

PILAR
Batch Mode
23.6.2014

Table of contents

1	Introduction	3
2	Set up	4
2.1	Header	4
2.2	<car>: Configuration file.....	4
2.3	<lic>: License	5
2.4	<log>: Activity log.....	5
2.5	<bgr> or <pl5>: Library.....	5
2.6	<ext>: Library extensions.....	5
2.7	<ext_db>: Library extensions.....	5
2.8	<evl>: Security profiles.....	6
2.9	Risk model.....	6
2.10	<tsv>	7
3	Execution	8
3.1	Valuation of domains	8
3.2	Valuation of assets	8
3.3	Valuation of threats.....	9
3.4	Valuation of safeguards.....	9
3.5	Set threat values	11
3.6	Apply tsv (threat standard values)	11
3.7	Search for vulnerabilities	12
3.8	Update vulnerabilities	12
3.9	Copy from another model	12
3.10	Remove phases	13
3.11	Output: save the model.....	13
4	Reports (RTF)	14
5	XML Reporting	15
5.1	Assets	15
5.2	Threats.....	16
5.3	Valuation of assets	17
5.4	Valuation of threats.....	18

5.5	Valuation of safeguards.....	19
5.6	Accumulated impact and risk	20
5.7	Deflected impact and risk.....	22
5.8	Security profile (.evl)	24
5.9	Delta reports	25
6	Database reporting.....	26
6.1	Accumulated impact and risk	26
6.2	Deflected impact and risk.....	26
6.3	Security profile (.evl)	27
7	Changes.....	28

1 Introduction

PILAR may be run in batch mode; that is, without graphical interface. This mode is useful for:

- unattended evaluation of risks (e.g. over night)
- reactive risk analysis (e.g. upon reporting of vulnerabilities)
- [re-]generation of reports

PILAR reads

1. a valid license (a ".lic" file)
2. a configuration file (a ".car" file)
3. a library (a ".bgr" file or a ".pl5" file)
4. [optionally] one or more library extensions (".kb" and ".lle" files)
5. [optionally] one or more security profiles (.evl files)
6. a system model (either from a ".mgr" file or from tables in a database)

Then you may apply changes to the model:

- load and apply one or more threat profile(s) (".tsv" files)
- import and set some values for assets, threats, ...
- generate reports in RTF, XML, or in database tables.

The working scenario is stored in plan files (".plan" files).

See an example:

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<pilar_batch
  working_dir="C:\Users\jam\pilar"
  bcm="false"
  quantitative="false"
>
  <lic>/Program Files/PILAR_5.3/lics/L30000.lic</lic>
  <car>/Program Files/PILAR_5.3/STIC_en.car</car>
  <log>batch.log</log>
  <p15>std_en_53.pl5</p15>
  <ext>ext_classes.lle</ext>

  <mgr>dmz.mgr</mgr>

  <report what="risk_down">
    risk_down.xml
  </report>
  <report what="risk_up" phase="target">
    risk_up_target.xml
  </report>
</pilar_batch>
```

2 Set up

This section covers the preliminary steps:

- .plan file header attributed

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<pilar_batch attributes>
```
- <car> to set configuration file
- <lic> to set license file
- <log> to save an activity report
- <bgr> or <pl5> to select a library
- <ext> to set library extensions from files
- <ext_db> to set library extensions from a database
- <tsv> to set threat values
see <apply-tsv>
- <mgr> or <db> to load a risk model

2.1 Header

The header tag is

```
pilar_batch
```

The following attributes are available:

```
working_dir="..."
```

mandatory

working directory

```
lib_dir="..."
```

optional (default read from .car file)

library directory, to load .bgr, .lle, and .kb

```
bcm="true | false"
```

optional (default: false)

if TRUE, pilar runs in “continuity of operations mode”;
else in standard risk analysis mode

```
quantitative="true | false"
```

optional (default: false)

if TRUE, pilar uses a quantitative mode;
else, a qualitative mode

2.2 <car>: Configuration file

Mandatory. Only one.

Specifies the configuration file (the .car file) to drive the risk analysis.

Format:

```
<car> absolute path name </car>
```

2.3 <lic>: License

Mandatory. Only one.

Provides the license (the .lic file).

Format

```
<lic> absolute path name </lic>
```

2.4 <log>: Activity log

Optional. Default is system console.

A file to record the batch activity.

