

# Package ‘mitre’

May 21, 2021

**Type** Package

**Title** Cybersecurity MITRE Standards Data and Digraphs

**Version** 1.0.0

**Maintainer** Humbert Costas <humbert.costas@gmail.com>

**Description** Extract, transform and load MITRE standards.

This package gives you an approach to cybersecurity data sets.

All data sets are build on runtime downloading raw data from MITRE public services.

MITRE <<https://www.mitre.org/>> is a government-funded research organization based in Bedford and McLean. Current version includes most used standards as data frames. It also provide a list of nodes and edges with all relationships.

**License** CC0

**URL** <https://github.com/motherhack3r/mitre>

**BugReports** <https://github.com/motherhack3r/mitre/issues>

**Encoding** UTF-8

**Imports** rlang, plyr, dplyr, igraph, stringr, jsonlite, RJSONIO, tidyr

**RoxygenNote** 7.1.1

**Suggests** rmarkdown, knitr, testthat (>= 3.0.0)

**VignetteBuilder** knitr

**Depends** R (>= 2.10)

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Humbert Costas [aut, cre]

**Repository** CRAN

**Date/Publication** 2021-05-21 07:20:03 UTC

## R topics documented:

attck.groups . . . . .	2
attck.mitigations . . . . .	3

attck.relations . . . . .	3
attck.software . . . . .	3
attck.tactics . . . . .	4
attck.techniques . . . . .	4
build_edges . . . . .	4
build_network . . . . .	5
build_nodes . . . . .	6
capec.categories . . . . .	6
capec.patterns . . . . .	7
capec.relations . . . . .	7
capec.views . . . . .	7
car.analytics . . . . .	8
car.coverage . . . . .	8
car.implementations . . . . .	8
car.model . . . . .	9
car.relations . . . . .	9
car.sensors . . . . .	9
cpe.nist . . . . .	10
cve.nist . . . . .	10
cwe.categories . . . . .	10
cwe.views . . . . .	11
cwe.weaknesses . . . . .	11
newEdge . . . . .	11
newNode . . . . .	12
shield.opportunities . . . . .	12
shield.procedures . . . . .	13
shield.relations . . . . .	13
shield.tactics . . . . .	13
shield.techniques . . . . .	14
shield.use_cases . . . . .	14

<b>Index</b>	<b>15</b>
--------------	-----------

---

attck.groups	<i>ATT&amp;CK Groups Objects.</i>
--------------	-----------------------------------

---

### Description

Full data set provided by MITRE

### Usage

attck.groups

### Format

A data frame with 11 variables.

---

attck.mitigations      *ATT&CK Mitigation Objects.*

---

**Description**

Full data set provided by MITRE

**Usage**

attck.mitigations

**Format**

A data frame with 12 variables.

---

attck.relations      *ATT&CK relations Objects.*

---

**Description**

Full data set provided by MITRE

**Usage**

attck.relations

**Format**

A data frame with 13 variables.

---

attck.software      *ATT&CK software Objects.*

---

**Description**

Full data set provided by MITRE

**Usage**

attck.software

**Format**

A data frame with 12 variables.

---

attck.tactics	<i>ATT&amp;CK tactics Objects.</i>
---------------	------------------------------------

---

**Description**

Full data set provided by MITRE

**Usage**

attck.tactics

**Format**

A data frame with 11 variables.

---

attck.techniques	<i>ATT&amp;CK techniques Objects.</i>
------------------	---------------------------------------

---

**Description**

Full data set provided by MITRE

**Usage**

attck.techniques

**Format**

A data frame with 15 variables.

---

build_edges	<i>Extract relationships between standards as edges in a data frame.</i>
-------------	--

---

**Description**

**from** : node id of edge start **to** : node id of edge end **from\_std** : standard id of edge start **to\_std** : standard id of edge end **value** : When a value is set, the nodes will be scaled using the options in the scaling object defined above. **title** : The title is shown in a pop-up when the mouse moves over the edge. **arrows** : To draw an arrow with default settings a string can be supplied. For example: 'to, from,middle' or 'to;from', any combination with any separating symbol is fine. If you want to control the size of the arrowheads, you can supply an object. **dashes** : When true, the edge will be drawn as a dashed line. **color** : Color for the node. **hidden** : When true, the node will not be shown. It will still be part of the physics simulation though!

**Usage**

```
build_edges(verbose = FALSE)
```

**Arguments**

verbose            logical, FALSE by default. Change it to see the process messages.

**Value**

data.frame

---

build_network	<i>Create a list of nodes and edges related to all standards in data folder.</i>
---------------	--

---

**Description**

Create a list of nodes and edges related to all standards in data folder.

**Usage**

```
build_network(verbose = FALSE, as_igraph = TRUE)
```

**Arguments**

verbose            logical, FALSE by default. Change it to see the process messages.

as\_igraph          logical, TRUE by default. Change it to get list of nodes and edges.

