

# 基于 Peach Fuzz 的媒体网关安全测试

姜文,刘立康

(西安电子科技大学 通信工程学院,陕西 西安 710071)

**摘要:**随着互联网技术和计算机技术的发展,软件在各行各业的应用越来越广泛。随之而来的软件安全问题也越来越突出。在常见的软件安全测试工具中,Peach Fuzz 是一款优秀的安全测试工具,能在多种操作系统上运行,支持多种协议的模糊测试。结合媒体网关安全测试工作实践,介绍了 Peach Fuzz 测试的基本结构和特点;介绍了 H. 248 协议和 SCTP 协议;叙述了媒体网关的测试模型;详细叙述了基于 Peach Fuzz 的媒体网关安全测试过程,其主要内容包括搭建和部署 Peach Fuzz 测试执行机、测试执行机对接测试环境、Peach Fuzz 测试套的调试和连跑、Peach Fuzz 测试的观测方法、测试结果分析。最后介绍了一个测试案例。工作实践表明采用 Peach Fuzz 工具进行媒体网关安全测试,有助于提高媒体网关产品的安全性和提升产品质量。

**关键词:**安全测试;模糊测试;Peach Fuzz;H. 248;媒体网关

**中图分类号:**TP311.5

**文献标识码:**A

**文章编号:**1673-629X(2020)05-0088-06

**doi:**10.3969/j.issn.1673-629X.2020.05.017

## Security Test of Media Gateway Based on Peach Fuzz

JIANG Wen,LIU Li-kang

(School of Telecommunication Engineering,Xidian University,Xi'an 710071,China)

**Abstract:**With the development of Internet and computer technology,the software is widely applied in all fields of life. Following by,the software security problem is becoming more and more prominent. Among the common software security test tools,Peach Fuzz is an excellent one which can run in multiple operating systems and support fuzzy test for multiple protocols. Combining with the work practice of media gateway security test,we introduce the basic structure and characteristics of the Peach Fuzz test and H. 248 protocol and SCTP protocol,and describe the media gateway test model. Especially,we describe the media gateway security test process based on Peach Fuzz in detail,mainly including building and deploying Peach Fuzz test execution machine,test execution machine docking test environment,debugging and running Peach Fuzz test set,Peach Fuzz test observation,and test results analysis. Finally a test case is given. Practice shows that the Peach Fuzz on media gateway security test helps to improve the safety and reliability of the media gateway products and product quality.

**Key words:**security test;fuzzy test;Peach Fuzz;H. 248;media gateway

## 0 引言

随着互联网技术和计算机技术的发展,软件在各行各业的应用越来越广泛。随之而来的软件安全问题<sup>[1-2]</sup>越来越突出,使得安全测试成为一个专门的测试领域。最近几年软件安全测试技术发展很快,出现了许多安全测试工具,常见的安全测试工具有:Nmap、Nessus、Openvas、Burpsuit、AWVS、Peach Fuzz、Code-nomicon<sup>[3-4]</sup>等。其中 Peach Fuzz 是一款优秀的安全测试工具,支持在多种操作系统上运行,支持多种协议的

安全测试。

Fuzz(模糊测试)<sup>[5]</sup>是一种通过提供非预期的输入并监视异常结果来发现软件安全漏洞的方法。基于网络协议的 Fuzz 测试利用“畸形数据包”测试网络环境的健壮性。基于网络协议的 Fuzz 测试过程如下:

- (a)获得测试协议的正常数据包;
- (b)用变异数据替换该数据包中的某些部分;
- (c)向被测试对象发送变异的数据包;
- (d)监控被测试对象的反应。

收稿日期:2019-05-18

修回日期:2019-09-19

网络出版时间:2020-01-10

基金项目:国家部委基础科研计划资助项目(A1120132007)

作者简介:姜文(1986-),女,高级工程师,硕士,CCF会员(E200032324M),研究方向为图像处理、软件工程和网络通信;刘立康,副教授,研究方向为数字通信、图像处理和软件工程。

