·网络空间安全· 文章编号：1000-3428(2023)12-0129-07 文献标志码：A 中图分类号：TP309

一种基于协议格式智能推断的灰盒测试技术

刘华玉[1]，甘水滔[2,3]，尹小康[1]，柳晓龙[2]，刘胜利[1]，李宏亮[4]

（1.战略支援部队信息工程大学 网络空间安全学院，郑州 450001；2.数学工程与先进计算国家重点实验室，江苏 无锡 214215；3.清华大学 网络研究院，北京 100084；4.江南计算技术研究所，江苏 无锡 214083）

摘 要：通信协议可保障网络应用和物联网设备之间的通信，但其在设计或实现中存在的脆弱性会带来严重的安全威胁和隐患。模糊测试技术作为一种软件安全分析的有效方法，在针对网络协议的脆弱性分析中表现出高效的性能和无可比拟的优势。现有的针对网络协议的灰盒测试技术仍依赖于人工识别协议格式来辅助测试，并且变异策略的设计更偏向于位和字节的变异，忽略了协议消息本身的格式信息，导致在测试时性能不佳。针对上述问题，提出一种基于对齐聚类的智能化协议格式推断模型ProCluster，用于指导灰盒测试中协议状态机构建和种子的变异。该模型通过自动提取协议关键字和推断相应类型，辅助协议灰盒测试模型构建更精准的种子变异策略，从而生成更符合协议规范的测试用例，以此加速提升模糊测试的代码覆盖能力和脆弱路径发现能力。实验结果表明，在对TinyDTLS、OpenSSL等程序的模糊测试中，与典型协议灰盒测试工具AFLNet相比，ProCluster的边覆盖率能够提升75%~182%，并且在TinyDTLS中发现一个缓冲区溢出漏洞样本。

关键词：灰盒测试；协议逆向；变异策略；网络协议；漏洞挖掘

开放科学（资源服务）标志码（OSID）：

中文引用格式：刘华玉，甘水滔，尹小康，等. 一种基于协议格式智能推断的灰盒测试技术[J]. 计算机工程，2023，49(12)：129-135，145.

英文引用格式：LIU H Y，GAN S T，YIN X K，et al. A gray-box test technology based on intelligent inference of protocol format[J]. Computer Engineering，2023，49(12)：129-135，145.

A Gray-box Test Technology Based on Intelligent Inference of Protocol Format

LIU Huayu[1]，GAN Shuitao[2,3]，YIN Xiaokang[1]，LIU Xiaolong[2]，LIU Shengli[1]，LI Hongliang[4]

（1.School of Cyberspace Security，Strategic Support Force Information Engineering University，Zhengzhou 450001，China；2.State Key Laboratory of Mathematical Engineering and Advanced Computing，Wuxi 214215，Jiangsu，China；3.Institute for Network Research，Tsinghua University，Beijing 100084，China；4.Jiangnan Institute of Computing Technology，Wuxi 214083，Jiangsu，China）

【Abstract】Communication protocols ensure secure communication between network applications and IoT devices. However，the fragility of their design and implementation can result in serious security threats and hidden dangers. Fuzzing，as an effective method for software security analysis，demonstrates efficient performance and unparalleled advantages in vulnerability analysis of network protocols. Nevertheless，existing stateful coverage-based grey-box fuzzing for network protocols still relies on manual identification of the protocol format to assist testing. In addition，the design of the mutation strategy is more prominent in the mutation of bits and bytes，disregarding the format information of the protocol message itself，resulting in suboptimal fuzzing performance. To address these issues，this study proposes an intelligent protocol format based on an aggregate class. This model uses high-efficiency and automatic extraction of protocol keywords and infers corresponding types to assist in building a more accurate seed mutation strategy for the protocol gray-box test model. This approach generates test cases that better conform to the specifications of protocols，thereby accelerating code coverage，improving illegal testing capability，enhancing the capacity of the fuzzy test，and increasing the ability to identify fragile paths. The experimental results demonstrate that，when fuzzing programs such as TinyDTLS and OpenSSL，ProCluster outperforms the typical stateful gray-box fuzzing tool AFLNet by increasing edge coverage by 75% to 182%. Furthermore，it successfully identified a buffer overflow vulnerability sample in TinyDTLS.

