A组对B组测试检查单

# 测试相关信息

测试用例版本：V1.8

测试人员：姬索肇 韩慧敏

测试时间：2019年5月27日

# 功能测试

2.1 功能测试用例结果

表格 2.1用例001测试结果

|  |  |  |  |
| --- | --- | --- | --- |
|  | 不解析模式的文件读取功能测试 | | |
| **简要描述** | 本测试主要用于极速模式所使用的不解析模式的文件读取功能能否正常运行 | | |
| **前提和约束** | 安装好扩展后Scapy及其运行环境 | | |
| **测试方法** | 白盒测试 | | |
| **测试步骤** | **1** | 准备好待切分的pcap文件 | |
| **2** | 实例化NoAnalysis\_PcapReader()文件解析对象，并传入pcap文件的路径 | |
| **3** | 调用get\_pcap\_head()方法获取文件头字节码数据 | |
| **4** | 调用read\_packet()方法读取数据块块头和块体字节码数据 | |
| **预期结果** | 步骤3 : 返回与pcap文件相应的文件头字节码数据  步骤4 : 返回pcap文件中相应的数据块的块头和块体字节码数据 | | |
| **异常情况** | 文件格式有误 | | 程序提示可能的异常信息，并终止运行 |
| **测试结果** |  | | |
| **测试结论** |  | | |

表格 2.2用例002测试结果

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | 不解析模式的文件写入功能测试 | | |
| **简要描述** | 本测试主要用于极速模式所使用的不解析模式的文件写入功能能否正常运行 | | |
| **前提和约束** | 安装好扩展后Scapy及其运行环境 | | |
| **测试方法** | 白盒测试 | | |
| **测试步骤** | **1** | 实例化NoAnalysis\_PcapWriter ()文件写入对象，并传入结果文件的路径 | |
| **2** | 构造文件头字节码字符串，调用\_write\_header()，写入文件头数据 | |
| **3** | 构造数据块字节码字符串元组，调用 \_write\_packet ()，写入数据块 | |
| **预期结果** | 步骤2 :构造的文件头字节码成功写入到结果文件中  步骤3 :构造的数据块字节码成功写入到结果文件中 | | |
| **异常情况** | 文件路径有误 | | 程序提示异常信息，并终止运行 |
| **测试结果** |  | | |
| **测试结论** |  | | |

表格 2.3用例003测试结果

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | PacketSession类单元测试 | | |
| **简要描述** | 本测试主要用于验证PacketSession类中的方法能否按照期望运行 | | |
| **前提和约束** | 安装好扩展后Scapy及其运行环境 | | |
| **测试方法** | 白盒测试 | | |
| **测试步骤** | **1** | 设置参数，实例化PacketSession对象 | |
| **2** | 使用unittest测试模块的assertEqual验证个成员属性初始化是否成功 | |
| **3** | 构造报文与时间实例，传入不同的报文对象和时间参数，使用assertEqual验证update方法是否正确更新了data属性 | |
| **4** | 构造时间实例，传入不同的时间参数，使用assertEqual验证scan方法是否正确处理了data属性 | |
| **5** | 使用assertEqual验证save方法不同模式下，是否正确导出了data属性 | |
| **预期结果** | 步骤2 :各成员属性正确初始化  步骤3 : update方法正确更新了data属性  步骤4 : scan方法正确处理了data属性  步骤5 : save方法正确导出了data属性 | | |
| **异常情况** | 文件路径有误 | | 程序提示异常信息，并终止运行 |
| **测试结果** |  | | |
| **测试结论** |  | | |

表格 2.4用例004测试结果

|  |  |  |
| --- | --- | --- |
| **名称** | TestTetradJudge类单元测试 | |
| **简要描述** | 本测试主要用于验证TetradJudge类中的方法能否按照期望运行 | |
| **前提和约束** | 安装好扩展后Scapy及其运行环境 | |
| **测试方法** | 白盒测试 | |
| **测试步骤** | **1** | 导入TetradJudge类 |
| **2** | 设置不同组合的IP过滤条件和IP对，使用unittest测试模块的assertEqual验证judge\_ip\_tetrad能否正确匹配过滤条件与IP对数据 |
| **3** | 设置不同组合的四元组过滤条件和四元组，使用unittest测试模块的assertEqual验证judge\_tetrad能否正确匹配过滤条件四元组与四元组数据 |
| **预期结果** | 步骤2 :各组合输入下能够正常返回True或False  步骤3 : 各组合输入下能够正常返回True或False | |
| **测试结果** |  | |
| **测试结论** |  | |

