TSV
Threat Standard Values

josé a. mañas

28.6.2018 (version 7.2)

1 Introduction

TSV files are used to model threats on assets. For each asset class and security dimension, it specifies the standard likelihood, and degradation of value.

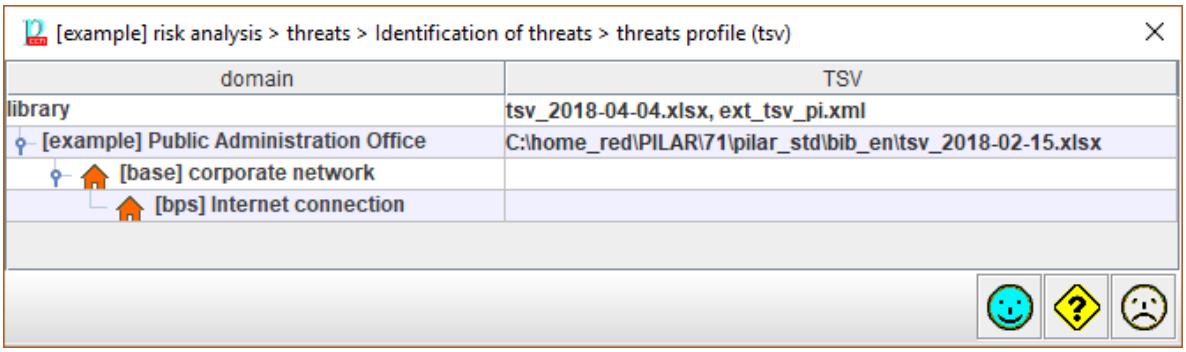
It may be specified as an excel file or as an XML file. Excel is easier to use, but it has less features than XML.

TSV files are routinely specified in CAR files to apply to projects open under a CAR instance, but you can specify a specific TSV for a project, and for security domains.

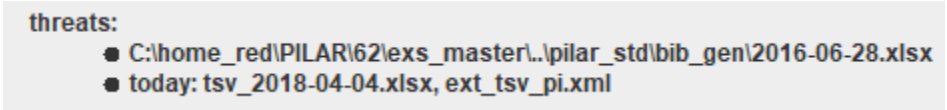
In the CAR file;

```
$ egrep -a -i tsv CIS_en.car
tsv= tsv_2018-04-04.xlsx
tsv= ext_tsv_pi.xml
tsv:EXT_L= tsv_log.xml, tsv_2018-04-04.xlsx
tsv:EXT_P= tsv_pps.xml, tsv_2018-04-04.xlsx
tsv:EXT_T= tsv_tempest.xml, tsv_2018-04-04.xlsx
```

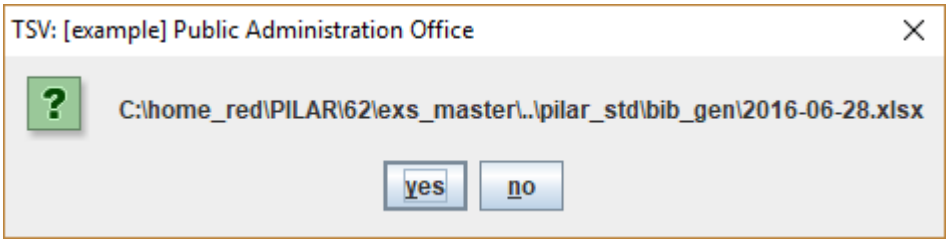
In graphical user interface:



When a project is open, PILAR checks whether the current TSV's in the library differ, and reports it



When there is a different TSV for the project or some security domain, PILAR tries to recover the file, or asks to continue using the specific TSV