【Key words】gray-box test；protocol reverse；mutation strategy；network protocols；vulnerability mining

DOI：10. 19678/j. issn. 1000-3428. 0066813

基金项目：中国博士后科学基金面上资助项目（2021M701942）。

作者简介：刘华玉（1997—），男，硕士研究生，主研方向为模糊测试、漏洞挖掘；甘水滔，副研究员、博士；尹小康，博士；柳晓龙，助理研究员、博士；刘胜利，教授、博士；李宏亮，研究员、博士。

收稿日期：2023-01-23 **修回日期：**2023-03-07 **E-mail：**jufenglhy@alumni.sjtu.edu.cn

0 概述

随着互联网的发展,网络安全问题近年来备受关注。网络协议作为互联网通信的基础,其安全性一直是学术界面临的重大难题。协议在设计和实现时可能存在安全问题,网络通信的易达性和传播性会严重威胁用户的隐私安全以及整个互联网络的安全。

2017年5月爆发的WannaCry勒索软件攻击,利用“永恒之蓝”漏洞广泛传播,短时间内超过150个国家和30万用户遭到攻击,给全球大量计算机用户造成数十亿美元的经济损失[1]。2020年,360CERT监测发现,国外研究团队发布的DNS协议实现的逻辑错误,攻击者通过发起指向恶意name-server的DNS查询请求,可以造成递归服务器和特定域名服务器拒绝服务影响[2]。这些协议级漏洞利用非法用户数据被服务器解析后,服务器产生非预期的执行行为,最终造成拒绝服务攻击或者远程执行攻击等。因此,针对网络协议实现的漏洞挖掘技术研究是当前的主流方向之一。

1990年,MILLER等[3]提出模糊测试概念,主要用来检测目标程序的鲁棒性。目前模糊测试技术[4-6]已经成为一种重要的程序脆弱性分析方法,在网络协议安全研究中具有自动化程度高、效率高等优势。现有的针对网络协议的模糊测试技术主要有以Peach[7]为主的黑盒测试方法和以AFLNet[8]为主的灰盒测试方法。Peach等黑盒测试需要协议的格式信息,依赖于人工知识储备,人工成本高,可扩展性较差。AFLNet是首次提出针对协议的灰盒测试方法,仅需要少量协议信息即可指导完成测试,但在测试过程中忽略了协议本身的格式信息,变异策略随机性强,导致测试效率低等问题。

为解决当前研究中存在的问题,本文提出一种基于协议逆向工程自动识别协议格式来辅助模糊测试的方法,主要包括:基于对齐聚类的智能化协议格式推断模型和协议格式信息辅助的变异策略优化。该方法在对TinyDTLS、OpenSSL等程序的模糊测试中不仅能够提高目标程序的边覆盖率,而且在最新版本TinyDTLS程序中触发一个缓冲区溢出类型崩溃,该崩溃在AFLNet中无法触发。

1 相关工作

1.1 网络协议模糊测试

近年来,针对网络协议的模糊测试得到了广泛的研究。现有的协议模糊测试研究方法主要分为基于测试用例生成和基于种子变异两类[9]。

目前基于测试用例生成的模糊测试主要以黑盒测试[10-11]为主,针对网络协议的测试,代表工具有Protos[12]以及Peach。2001年,KAKSONEN等[12]提出了Protos,实现了针对网络协议的模糊测试,这也标志着模糊测试技术成为程序脆弱性分析的实用性工具。Protos为不同的网络协议提供了不同的测试用例集,通过故障注入的方式测试协议软件安全性,测试人员通过指定协议格式中的字段来生成测试用例。Peach刚开始主要被用于文件的模糊测试中,经过多次升级改进,目前适用于测试网络协议等多种类型程序,而且至今仍然在被使用。

基于种子变异的协议模糊测试方法以灰盒测试为主。近年来,随着AFL[13]的提出,通过程序轻量级插桩实现基于覆盖率引导的灰盒测试方法成为主流模糊测试方式。AFLNet是针对状态网络协议的灰盒测试工具,AFLNet以覆盖率为导向,采用插桩的方法监控程序边路径覆盖情况,引导种子变异以探索更多的路径。AFLNet仅需要少量的协议知识用于提取请求消息和构建协议状态机,自主性高,并且易于扩展,是目前成熟且应用最为广泛的协议模糊工具。此后,针对网络协议的灰盒测试技术得到发展,研究人员提出并实现了许多工具。StateAFL[14]是一种将被测程序内存状态映射为唯一的状态标识符,从而推断目标服务器的当前协议状态,无须手动为协议定制。SPFuzz[15]通过预定义协议规范等来增强测试用例的有效性。SNPSFuzzer[16]提出一种基于快照的测试方法,通过在特定位置存储程序的上下文信息,并在需要模糊相关状态时恢复上下文信息,提升测试速度。这些工具通过协议状态推断、协议分析、种子功率调度[17]等多方面的改进,在实际应用过程中取得一定进展。

1.2 协议逆向

协议逆向技术是在没有或少量协议规范知识的条件下,通过对协议实体的网络输入输出和指令执行流程等进行跟踪分析,提取网络协议基本描述和状态机等信息[18-20]。根据现有的研究方法,协议逆向技术主要分为基于Network Trace和基于Tainted Data两类。