**Value**

list, containing nodes and edges as data frames

**Examples**

```
mitrenet <- mitre::build_network(as_igraph = FALSE)
```

---

build_nodes	<i>Transform all standards as nodes in a data frame.</i>
-------------	--

---

### Description

**id** : The id of the node unique value for all standard elements. **label** : The label is the piece of text shown in or under the node, depending on the shape. **group** : When not undefined, the group of node(s) **type** : Used as subgroup to classify different object from **value** : When a value is set, the nodes will be scaled using the options in the scaling object defined above. **title** : Title to be displayed when the user hovers over the node. The title can be an HTML element or a string containing plain text or HTML. **standard** : The id of the standard shape : The shape defines what the node looks like. The types with the label inside of it are: ellipse, circle, database, box, text. The ones with the label outside of it are: image, circularImage, diamond, dot, star, triangle, triangleDown, square and icon. **color** : Color for the node. **hidden** : When true, the node will not be shown. It will still be part of the physics simulation though! **mass** : Default to 1. The barnesHut physics model (which is enabled by default) is based on an inverted gravity model. By increasing the mass of a node, you increase it's repulsion. Values lower than 1 are not recommended. **description** : Description could include extra information or nested data which include other columns from original data frame observation.

### Usage

```
build_nodes(verbose = FALSE)
```

### Arguments

**verbose**            logical, FALSE by default. Change it to see the process messages.

### Value

data.frame

---

capec.categories	<i>CAPEC categories Objects.</i>
------------------	----------------------------------

---

### Description

Full data set provided by MITRE

### Usage

```
capec.categories
```

### Format

A data frame with 4 variables.

---

capec.patterns	<i>CAPEC patterns Objects.</i>
----------------	--------------------------------

---

**Description**

Full data set provided by MITRE

**Usage**

capec.patterns

**Format**

A data frame with 16 variables.

---

capec.relations	<i>CAPEC relations Objects.</i>
-----------------	---------------------------------

---

**Description**

Full data set provided by MITRE

**Usage**

capec.relations

**Format**

A data frame with 4 variables.

---

capec.views	<i>CAPEC views Objects.</i>
-------------	-----------------------------

---

**Description**

Full data set provided by MITRE

**Usage**

capec.views

**Format**

A data frame with 5 variables.

---

car.analytics      *CAR analytics Objects.*

---

**Description**

Full data set provided by MITRE

**Usage**

car.analytics

**Format**

A data frame with 17 variables.

---

car.coverage      *CAR coverage Objects.*

---

**Description**

Full data set provided by MITRE

**Usage**

car.coverage

**Format**

A data frame with 4 variables.

---

car.implementations      *CAR implementations Objects.*

---

**Description**

Full data set provided by MITRE

**Usage**

car.implementations

**Format**

A data frame with 7 variables.



---

car.model	<i>CAR data model Objects.</i>
-----------	--------------------------------

---

**Description**

Full data set provided by MITRE

**Usage**

car.model

**Format**

A data frame with 8 variables.

---

car.relations	<i>CAR relations Objects.</i>
---------------	-------------------------------

---

**Description**

Full data set provided by MITRE

**Usage**

car.relations

**Format**

A data frame with 2 variables.

---

car.sensors	<i>CAR sensors Objects.</i>
-------------	-----------------------------

---

**Description**

Full data set provided by MITRE

**Usage**

car.sensors

**Format**

A data frame with 5 variables.

---

cpe.nist	<i>Common Platform Enumeration.</i>
----------	-------------------------------------

---

**Description**

Full data set provided by NIST.

**Usage**

cpe.nist

**Format**

A data frame with 16 variables: title, cpe.22, cpe.23, and all separated values.

---

cve.nist	<i>Common Vulnerability Enumeration.</i>
----------	--

---

**Description**

Full data set provided by NIST.

**Usage**

cve.nist

**Format**

A data frame with 34 variables: cve.id, problem.type which is related to CWE, description, vulnerable.configuration which is related to CPE, references, cvss3, cvss2 and all separated values.

---

cwe.categories	<i>CWE categories Objects.</i>
----------------	--------------------------------

---

**Description**

Full data set provided by MITRE

**Usage**

cwe.categories

**Format**

A data frame with 7 variables.

---

cwe.views

*CWE views Objects.*


---

**Description**

Full data set provided by MITRE

**Usage**

cwe.views

**Format**

A data frame with 7 variables.

---

cwe.weaknesses

*CWE Weaknesses Objects.*


---

**Description**

Full data set provided by MITRE

**Usage**

cwe.weaknesses

**Format**

A data frame with 24 variables.