网络出版地址:<http://kns.cnki.net/kcms/detail/61.1450.TP.20200110.1118.002.html>

1 Peach Fuzz 工具

Peach Fuzz<sup>[6-11]</sup>是开源的、跨平台的模糊测试框架,最初采用 Python 语言编写,发布于 2004 年。第三版使用 C#重写了整个框架。Windows 下使用 Peach 3.0 之后的版本需要预先安装 .net 4 和 Python 软件;Linux、OS X 下需要安装 Mono .net 开发框架和

Python 软件。

1.1 Peach Fuzz 测试基本结构

Peach Fuzz 的关键是使用 XML 语言编写 Pit 文件<sup>[12-13]</sup>,通过配置文件实现与测试环境对接,来完成测试过程。Peach Fuzz 测试框架主要包括三部分,如图 1 所示。

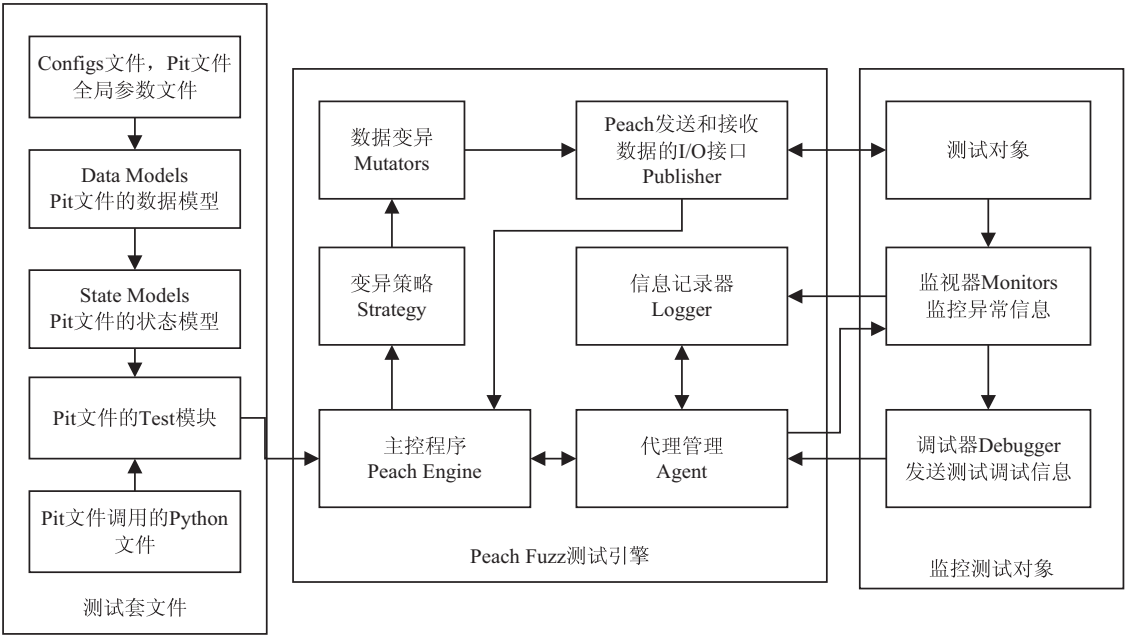


图 1 Peach Fuzz 测试基本结构

1.1.1 测试套文件

测试套文件由 Pit 文件和测试需要调用的 Python 文件组成。各文件的功能如下:

(a)Config 文件:采用 XML 语言编写,包含 Pit 文件的全局参数,是 Pit 文件与测试环境对接的接口文件;

(b)Pit 的 DataModel 模块:根据报文格式,描述数据类型信息,关系(大小、数量、偏移量)一个 Pit 文件包含一个或多个 DataModel;

(c)Pit 的 StateModel 模块:用于定义测试的逻辑,实际上相当于一个状态机。它定义了怎么给目标发送和接收数据,由一个或多个 State 组成。State 由 Action 组成。Action 可以执行多种操作。Action 是发送命令给 Publisher 的一种主要方式,它能发送输出,接收输入或打开一个连接;

(d) Pit 的 Test 模块:设置 Agent、Monitor、StateModel、Publisher、数据变异策略(Strategy),日志文件路径等;