表格 2.5用例101测试结果

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | 端口报文监听测试 | | |
| **简要描述** | 本测试用例主要测试Scapy的端口监听功能，包括四个方面：抓取流量包，设置监听参数，设置停止监听条件和核对返回报文数据。 | | |
| **前提和约束** | 正确配置python环境、安装Scapy包 | | |
| **测试方法** | 黑盒测试 | | |
| **测试步骤** | **1** | 在python代码中导入scapy | |
| **2** | 在sniff函数中配置iface参数，设置嗅探器所要嗅探的网卡，接受返回的数据包。 | |
| **3** | 在sniff函数中配置filter参数设置过滤规则。filter的规则使用BPF语法，可以限制抓取报文的IP等信息。 | |
| **4** | 在sniff函数中设置count参数的值，以限定嗅探包的数目。 | |
| **5** | 在sniff函数中设置stop\_filter，传入自定义的函数，当满足函数要求时scapy停止嗅探。 | |
| 6 | 将以上几个步骤中嗅探到的报文打印出来以进行核对。 | |
| **预期结果** | 步骤2 :设置iface参数嗅探到的报文与iface的设置一致。  步骤3 :设置filter参数嗅探到的报文符合设置的BPF规则  步骤4 :设置count参数嗅探到的报文数量与count一致  步骤5 :设置stop\_filter参数嗅探到的数据包只有最后一个报文符合自定义函数的条件 | | |
| **异常情况** | iface参数设置有误 | | 输出错误信息提示，并停止运行 |
| **测试结果** |  | | |
| **测试结论** |  | | |

表格 2.6用例102测试结果

|  |  |  |
| --- | --- | --- |
| **名称** | 数据报文解析测试 | |
| **简要描述** | 本测试用例检查scapy是否能正确解析pcap文件 | |
| **前提和约束** | 正确配置python环境、安装Scapy包、格式正确的pcap文件，安装wireshark | |
| **测试方法** | 黑盒测试 | |
| **测试步骤** | **1** | 在python代码中导入scapy |
| **2** | 利用scapy的rdpcap函数读入pcap文件 |
| **3** | 利用repo函数得到数据包的规范化字符串 |
| **4** | 将解析后的报文内容打印出来 |
| **5** | 在wireshark中打开pcap文件，并与scapy的解析结果进行对比 |
| **预期结果** | Scapy解析后的报文与wireshark中的报文各个字段的值都相同 | |
| **测试结果** |  | |
| **测试结论** |  | |

表格 2.7用例103测试结果

|  |  |  |
| --- | --- | --- |
| **名称** | 单层数据报文构造测试 | |
| **简要描述** | 本测试用例测试Scapy构造单层报文的正确性 | |
| **前提和约束** | 正确配置python环境、安装Scapy包 | |
| **测试方法** | 黑盒测试 | |
| **测试步骤** | **1** | 在python代码中导入scapy |
| **2** | 构造TCP、IP等类型的报文，并指定des，scr等字段 |
| **3** | 通过pack.show()方法查看构造的报文 |
| **预期结果** | 构造的报文字段与设定一致 | |
| **评价准则** | 实际结果与预期结果一致 | |
| **测试结果** |  | |
| **测试结论** |  | |

表格 2.8用例104测试结果

|  |  |  |
| --- | --- | --- |
| **名称** | 多层数据报文构造测试 | |
| **简要描述** | 本测试用例测试Scapy构造多层报文的正确性 | |
| **前提和约束** | 正确配置python环境、安装Scapy包 | |
| **测试方法** | 黑盒测试 | |
| **测试步骤** | **1** | 在python代码中导入scapy |
| **2** | 使用“/”操作符构造TCP、IP等类型堆叠的报文，可以单独指定每个类型报文的des，scr等字段 |
| **3** | 通过pack.show()方法查看构造的报文 |
| **预期结果** | 构造的报文字段与设定一致，堆叠方式与设定一致 | |
| **测试结果** |  | |
| **测试结论** |  | |

