****

软 件 工 程 实 验

软件产品设计与实现

基于Scapy的分析与扩展

Version 0.2

实 验 小 组 B组

小 组 成 员 陈鸿超（SY1806214）

李铎坤（SY1806219）

刘 颖（SY1806418）

袁梦阳（BY1806157）

编制时间： 2019年4月

版本变更历史

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版本 | 提交日期 | 编制人 | 修改说明 | 审核人 | 版本说明 |
| v0.1 | 2019/4/17 | 袁梦阳 | 初版 |  | 初版的2、4.1、4.2 |
| v0.2 | 2019/4/17 | 刘颖 | 初版 |  | 初版的1、2、4.3、4.4 |

目录

[版本变更历史 I](#_Toc6246039)

[1 引言 1](#_Toc6246040)

[2 总体设计 2](#_Toc6246041)

[3 系统架构 3](#_Toc6246042)

[4 扩展功能设计 4](#_Toc6246043)

[5 实现方案 6](#_Toc6246044)

[6 集成方案 7](#_Toc6246045)

# 引言

## 编写目的

本软件设计说明书针对需求分析阶段提出的Scapy扩展功能进行描述。

本文档是开发者前期开发的重要依据，是为了明确所开发的扩展模块的具体功能而设计的实现方案。除此之外，本文档也是后期交付时软件是否符合标准的一个重要参考。

## 定义

表 1.1 术语与缩略词表

|  |  |
| --- | --- |
| 术语 | 说明 |
| IP | 互联网协议地址，是分配给用户上网使用的网际协议的设备的数字标签 |
| pcap | pcap 是packet capture 的缩写，pcap文件格式是常用的数据报存储格式 |
| 报文 | 报文(message)是网络中交换与传输的数据单元，即站点一次性要发送的数据块 |
| 端口 | 是一种经由软件创建的服务，在一个计算机操作系统中扮演通信的端点 |
| 端口监听 | 端口监听是指对客户端(个人机器)所操作的一种信息记录，还用于实现对共享目录访问的监测和控制 |
| 接口 | API，是一些预先定义的函数，目的是提供应用程序与开发人员基于某软件或硬件得以访问一组例程的能力，而又无需访问源码，或理解内部工作机制的细节 |
| SMTP |  |
| HTTP |  |
| tcpdump、WinPcap |  |
| BPF |  |

## 参考资料

[1]吕云翔，王昕鹏，邱玉龙.软件工程——理论与实践.北京：人民邮电出版社，2012.

[2]需求规格说明书2.4.docx

[3]虞平. 一种面向会话的入侵检测模型[D].东南大学,2004.

# 总体设计

## 需求规定

本项目基于Scapy开源项目进行扩展，面向人群为计算机网络方面的开发与测试人员，主要目的是便于用户快捷地进行报文的获取、解析和发送等操作。为了满足此需要，本项目考虑实现一个更全面、快速的报文操作工具。通过对Scapy工具的扩展，在Scapy开源项目中扩展出开发人员需求但却缺少的功能，使得Scapy更加易用与全面。

## 硬件运行环境

1. CPU：Intel i5-4590及以上，AMD Ryzen 3 1200及以上
2. 内存：4G内存及以上（系统最低2GB，Scapy报文解析较占用内存空间）
3. 硬盘：20G硬盘及以上（Windows 7 64位安装最低配置）

## 软件运行环境

1. 操作系统：
   * 1. Windows 7及以上： 已安装Npcap/WinPcap，
     2. Ubuntu 16.04及以上： 已安装tcpdump
2. Anaconda（可创建多版本Python虚拟环境），Python 2.7+，Python 3.4+
3. 第三方Python库：matplotlib，pyx，vpython，cryptography等

## 编程语言与工具

1. 编程语言：Python
2. 编程工具：pycharm

# 系统架构

作为一个数据包处理工具，Scapy的系统架构比较简单，主要由底层系统、基础模块、应用模块三部分组成。其中，底层系统是操作系统本身自带的功能，基础模块和应用模块则是由Scapy所实现。Scapy的系统架构如图 3.1所示。