Format:

```
<log> path relative to working_dir </log>
```

2.5 <bgr> or <pl5>: Library

~~Mandatory~~. Optional: by default uses .CAR.

Specifies the library (the .bgr file).

Format:

```
<bgr> path relative to lib_dir </bgr>
```

Since PILAR version 5, the BGR files have the extension PL5. The following tag is as synonymous:

```
<p15> path relative to lib_dir </p15>
```

2.6 <ext>: Library extensions

By default, PILAR loads according to .CAR.

Optional: 0 or more.

Loads library extensions (.lle files) and specific protections (.kb files).

Format:

```
<ext> path relative to lib_dir </ext>
```

2.7 <ext_db>: Library extensions

Optional: 0 or more. You need a license that enables database access.

Loads library extensions from a database.

Format:

```
<ext_db user="..." password="..."> jdbc url </ext_db>
```

In order to read from and write to a database, PILAR uses a JDBC connector that must be provided:

```
<jar> absolute path to .jar connector </jar>
<class> Driver class in jar </class>
```

Example:

```
<jar> /java/mysql-connector-java-5.1.8-bin.jar </jar>
<class> com.mysql.jdbc.Driver </class>
```

2.8 <evl>: Security profiles

By default, PILAR loads according to .CAR.

Optional: 0 or more.

Loads the security profile (.evl files).

Format:

```
<evl> path relative to lib_dir </ext>
```

2.9 Risk model

It is mandatory to load a model.

Formats:

```
<mgr> path relative to working_dir </mgr>
```

If needed, add a password:

```
<mgr password="..."> path relative to working_dir </mgr>
```

For databases:

```
<db user="..." password="..."> jdbc url </db>
```

Examples:

```
<mgr>risk-model_2000-12-28.mgr</mgr>
```

```
<db user="john" password="password">
  jdbc:mysql://localhost/risk_model
</db>
```

In order to read from and write to a database, PILAR uses a JDBC connector that must be provided:

```
<jar> absolute path to .jar connector </jar>
<class> Driver class in jar </class>
```

Example:

```
<jar> /java/mysql-connector-java-5.1.8-bin.jar </jar>
<class> com.mysql.jdbc.Driver </class>
```

2.10 <tsv>

Optional: 0 or more.

This tag is partly obsolete. Currently, PILAR reads it as part of the scripting part; that is, the TSV file is applied when found. By default, PILAR uses the threat valuation in the working risk model. When this tag is found, the TSV specification is applied on top of the previously existing one. If there is more than one TSV, each one is applied in order.

See <apply-tsv>, a synonymous.

3 Execution

Once the working framework has been established with the header commands, you may specify a number of actions to execute. These are executed in the order they appear in the plan file. You may refer to it as “scripting”.

3.1 Valuation of domains

Deprecated. Use essential assets to valuate.

Format:

```
<import> filename in working_dir </import>
```

3.2 Valuation of assets

Format:

```
<import> filename in working_dir </import>
```

Pilar recognizes the following formats:

standard risk	continuity
<pre>file ::= <pilar_asset_values> { value }0+ </pilar_asset_values> value ::= <set asset="code" dimension="code" [vl="level"] [vn="number"] > { why }0+ [comment] </set> why ::= <why> criterion </why> comment ::= <comment> comment </comment></pre>	<pre>file ::= <pilar_bcm_asset_values> { value }0+ </pilar_bcm_asset_values> value ::= <set asset="code" step="seconds" [vl="level"] [vn="number"] > { why }0+ [comment] </set> why ::= <why> criterion </why> comment ::= <comment> comment </comment></pre>

See examples in [“valuation of domains”](#).

3.3 Valuation of threats

Format:

```
<import> filename in working_dir </import>
```

Pilar recognizes the following formats:

standard risk	continuity
<pre>file ::= <pilar_threats> { value }0+ </pilar_threats> value ::= <set asset="code" threat="code" frequency="value" > { degradation }0+ </set> degradation ::= <degradation dimension="code" degradation="percent" > </degradation></pre>	<pre>file ::= <pilar_bcm_threats> { value }0+ </pilar_bcm_threats> value ::= <set asset="code" threat="code" frequency="value" [step="seconds"] > </set></pre>

Frequency:

e.g. "1.5" (floating point, English notation)

Dimension code:

e.g. "C" for confidentiality

Please, note that code is language dependent.