表格 2.9用例105测试结果

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | 数据包字符展示测试 | | |
| **简要描述** | 本测试验证开发人员是否可以按照要求导出数据包的字符展示结果。 | | |
| **前提和约束** | 正确配置python环境、正确安装Scapy包 | | |
| **测试方法** | 黑盒测试 | | |
| **测试步骤** | **1** | 在python代码中导入Scapy | |
| 2 | 利用Scapy的rdpcap函数读入待测试pcap文件 | |
| **3** | 调用报文数据的show方法显示数据的字符展示 | |
| **预期结果** | 输出报文数据包的字符展示 | | |
| **评价准则** | 报文字符展示与wireshark解析结果一致且完整，即为正确。 | | |
| **异常情况** | 文件格式有误 | | 程序提示可能的异常信息，并终止运行 |
| 报文数据有误 | | 在会影响程序正常执行的情况下提示异常信息，并终止运行 |
| **测试结果** |  | | |
| **测试结论** |  | | |

表格 2.10用例106 测试结果

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | 数据包图形展示测试 | | |
| **简要描述** | 本测试验证开发人员是否可以按照要求导出数据包的图形展示PDF。 | | |
| **前提和约束** | 正确配置python环境、正确安装Scapy和PyX包 | | |
| **测试方法** | 黑盒测试 | | |
| **测试步骤** | **1** | 在python代码中导入Scapy | |
| **2** | 利用Scapy的rdpcap函数读入待测试pcap文件 | |
| **3** | 设定layer\_shift参数 | |
| **4** | 调用pdfdump函数生成PDF | |
| **预期结果** | 正确导出报文数据包的图形展示 | | |
| **评价准则** | 报文图形展示内容与wireshark解析结果一致且完整，即为正确。 | | |
| **异常情况** | 文件格式有误 | | 程序提示可能的异常信息，并终止运行 |
| 报文数据有误 | | 在会影响程序正常执行的情况下提示异常信息，并终止运行 |
| **测试结果** |  | | |
| **测试结论** |  | | |

表格 2.11用例107 测试结果

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | pcap文件读取测试 | | |
| **简要描述** | 本测试验证开发人员是否可以正确读入pcap文件。 | | |
| **前提和约束** | 正确配置python环境、正确安装Scapy包 | | |
| **测试方法** | 黑盒测试 | | |
| **测试步骤** | **1** | 在python代码中导入Scapy | |
| **2** | 设置待读入文件存储位置、读取报文数量 | |
| **3** | 利用Scapy的rdpcap函数读入待测试pcap文件 | |
| **4** | 调用报文数据的show方法显示数据的字符展示 | |
| **预期结果** | 正确读入预期数量的报文数据 | | |
| **评价准则** | 读入后的pcap包数据展示结果与wireshark打开的结果相同即为正确，同时计算是否读入了指定数量的报文。 | | |
| **异常情况** | 文件格式有误 | | 程序提示可能的异常信息，并终止运行 |
| **测试结果** |  | | |
| **测试结论** |  | | |

表格 2.12用例108 测试结果

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | 16进制编码导入测试 | | |
| **简要描述** | 本测试验证开发人员是否可以按照要求以16进制编码格式导入报文数据。 | | |
| **前提和约束** | 正确配置python环境、正确安装Scapy包 | | |
| **测试方法** | 黑盒测试 | | |
| **测试步骤** | **1** | 在python代码中导入Scapy | |
| **2** | 利用Scapy的rdpcap函数读入待测试pcap文件 | |
| **3** | 调用报文数据的show方法显示原始数据的字符展示 | |
| 4 | 调用str函数将报文数据转换为16进制格式 | |
| **5** | 调用报文数据的show方法显示16进制转化出的数据的字符展示 | |
| **预期结果** | 正确地将16进制编码格式的报文数据读入为Scapy解析格式 | | |
| **评价准则** | 步骤3与步骤5所得结果相同 | | |
| **异常情况** | 文件格式有误 | | 程序提示可能的异常信息，并终止运行 |
| **测试结果** |  | | |
| **测试结论** |  | | |