基于Network Trace的协议逆向技术,即基于报文的协议逆向技术,依赖于wireshark等抓包工具捕获协议流量进行逆向分析的技术。该技术针对协议字段的取值变化和特征推断协议的信息,典型的代表是由文献[21]在2004年启动并发布的PI(Protocol Informatics)项目。该项目引入了多序列对比算法[22],并根据相同类型报文聚类的统计特征对报文格式进行分析。Discover[23]使用报文序列分析方法实现完整的协议格式提取,该技术在推断消息格式时模拟报文解析的过程,能够有效识别依赖于消息本身的字段。

基于Tainted Data的协议逆向技术,也即基于指令序列的协议逆向技术,主要通过动态污点分析技术跟踪分析服务器程序对报文的解析流程,根据程序对报文的解析来推断协议格式。Polyglot[24]提出利用该技术自动解析协议格式,Polyglot采用

shadowing方法监视程序对应用数据的处理过程，根据数据处理过程中的二进制信息来提取有关字段边界和关键字的信息。AutoFormat[25]提出基于指令轨迹的协议识别技术，该技术可分为上下文感知的执行监控模块和协议字段识别模块。

基于Tainted Data的协议逆向技术利用协议实现的详细数据处理流程，可以在逆向工程中实现较高的精度，但存在固件获取、二进制程序混淆等困难。基于Network Trace的协议逆向技术只需考虑通信过程的数据包，不需要程序源码，具有良好的逆向效果。

1.3　存在的问题

针对日益复杂的网络协议，模糊测试技术是目前最为实用的安全分析技术之一，但现有的研究方案依然存在以下问题：

1）人工成本高。现有的网络协议黑盒测试工具需要大量的协议知识辅助测试，依赖人工构建协议模型和数据模型。实验时要求测试人员需要精确掌握协议规范，还要熟悉工具的使用，以此才能够构建耦合效果良好的模型，这不仅给测试人员增加了大量的工作，而且协议测试的效果完全取决于模型的好坏，这使得工具的可扩展性变差。

2）变异策略缺乏协议格式信息指导。代码块1为TinyDTLS程序中处理收到消息的部分源码。

```
1.static unsigned int
2.is_record(uint8 *msg,size_t msglen) {
3.unsigned int rlen = 0;
4. if (msglen >= DTLS_RH_LENGTH /* FIXME allow empty records? */
5.#ifdef DTLS_CHECK_CONTENTTYPE
6.&& strchr(content_types,msg[0])
7.#end if
8.&& msg[1] == HIGH(DTLS_VERSION)
9.&& msg[2] == LOW(DTLS_VERSION))
10.{
11.rlen = DTLS_RH_LENGTH +
12.dtls_uint16_to_int(DTLS_RECORD_HEADER(msg)->length);
13.
14./* we do not accept wrong length field in record header */
15.if(rlen > msglen)
16.rlen = 0;
17.}
18.return rlen;
19.}
```

研究发现，程序在这部分会将收到消息的实际长度msglen与消息内的长度字段length进行比较（第15行）。如果变异得到的测试用例无法通过此校验，将无法探索更深的路径。然而现有的状态协议灰盒测试方法大多是作为AFL的扩展，直接采用AFL中的随机变异策略，忽略了协议格式信息，很多测试用例无法通过服务器的初步数据校验（第8、9、15行），造成大量的无效测试用例，降低了测试的性能。

为解决上述问题，本文提出一种基于对齐聚类的智能化协议格式推断模型ProCluster，用于指导灰盒测试中协议状态机构建和种子的变异。该模型通过自动提取协议关键字和推断相应类型，来辅助种子变异策略生成更符合协议规范的测试用例，以此加速提升模糊测试的代码覆盖能力和脆弱路径发现能力。

2　本文方法

为了解决面向通信协议的模糊测试中代码覆盖率低、变异策略缺乏协议信息指导等问题，本文将协议逆向技术应用到灰盒测试中，提出一种基于对齐聚类的智能化协议灰盒测试方法ProCluster。考虑到协议逆向的成本，利用基于Network Trace的协议逆向技术获取协议格式，然后根据设计的变异策略在协议格式信息的指导下对种子进行变异，得到更多符合预期的测试用例用于对网络协议的测试。

2.1　框架设计

本文所提灰盒测试方法的框架如图1所示，主要包括协议识别、状态机构造以及种子变异3个模块。ProCluster首先利用协议识别模块将前期通过wireshark等工具抓取的数据包进行自动化解析，然后将得到的协议格式等结果作为模糊测试引擎的输入，来辅助后续模糊测试工作。