Degradation:

An integer between 0 and 100.

Step:

examples			
1 hour	"3600"	1,5 hours	"5400"
	"3600s"		"5400s"
	"60m"		"90m"
	"1h"		"1h30m"

3.4 Valuation of safeguards

Format:

```
<import> filename in working_dir </import>
```

Pilar recognizes the following format:

safeguards
<pre> file ::= <pilar_safeguards> { per_domain per_asset }0+ </pillar_safeguards> per_domain ::= <domain code="code" > { per_phase }0+ </domain> per_asset ::= <asset code="code" > { per_phase }0+ </asset> per_phase ::= <phase code="code" > { value }0+ </phase> value ::= <safeguard code="code" value="maturity" > [comment] </safeguard> </pre>

Maturity:

examples	
blank	“”
...?	“?”
n.a.	“na”
L0	“L0”
L1	“L1”
L2	“L2”
L3	“L3”
L4	“L4”
L5	“L5”

3.5 Set threat values

Format:

```
<set values />
```

Pilar recognizes the following formats:

<pre><set A="asset code" Z="threat code" freq="frequency value" /></pre>	<pre><set A="asset code" Z="threat code" deg="degradation value" D="dimension code" /></pre>
<pre><set A="asset code" Z="threat code" freq="frequency value" deg="degradation value" D="dimension code" /></pre>	
<pre><set A="asset code" Z="threat code" step="seconds" /></pre>	<pre><set A="asset code" Z="threat code" freq="frequency value" step="seconds" /></pre>

Setting the frequency and/or the degradation of a given threat on a given asset.

Or the interruption step.

Frequency:

e.g. "1.5" (floating point, English notation)

Dimension code:

e.g. "C" for confidentiality

Please, note that code is language dependent.

Degradation:

An integer between 0 and 100.

Step:

examples			
1 hour	"3600"	1,5 hours	"5400"
	"3600s"		"5400s"
	"60m"		"90m"
	"1h"		"1h30m"

3.6 Apply tsv (threat standard values)

Format:

```
<tsv> filename in working_dir </tsv>
```

or

```
<apply-tsv> filename in working_dir </apply-tsv>
```

Loads the file, and applies the matching values. The content of the file is a standard TSV extension file.

3.7 Search for vulnerabilities

Format:

```
<search-vulnerabilities>
  filename in working_dir
</search-vulnerabilities>
```

See documentation on how PILAR searches for vulnerabilities using and NVD formats, and matching according to CPE names.

Matching vulnerabilities are assigned to assets.

3.8 Update vulnerabilities

Format:

```
<update-vulnerabilities>
  filename in working_dir
</update-vulnerabilities>
```

See documentation on how PILAR specifies CVSS refinements, and applies to vulnerabilities identified by the CVE-ID.

Matching vulnerabilities are updated.

3.9 Copy from another model

You may import the valuation of safeguards and security profiles from another model.

```
copy ::=
  <copy
    from="model.mgr"
  >
    [ safeguards ]
    [ evl ]
    [ plane ]
    [ phase ]
  </copy>

safeguards ::=
  <safeguards />

evl ::=
  <evl
    code="code"
  />

plane ::=
  <plane
    code="code"
```

```

    [ target="code" ]
  />

phase ::=
  <phase
    code="code"
    [target="code" ]
  </phase>

```

Safeguards are copied, if the tag “safeguards” is present.

Evaluation profiles are copied, if mentioned. Only those mentioned.

Only values in the mentioned planes and phases are copied. If a target plane is specified, PILAR copies values from the plane in the from-model into the target plane in the working model. If a target phase is specified, PILAR copies from the phase in the from-model into the target phase in the working model. If the target plane or the target phase does not exist, they are created and added to the end of the current planes and phases, respectively.

3.10 Remove phases

You may remove one or more phases:

```

remove ::=
  <remove>
    { phase }0+
  </remove>

phase ::=
  <phase
    code="code"
  />

```

The phases are removed, along with the associated valuations.

3.11 Output: save the model

To save the working model with the modifications applied while scripting:

```
<output file="mgr file" [ password="..." ] />
```

The file is relative to the working directory. If no file is specified, the model is saved as “output.mgr”.