表格 2.13用例109测试结果

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | pcap文件导出测试 | | |
| **简要描述** | 本测试验证开发人员是否可以正确导出pcap文件 | | |
| **前提和约束** | 正确配置python环境、正确安装Scapy包 | | |
| **测试方法** | 黑盒测试 | | |
| **测试步骤** | **1** | 在python代码中导入Scapy | |
| **2** | 利用Scapy的rdpcap函数读入待测试pcap文件 | |
| **3** | 调用wrpcap函数导出pcap文件 | |
| **预期结果** | 正确导出pcap文件 | | |
| **评价准则** | 导出后的pcap包数据与原始数据在wireshark中打开的结果相同。 | | |
| **异常情况** | 文件格式有误 | | 程序提示可能的异常信息，并终止运行 |
| **测试结果** |  | | |
| **测试结论** |  | | |

表格 2.14用例110 测试结果

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | 16进制编码格式导出测试 | | |
| **简要描述** | 本测试验证开发人员是否可以导出正确的16进制编码格式的报文数据。 | | |
| **前提和约束** | 正确配置python环境、正确安装Scapy包 | | |
| **测试方法** | 黑盒测试 | | |
| **测试步骤** | **1** | 在python代码中导入Scapy | |
| **2** | 利用Scapy的rdpcap函数读入待测试pcap文件 | |
| **3** | 调用str函数 | |
| **4** | print输出转换后结果 | |
| **预期结果** | 正确导出16进制格式的报文数据 | | |
| **评价准则** | 与wireshark中16进制结果作对比，相同 | | |
| **异常情况** | 文件格式有误 | | 程序提示可能的异常信息，并终止运行 |
| **测试结果** |  | | |
| **测试结论** |  | | |

表格 2.15用例111测试结果

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | 发送报文测试 | | |
| **简要描述** | 本测试用例测试Scapy发送报文的正确性 | | |
| **前提和约束** | 正确配置python环境、安装Scapy包 | | |
| **测试方法** | 黑盒测试 | | |
| **测试步骤** | **1** | 在python代码中导入scapy | |
| **2** | 构造网络层报文，并作为参数传入send函数 | |
| **3** | 构造数据链路层报文，并作为参数传入sendp函数，同时设置iface参数指定发送数据包的网卡。 | |
| **4** | 分别执行send和sendp函数 | |
| **预期结果** | 1. 运行send()函数后，IDE会显示成功发送一个数据包 2. 运行sendp()函数后，IDE会显示成功发送一个数据包 | | |
| **异常情况** | iface参数设置有误 | | 输出错误提示信息，并停止运行 |
| 报文构造有误 | | 输出错误提示信息，并停止运行 |
| **测试结果** |  | | |
| **测试结论** |  | | |

表格 2.16用例112测试结果

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | 发送并接收报文测试 | | |
| **简要描述** | 本测试用例测试Scapy发送并接收报文的正确性 | | |
| **前提和约束** | 正确配置python环境、安装Scapy包 | | |
| **测试方法** | 黑盒测试 | | |
| **测试步骤** | **1** | 在python代码中导入scapy | |
| **2** | 构造网络层报文，报文的目的地址设置为一个不可达的IP。将报文作为参数传入sr函数，并通过timeout和retry配置超时时间和重传次数。 | |
| **3** | 构造网络报文，并通过变量ans,unasns接受其返回值。然后通过通过两个变量的summary()方法显示受到返回值的报文和没有受到返回值的报文。 | |
| **4** | 分别执行2和3中的操作 | |
| **预期结果** | 1. 执行2中的操作后，可以看到scapy不断尝试发报文，尝试的次数为retry中配置的次数，并最终显示报文未送达，接收到0个返回值。 2. 调用ans变量的summary方法后显示出接收到返回值的报文，调用unans变量的summary方法后显示出未受到返回值的报文。 | | |
| **异常情况** | iface参数设置有误 | | 输出错误提示信息，并停止运行 |
| 报文构造有误 | | 输出错误提示信息，并停止运行 |
| **测试结果** |  | | |
| **测试结论** |  | | |