(e)Python 文件:对于一些复杂的 Fuzz 测试,根据特殊需要,为 Pit 文件编写专用的 Python 函数模块。

1.1.2 Peach Fuzz 测试引擎

Peach Fuzz 测试引擎<sup>[14]</sup>主要包括 Peach Engine、变异策略 Strategy、数据变异 Mutator、代理管理 Agent、

Publisher、Logger。

(a)Peach Engine:Peach Fuzz 测试的主控程序;

(b)变异策略 Strategy:Peach 工具有三种变异策略,分别为 Sequential、RandomDeterminstic、Random,在文中的媒体网关测试中选择 Random 作为变异策略;

(c)数据变异 Mutator:对传送报文数据进行变异处理;

(d)代理管理 Agent:Agent 能够运行在本地或者远程的 Peach 进程,这些进程可以启动监视器 Monitor,监控被测试目标,捕获 Crash 信息;

(e)Publisher:是 Peach 发送和接收数据的 I/O 接口,每个测试套至少包含一个 Publisher;执行 Action 需要指定 Publisher;

(f)Logger:记录各种相关信息。

1.1.3 监控测试对象

通过 Publisher 发送畸形报文测试被测试对象,通过 Monitors 监测被测试对象的反应。通过 Debugger 发送调试信息。

1.2 Peach Fuzz 工具的特点

优点:可以对文件格式、ActiveX、网络协议、API 等进行 Fuzz 测试;工具由多个组件组成,各组件层次独立,有助于各组件独立扩展;提供 SDK,可扩展和更新 Fuzz 变异算法、监控器、Publisher 发包器等各组件;

变异能力强。

缺点:规则比较复杂,测试人员需要一定的时间掌握 Peach 的数据建模方法。目前 Peach Fuzz 已经支持的协议有 110 多个,提供的协议测试套不满足产品测试需求时,需要自己扩展开发。

## 2 H. 248 协议和 SCTP 协议

### 2.1 H. 248 协议简介

H. 248<sup>[15]</sup>是一种媒体网关控制协议,是软交换网络中控制层的软交换设备(媒体网关控制器 MGC)和接入层中各种媒体网关(MG)的标准接口协议。H. 248协议是一种主从协议,在 MGC 与 MG 的交互中,MGC 控制呼叫建立的过程,MG 只是被动地接收 MGC 下发的各种指令,然后完成相应的动作。H. 248 协议是一种双向式的交互协议,协议消息分为命令和

响应两种类型,每个命令需要接受对方回送响应。

H. 248 协议底层传输机制可以采用 UDP/TCP/SCTP。由于 H. 248 协议对网络延时和丢包不太敏感,目前工程上通常采用 UDP 协议传送报文。对于报文传输可靠性要求比较高的情况,采用 SCTP 作为传输层协议。

一个 H. 248 消息的信息体通常由一个或多个事务(Transaction)组成。每个事务由一个 TransactionID 来标识。事务由一个或多个动作(Action)组成。Action 由一系列命令组成。每个命令可带多个参数(描述符)。

H. 248 协议使用命令对关联域(Context)和终端(Termination)进行控制。H. 248 协议所包含的命令、命令功能以及命令的传输方向如表 1 所示。

表 1 H. 248 协议命令

命令名称	命令功能	命令方向
Add	向一个关联域添加终端	MGC 到 MG
Modify	修改终端的属性、事件与信号	MGC 到 MG
Subtract	从关联域中删除一个终端	MGC 到 MG
Move	将一个终端从一个关联域转移到另一个关联域	MGC 到 MG
AuditValue	获取相关终端的当前属性、事件、信号以及统计信息	MGC 到 MG
Auditcapabilities	获取媒体网关所允许的终端的属性、事件以及信号的所有可能值的信息	MGC 到 MG
Nofity	向媒体网关控制器报告媒体网关中发生的事件	MG 到 MGC
ServiceChange	向媒体网关报告一个终端进入或退出服务	双向均可
	向媒体网关控制器报告终端退出或进入服务	MGC 到 MG
	向媒体网关控制器进行注册	MG 到 MGC

