tag{6}$$

2)漏洞挖掘效率:对于每次模糊测试,记录从第 1 个请求发起,到第 1 次检出异常错误码(如错误代码 3、从站无响应等)的时间间隔。该间隔越短说明漏洞挖掘效率越高。挖掘效率可用时间(t_D)表示,t_D 越大,挖掘效率越低,其计算过程如下:

$$t_D = t_{\text{abnormal}} - t_1 \tag{7}$$

4.2 合法率分析

将 GPF 和 2 种采样方式的 Fuzzy-RNN 进行对比,实验结果如图 6 ~ 图 9 所示。水平线代表 GPF 在该仿真软件上的表现,由于 GPF 不涉及学习过程,因此没有变化趋势。

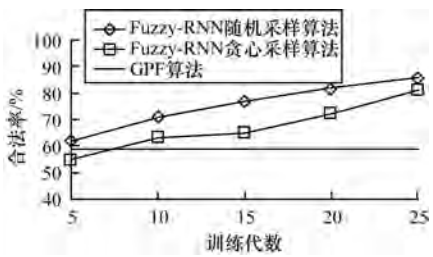

图 6 Modbus Slave 上的指令合法率

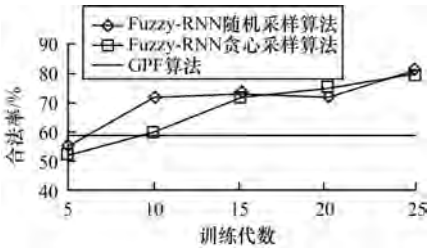

图 7 xMasterSlave 上的指令合法率

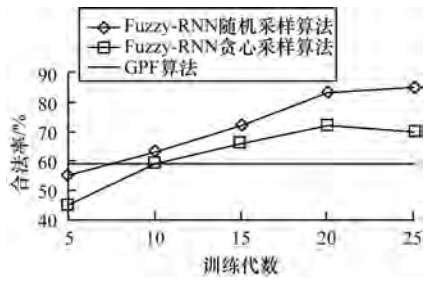

图 8 pymodbus 上的指令合法率

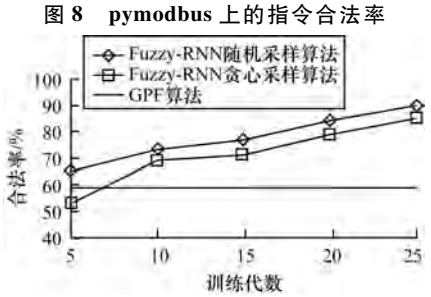

图 9 MOD_RSSIM 上的指令合法率

从 4 个实验整体来看,合法性指标上都有 GPF < Fuzzy-RNN(随机)≈ Fuzzy-RNN(贪心)。Fuzzy-RNN 算法在训练超过 10 代之后,合法率将明显超越 GPF 算法,其合法率上升趋势在 15 代以后显著放缓。GPF 算法的合法率平均数为 58.5%,而 Fuzzy-RNN 算法最终到达的合法率在 75% ~90%。Fuzzy-RNN 算法的贪心采样在 pymodbus 上最终合法率明显低于其他 3 组,随机采样在训练代数小于 20 时,其合法率也出现该现象。其原因在于该仿真软件支持的数据模式与参数范围有限,增加了随机生成中正常抛出的可能性。对于 15 代 ~25 代的 Fuzzy-RNN 算法,贪心采样的平均合法率比随机采样高 3.92%。由于 Modbus 协议的参数取值空间较大,随机采样造成的合法性下降对其影响不大,从侧面说明了学习模块的概率密度模型适用于针对该协议的预测。

4.3 漏洞挖掘效率分析

使用训练了 15 代的 Fuzzy-RNN 所产生的合法包对 4 个 Modbus 仿真器进行漏洞挖掘,其获得异常响应的耗时情况如表 2 和表 3 所示。

表 2 漏洞挖掘效率按帧数对比 帧

待测平台	Fuzzy-RNN 随机算法	Fuzzy-RNN 贪心算法	GPF 算法
Modbus Slave	700	1 300	1 500
xMasterSlave	200	380	500
pymodbus	1 000	2 000	12 000
MOD_RSSIM	200	220	300

表 3　漏洞挖掘时间对比　　min

待测平台	Fuzzy-RNN 随机算法	Fuzzy-RNN 贪心算法	GPF 算法
Modbus Slave	6.00	11.00	13.00
xMasterSlave	1.25	2.00	3.00
pymodbus	8.00	15.00	98.00
MOD_RSSIM	1.20	1.25	1.45

由表 2 和表 3 可知,使用随机采样的 Fuzzy-RNN 算法使得 Modbus Slave、xMasterSlave、pymodbus 的挖掘时间缩短了 50% 以上。而对于 MOD_RSSIM,由于其可支持的 Function Code 较多且仿真边界值设定范围较广,因此 RNN 阶段所能学习到的条件概率模型并不强,其时间增益主要来自算法 1 所描述的反随机策略。使用贪心采样的 Fuzzy-RNN 算法,其所带来的时间增益则较差。但总体来看,Fuzzy-RNN 算法在绝大部分 Modbus 仿真系统中都实现了模糊效率的提高。