表格 2.17用例113测试结果

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | 文件切分功能的兼容模式测试 | | |
| **简要描述** | 本测试主要用于验证pcap文件切分功能的兼容模式能否正常运行 | | |
| **前提和约束** | 安装好扩展后Scapy及其运行环境 | | |
| **测试方法** | 黑盒测试 | | |
| **测试步骤** | **1** | 准备好待切分的pcap文件 | |
| **2** | 设置基本参数，包括监听模式、数据路径、结果路径、批大小，测试基本切分功能 | |
| **3** | 再设置文件名前缀参数，测试结果命名功能 | |
| **4** | 再设置链路层类型、存储类型参数，测试兼容模式下的自定义文件格式功能 | |
| **预期结果** | 步骤2 : 结果目录下产生切分好的pcap文件  步骤3 : 结果目录下产生切分好的、指定文件名前缀的pcap文件  步骤4 : 结果目录下产生切分好的、指定格式的pcap文件 | | |
| **异常情况** | 文件格式有误 | | 程序提示可能的异常信息，并终止运行 |
| 报文数据有误 | | 在会影响程序正常执行的情况下提示异常信息，并终止运行 |
| **测试结果** |  | | |
| **测试结论** |  | | |

表格 2.18用例114测试结果

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | 文件切分功能的极速模式测试 | | |
| **简要描述** | 本测试主要用于验证pcap文件切分功能的极速模式能否正常运行 | | |
| **前提和约束** | 安装好扩展后Scapy及其运行环境 | | |
| **测试方法** | 黑盒测试 | | |
| **测试步骤** | **1** | 准备好待切分的pcap文件 | |
| **2** | 设置基本参数，包括监听模式、数据路径、结果路径、批大小，测试基本切分功能 | |
| **3** | 设置文件名前缀参数，测试结果命名功能 | |
| **预期结果** | 步骤2 : 结果目录下产生切分好的pcap文件  步骤3 : 结果目录下产生切分好的、指定文件名前缀的pcap文件 | | |
| **异常情况** | 文件格式有误 | | 程序提示可能的异常信息，并终止运行 |
| 报文数据有误 | | 在会影响程序正常执行的情况下提示异常信息，并终止运行 |
| **测试结果** |  | | |
| **测试结论** |  | | |

表格 2.19用例115测试结果

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | 监听持久化功能的兼容模式测试 | | |
| **简要描述** | 本测试主要用于验证监听持久化功能的兼容模式能否正常运行 | | |
| **前提和约束** | 1. 安装好扩展后Scapy及其运行环境 2. 确保运行环境的设备和网络正常 | | |
| **测试方法** | 黑盒测试 | | |
| **测试步骤** | **1** | 设置基本运行参数，包扩监听模式、结果路径、批大小 | |
| **2** | 设置停止监听参数，测试不同的停止监听功能 | |
| **3** | 设置文件名前缀参数，测试结果命名功能 | |
| **4** | 设置端口参数，测试指定监听端口功能 | |
| **5** | 设置报文筛选条件，测试筛选报文功能 | |
| **6** | 打印出监听到的报文，测试监听到的报文是否被解析 | |
| **预期结果** | 步骤2 : 按照停止条件停止监听，并在结果目录下存储了监听到的报文数据文件  步骤3 : 结果目录下产生了指定文件名前缀的pcap文件  步骤4 : 结果文件中的报文数据来自于指定硬件端口  步骤5 : 结果文件中的报文数据符合筛选规则  步骤6 : 打印出的报文是Scapy已经解析后的格式 | | |
| **异常情况** | 监听到的报文数据有问题，无法正常解析 | | 输出错误日志，然后继续监听 |
| **测试结果** |  | | |
| **测试结论** |  | | |

表格 2.20用例116测试结果

|  |  |  |
| --- | --- | --- |
| **名称** | 监听持久化功能的极速模式测试 | |
| **简要描述** | 本测试主要用于验证监听持久化功能的极速模式能否正常运行 | |
| **前提和约束** | 1. 安装好扩展后Scapy及其运行环境 2. 确保运行环境的设备和网络正常 | |
| **测试方法** | 黑盒测试 | |
| **测试步骤** | **1** | 设置基本运行参数，包扩监听模式、结果路径、批大小 |
| **2** | 设置停止监听参数，测试不同的停止监听功能 |
| **3** | 设置文件名前缀参数，测试结果命名功能 |
| **4** | 设置端口参数，测试指定监听端口功能 |
| **5** | 设置报文筛选条件，测试筛选报文功能 |
| **6** | 打印出监听到的报文，测试监听到的报文是否按照自定义格式处理 |
| **预期结果** | 步骤2 : 按照停止条件停止监听，并在结果目录下存储了监听到的报文数据文件  步骤3 : 结果目录下产生了指定文件名前缀的pcap文件  步骤4 : 结果文件中的报文数据来自于指定硬件端口  步骤5 : 结果文件中的报文数据符合筛选规则  步骤6 : 打印出的报文是自定义的字典格式 | |
| **测试结果** |  | |
| **测试结论** |  | |

