# Package 'rpcap'

February 17, 2016

**Type** Package

**Title** A package to process network packet header data

**Version** 1.0

**Date** 2016-01-21

**Author** Denis Hock

**Maintainer** Denis Hock <dehock@fb2.fra-uas.de>

**Description** The package rpcap contains example traces of network traffic extracted from the DARPA Intrusion Detection Evaluation Data Set and a set of functions to easily generate statistics to interpret and analyze packet header data.

**URL** https://github.com/fg-netzwerksicherheit/



**License** GPL-3

## R topics documented:

rpcap-package *Package rpcap*

## Description

A Package to process network packet header data.

## Details

The package rpcap contains example traces of network traffic extracted from the DARPA Intrusion Detection Evaluation Data Set (see: packetHeader), a two list which map protocol number to protocol name (see: protocol) as well as port number to name (see: port) extracted from IANA. The provided functions aim to easily generate statistics to interpret and analyze packet header data in csv format.

## Author(s)

Frankfurt University of Applied Sciences Maintainer: Denis Hock

## References

Hock, D., and M. Kappes. "Using R for Anomaly Detection in Network Traffic." Fifth International Conference on Internet Technologies & Applications. 2013.

aggregatePerTimeUnit *Aggregate data per TimeUnit*

## Description

Summarize the packets by time windows and an optional filter. Returns the result of a given function.

## Usage

```
aggregatePerTimeUnit(vector, frame.time, unit = "sec", filter = vector == vector, FUN = length)
```

## Arguments

| | |
|---|---|
| vector | Field to summarize |
| frame.time | Vector with POSIX time |
| unit | sec, min, ... |
| filter | e.g. packetHeader$tcp.dstport == 80, ... |
| FUN | A function to summarize each the vector for each time window |

## Examples

```
data(packetHeader)
aggregatePerTimeUnit(packetHeader$tcp.dstport, packetHeader$frame.time)
```

---

`frequencyPerValue` *Count the occurences of each unique value*

---

## Description

This function counts the occurences of each unique value and is e.g. used to analyze categorical data such as IP addresses.

## Usage

```
frequencyPerValue(vector)
```

## Arguments

| | |
|---|---|
| `vector` | Your input variable. |

## Value

Returns a data.frame with each unique value and the number of occurences

## Examples

```
data(packetHeader)
frequencyPerValue(packetHeader$ip.src)
```

---

`packetHeader` *Data Set: PacketHeader*

---

## Description

This data contains packet header data of a single day from the 1999 DARPA Intrusion Detection Evaluation Data Set.

## Usage

```
data("packetHeader")
```

## Format

packetHeader is a data.frame containing following vectors: ip.src, ip.dst, ip.proto, frame.len, frame.time, tcp.flags, tcp.dstport, tcp.srcport, udp.dstport, udp.srcport

## Details

The Cyber Systems and Technology Group of MIT Lincoln Laboratory, under Defense Advanced Research Projects Agency and Air Force Research Laboratory sponsorship, has collected and distributed the first standard corpora for evaluation of computer network intrusion detection systems.

A test bed generated live background traffic similar to that on a government site containing hundreds of users on thousands of hosts. More than 200 instances of 58 attack types were launched against victim UNIX and Windows NT hosts in three weeks of training data and two weeks of test data.

The data set has the objectives to to measure the effectiveness of an IDS in detecting intrusive behavior in the presence of normal computer and network activity and to measure the effectiveness of response mechanisms and their impact on normal users.

Following types of misuse can be present: Denial of service, Unauthorized access from a remote machine, Unauthorized transition to root by an unprivileged user, Surveillance and probing, Anomalous user behavior. Many of these attacks are taken directly from Kris Kendall's MIT Master's thesis.

## Source

https://www.ll.mit.edu/ideval/index.html

## References

Haines, Joshua W., et al. DARPA intrusion detection system evaluation: Design and procedures. Technical Report 1062, MIT Lincoln Laboratory, 2001.

## Examples

```
data(packetHeader)
plot(packetHeader$tcp.dstport)
```

---

`port` *Data Set: Port*

---

## Description

This data.frame provides a mapping between port numbers and names, as assigned by IANA

## Usage

```
data("port")
```

## Format

A data frame with 11485 observations on the following 3 variables.

`Service.Name` a character vector

`Port.Number` a numeric vector

`Transport.Protocol` a character vector

## Details

Service names and port numbers are used to distinguish between different services that run over transport protocols such as TCP, UDP, DCCP, and SCTP.