从表 3 可知,随机采样相比贪心采样,漏洞挖掘时间平均降低了 43.74%,相对于 3.92% 的合法率减益,这是一个更加显著的性能提升。因此,本文涉及的最优 Modbus 模糊器模型是使用随机采样的 Fuzzy-RNN 算法。

4.4　测试覆盖率分析

由于测试的分支覆盖率在不同的训练集上不尽相同,难以直接测量,因此本文试图从理论上进行说明。

由算法 1 可知,本文算法是针对定长字符串的相互关系进行预测,状态机结构相对简单,同时该算法对出现概率较低的内容进行了概率补偿,可以在一定程度上降低覆盖率达 100% 时测试时间的上限。因此,只要测试的时间在可接受的范围内足够长,测试生成的包将覆盖训练集的所有分支,达到较高的分支覆盖率。

5　结束语

本文设计了基于 RNN 的 Modbus/TCP 协议模糊测试算法。结合模糊测试的思想与循环神经网络的预测性质来对 Modbus/TCP 协议的安全漏洞进行挖掘。理论分析和实验结果表明,该算法在节省特征工程的同时,能够显著提升生成消息的合法率,并大幅提高漏洞探查效率。下一步将把该模型扩展到其他 ICP 中,同时改进采样函数,使得该算法在其他协议中仍能得到较好效果,保障工业控制领域的网络安全。

参考文献

[1] 潘洪跃. 基于 MODBUS 协议通信的设计与实现[J]. 计量技术,2002(4):35-36.

[2] 李伟明,张爱芳,刘建财,等. 网络协议的自动化模糊测试漏洞挖掘方法[J]. 计算机学报,2011,34(2):242-255.

[3] MCNALLY R, YIU K, GROVE D, et al. Fuzzing: the state of the art[EB/OL]. [2018-04-01]. http://pdfs.semanticscholar.org/a80c/60dbf12377144083e05146117881a961e459.pdf.

[4] HANFORD K V. Automatic generation of test cases[J]. IBM Systems Journal,1970,9(4):242-257.

[5] 刘为. 基于模糊测试的 XSS 漏洞检测系统研究与实现[D]. 长沙:湖南大学,2010.

[6] LÄMMEL R, SCHULTE W. Controllable combinatorial coverage in grammar-based testing[C]//Proceedings of IFIP International Conference on Testing of Communicating Systems. Berlin, Germany: Springer, 2006: 19-38.

[7] GURUTEJA V V, BISWAS M. Edge detection algorithm using dynamic fuzzy interface system[C]//Proceedings of Innovations in Electronics and Communication Engineering. Berlin, Germany: Springer, 2018: 287-295.

[8] 高刃,唐龙,伍爵博. 基于神经网络的无线传感器网络数据预测应用研究[J]. 计算机科学,2012,39(5):44-47.

[9] PARISOTTO E, MOHAMED A R, SINGH R, et al. Neuro-symbolic program synthesis[EB/OL]. [2018-04-01]. https://arxiv.org/pdf/1611.01855.pdf.

[10] 郝怡然,盛益强,王劲林,等. 基于递归神经网络的网络安全事件预测[J]. 网络新媒体技术,2017,6(5):54-58.

[11] BHATIA S, SINGH R. Automated correction for syntax errors in programming assignments using recurrent neural networks[EB/OL]. [2018-04-01]. https://arxiv.org/pdf/1603.06129.pdf.

[12] 李武波,张蕾,舒鑫. 基于 Seq2Seq 的生成式自动问答系统应用与研究[J]. 现代计算机(专业版),2017,36(12):57-60.

[13] HUITSING P, CHANDIA R, PAPA M, et al. Attack taxonomies for the Modbus protocols[J]. International Journal of Critical Infrastructure Protection, 2008(1): 37-44.

[14] 张典波. Peach 在工业控制系统漏洞挖掘中的改进及应用[D]. 北京:北京邮电大学,2016.

[15] 张亚丰,洪征,吴礼发,等 基于范式语法的工控协议 Fuzzing 测试技术[J]. 计算机应用研究,2016,33(8):2433-2439.

[16] XIONG Qi, LIU Hui, XU Yuan, et al. A vulnerability detecting method for Modbus-TCP based on smart fuzzing mechanism[C]//Proceedings of 2015 IEEE International Conference on Electro/Information Technology. Washington D.C., USA: IEEE Press, 2015: 404-409.

[17] 朱小襄. ModBus 通信协议及编程[J]. 电子工程师,2005,31(7):42-44.

[18] BENGIO Y, SIMARD P, FRASCONI P. Learning long-term dependencies with gradient descent is difficult[J]. IEEE Transactions on Neural Networks, 1994, 5(2): 157-166.

[19] SUNDERMEYER M, SCHLÜTER R, NEY H. LSTM neural networks for language modeling[C]//Proceedings of the 13th Annual Conference of the International Speech Communication Association. [S.l.]: International Speech Communication Association, 2012: 1-4.

[20] DEMOTT J D, ENBODY R J, PUNCH W F. Revolutionizing the field of grey-box attack surface testing with evolutionary fuzzing[EB/OL]. [2018-04-01]. https://www.vdalabs.com/tools/EFS.pdf.

编辑　樊丽娜