表格 2.21用例117测试结果

|  |  |  |
| --- | --- | --- |
| **名称** | 协议类型获取功能测试 | |
| **简要描述** | 本测试主要用于验证协议类型获取功能能否正常运行 | |
| **前提和约束** | 1. 安装好扩展后Scapy及其运行环境 | |
| **测试方法** | 黑盒测试 | |
| **测试步骤** | **1** | 准备好包含多种报文数据的pcap文件 |
| **2** | 使用Scapy本身的文件读取功能读取并解析报文数据 |
| **3** | 调用协议类型获取功能输出报文所有层协议类型、最高层可识别协议类型、指定层的协议类型 |
| **预期结果** | 能够正确输出Scapy解析后报文的指定协议信息 | |
| **测试结果** |  | |
| **测试结论** |  | |

表格 2.22用例118 会话提取功能测试

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | 会话提取功能测试 | | |
| **简要描述** | 本测试验证开发人员是否可以正确进行会话提取。主要测试包括四元组条件会话过滤功能，会话时间切分模式选择功能是否正常、切分的小文件正确性核对、分批处理的大小设置功能是否正常。 | | |
| **前提和约束** | 正确配置python环境、正确安装Scapy包 | | |
| **测试方法** | 黑盒测试 | | |
| **测试步骤** | **1** | 在python代码中导入Scapy | |
| **2** | 实例化SessionExtractor()类 | |
| **3** | 调用SessionExtractor()类中run方法，其中要设定待提取文件路径、会话文件输出路径、分批读入报文个数、存储文件前缀、四元组过滤器、会话切分时间阈值、会话切分模式、信息显示级别参数。即可提取出会话文件。   1. 改变会话过滤器参数，测试过滤功能。 2. 改变切分模式，测试切分模式选择功能是否正常。 3. 改变分批读入报文个数参数，测试分批处理功能是否正常。 | |
| **预期结果** | 按照设定要求正确提取目标文件中的会话。 | | |
| **评价准则** | 与wireshark中解析结果作对比，判断会话是否正确提取 | | |
| **异常情况** | 文件格式有误 | | 程序提示可能的异常信息，并终止运行 |
| **测试结果** |  | | |
| **测试结论** |  | | |

2.2 非功能测试用例结果

表格 2.23用例201测试结果

|  |  |  |
| --- | --- | --- |
| **名称** | 操作系统兼容性 | |
| **简要描述** | 本测试用例用于验证Scapy工具库是否能够运行于不同的操作系统上（Windows，Linux等）上，即在不同系统上Scapy各个基础功能模块是否可以按照预期方式执行，是否具备操作系统无关性。 | |
| **前提和约束** | 目标操作系统上已安装Python环境和Scapy工具库，准备好测试数据。 | |
| **测试方法** | 黑盒测试/非功能测试 | |
| **测试步骤** | **1** | 新建Python脚本，在其中编写依次调用各个主要功能模块对测试数据进行处理并输出结果的代码 |
| **2** | 在不同的系统下，执行该Python代码，并记录其输出结果 |
| **3** | 将各系统下的输出结果和预期结果进行对比 |
| **预期结果** | 不同系统下，执行没有出现运行时错误，且各输出结果和预期输出一致。 | |
| **测试结果** |  | |
| **测试结论** |  | |

表格 2.24用例202 测试结果

|  |  |  |
| --- | --- | --- |
| **名称** | Python版本兼容性 | |
| **简要描述** | 本测试用例用于验证Scapy工具库是否提供了能运行于要求的不同版本Python环境上的代码（Python2.7+，Python3.4+），即Scapy工具库是否提供了Python2/3环境下的支持，以适应实际复杂的生产环境。 | |
| **前提和约束** | 目标操作系统上已安装Anaconda环境，并以此创建了要求的不同版本Python虚拟环境（2.7，3.4，…，3.7） | |
| **测试方法** | 黑盒测试/非功能测试 | |
| **测试步骤** | **1** | 在不同虚拟环境下，执行pip install Scapy命令安装Scapy工具库 |
| **2** | 安装完成后，运行虚拟环境python解释器，执行import scapy.all as scapy |
| **3** | 运行Scapy控制台工具 |
| **预期结果** | 不同版本环境下，执行Scapy库导入成功，控制台运行成功 | |
| **测试结果** |  | |
| **测试结论** |  | |