---

newEdge

*Create an empty node*


---

**Description**

from : node id of edge start to : node id of edge end from\_std : standard id of edge start to\_std : standard id of edge end title : The title is shown in a pop-up when the mouse moves over the edge. value : When a value is set, the nodes will be scaled using the options in the scaling object defined above. label : The label of the edge. HTML does not work in here because the network uses HTML5 Canvas. arrows : To draw an arrow with default settings a string can be supplied. For example: 'to, from,middle' or 'to;from', any combination with any separating symbol is fine. If you want to control the size of the arrowheads, you can supply an object. dashes : When true, the edge will be drawn as a dashed line. hidden : When true, the node will not be shown. It will still be part of the physics simulation though! color : Color for the node. hidden : When true, the node will not be shown. It will still be part of the physics simulation though!

**Usage**

```
newEdge()
```

**Value**

```
data.frame
```

---

```
newNode
```

```
Create an empty node
```

---

**Description**

id : The id of the node unique value for all standard elements. label : The label is the piece of text shown in or under the node, depending on the shape. group : When not undefined, the group of node(s) type : Used as subgroup to classify different object from value : When a value is set, the nodes will be scaled using the options in the scaling object defined above. title : Title to be displayed when the user hovers over the node. The title can be an HTML element or a string containing plain text or HTML. standard : The id of the standard shape : The shape defines what the node looks like. The types with the label inside of it are: ellipse, circle, database, box, text. The ones with the label outside of it are: image, circularImage, diamond, dot, star, triangle, triangleDown, square and icon. color : Color for the node. hidden : When true, the node will not be shown. It will still be part of the physics simulation though! mass : Default to 1. The "barnesHut" physics model (which is enabled by default) is based on an inverted gravity model. By increasing the mass of a node, you increase it's repulsion. Values lower than 1 are not recommended. description : Description could include extra information or nested data which include other columns from original data frame observation.

**Usage**

```
newNode()
```

**Value**

```
data.frame
```

---

```
shield.opportunities SHIELD opportunities Objects.
```

---

**Description**

Full data set provided by MITRE

**Usage**

```
shield.opportunities
```

**Format**

A data frame with 2 variables.

---

shield.procedures      *SHIELD procedures Objects.*

---

**Description**

Full data set provided by MITRE

**Usage**

shield.procedures

**Format**

A data frame with 2 variables.

---

shield.relations      *SHIELD relations Objects.*

---

**Description**

Full data set provided by MITRE

**Usage**

shield.relations

**Format**

A data frame with 3 variables.

---

shield.tactics      *SHIELD tactics Objects.*

---

**Description**

Full data set provided by MITRE

**Usage**

shield.tactics

**Format**

A data frame with 4 variables.

---

shield.techniques      *SHIELD techniques Objects.*

---

**Description**

Full data set provided by MITRE

**Usage**

shield.techniques

**Format**

A data frame with 4 variables.

---

shield.use\_cases      *SHIELD use cases Objects.*

---

**Description**

Full data set provided by MITRE

**Usage**

shield.use\_cases

**Format**

A data frame with 2 variables.

# Index

## \* datasets

- attck.groups, 2
  - attck.mitigations, 3
  - attck.relations, 3
  - attck.software, 3
  - attck.tactics, 4
  - attck.techniques, 4
  - capec.categories, 6
  - capec.patterns, 7
  - capec.relations, 7
  - capec.views, 7
  - car.analytics, 8
  - car.coverage, 8
  - car.implementations, 8
  - car.model, 9
  - car.relations, 9
  - car.sensors, 9
  - cpe.nist, 10
  - cve.nist, 10
  - cwe.categories, 10
  - cwe.views, 11
  - cwe.weaknesses, 11
  - newEdge, 11
  - newNode, 12
  - shield.opportunities, 12
  - shield.procedures, 13
  - shield.relations, 13
  - shield.tactics, 13
  - shield.techniques, 14
  - shield.use\_cases, 14
- 
- attck.groups, 2
  - attck.mitigations, 3
  - attck.relations, 3
  - attck.software, 3
  - attck.tactics, 4
  - attck.techniques, 4
- 
- build\_edges, 4
  - build\_network, 5
  - build\_nodes, 6
- 
- capec.categories, 6
  - capec.patterns, 7
  - capec.relations, 7
  - capec.views, 7
  - car.analytics, 8
  - car.coverage, 8
  - car.implementations, 8
  - car.model, 9
  - car.relations, 9
  - car.sensors, 9
  - cpe.nist, 10
  - cve.nist, 10
  - cwe.categories, 10
  - cwe.views, 11
  - cwe.weaknesses, 11
  - newEdge, 11
  - newNode, 12
  - shield.opportunities, 12
  - shield.procedures, 13
  - shield.relations, 13
  - shield.tactics, 13
  - shield.techniques, 14
  - shield.use\_cases, 14