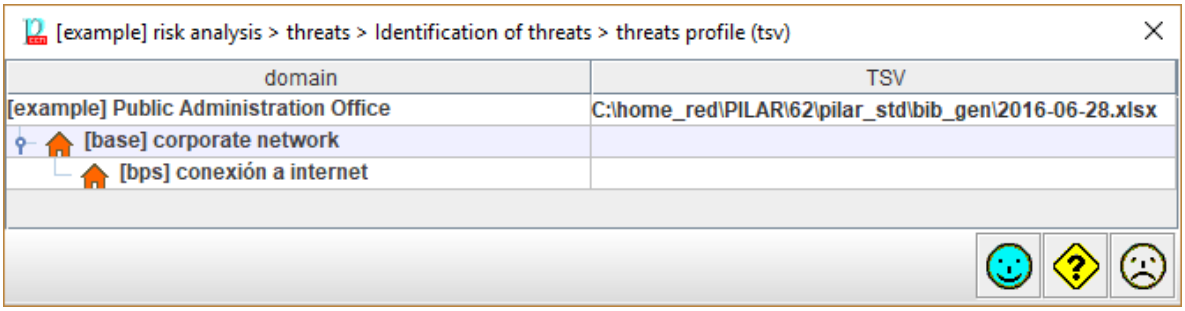


You may say YES to use the specific TSV for the project; or NO, to use library default (as specified in CAR file).

1.1 Previous versions of PILAR

Before version 7.2, PILAR was saving a single information on the TSV used by the project, without making a difference between inheriting it from the context (CAR) o loading an explicit one. As a consequence, after every version update, PILAR was asking for either retaining old context or jumping into new one.

And, in selecting specific TSVs, there was no clear distinction between context and project:



1.2 Specification for a class

TSV specifies information per asset class. When something is specified for a class, it applies to every derivation (children) unless refined. For every class, the most refined data applies.

For example, for HW.* classes, on threat A.25 (threat)

app	family	threat	likely	step	D=en:A	D=en:I	D=en:C
	HW	A.25	0.5	7d	100%		50%
	HW.host	A.25	0.1	15d	100%		100%

	HW.mid	A.25	0.5	7d	100%		100%
	HW.pc	A.25	5	1h	5%		10%
	HW.mobile	A.25	20	30m	1%		10%
	HW.vhost	A.25	0				
	HW.data	A.25	-1	2d	100%		100%
	HW.network	A.25	-1	1d	20%		

Read it:

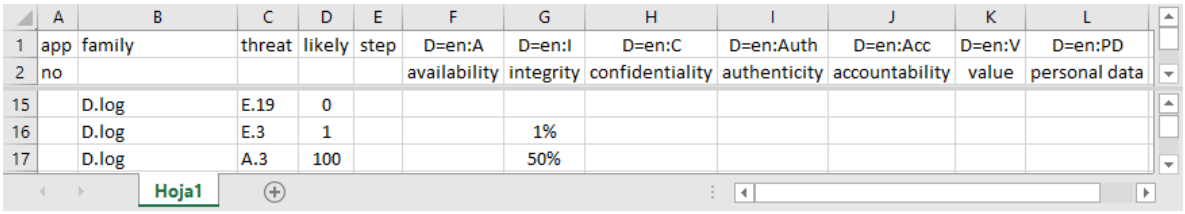
- for HW assets, A.25 is likely to occur .5 times per year with an impact of 100% on availability, and 50% on confidentiality
- for HW.host, likelihood is .1 per year, 100% on availability, and 100% on confidentiality
- for HW.pc, likelihood is 5 per year, 5% on availability, and 10% on confidentiality
- for HW.mobile, likelihood is 20 per year, 1% on availability, and 10% on confidentiality
- for HW.vhost, the threat is not expected to occur
- for HW.data, likelihood is decided based on other classes, while impact is 100% on availability and confidentiality

When several classes apply to an asset, PILAR applies the highest likelihood, and the highest degradation.

1.3 Refinement

Several TSV files may be specified one after another. The ordering makes a difference since TSVs are loaded sequentially, and new ones hide previous. This behavior is convenient in order to use the standard library as default and refine only a few.

