Курсова работа
по “Мрежова Сигурност I”
СУ “Св. Климент Охридкси”
Факултет по Математика и Информатика

PCAP-FILTER

Станислав Петров, 61055
Красимир Байлов, 61080

Дата: 15 Януари 2011

Съдържание

1. Задание

Заданието е копирано от сайта “Учебен център на ISECA”:

PCAP филтър (6/10, двама души)
Да се напише филтър за потоци от пакети, който да променя съдържанието им, като запазва валидността им (дължина, headers, checksums и т.н.)

Вход: PCAP поток - указан мрежови интерфейс, файл или стандартен вход
Изход: PCAP поток - файл или стандартен изход; незадължително: изпращане на пакетите по мрежови интерфейс
Статистика: брой открити и анализирани потоци, брой променени потоци, брой променени пакети във всеки поток, статистически разпределения

Основна функционалност: промяна на UDP-пакети, носещи информация за SIP-сесии, като премахват от тях полето Billing-Credit-Time или друго, зададено в конфигурационен файл или промяна на UDP-пакети, носещи информация за RADIUS-сесии, като променят в тях полето NAS-IP-Address на зададен адрес или други, зададени в конфигурационен файл
Допълнителна функционалност (за бонус точки): промяна на TCP-пакети, носещи информация за HTTP-сесии, като премахват от тях хедъра X-Forwarded-For и променят полето User-Agent или други, зададени в конфигурационен файл, като резултатът продължава да бъде валиден TCP-поток (offsets, sliding window и т.н.).

2. Изтегляне на приложението от хранилище (repository)

Проектът “**pcap-filter**” е качен в сайта “Google Code” и може да бъде изтеглен в режим на преглед от адрес http://code.google.com/p/pcap-filter/

3. Описание на приложението

a. Изисквания

Преди да стартирате проложението трябва да имате инсталиран следния софтуер:

- Java Runtime Environment 6 (ако искате да компилирате source code - JDK 6)
- WinPcap (windows) или libpcap(UNIX/Linux)
- Jpcap - пакетът може да бъде свален от следния линк: http://netresearch.ics.uci.edu/kfujii/Jpcap/doc/download.html

b. Функционалност

Приложението “**pcap-fitler**” прихваща пакети от указан мрежови интерфейс или

входен файл с данни. Променя UDP пакети, които носят информация за SIP сесии и TCP пакети, които носят информация за HTTP сесии.

c. Конфигурация

Всички конфигурационни файлове са с разширение *.properties, т.нар. properties файлове, които се използват от програмния език Java. Важно е да се отбележи, че за коментар се счита знакът "#" поставен в началото на реда. Долуописаните файлове имат предоставена помощна информация за ползването им.

Приложението се конфигурира от файл "**pcap_filter_config.properties**".

Конфигурационните ключове са следните:

Конфигурационен ключ	Описание
filter	Задава се правило на филтриране на потока от пакети
sip_mask_file	Име на конфигурационен файл, в който се задават какви полета ще се променят в SIP пакет
http_mask_file	Име на конфигурационен файл, в който се задават какви полета ще се променят в HTTP пакет

Описание на стойностите на конфигурационния ключ "**filter**":

Стойност	Описание
portrange 5060-5080	Филтър за прихващане на UDP пакети, които носят SIP сесии. Портовете на UDP пакетите попадат в зададения интервал от номера на портове
port 80	Филтър за прихващане на пакети с порт 80, т.е. пакети, които носят информация за HTTP сесии

Описание на стойностите на конфигурационните ключове "**sip_mask_file**" и "**http_mask_file**":

Конфигурационен ключ	Стойност	Описание

sip_mask_file	SIP_MASK.properties	Име на конфигурационен файл за работа със SIP пакети
http_mask_file	HTTP_MASK.properties	Име на конфигурационен файл за работа с HTTP пакети

d. Компилиране

Приложението е предварително компилирано в предоставения "pcap-filter.jar" файл, който се намира в директория "pcap-filter/application/pcap-filter_jar". За да компилирате source кода, трябва да сте покрили следните изисквания:
- Инсталиран JDK 6 (jdk1.6.0_16)
- конфигуриран пълен път до библиотеката "**Jpcap**" в променливата (environment variable) "**CLASSPATH"**

Влезте в директория "**pcap-filter/application/pcap-filter/src**" и компилирайте с команда "**javac**". Друг начин за компилиране е като направите проект от наличния source code в Eclipse 3.5 (или по-нов).

e. Изпълнение

За да изпълните приложението трябва да имате конфигуриран път за достъп до командата "**java**" през команден ред. Влезте в директория "**pcap-filter/application/pcap-filter_jar**" и изпълнете следната команда, за да стартирате приложението:

```
C:\PcapFilterDir java -jar pcap-filter.jar
```