### 2.2 SCTP 协议简介

SCTP 协议(流传送控制协议)既能增强 UDP 业务并提供数据报文的可靠传输,又能克服 TCP 的某些局限,可以在不可靠的分组网络(IP 网)上提供可靠的数据传输。

SCTP 协议的主要功能有:

- (a) 在确认方式下无差错、无重复地传送用户数据;
- (b) 可以将多个小用户数据分组封装到一个 SCTP 的数据块中;
- (c) 根据通道最大传输单元(MTU)的限制,可以将大用户数据分拆和重组为多个数据块;
- (d) 在多个流上保证用户数据的顺序提交;
- (e) SCTP 的设计中包含避免拥塞、避免遭受泛播和匿名攻击的功能。

SCTP 协议通过在两个 SCTP 端点间建立偶联,通过 IP 地址+SCTP 端口号来为两个 SCTP 用户提供可

靠的消息传送业务,消息在数据块中进行传送。

## 3 基于 H. 248 协议的媒体网关安全测试

文中采用 Peach Fuzz 工具对云化媒体网关进行安全测试,采用 SCTP 作为传输层协议。依据 MG 与 MGC 的控制与连接关系,采用 Peach Fuzz 执行机模拟媒体网关控制器,来完成基于 H. 248 协议的云化媒体网关安全测试。

工作内容包括云化执行机搭建,执行机模拟 MGC;在执行机上安装 Peach 工具和其他相关工具,搭建测试环境;编写 Peach 测试套;Peach Fuzz 测试套调试和连跑;测试的观测方法;测试结果分析。

### 3.1 测试模型

#### 3.1.1 基于 H. 248 协议的网络结构

媒体网关控制器 MGC 通过 H. 248 协议中的命令控制媒体网关 MG,在终端 1 与终端 2 之间进行媒体数据传输,网络结构如图 2 所示。

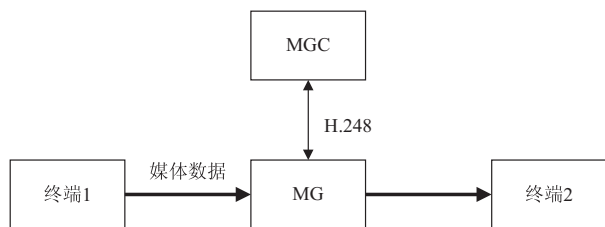


图2 基于 H.248 协议的网络结构

搭建的 Peach 执行机用来模拟 MGC,选择 MG 为被测试的媒体网关,配置媒体数据终端,形成 Peach Fuzz 测试环境。

### 3.1.2 报文变异处理

为了简化测试模型,选择 AddRequest、Modify Request、SubtractRequest、MoveRequest、ServiceChange Request、NofityReply 六个命令的报文进行变异处理,从 MGC 向媒体网关 MC 发送报文,进行 Peach Fuzz 安全测试。

## 3.2 搭建云化执行机

使用 VMWARE 虚拟化软件,制作 64 位 Suse Linux 11.3 操作系统的虚拟机。登录 vCenter 完成虚拟机的 CPU、内存大小、数据盘大小、操作系统安装、网络配置等步骤;通过 yast 命令配置虚拟机的 IP 地址,开启 SSH 服务。完成安装配置之后,虚拟机能够使用 PUTTY 工具登录。云化的执行机上通常需要配置两个网络,一个是维护网络,一个是业务网络,通常使用业务网络进行 Peach Fuzz 测试。云化的执行机用来模拟图 2 中的 MGC。

## 3.3 执行机上安装相关工具

需要安装的相关工具有:Peach 工具、mono 工具和 SCTP 协议的发包器。

### 3.3.1 Peach 工具安装部署

将 Peach 工具版本包拷贝到/opt 目录下,直接解压即可。

解压命令:unzip peach-pro-4.3.235-linux\_x86\_64\_release.zip。

### 3.3.2 mono 工具部署

自 Peach 3.0 之后的版本 Peach 工具使用 C#语言进行开发,在 Linux 操作系统下 mono 工具必须与 Peach 一起安装,否则 Peach 工具不能使用。mono 工具的安装步骤如下所示:

(a)将 mono 工具版本包放到指定的目录下,解压软件包:tar xzvf mono-4.8.1.tar.gz;

(b)添加环境变量:环境变量的设置有两种情况,一种是临时情况,shell 退出之后就失效了;一种是永久的,一次设定之后,系统重启,或者 shell 退出或切换,环境设置都生效。

增加临时环境变量:执行 export PATH=/opt/

peach43235:/opt/mono/bin:\$PATH。

增加永久环境变量:打开 profile 文件;在文件中添加命令行:export PATH=/opt/peach43235:/opt/mono/bin:\$PATH。

### 3.3.3 配置 SCTP 协议的发包器文件 sctplister.so

Peach Fuzz 测试使用 SCTP 协议传送报文,需要提供 SCTP 的发包器文件。开发工程师采用 C 语言编写 SCTP 发包器源代码,使用 GCC 4.3.3 版本编译器,编译生成发包器文件 sctplister.so。

### 3.3.4 部署出报告的库文件

Peach Fuzz 运行完成的结果需要通过 ibgdiplus.so.0 生成 \*.pdf 文件格式的报告。这个文件保存在/opt/mono/lib 文件夹下。

### 3.3.5 测试环境的网络部署和对接

Peach Fuzz 执行机上的测试工具部署完成之后,需要通过 MML 命令与被测试的媒体网关对接起来。在图 2 中选择 MG 作为被测试的网关,需要为 MG、MGC 配置传输的 H.248 协议的信令 IP 地址和端口号,以及媒体数据传输所需的媒体域与媒体终端的相关信息。

(1)MG、MGC 的相关配置。

(a)配置 MG1 和 MGC 的 IP 地址与使用的端口号;

(b)添加信令地址,媒体网关 MG 的 IP 地址也可用作信令地址;

(c)添加 H.248 链路,配置 H.248 链路时,需要将 MG、MGC、信令地址、端口号以及 SCTP 协议全部关联起来;

(d)执行媒体网关 MG 激活,激活之后 MG 处于运行状态(正常状态)。

(2)媒体域、媒体的相关配置。

(a)创建媒体域(Context),配置媒体域的关联标识 Context ID 值;

(b)将媒体作为终端(Termination)加入媒体域中,配置媒体终端地址。

保存在 H.248\_binary\_Mg.xml.config 中的源 IP 和源端口号是 MGC 的 IP 地址和端口号,目的 IP 和目的端口号是 MG 的 IP 地址和端口号。

## 3.4 Peach Fuzz 测试套

Peach 工具提供的 H.248 协议测试套不能满足测试的要求,需要自行编写 H.248 协议的测试套。编写完成的 H.248 协议测试套由 5 个文件组成,如表 2 所示。

(a)文件 H.248\_binary\_data\_mg.xml 为 Pit 文件的数据模型文件,主要包括命令请求和命令回复的数据模型。



表 2 Peach Fuzz 测试套

序号	测试套文件名称	文件类型
1	H. 248_binary_data_mg. xml	Pit 的数据模型文件
2	H. 248_binary_state_mg. xml	Pit 的状态模型文件
3	H. 248_binary_mg. xml	Pit 的 Test 文件
4	H. 248_binary_mg. xml. config	和测试环境对接的相关配置文件
5	H. 248_bin_mg. py	Pit 文件调用的 Python 文件

(b)文件 H. 248\_binary\_state\_mg. xml 为 Pit 文件的状态模型文件,主要包括发送报文的状态模型(State)和接收报文的状态模型(State)。在各(State)中的 Action 中给出了测试启动的初始参数,类似于种子文件。

(c)H. 248\_binary\_mg. xml 为 Pit 文件的 Test 文件,包含测试所需要的 Agent、Publisher。变异策略 Strategy、Logger 的配置。

```
Test 文件中的主要代码如下:
代理管理 Agent:在 Agent 中定义了监控器;
<Agent name="LocalAgent">
<Monitor class="Ping">
.....
</Monitor>
</Agent>
配置 Publisher、发包器为 Sctp_listener。
<Test name="Default" maxOutputSize="100000000" targetLifeTime="session">
<StateModel ref="H248:MgcToMg"/>
<Publisherclass="Sctp_listener"/>
.....
</Publisher>
配置变异策略、Logger 路径。
<strategy class="Random"/>
<Logger class="File">
<Param name="Path" value="##LoggerPath##"/>
</Logger>
```