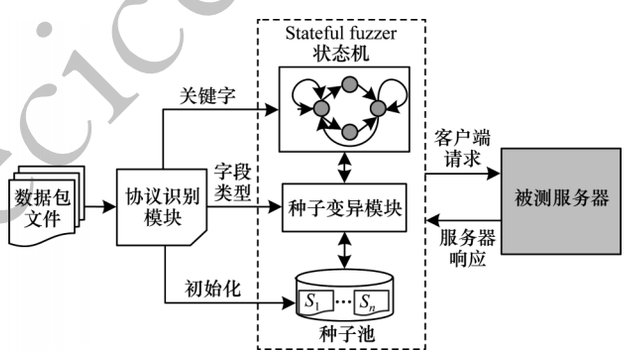

图1　基于协议格式智能推断的灰盒测试技术框架

Fig.1　Framework of gray-box test technology based on intelligent inference of protocol format

协议识别、状态机构造以及种子变异3个模块的介绍如下：

1）协议识别模块

该模块基于Network Trace的协议逆向技术实现。该模块以原始网络数据包为输入，经过预处理得到消息序列作为初始种子，进一步推断解析得到协议关键字和协议格式，用于后续构造状态机和种子变异。

2）状态机构造模块

该模块主要利用前一阶段中识别的关键字作为

响应消息的状态识别码,以此在测试过程中自动化构建状态转移图,更进一步指导变异过程。与AFLNet相比,本文提出的方法省略了人工提取协议状态的过程,无须手动为协议定制,使得测试更为自动化。

3)种子变异模块

针对变异策略缺乏协议格式信息指导问题,该模块以AFLNet为原型,在原有变异规则基础上设计消息序列变异策略,并根据协议格式信息实现针对协议字段的变异策略,使模糊工具能够更加高效地发现一些常规的安全漏洞。

2.2 协议格式识别模块

本文采用基于Network Trace的协议识别技术。分析此类技术现有的研究方法又可以分为基于对齐聚类方法和基于token的方法。基于对齐聚类方法,现有的研究主要通过利用各种对齐算法对齐消息并计算消息的相似性分数,得到的结果与阈值进行比较,依据此进行聚类并分析。但是预设置的阈值对协议本身敏感,因此这些方法鲁棒性低。基于token的研究方法将消息拆分为token,然后根据token值对消息进行分组。这类方法依赖于预定义的token分隔符,而且将字段限制在token级别,在分析过程中很容易造成冗余,降低正确率。

本文考虑在对齐消息的基础上,利用概率分析的方法提取状态协议中的消息状态,即消息中的关键字,然后依据关键字完成消息聚类,最后对簇内的消息进行格式推断解析。协议格式识别算法如下:

算法1 协议格式识别算法

输入 wireshark等工具抓取的pcap数据包

输出 protocol format

```
1./*获取协议消息数据*/
2.M←get_message(*.pcap)
3./*执行多序列比对*/
4.M←MSA(M)
5.K←∅
6.for field in M do
7.if field.type is dynamic then
8./*生成候选关键字*/
9.K←K∪field
10.end if
11.end for
12.for K in K do
13./*计算候选关键字的概率*/
14.calculate(p_k)
15.end for
16./*选择概率最大的作为关键字*/
17.K←K[max(p_k)]
18./*聚类后按字段恢复格式*/
19.clustering(k) and format inference
```

1)多序列比对算法

现有研究中针对协议序列比对大多使用双序列比对算法[26-27],而考虑到报文序列较多情况下会影响整体的性能,本文使用多序列比对算法对报文序列进行对齐操作。文献[28]针对三类多序列比对算法在报文序列上的性能和效果做出了分析和测试,本文在此基础上考虑使用基于迭代细化策略的渐进比对算法。该方法首先对齐最相似的报文序列,然后将其他序列逐步添加到对齐结果中,随后迭代地重新排列初始全局比对结果的序列子集,以提高精度。

2)生成候选关键字

本节考虑为了使生成的候选关键字尽最大可能包含协议的真实关键字,需要保守地构造候选关键字列表。

利用上文多序列对齐的结果,将报文分割为字段。针对文本类协议主要依据“空格”等分隔符进行字段划分,针对二进制协议,以字节为单位进行字段划分。根据前期初步对齐结果分析所有字段在所有消息中值的变化。对此,可以区分出静态字段(该字段在所有消息中不发生变化,标记为“S”)和动态字段(标记为“D”),对于连续的静态字段可以合并成更大的静态字段。为了构建保守的候选关键字列表,本文不仅将单位字节的动态字段视为候选关键字,而且在单位字节的基础上考虑将连续的动态字段整体作为一个候选关键字。考虑实际的协议设计以及避免造成后续计算浪费,本文设置候选关键字长度的阈值为3。至此,生成了一个保守的候选关键字列表。

3)概率分析

本节对候选关键字进行概率分析,筛选出最为可能的协议关键字。本文主要基于两个直观的观察来对候选关键字进行概率分析,分别是根据候选关键字聚类后簇的相似度得分以及簇规模。