Командата е описана във файл **pcap-filter_jar/start_pcap_filter.bat**

След като програмата бъде стартирана ще бъде визуализирано конзолно меню, което съпътства работата на потребителите. При следване на инструкциите трябва да се въведат следните данни:

- Тип на източника, от който се четат пакети (файл, мрежова карта)
- източник от който се четат пакетите (име на файла или номер на мрежовия интерфейс)
- Тип на устройството, където ще се пишат/изпзращат пакетите
- Устройтсвото, където ще се пишат/изпращат пакетите

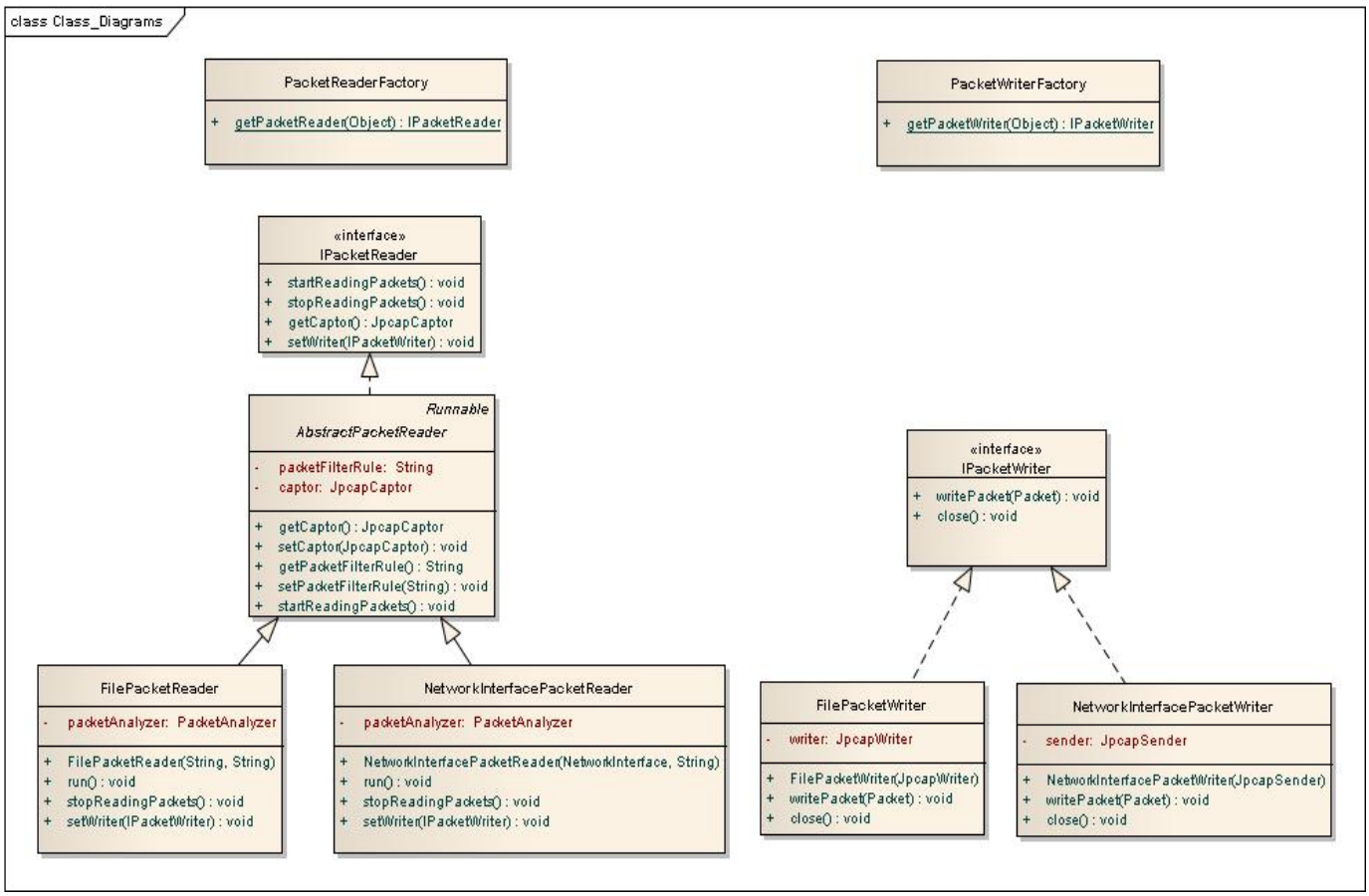
```
E:\Krasi\Lectures\Course_4\Network_Security\Project_Packet_Captu
ring\pcap-filter
\application\pcap-filter_jar>java -jar pcap-filter.jar
select input source:
type "file" for file
```