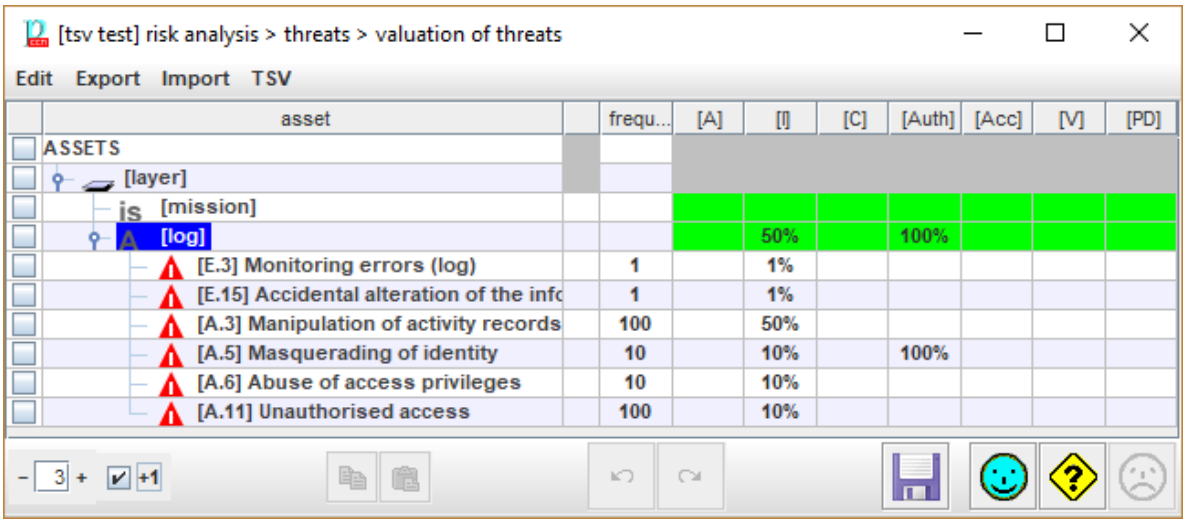
For example, this is the standard library for log assets



	A	B	C	D	E	F	G	H	I	J	K	L
1	app	family	threat	likely	step	D=en:A	D=en:I	D=en:C	D=en:Auth	D=en:Acc	D=en:V	D=en:PD
2	no					availability	integrity	confidentiality	authenticity	accountability	value	personal data
15		D.log	E.19	0								
16		D.log	E.3	1			1%					
17		D.log	A.3	100			50%					

Hoja1

And this is the automatic threats on log assets



Now, lets load a new TSV after the standard one

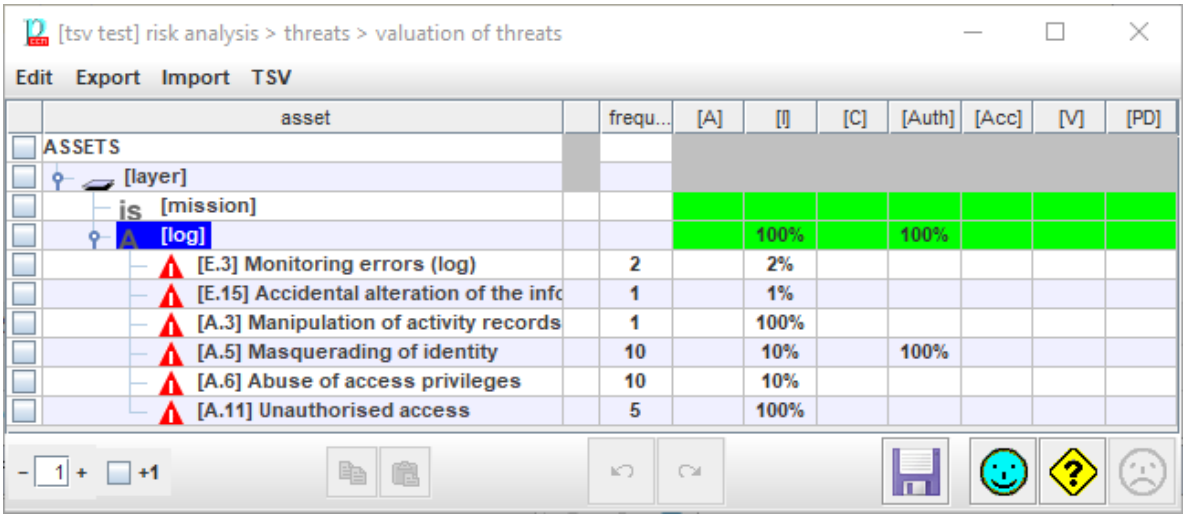
ext_tsv_conf.xml

```
<?xml version="1.0" encoding="utf-8" ?>

<threat-standard-values>

  <family F="D.log">
    <threat Z="E.3" f="2.0">
      <set D="en:I" deg="0.02" />
    </threat>
    <threat Z="A.3" f="1.0">
      <set D="en:I" deg="1.00" />
    </threat>
    <threat Z="A.11" f="5.0">
      <set D="en:I" deg="1.00" />
    </threat>
  </family>

</threat-standard-values>
```

XXX.car

```
tsv= tsv_2018-04-04.xlsx
tsv= ext_tsv_conf.xml
```

Now, the automatic threats look like this



2 Excel format

There is one row per asset class, while columns specify the class, the likelihood, and the degradation. Header row specifies the meaning of each column.