表格 2.25用例203测试结果

|  |  |  |
| --- | --- | --- |
| **名称** | 对于新协议的扩展性 | |
| **简要描述** | 本测试用例用于验证基于Scapy的分析与扩展项目的最终产品是否具有对于新协议的扩展性，即是否能够在尽量少修改项目代码的基础上完成对一个全新协议的支持。 | |
| **前提和约束** | Scapy正常运行所需要的环境均已配置好，且准备好待添加的新协议字段格式。 | |
| **测试方法** | 黑盒测试/非功能测试 | |
| **测试步骤** | **1** | Python代码中创建新协议类，继承自Packet父类，并依据新协议格式定义字段名称、类型及内容 |
| **2** | 使用bind\_layers函数将新协议类和已存在的协议层进行绑定（例如将新协议插入在TCP层的payload中） |
| **3** | 使用Scapy数据构造模块构造出包含该新协议的报文对象 |
| **4** | 使用数据存储模块将报文对象存储为pcap文件 |
| **5** | 使用数据导入模块导入该pcap文件为报文对象 |
| **6** | 对比直接构造的报文对象和导入的报文对象 |
| **预期结果** | 整个过程中，Scapy可以正常构造、存储，导入和解析包含该新协议的报文，且前后的报文对象一致，对该新协议的扩展无需修改Scapy原本的项目代码。 | |
| **测试结果** |  | |
| **测试结论** |  | |

表格 2.26用例204测试结果

|  |  |  |
| --- | --- | --- |
| **名称** | 非法数据解析 | |
| **简要描述** | 本测试用例用于验证基于Scapy的分析与扩展项目的最终产品是否能够处理非法的数据报文，即对于包含非法字段或者非法的协议层是否进行原始保留处理，将其封装入Raw类型对象中。 | |
| **前提和约束** | Scapy正常运行所需要的环境均已配置好，且准备好包含非法内容的数据报文。 | |
| **测试方法** | 黑盒测试/非功能测试 | |
| **测试步骤** | **1** | 调用数据导入和解析模块对非法数据报文文件进行导入和解析 |
| **2** | 调用解析后报文对象的show()方法查看解析的各层协议类型 |
| **3** | 将得到的协议类型和数据文件的真实情况进行对比 |
| **预期结果** | 导入和解析过程中没有出现运行时错误，且从包含非法数据的协议层向上的数据部分被封装为Raw协议层类型 | |
| **测试结果** |  | |
| **测试结论** |  | |

表格 2.27用例205测试结果

|  |  |  |
| --- | --- | --- |
| **名称** | 交互模式函数调用 | |
| **简要描述** | 本测试用例用于验证基于Scapy的分析与扩展项目的最终产品是否能成功进入交互模式，并调用各个功能函数进行处理。 | |
| **前提和约束** | Scapy正常运行所需要的环境均已配置完毕 | |
| **测试方法** | 黑盒测试/非功能测试 | |
| **测试步骤** | **1** | 在命令行下输入命令scapy，进入Scapy交互模式 |
| **2** | 使用a=IP(ttl=10) Ether()/IP()/IP()/UDP()等类似命令进行packet创建或进行packet堆叠操作 |
| **3** | 使用a=rdpcap("/spare/captures/isakmp.cap")类似命令读取pcap文件，使用wrpcap写入pcap文件 |
| **4** | 使用pdfdump，psdump进行报文可视化展示 |
| **5** | 类似的调用Scapy中其他的功能函数（数据监听与发送，扩展部分功能函数等）进行报文处理 |
| **6** | 将各条命令执行结果和import导入方式函数执行结果进行对比 |
| **预期结果** | 在交互模式下各个功能函数完成了与库导入形式下相同的操作 | |
| **测试结果** |  | |
| **测试结论** |  | |