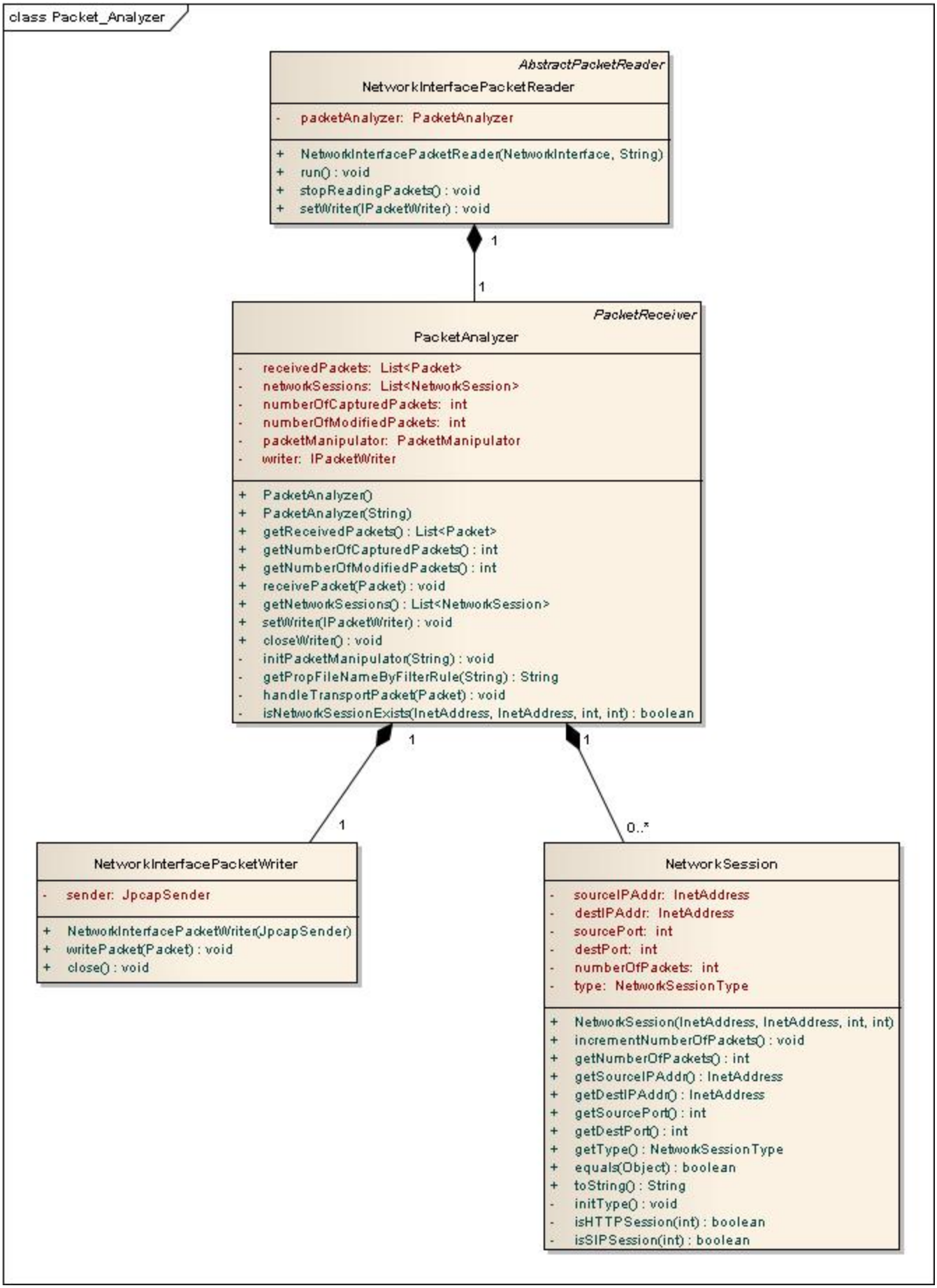
```
type "network" for network interface
type "exit" for exit
file                                    определяме, че ще четем от файл
Enter file name:
sip_session_captured.pcap               посочваме файла
Select packet destination:
type "file" for file
type "network" for network interface
type "system" for System.out
type "exit" for exit
network                 ще изпращаме пакетите през мрежовия интерфейс
Please choose device from list below:           предоставя ни се
1, VMware Virtual Ethernet Adapter          списък от налични
2, Microsoft                                    мрежови интерфейси
3, VMware Virtual Ethernet Adapter
4, Realtek RTL8101E PCI-E Fast Ethernet NIC

Enter number: 4                     избираме мрежови интерфейс
Start listening for packets...
```