根据前期的对齐结果,计算两两报文序列的相似度得分,并以此构造报文序列的相似度得分矩阵。然后利用相似度矩阵计算聚类后簇内的相似度得分和簇间的相似度得分。直观地认为簇内的相似度得分应该总是大于簇间的相似度得分,然而情况并不理想,两类得分会出现重叠的现象。

现有研究中大多直接根据相似度得分与阈值之间的大小关系来进行判断,但在实际应用中阈值对协议依赖性强,固定的阈值在协议逆向分析中很难得到置信的结果。基于此,本文提出假设:基于阈值的聚类结果中存在同类型消息被聚类到不同的簇中,不同类型的消息被聚类到同一簇中,即错误不匹配(False Non-Match, FNM)和错误匹配(False Match,FM)。根据针对多序列比对的研究[29],本文设置取值从0~1的阈值t,通过错误匹配率曲线和错误不匹配率曲线得到等错误率(Equal Error Rate, EER)值,该值描述聚类结果总体准确性。分别给出错误匹配率和错误不匹配率的计算公式,如式(1)、式(2)所示:

$$R_{\mathrm{FM}}=\frac{\mathrm{num}(S_{\mathrm{inter}\_i}>t)}{N} \tag{1}$$

$$R_{\mathrm{FNM}}=\frac{\mathrm{num}(S_{\mathrm{inner}\_j}<t)}{M} \tag{2}$$

其中：R_{FM}表示错误匹配率；R_{FNM}表示错误不匹配率；S_{inter_i}为第i个簇间相似度分数；N为簇间相似度分数的个数，$i \in [1,N]$；S_{inner_j}为第j个簇内相似度分数；M为簇内相似度分数的个数，$j \in [1,M]$。

最后根据式(3)得到一个关于当前候选关键字基于相似度分数的置信度值p：

$$p = 1 - E \tag{3}$$

其中：E为等错误率值。

本文在研究过程中发现，对于基于相似度分数的概率，存在一个突出问题。当存在某个候选关键字聚类后有过多的簇时，此时该候选关键字的置信度会比较高，对后续格式分析干扰严重。因此，对该值设置一个基于簇规模的权重。

针对簇规模，本文分析真正的关键字聚类后的结果，得出以下结论：针对非刻意捕获的报文序列，生成的簇数量适中，且很少存在单个消息为独立簇。根据上述结论，给出候选关键字基于簇规模的权重配置w：

$$w = 1 - \frac{N_{single-message}}{N_{cluster}} \tag{4}$$

其中：$N_{single-message}$表示仅有一条消息的簇数量；$N_{cluster}$表示簇总数。

最后本文利用因子图计算得到候选关键字的后验概率，概率最大的即为协议关键字。假设多序列对比没有正确对齐，因此无法正确识别关键字。尽管如此，根据前面分析得到的关键字进行聚类，簇内消息的结构差异可能会减少。因此，对于每个簇，执行多序列对比和概率关键字分析。随后将新的关键字与原始关键字进行比较，如果新的关键字可以在所有消息获得更好的区分效果，则用新的关键字替换原始关键字。重复该过程，直到无法识别更好的关键字。

4)格式推断

对于最终的聚类结果，每个簇中均为同一类型消息。最后本文利用启发式方法对消息中的长度、cookie等字段进行恢复。对于长度字段(标记为“L”)，本文主要基于消息长度的差值与同一类型消息中对齐字段的差值进行比较确定。针对cookie字段(标记为“C”)，使用RolePlayer[30]中的启发式方法来识别。

利用wireshark抓取的DTLS(Datagram Transport Layer Security)协议数据报文对本文提出的模型进行测试。如图2所示，本文提出的模型能够区分协议数据中的静态字段和动态字段，准确识别协议数据的关键字，并在后续推断分析时能够识别到DTLS协议中的长度字段。本文的目标是利用协议逆向技术来辅助模糊测试，不进行精准的协议格式解析工作，因此图2展示的结果中大多只分析静态字段或动态字段，不进行深入解析。

字段位置	字段值	本文方法解析结果	Wireshark解析结果
0	14	key word	Content Type
1	fe	S	Version
2	fd		
3	00		Epoh
4	00	D	
5	00	S	Sequence Number
6	00		
7	00		
8	00		
9	00	D	
10	05	D	
11	00	L	Length
12	01		
…			

图2　协议识别结果

Fig.2　The results of protocol identification

2.3　种子变异模块

本模块在AFLNet原有的变异策略基础上分析其中的不足之处，并提出新的优化变异策略。AFLNet现有的变异策略包括确定性变异和随机变异阶段。确定性变异阶段主要包括位和字节的变异，这类针对数据报文的变异策略在实际中很难有出色的表现，而且会耗费大量的时间。因此，大多在实际操作过程中禁用确定性变异阶段，直接进入随机变异阶段。在运行过程中，随机变异阶段通常占有较大比例，而且该阶段也通常会覆盖更多的新路径。但是随机变异阶段对种子的变异盲目随机，这会产生大量无效的种子，从而造成模糊效率低下的问题。本文提出消息序列变异策略，并针对原有随机变异策略进行优化。