图 3.1 系统架构图

基础模块是应用模块的实现基础，对于用户而言是不可见的，主要包括环境配置模块、插件调用模块、网络标准模块、协议格式模块。其中环境配置模块主要负责获取运行环境信息，初始化其他模块所需的配置参数；插件调用模块主要负责调用各种插件，比如tcpdump、WinPcap等；网络标准模块主要负责封装计算机网络领域所使用的编码、传输、解码标准；协议格式模块主要封装了Scapy所支持的所有协议的具体格式，包括所有协议字段名称、数据类型、大小、位置等信息。

应用模块是用户可以直接使用的模块，主要包括数据导入模块、数据导出模块、数据展示模块、数据解析模块、数据构造模块、端口监听模块、数据传输模块。这些模块实现了用户进行数据包处理所需的各种功能，包括数据获取、数据构造、数据存储、数据发送和数据展示等。

我们在应用层进行扩展，添加了四个扩展模块。

# 扩展功能设计

## pcap文件切分功能设计

pcap文件的报文使用字节码存储，所占用的存储空间很小。但对于解析后的报文数据而言，每条报文都包含大量对象、字段和方法，使得解析后的数据占用的存储空间是原始报文大小的数十倍。

由于解析pcap文件对内存造成极大的负担，而且内存占用率过高会影响程序的运行效率，那么，对于较大的pcap文件，将其切分成若干个小文件，再进行批处理非常必要。

本项目扩展了Scapy对大pcap文件进行切分的功能。开发人员只需提供所要切分的pcap文件路径、切分后的存储目录以及切分批大小即可按照默认设定简便地对pcap文件进行切分。同时开发人员还可自行设定以下切分过程中的需求：

1. 开发人员可以通过更改切分模式设置来设定切分模式。本项目的pcap文件切分功能提供两种切分模式供开发人员选择。第一种为兼容模式，调取文件解析模块解析数据后再进行切分，切分后可存储为与源文件不同的格式，但效率较低。另一种为极速模式，无需数据解析，对源数据进行切分，速度较快，但不可转换为与源文件不同的格式。默认为兼容模式。
2. 开发人员可以通过更改命名前缀设定来设置切分后的文件命名。默认情况下，新生成的文件命名方式为：源文件名\_00000~源文件名\_99999。当用户设定了命名前缀的情况下，新生成的文件命名方式为：前缀\_00000~前缀\_99999。
3. 开发人员可以设定报文数据的链路类型。本设置只有在切分模式为兼容模式时有效。
4. 开发人员可以设定切分后的报文的存储方式。开发人员可以选择报文字节码的存储方式是大端存储还是小端存储。本设置只有在兼容模式下游侠。

## 获取报文协议类型功能设计

Scapy根据报文的协议层次将报文数据分层解析，解析的结果也按照协议层堆叠起来。解析结果的每一层是一个单独的对象，拥有一个payload变量，用于保存下一层协议的对象实例。

比如一条DNS报文pkt，其解析结果是一个物理层的Ether对象实例，pkt.payload是一个网络层的IP对象实例，pkt.payload.payload是一个传输层的UDP对象实例，依次类推。

在处理多协议流量数据时，用户通常需要对不同协议类型的报文进行不同的处理操作。但用户无法直接从Scapy解析的结果中获取报文的协议类型，必须自行设计一种判定逻辑来得到报文的协议类型。因此，为解析后的报文对象内置一个获取协议类型的函数可以为用户提供极大的方便。

本项目扩展了Scapy获取报文协议类型的功能。开发人员可以直接通过本功能来获得报文的协议类型。用户可以选择两种报文协议类型的返回方式：一种是返回单层协议类型，另一种是返回所有层协议类型。

同时当开发人员选择的返回类型为单层时，可以选择一个的网络层，表示需要返回该层协议的类型。默认为返回可识别的最高层的协议。

## 监听持久化

Scapy的监听存储策略为将监听到的所有报文都存在内存中，监听结束后再一起存储到硬盘上。由于pcap文件通常都很大，因此Scapy原有的监听存储功能很容易造成内存泄漏。另外，如果突然宕机，数据就会全部丢失。基于该情况我们利用内存快照设计了监听持久化模块。