Забележка: За да спрете изпълнението на праграмата, когато четете от мрежовия интерфейс въведете команда **`stop`** и натиснете бутона Enter.

f. UML Class диаграми на приложението


class Class_Diagrams
PacketReaderFactory
+ getPacketReader(Object) : IPacketReader
PacketWriterFactory
+ getPacketWriter(Object) : IPacketWriter
«interface»
IPacketReader
+ startReadingPackets() : void
+ stopReadingPackets() : void
+ getCaptor() : JpcapCaptor
+ setWriter(IPacketWriter) : void
Runnable
AbstractPacketReader
- packetFilterRule: String
- captor: JpcapCaptor
+ getCaptor() : JpcapCaptor
+ setCaptor(JpcapCaptor) : void
+ getPacketFilterRule() : String
+ setPacketFilterRule(String) : void
+ startReadingPackets() : void
«interface»
IPacketWriter
+ writePacket(Packet) : void
+ close() : void
FilePacketReader
- packetAnalyzer: PacketAnalyzer
+ FilePacketReader(String, String)
+ run() : void
+ stopReadingPackets() : void
+ setWriter(IPacketWriter) : void
NetworkInterfacePacketReader
- packetAnalyzer: PacketAnalyzer
+ NetworkInterfacePacketReader(NetworkInterface, String)
+ run() : void
+ stopReadingPackets() : void
+ setWriter(IPacketWriter) : void
FilePacketWriter
- writer: JpcapWriter
+ FilePacketWriter(JpcapWriter)
+ writePacket(Packet) : void
+ close() : void
NetworkInterfacePacketWriter
- sender: JpcapSender
+ NetworkInterfacePacketWriter(JpcapSender)
+ writePacket(Packet) : void
+ close() : void


class Packet_Analyzer
AbstractPacketReader
NetworkInterfacePacketReader
- packetAnalyzer: PacketAnalyzer
+ NetworkInterfacePacketReader(NetworkInterface, String)
+ run() : void
+ stopReadingPackets() : void
+ setWriter(IPacketWriter) : void
1
1
PacketReceiver
PacketAnalyzer
- receivedPackets: List<Packet>
- networkSessions: List<NetworkSession>
- numberOfCapturedPackets: int
- numberOfModifiedPackets: int
- packetManipulator: PacketManipulator
- writer: IPacketWriter
+ PacketAnalyzer()
+ PacketAnalyzer(String)
+ getReceivedPackets() : List<Packet>
+ getNumberOfCapturedPackets() : int
+ getNumberOfModifiedPackets() : int
+ receivePacket(Packet) : void
+ getNetworkSessions() : List<NetworkSession>
+ setWriter(IPacketWriter) : void
+ closeWriter() : void
- initPacketManipulator(String) : void
- getPropFileNameByFilterRule(String) : String
- handleTransportPacket(Packet) : void
- isNetworkSessionExists(InetAddress, InetAddress, int, int) : boolean
1
1
1
0..*
NetworkInterfacePacketWriter
- sender: JpcapSender
+ NetworkInterfacePacketWriter(JpcapSender)
+ writePacket(Packet) : void
+ close() : void
NetworkSession
- sourceIPAddr: InetAddress
- destIPAddr: InetAddress
- sourcePort: int
- destPort: int
- numberOfPackets: int
- type: NetworkSessionType
+ NetworkSession(InetAddress, InetAddress, int, int)
+ incrementNumberOfPackets() : void
+ getNumberOfPackets() : int
+ getSourceIPAddr() : InetAddress
+ getDestIPAddr() : InetAddress
+ getSourcePort() : int
+ getDestPort() : int
+ getType() : NetworkSessionType
+ equals(Object) : boolean
+ toString() : String
- initType() : void
- isHTTPSession(int) : boolean
- isSIPSession(int) : boolean

4. Как беше написан този документ

Целият материал в този документ е написан в Google Docs. Оформен е с шрифтове Times New Roman, Courier New. Резултатът е свален в PDF формат. По желание на преподавателите може да им бъде предоставен достъп за преглед на документа в Google Docs.

5. Използвани източници

http://netresearch.ics.uci.edu/kfujii/Jpcap/doc/
http://www.winpcap.org/install/default.htm
http://sourceforge.net/projects/libpcap/
RFC 3261 SIP: Session Initiation Protocol (http://www.ietf.org/rfc/rfc3261.txt)
RFC 2616 Hypertext Transfer Protocol -- HTTP/1.1 (http://www.w3.org/Protocols/rfc2616/rfc2616.html)
http://www.apl.jhu.edu/~hall/java/beginner/settingup.html
http://download.oracle.com/javase/1.4.2/docs/tooldocs/windows/javac.html
http://training.iseca.org/