首先本文设计并实现消息序列变异策略。为了能够变异得到高度切合程序处理的消息，将当前选择的消息完整地插入该消息序列的任意位置，保持了消息的完整结构，也期待能够获得更多的状态转换。

然后优化随机变异阶段对种子的变异。结合前面对网络报文的分析，得到在报文序列中存在一些静态字段，而且可以直观地感受到编程人员在处理这些字段时并不会花费过多的篇幅。也就是在模糊测试过程中变异这类字段，很难会覆盖到新的执行路径，那么在随机变异阶段再对这类字段进行变异，往往没有收获，还降低了模糊测试的效率。因此，为了降低随机变异阶段种子变异的盲目性，本文提出大幅减小甚至不去对种子的静态字段进行变异操作，降低了目标程序的执行次数。

另外,前期协议逆向中除了区分静态字段和动态字段外,也分析得到了长度字段。针对长度字段,本文提出的变异方法是利用一些特殊值替换该字段的原有值,以此来测试该字段能否触发程序崩溃,如表1所示。

表1 字段变异策略

Table 1 Field mutation strategy

字段类型	策略
长度	随机值 边界值
静态	0 NULL

对于报文中的其他字段,本文仍以AFLNet中的随机变异机制为主要变异方法。为降低变异过程中的随机性,本文针对前面提到的静态字段和长度字段进行调整,恢复静态字段的原有值,并将长度字段赋值为变异后报文的长度。

3 实验评估

本文的实验是在一台Ubuntu 18.04系统上完成的。实验中提出两个指标来评价模糊测试工具的性能:一个是触发程序崩溃的数量;另一个是路径覆盖,即在测试过程中被测程序覆盖的路径数。本文针对DTLS协议和TLS(Transport Layer Security)协议进行了测试评估:在Eclipse的TinyDTLS公开库上测试DTLS协议;在OpenSSL公开库上测试TLS协议。为了保证实验的公平性,本文选取相同的输入在同一台虚拟机中完成实验。

3.1 边覆盖能力评估

本文以AFLNet作为基准程序,利用边覆盖对所提方法进行评估。利用AFLNet和本文设计的工具ProCluster分别对TinyDTLS和OpenSSL公开库进行24 h测试,并分析了测试结果。表2所示为最终实验结果。

表2 在TinyDTLS和OpenSSL上的测试结果

Table 2 Test results on TinyDTLS and OpenSSL

程序	AFLNet	ProCluster
TinyDTLS	208	587(182.2%↑)
OpenSSL	305	534(75.1%↑)

由于这两类协议均为加密传输协议,测试主要集中在协议握手阶段,很难覆盖到后续的状态转换,因此覆盖率都保持相对较低的水平。可以看到,本文所提方法ProCluster在两个公开库的测试中均表现优于AFLNet。结果表明,本文所提的方法具有更好地发现更多路径的能力。