用户指定好pcap文件的存储目录后，模块允许用户自定义的内容包括：

* 数据批大小：捕获数据时达到该阈值大小就对内存中数据进行一次存储写入
* 监听端口：监听端口或端口列表，默认监听所有端口
* 筛选规则：采用BPF规则，只监听符合要求的报文
* 停止监听的报文个数：当监听到用户指定的报文个数时，停止监听
* 停止监听结束时间：到达用户指定的监听结束时间时，停止监听
* 停止监听判定函数：用户可以自定义一个判定函数，当某条报文符合该判定函数的条件时，模块停止监听

## 3.2 会话提取功能

首先我们需要对会话的概念加以解释，此处的会话与HTTP中的会话有所区别。一次会话由互相通信的两个实体之间的交互过程组成。根据会话的通信性质，可将会话的两个实体分为客户端和服务端。一次会话过程由客户端和服务端的多次交互组成，并且客户端的一次请求总是对应于服务端的一次回应。为了论述方便，在本文中定义客户端向服务端发出的请求或服务器向客户机的回应为交互过程的一个句子。图 4.1展示了一个SMTP实际交互的例子。在客户端向服务端发起TCP连接请求并被接受后，服务端和客户端开始了基于TCP和SMTP会话，图片中反映了客户端和服务端的多次交互过程。简而言之，句子是一个以报文为基础的客户端和服务端通信单位，由一个或多个相互交互的句子构成了会话。

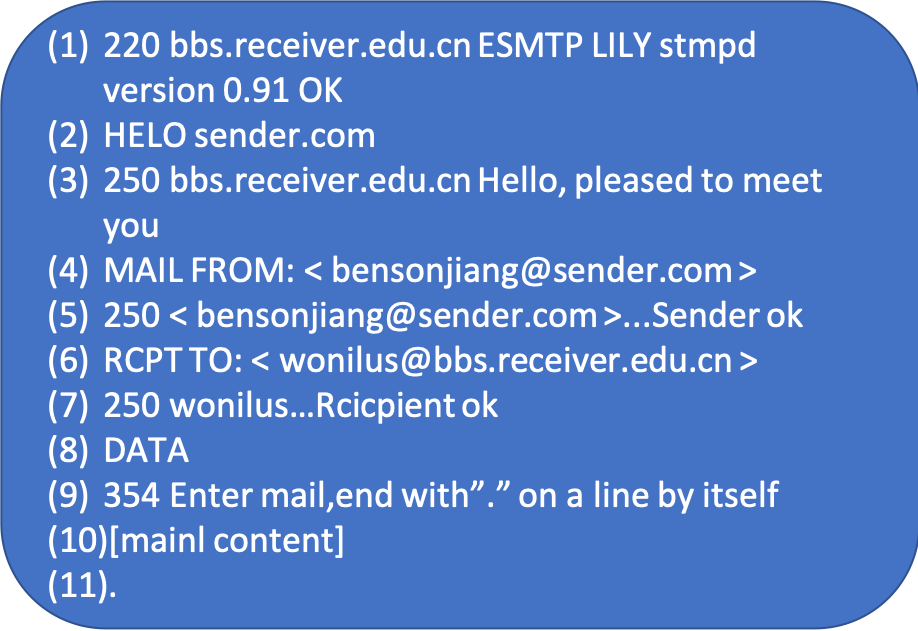


图 4.1 SMTP交互实例

一个pcap文件中包含大量会话，有时我们只需要其中一部分或是只需要从某一IP发往另一个IP的会话。基于这种应用场景，我们设计了会话提取功能：从指定的pcap文件中提取出符合特定条件的会话，并进行存储。提取的条件有两种：四元组提取和时间窗口提取，下面将分别介绍。

四元组提取指的是源IP地址、源端口、目的IP地址和目的端口四项条件，是最常用的会话的筛选条件。四元组的四个条件均为可选参数，即当参数不为空时根据传入的参数筛选符合条件的会话，当传入的参数为空时可以任意取值。

时间窗口提取是指根据用户设置的时间窗口大小将间隔时间在时间窗内的会话数据提取出来放入一个文件中。

# 实现方案

# 集成方案