	A	B	C	D	E	F	G	H	I	J
1	app	family	threat	likely	step	D=en:A	D=en:I	D=en:C	D=en:Auth	D=en:A
2	no					availability	integrity	confidentiality	authenticity	accounta
3		D	E.15	1			1%			
4		D	E.18	1	1d	1%				
5		D	E.19	1				10%		
6		D	A.5	10			10%	50%	100%	
7		D	A.6	10	1d	1%	10%	50%		
8		D	A.11	100			10%	50%		
9		D.conf	E.15	0						
10		D.conf	E.19	0						
11		D.conf	E.4	1			1%			
12		D.conf	A.6	0						

2.1 Headers

app

This column is for applicability. Rows labelled as “no” are skipped. You may regard those rows as comments.

family

The asset class to which the row applies.

threat

The threat to which the row applies.

likely

The likelihood expressed an annual rate of occurrence (ARO). 1 stands once per year; 10 for once per month; 100 for every day.

The value may be -1, meaning this row does not specify the likelihood, but it is decided from other rules that apply.

step

The default interruption step for business continuity.

D=…

After "D=" specify a dimension, as language:acronym. For instance, D=en:A will specify values for availability.

The value may be -1, meaning this row does not specify the degradation, but it is decided from other rules that apply.

3 XML format

```
 1  file ::=
 2    <threat-standard-values>
 3      { include | family }0+
 4    </threat-standard-values>
 5
 6  include ::=
 7    <include>
 8      filename
 9    </include>
10
11  family ::=
12    <family F="class(es)" >
13      { threat }0+
14    </threats>
15
16  threat ::=
17    <threat Z="code" f="aro" s="step" >
18      { set }0+
19    </threat>
20
21  set ::=
22    <set D="code" deg="deg" />
```

INCLUDE is used to read another file in place, as a macro replacement.

3.1 Attributes

line	attribute	meaning
12	F	one or more asset classes (comma separated names)
17	Z	one or more threats (comma separated values)
17	f	likelihood expressed as ARO (annual rate of occurrence; 1 for once per year) or special value -1

17	s	interruption step
22	D	security dimension code
22	deg	degradation expressed as a real number between 0.0 and 1.0 or special value -1

3.2 Example

```
- <threat-standard-values>
    - <family F="arch.ip">
        - <threat f="1.0" Z="E.15">
            <set deg="0.1" D="I"/>
          </threat>
        - <threat f="1.0" Z="E.18" s="1d">
            <set deg="0.1" D="D"/>
          </threat>
        - <threat f="1.0" Z="E.19">
            <set deg="0.1" D="C"/>
          </threat>
```

4 Border profile

For border, the specification specifies (1) how to pass through the border, and (2) how to act on internal assets.

First, you must identify the attacker in the external zone. For instance

```
attacker= [EXT_L] External attackers (cyber)
```

Then, specify the capabilities of the attacker

```
tsv:EXT_L= tsv_log.xml, tsv_2018-04-04.xlsx
```

Let's study the first part, devoted to passing through the border

```
 1  <threat-standard-values>
 2    <filter>
 3      <families>
 4          arch.ip SW HW
 5      </families>
 6      <threats>
 7          A.3 A.4 A.8 A.11 A.15 A.18 A.19 A.22
 8      </threats>
 9    </filter>
10
11    <family F="arch.ip">
12      <threat Z="A.8" f="10" />
13      <threat Z="A.11" f="10" />
14    </family>
15
16  </threat-standard-values>
```

line	meaning
2-9	filters the asset classes, and threats that the attacker may exercise after getting into the zone
11-14	specifies the options of the attacker to transit from one zone into another
11	the attacker may use arch.ip assets to move into another zone

12	on the arch.ip asset, it may apply an A.8 threat with a likelihood of 10
13	on the arch.ip asset, it may apply an A.11 threat with a likelihood of 10
4, 7	once the attacker is inside
4	assets this attacker may attack after passing the border
7	threats this attacker may exercise after passing the border

5 CAR file

The CAR file is used to configure PILAR. Among other subjects, it is used to specify default TSV for the projects.