本文进一步比较分析了边覆盖数在两个公开库中的实时性能,如图3所示。

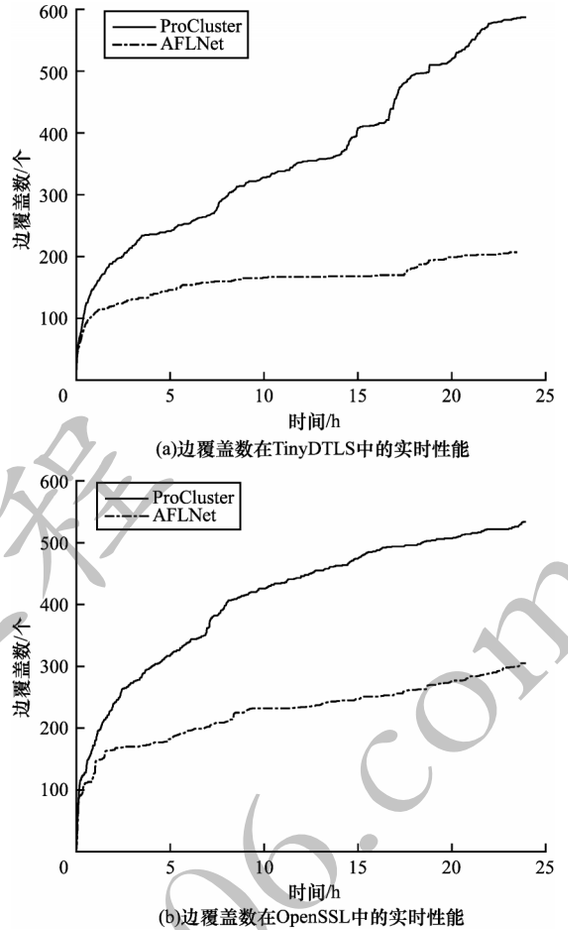

图3 边覆盖数的实时性能比较

Fig.3 Real-time performance comparison of the number of edge coverages

AFLNet在随机变异阶段随机性强,而且变异得到的消息很难保持消息内的依赖关系,这样会产生大量无效测试用例,最终导致性能不佳。根据观察,本文方法ProCluster能够发现更多的状态转换和路径覆盖,主要依赖于设计的基于消息序列的变异策略,以及在随机变异得到新的消息时会尝试恢复静态字段、长度字段等消息内依赖关系,使得消息更可能符合规范。ProCluster能够解决代码块1中所处理的长度字段等消息内依赖关系,变异得到测试用例后,恢复消息内的静态字段(代码块1中第8行msg[1]的数据和9行msg[2]的数据),并根据消息实际长度对消息内长度字段length(代码块1中第12行的数据)赋值,使测试用例可以通过目标程序对这些特殊字段的校验(代码块1第8、9、15行),更进一步去深入探索被测程序的功能路径。而AFLNet利用随机变异策略得到的测试用例,几乎无法通过相应的数据校验(代码块1第8、9、15行),不能更深入地探索程序路径。因此,本文的方法在实验时能够获得更高的程序覆盖率。

3.2 触发崩溃路径产生的能力评估

DTLS即数据包传输层协议。该协议在TLS协议基础上进行扩展,使之能够支持UDP协议。

TinyDTLS是DTLS协议的一个公开实现。在针对TinyDTLS公开库的模糊测试过程中,ProCluster多次触发程序崩溃,而基准对比工具AFLNet在测试时没有触发任何崩溃。

对程序崩溃进行分析,发现当服务器接收数据的长度大于设置的最大长度时,服务器会自动截断多余数据,但在后续处理时只考虑接收数据的大小和消息内长度依赖之间的关系,忽略了实际保存在指针中的数据长度小于该值,最终造成缓冲区溢出。因为服务器在处理过程中会校验消息的长度字段值,ProCluster能够在变异得到测试用例后,根据消息的实际长度自动对消息内长度字段进行赋值,而AFLNet变异得到的测试用例无法通过代码块1中的长度字段校验,因此在测试过程中不能触发这类崩溃。

4 结束语

本文结合协议逆向分析技术设计并实现一种新的状态协议灰盒测试方法。ProCluster基于对齐聚类实现对协议格式的自动推断,能够实现在消息字段级别进行变异操作,以生成更多符合规范的消息,并最大可能减少无效消息,从而获得更好的性能。将该方法与AFLNet在TinyDTLS和OpenSSL两个公开库上进行测试,结果表明,本文所提的方法在路径覆盖以及崩溃触发各方面都优于AFLNet。在针对协议实现的模糊测试研究中,基于字段的变异策略不仅能够产生更符合程序预期的消息,而且能够避免同基于位和字节的变异策略一样产生大量无效的消息。本文方法在字段划分时依然不够精细化,下一步将研究实现像wireshark一样的消息解析器,以区分出更细粒度的字段。

参考文献

[1] Kaspersky. What is wannacry ransomware?[EB/OL]. [2022-11-20]. https://www.kaspersky.com.au/resourcecenter/threats/ransomwarewannacry.

[2] 360-CERT. NXNSAttack:DNS协议安全漏洞通告[EB/OL]. [2022-11-20]. https://mp.weixin.qq.com/s/jYxHvIgPyeQknX0dNTZ0Zg.
360-CERT. NXNSAttack:notification of DNS protocol security vulnerabilities[EB/OL]. [2022-11-20]. https://mp.weixin.qq.com/s/jYxHvIgPyeQknX0dNTZ0Zg. (in Chinese)

[3] MILLER B P,FREDRIKSEN L,SO B. An empirical study of the reliability of UNIX utilities[J]. Communications of the ACM,1990,33(12):32-44.

[4] MANÈS V J M,HAN H,HAN C,et al. The art,science,and engineering of fuzzing:a survey[J]. IEEE Transactions on Software Engineering,2021,47(11):2312-2331.

[5] GODEFROID P. Fuzzing[J]. Communications of the ACM,2020,63(2):70-76.

[6] BOEHME M,CADAR C,ROYCHOUDHURY A. Fuzzing:challenges and reflections[J]. IEEE Software,2020,38(3):79-86.

[7] EDDINGTON M. Peach fuzzer[EB/OL]. [2022-11-20]. https://peachtech.gitlab.io/peach-fuzzer-community.