Service names are assigned on a first-come, first-served process, as documented in [RFC6335].

Port numbers are assigned in various ways, based on three ranges: System Ports (0-1023), User Ports (1024-49151), and the Dynamic and/or Private Ports (49152-65535); the difference uses of these ranges is described in [RFC6335]. System Ports are assigned by IETF process for standards-track protocols, as per [RFC6335]. User Ports are assigned by IANA using the "IETF Review" process, the "IESG Approval" process, or the "Expert Review" process, as per [RFC6335]. Dynamic Ports are not assigned.

The registration procedures for service names and port numbers are described in [RFC6335].

## Source

https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.txt

## Examples

```
data(port)
portNumberToName(c(80,21))
```

---

# portNumberToName *Resolves port numbers*

---

## Description

Expects one or more port numbers and returns a vector of port names.

## Usage

```
portNumberToName(vector)
```

## Arguments

| | |
|---|---|
| vector | A vector of port numbers |

## Details

This function maps the assigned port numbers to the corresponding service name.

## Value

Returns a vector of port names.

## See Also

port

## Examples

```
data(port)
portNumberToName(port$Port.Number)
```

---

`protocol` *Data Set: Protocol*

---

## Description

This data.frame provides a mapping between protocol numbers and names, as assigned by IANA

## Usage

```
data("protocol")
```

## Format

A data frame with 147 observations on the following 2 variables.

`Decimal` a numeric vector

`Keyword` a character vector

## Details

In the Internet Protocol version 4 (IPv4) [RFC791] there is a field called "Protocol" to identify the next level protocol. This is an 8 bit field. In Internet Protocol version 6 (IPv6) [RFC2460], this field is called the "Next Header" field.

## Source

http://www.iana.org/assignments/protocol-numbers/protocol-numbers.txt

## Examples

```
data(protocol)
protocolNameToNumber(c('icmp','tcp'))
protocolNumberToName(c(1,6))
```

## protocolNameToNumber *Resolves protocol names*

### Description

Expects one or more protocol names and returns a vector of protocol numbers.

### Usage

```
protocolNameToNumber(vector)
```

### Arguments

| | |
|---|---|
| vector | A vector of protocol keywords |

### Details

This function maps the assigned Internet protocol keywords to decimal numbers.

### Value

Returns a vector of protocol numbers.

### See Also

protocol, protocolNumberToName

### Examples

```
data(protocol)
protocolNameToNumber('icmp')
```

## protocolNumberToName *Resolves protocol numbers*

### Description

Expects one or more protocol numbers and returns a vector of protocol names.

### Usage

```
protocolNumberToName(vector)
```

### Arguments

| | |
|---|---|
| vector | A vector of protocol numbers |

### Details

This function maps the assigned Internet protocol numbers to a keyword.

### Value

Returns a vector of protocol names.

### See Also

protocol, protocolNameToNumber

### Examples

```
data(protocol)
protocolNumberToName(1)
```

---

## readPcapCsv *Read and prepare a pcap-files*

---

### Description

Reads a csv-file with packet header data, changes the timestamp format from wireshark "Jul 29, 2015 10:50:11.596093000 CEST" to POSIX and replaces missing values (e.g. ports) with -1. Please note that it is required to install this locale on your operating system! If you don't need this data preprocessing or don't want to use frame.time in your data, use the normal read.csv().

### Usage

```
readPcapCsv(filename, sep = ",")
```

### Arguments

| | |
|---|---|
| filename | filname of your csv-file |
| sep | your separator (A semicolon probably works best, the comma, dot or whitespaces are included in wiresharks times-format) |

### Value

Returns a data.frame

### Note

You can export pcap-files using the following commands: "apt-get install tshark" "tshark -r test.pcap -T fields -e frame.time -e ip.src -e ip.dst -e ip.proto -E header=y -E separator=',' > test.csv" The keywords for the fields to export are equal to the expressions you can use in wireshark.

You can install the locale en_US.utf8 using "sudo dpkg-reconfigure locales"

## References

http://linux.die.net/man/1/tshark

---

`stats` *Min, Max, Mean*

---

## Description

A function to summarize packet header fields.

## Usage

```
stats(vector)
```

## Arguments

| | |
|---|---|
| `vector` | Accepts any numeric vector |

## Value

Returns an dataframe with following values:

`min`

`max`

`mean`

## Examples

```
data(packetHeader)
stats(packetHeader$frame.len)
```

# Index