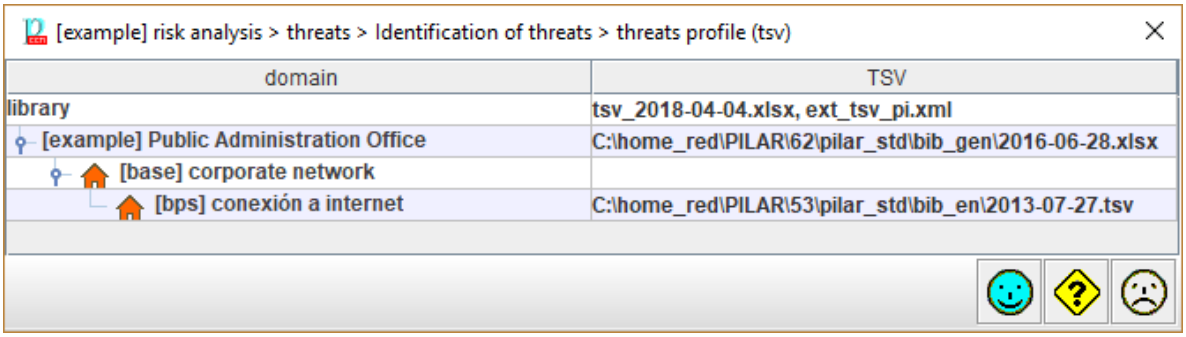
A typical CAR file includes several references to TSV files

```
$ egrep -i tsv STIC_ens.car
tsv= tsv_2018-04-04.xlsx
tsv= ext_tsv_pi.xml
tsv= ext_tsv_pi_aepd.xml
tsv:EXT_L= tsv_log.xml, tsv_2018-04-04.xlsx
tsv:EXT_P= tsv_pps.xml, tsv_2018-04-04.xlsx
tsv:EXT_T= tsv_tempest.xml, tsv_2018-04-04.xlsx
```

The first line loads a base association of threats to asset classes. The second one adds threats for personal data. The third one adds, to the save TSV table, more threats on personal data. Lines 4, 5, and 6 specify threats from external attackers.

After starting PILAR, those TSV are applied by default to projects. Nevertheless, projects may replace them by specific ones, either for the project as a whole, or for specific security domains.

In threat identification and threat valuation screens,



6 Database

The TSV information is saved in two tables

- modelattr, for library and project information
- domainattr, for security domain information

This is a project (modelattr)

- with a default TSV from CAR file
- with a specific TSV for the project

line	project	mykey	val
1	example	lib_tsv_name_2	0:tsv_2018-04-04.xlsx
2	example	lib_tsv_path_2	0:C:\home_red\PILAR\72\pilar_std\bib_en\tsv_2018-04-04.xlsx
3	example	lib_tsv_sign_2	0:92af3fd21e27e2b49bae779934387ce4
4	example	lib_tsv_name_2	1:ext_tsv_pi.xml
5	example	lib_tsv_path_2	1:C:\home_red\PILAR\72\pilar_std\bib_en\ext_tsv_pi.xml
6	example	lib_tsv_sign_2	1:beb3e81eaa3fab442cc45a61d132f42e
7	example	tsv_name_2	0:2016-06-28.xlsx
8	example	tsv_path_2	0:C:\home_red\PILAR\62\pilar_std\bib_gen\2016-06-28.xlsx
9	example	tsv_sign_2	0:218e8845e1302adb518c591acf8fcf34

And a specific setting for a domain (domainattr)

line	project	domain	attr	value
1	example	bps	tsv_name_2	0:2013-07-27.tsv
2	example	bps	tsv_path_2	0:C:\home_red\PILAR\53\pilar_std\bib_en\2013-07-27.tsv
3	example	bps	tsv_sign_2	0:02516d68dda7da629c86dd145f67b4db

The information appears in three pieces

- tsv_name: for the name of the file
- tsv_path: for the absolute path of the file
- tsv_sign: a hash signature of the file (to detect changes)

These pieces appear in chunks (1-3, 4-6, 7-9) where the position of the piece (load ordering into TSV table) is the value, an integer before a colon.

position : value

PILAR reads the values in groups of three, sorts by position, and loads into specific tables.