[8] PHAM V T,BÖHME M,ROYCHOUDHURY A. AFLNET:a grey-box fuzzer for network protocols[C]//Proceedings of the 13th IEEE International Conference on Software Testing,Validation and Verification. Washington D. C.,USA:IEEE Press,2020:460-465.

[9] SUTTON M,GREENE A,AMINI P. Fuzzing:brute force vulnerability discovery[M]. [S. 1.]:Addison-Wesley,2007.

[10] AITEL D. The advantages of block-based protocol analysis for security testing[EB/OL]. [2022-11-20]. https://citeseerx.ist.psu.edu/viewdoc/download;jsessionid.

[11] BANKS G,COVA M,FELMETSGER V,et al. SNOOZE:toward a stateful NetwOrk prOtocol fuzZEr[M]. Berlin,Germany:Springer,2006.

[12] KAKSONEN R,LAAKSO M,TAKANEN A. Software security assessment through specification mutations and fault injection[M]. Berlin,Germany:Springer,2001.

[13] ZALEWSKI M. American fuzzy lop fuzzer[EB/OL]. [2022-11-20]. https://lcamtuf.coredump.cx/afl/.

[14] NATELLA R. StateAFL:grey-box fuzzing for stateful network servers[J]. Empirical Software Engineering,2022,27(7):191.

[15] SONG C X,YU B,ZHOU X,et al. SPFuzz:a hierarchical scheduling framework for stateful network protocol fuzzing[J]. IEEE Access,2019,7:18490-18499.

[16] LI J Q,LI S Y,SUN G,et al. SNPSFuzzer:a fast greybox fuzzer for stateful network protocols using snapshots[EB/OL]. [2022-11-20]. https://arxiv.org/abs/2202.03643.pdf.

[17] LIU D G,PHAM V T,ERNST G,et al. State selection algorithms and their impact on the performance of stateful network protocol fuzzing[C]//Proceedings of IEEE International Conference on Software Analysis,Evolution and Reengineering. Washington D. C.,USA:IEEE Press,2022:720-730.

[18] WONDRACEK G,COMPARETTI P M,KRÜGEL C,et al. Automatic network protocol analysis[C]//Proceedings of NDSS'08. Washington D. C.,USA:IEEE Press,2008:235-247.

[19] COMPARETTI P M,WONDRACEK G,KRUEGEL C,et al. Prospex:protocol specification extraction[C]//Proceedings of the 30th IEEE Symposium on Security and Privacy. Washington D. C.,USA:IEEE Press,2009:110-125.

[20] LIN Z,ZHANG X,XU D. Automatic reverse engineering of data structures from binary execution[C]//Proceedings of the 11th Annual Information Security Symposium. Washington D. C.,USA:IEEE Press,2010:105-117.

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

[22] FENG D F,DOOLITTLE R F. Progressive sequence alignment as a prerequisitetto correct phylogenetic trees[J]. Journal of Molecular Evolution,1987,25(4):351-360.

[23] CUI W D,KANNAN J,WANG H J. Discoverer:automatic protocol reverse engineering from network traces[EB/OL]. [2022-11-20]. https://www.researchgate.net/publication/228641854.

(下转第145页)

（上接第135页）

[24] CABALLERO J, YIN H, LIANG Z K, et al. Polyglot: automatic extraction of protocol message format using dynamic binary analysis[C]//Proceedings of the 14th ACM Conference on Computer and Communications Security. New York, USA: ACM Press, 2007: 317-329.

[25] LIN Z Q, JIANG X X, XU D Y, et al. Automatic protocol format reverse engineering through context-aware monitored execution [C]//Proceedings of NDSS'08. Washington D. C., USA: IEEE Press, 2008: 1-15.

[26] LEITA C, MERMOUD K, DACIER M. ScriptGen: an automated script generation tool for Honeyd [C]// Proceedings of the 21st Annual Computer Security Applications Conference. Washington D. C., USA: IEEE Press, 2006: 203-214.

[27] BOSSERT G, GUIHÉRY F, HIET G. Towards automated protocol reverse engineering using semantic information [C]//Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security. New York, USA: ACM Press, 2014: 51-62.

[28] 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. LI W M, ZHANG A F, LIU J C, et al. An automatic network protocol fuzz testing and vulnerability discovering method[J]. Chinese Journal of Computers, 2011, 34(2): 242-255. (in Chinese)

[29] PHILLIPS P J, MARTIN A, WILSON C L, et al. An introduction evaluating biometric systems[J]. Computer, 2000, 33(2): 56-63.

[30] CUI W D, PAXSON V, WEAVER N, et al. Protocol-independent adaptive replay of application dialog [C]// Proceedings of NDSS'08. Washington D. C., USA: IEEE Press, 2008: 235-246.

编辑　索书志