(d)H. 248\_binary\_Mg. xml. config 为 Pit 的相关参数配置文件,主要包含 SourceIPV4、SourcePort、TargetIPV4、TargetPort、Timeout、Strategy、PitLibrary Path、LoggerPath 等参数。

(e)H. 248\_bin\_mg. py 为 Pit 文件测试运行时调用的 Python 文件,提供测试过程 StateModel 文件中各个状态(State)的调度、控制和响应函数。

3.5 Peach Fuzz 测试套调试和连跑

3.5.1 测试套调试

测试套调试过程如下:  
进入/opt/peach43235/pits 文件夹

执行:peach ./Net/ H. 248-Binary\_Mg. xml -- debug -1

调试过程中,使用网络抓包工具 tcpdump,对各个命令的请求和回复开展抓包工作,抓包得到各个命令请求和命令回复的报文流程如图 3 所示,表示测试套调试成功。

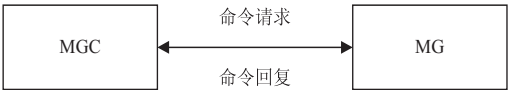


图 3 Peach Fuzz 测试消息流程

3.5.2 测试套以连跑

进入/opt/peach43235/pits 文件夹  
执行:peach ./Net/ H. 248\_binary\_mg. xml -- range -1,1500000(1500000 为迭代次数)。

3.6 Peach Fuzz 测试的观测方法

运行 Peach Fuzz 工具前后,都需要对云化 MG 环境进行观测。

3.6.1 执行 Peach Fuzz 测试之前进行的观测

- 测试前进行如下观测:
- (a)在图 2 所示的网络环境中,观测并记录各虚拟机的资源占用情况;
  - (b)观测并记录 MG 环境上已经存在的告警信息;
  - (c)执行调试命令,打开云化 MG 环境中相关进程的打印协议栈日志的调试开关,这样 Peach Fuzz 测试结束之后,可以打印处日志的 error 信息;

(d)在运行 Peach Fuzz 测试的执行机上,对测试环境对接的网卡进行抓包,可以通过 ifconfig 文件进行网卡查询。

抓包命令:tcpdump -v -n -i ethx -w /opt/sctp\_x.cap -s 0 port xxxx  
(端口号与已经配置的 H248 链路一致)。

3.6.2 执行 Peach Fuzz 测试之后需要进行的观测

- 测试后进行如下观测:
- (a)在图 2 所示的网络环境中观测并记录各虚拟机的资源占用情况;
  - (b)在 MG 环境上观测并记录告警,并与测试执行之前环境上上报的告警进行对比,查看是否有新的异常告警出现;
  - (c)执行 MG 产品查询内存的调试命令,记录执行结果,查看执行结果的最后一列,如果有非 0 的值就需要联系开发人员,定位导致内存泄漏的原因。

3.7 测试结果分析

Peach 测试完成之后,对测试环境中上报的告警进行分析,查看是否有新增告警信息。执行 GET SYSINFO 命令收集进程或业务虚拟机相关的日志记

录,将日志记录反馈给软件开发工程师和相关人员;分析产生告警的原因,定位和处理存在的各种问题。

## 4 典型案例

使用 VMware 软件构建虚拟机来模拟 MGC,搭建两个网卡、4CPU、8 G 内存、100 G 存储空间、64 位 Suse Linux 11.3 操作系统的虚拟机。Peach 版本为 4.3.235、配套的 mono 版本为 4.8.1。Peach 工具自带的 H.248 测试套不能满足 S 产品媒体网关 MG 测试的要求,需要自行编写测试套文件。