In order to save on a database, use the following format:

```
<output db="url" user="..." password="..." />
```

as in the following example

```

<output db="jdbc:mysql://localhost/risk_model"
  user="john"
  password= "password"
/>

```

4 Reports (RTF)

You may generate a report using a template. See specification of templates.

```
<report template="file" output="file" />
```

or

```
<report template="file">output_file</report>
```

where the template and the output files are relative to the working directory.

The template may need further information requested by means of the tags. The value to these requests may be provided as attributes of the report tag:

template request	<report> attribute
<ask.domains />	ask.domains the value is a comma-separated list of domain codes
<ask.domain name="X" />	ask.domain.X the value is the code of a domain
<ask.phases />	ask.phases the value is a comma-separated list of phase codes, and NULL for the potential risk
<ask.phase name="X" />	ask.phase.X the value is the code of a phase, and NULL for the potential risk

Example

```
<report template="C:\Users\jam\pilar\report_tpl.rtf"
  ask.domains="base"
  ask.phases="null, current, target"
  ask.phase.target="target"
  ask.domain.ESE="base"
  ask.domain.LSE="base"
  ask.domain.GSE="base"
>
  report_out.rtf
</report>
```

5 XML Reporting

The XML syntax is presented using a variant of BNF notation, namely:

notation	meaning
$x \mid y$	choice “x” or “y”
$[x]$	zero or one occurrence of “x”; that is, “x” is optional
$\{ x \}0+$	zero or more occurrences of “x”
$\{ x \}1+$	one or more occurrences of “x”

5.1 Assets

Format:

```
<report what="assets">
  filename in working_dir
</report>
```

assets
<pre>file ::= <assets> { asset }0+ </assets> asset ::= <asset code="..."> name </asset></pre>

5.2 Threats

Format:

```
<report what="threats">
  filename in working_dir
</report>
```

threats

```
file ::=
  <threats>
    { threat }0+
  </threats>

threat ::=
  <threat code="...">
    name
  </threat>
```

5.3 Valuation of assets

Format:

```
<report what="valuation_assets">
  filename in working_dir
</report>
```

standard risk	continuity
<pre>file ::= <pilar_asset_values> { value }0+ </pilar_asset_values> value ::= <set asset="code" dimension="code" [vl="level"] [vn="value"] > { why }0+ [comment] </set> why ::= <why> criterion </why> comment ::= <comment> comment </comment></pre>	<pre>file ::= <pilar_bcm_asset_values> { value }0+ </pilar_bcm_asset_values> value ::= <set asset="code" step="seconds" [vl="level"] [vn="value"] > { why }0+ [comment] </set> why ::= <why> criterion </why> comment ::= <comment> comment </comment></pre>

5.4 Valuation of threats

Format:

```
<report what="valuation_threats">
  filename in working_dir
</report>
```

standard risk	continuity
<pre>file ::= <pilar_threats> { value }0+ </pilar_threats> value ::= <set asset="code" threat="code" frequency="value" > { degradation }0+ </set> degradation ::= <degradation dimension="code" degradation="percent" </degradation></pre>	<pre>file ::= <pilar_bcm_threats> { value }0+ </pilar_bcm_threats> value ::= <set asset="code" threat="code" frequency="value" step="seconds" > </set></pre>

5.5 Valuation of safeguards

Format:

```
<report
  what="valuation_safeguards"
  [ domain="comma-separated list of domain codes" ] >
  [ phase="comma-separated list of phase codes" ] >
  filename in working_dir
</report>
```

If no "domain" is specified, all the domains are used.

If no "phase" is specified, all the phases are used.

safeguards

```
file ::=
  <pilar_safeguards>
    { per_domain | per_asset }0+
  </pilar_safeguards>

per_domain ::=
  <domain
    code="code"
  >
  { per_phase }1+
  </domain>

per_asset ::=
  <asset
    code="code"
  >
  { per_phase }1+
  </asset>

per_phase ::=
  <phase
    code="code"
  >
  { value }1+
  </phase>

value ::=
  <safeguard
    code="code"
    value="maturity"
  >
  [ comment ]
  </safeguard>
```

maturity

"L0" | "L1" | "L2" | "L3" | "L4" | "L5" | "" | "na" | "?"

5.6 Accumulated impact and risk

Format:

```
<report what="risk_down"
  [ domain="comma-separated list of domain codes" ] >
  [ phase="comma-separated list of phase codes" ] >
  filename in working_dir
</report>
```

If no “domain” is specified, all the domains are used.