使用一个执行机对 MG 进行 15 万次 Peach Fuzz 测试需要 30 天时间。为了提高测试效率,共搭建了 10 个测试执行机(MGC),同时分别添加 10 个云化 MG 和 H248 链路。每个执行机上分别启动 1.5 万次 Peach Fuzz 测试,4 天时间完成测试连跑。

测试完成之后,各虚拟机资源占用正常、无异常告警以及内存泄漏则表示测试通过。对于测试中出现的一些问题,联系相关人员对问题进行定位分析,使问题得到有效处理。

## 5 Peach Fuzz 测试过程中遇到的问题实例

执行 Peach Fuzz 测试过程中遇到的几个典型问题与解决方法如下:

(a) 运行测试套时,找不到 stcplstener. so 文件。

解决方法:编译生成该文件,同时进行路径部署和环境变量设置。

(b) 运行测试套时,H.248 消息报错为 ADD Response 消息匹配数据失败,H.248 跟踪消息中报错信息为“资源不足”。查看运行 Peach 对接的测试环境,发现环境中没有配置媒体域和媒体终端。

解决方法:对接测试网络环境时,采用 MML 命令添加媒体域与媒体终端。

(c) 从 Peach FUZZ 的 debug. log 中可以看出,Add Request 消息复制失败,没有收到消息;从抓包结果来看,执行 Peach Fuzz 测试时会出现自动拆链的情况。

解决方法:确认是由于 Add Request 消息发消息超时导致的,可以通过增大超时时间来解决。

## 6 结束语

介绍了 Peach Fuzz 测试基本结构和特点,H.248 协议和 SCTP 协议;叙述了媒体网关安全测试模型。

结合媒体网关安全测试的工作实践,叙述了基于 Peach Fuzz 的媒体网关测试过程。介绍了大量基于 Peach Fuzz 测试的技术细节。工作实践表明对媒体网关进行 Peach Fuzz 安全测试,有助于提高媒体网关产品的安全可靠性和提高媒体网关产品开发与测试的效率,降低开发成本,提升产品质量。

### 参考文献:

- [1] 苏璞睿,应凌云,杨 轶. 软件安全分析与应用[M]. 北京:清华大学出版社,2017.
- [2] SUTTON M, GREENE A, AMINI P. 模糊测试强制发掘安全漏洞的利器[M]. 段 念,赵 勇,译. 北京:电子工业出版社,2013.
- [3] 伊胜伟,张翀斌,谢 丰,等. 基于 Peach 的工业控制网络协议安全分析[J]. 清华大学学报:自然科学版,2017,57(1):50-54.
- [4] 章 焯. Fuzz 安全测试技术研究[D]. 西安:西安电子科技大学,2010.
- [5] MA Rui, REN Shuaimin, MA Ke, et al. Semi-valid fuzz testing case generation for stateful network protocol[J]. Tsinghua Science and Technology, 2017, 22(5):458-468.
- [6] 张亚丰,洪 征,吴礼发,等. 基于状态的工控协议 Fuzzing 测试技术[J]. 计算机科学,2017,44(5):132-140.
- [7] 马金鑫,张 涛,李舟军,等. Fuzzing 过程中的若干优化方法[J]. 清华大学学报:自然科学版,2016,56(5):478-483.
- [8] YANG Huan, ZHANG Yuqing, HU Yupu, et al. IKE vulnerability discovery based on fuzzing[J]. Security and Communication Networks, 2012, 6(7):889-901.
- [9] 赵丽娟. Fuzz 安全测试技术研究[D]. 北京:北京邮电大学,2011.
- [10] 王 浩. PeachFuzz 模糊测试平台的研究与改进[D]. 北京:北京邮电大学,2013.
- [11] 赵 凯. 基于 Peach 的 Fuzz 策略优化研究与实现[D]. 北京:北京邮电大学,2015.
- [12] 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.
- [13] 耿秋菊,卜 哲. H.248/MEGACO 协议安全性分析与测试[J]. 电信网技术,2009(3):13-18.
- [14] 桂海源,张碧玲. 信令系统[M]. 北京:北京邮电大学出版社,2008.
- [15] 李静林,孙其博,杨放春. 下一代网络通信协议分析[M]. 北京:人民邮电出版社,2010.