If no “phase” is specified, all the phases are used. The potential values are reported under phase “null”.

standard risk	continuity
<pre>file ::= <risks_down> { per_phase }0+ </risks_down> per_phase ::= <phase code="code"> { item }0+ </phase> item ::= <item asset="code" [dcode="code"] dimension="code" value="a_value" accumulated="a_value" threat="code" degradation="percent" frequency="value" impact="a_value" risk="b_value" ></pre>	<pre>file ::= <risks_down bcm="true"> { per_phase }0+ </risks_down> per_phase ::= <phase code="code"> { item }0+ </phase> item ::= <item asset="code" threat="code" frequency="value" step="seconds" impact="a_value" risk="b_value" ></pre>

dcode

Optional: the internal code used by Pilar to identify the dimension in any language. Please, note that the ‘code’ in attribute ‘dimension’ depends on the language.

percent

An integer between 0 and 100.

a_value

The value of the assets, and of the impact may be numerical (in quantitative analysis) or a value level (in qualitative analysis; e.g. “[7]”).

b_value

The risk value may be numerical (in quantitative analysis) or a criticality level (in qualitative analysis; e.g. "{4.8}").

5.7 Deflected impact and risk

Format:

```
<report what="risk_up"
  [ domain="comma-separated list of domain codes" ] >
  [ phase="comma-separated list of phase codes" ] >
  filename in working_dir
</report>
```

If no “domain” is specified, all the domains are used.

If no “phase” is specified, all the phases are used. The potential values are reported under phase “null”.

standard risk	continuity
<pre>file ::= <risks_up> { per_phase }0+ </risks_up> per_phase ::= <phase code="code"> { item }0+ </phase> item ::= <item above="code" below="code" [dcode="code"] dimension="code" value="a_value" threat="code" degradation="percent" frequency="value" impact="a_value" risk="b_value" ></pre>	<pre>file ::= <risks_up bcm="true"> { per_phase }0+ </risks_up> per_phase ::= <phase code="code"> { item }0+ </phase> item ::= <item above="code" below="code" threat="code" frequency="value" step="seconds" impact="a_value" risk="b_value" ></pre>

dcode

Optional: the internal code used by Pilar to identify the dimension in any language. Please, note that the ‘code’ in attribute ‘dimension’ depends on the language.

percent

An integer between 0 and 100.

a_value

The value of the assets, and of the impact may be numerical (in quantitative analysis) or a value level (in qualitative analysis; e.g. “[7]”).

b_value

The risk value may be numerical (in quantitative analysis) or a criticality level (in qualitative analysis; e.g. "{4.8}").

5.8 Security profile (.evl)

Format:

```
<report evl="evl filename in working_dir"
  [ domain="comma-separated list of domain codes" ] >
  [ phase="comma-separated list of phase codes" ] >
  filename in working_dir
</report>
```

If no “domain” is specified, all the domains are used.

If no “phase” is specified, all the phases are used.

security profile
<pre>file ::= <controls code="code" > { per_domain }0+ </controls> per_domain ::= <domain code="code" > { per_phase }1+ </domain> per_phase ::= <phase code="code"> { value }0+ </phase> value ::= <control code="code" value="percent" > [comment] </control></pre>

percent

An integer between 0 and 100.

5.9 Delta reports

You may evaluate risks at some point, change some values, recalculate risks, and report only on differences. So you can analyse the consequences of some change of valuation of assets, threats or safeguards.

Pilar takes a snapshot of risks

```
<mark label="name" />
```

and when a report is produced, it may be instructed to compare with a previous mark

```
<report diff="name" what="risk_down">  
  risk_down.xml  
</report>
```

```
<report diff="name" what="risk_up">  
  risk_up.xml  
</report>
```

The format is the standard one, but only changes are reported, showing both old and new values. For instance

```
<item asset="LAN" dcode="D" dimension="A"  
  value=" " accumulated="[5]"  
  threat="E.9" degradation="100"  
  frequency="10.0"  
  old_impact=" " impact="[5]"  
  old_risk="{1.2}" risk="{4.8}"  
>
```

“old_impact” is shown when the impact changes. “old_risk” is shown when risk changes. The “item” is shown when either impact or risk change.

There may be several marks; PILAR stores the risks at the labelled mark, and later on compares the current situation with the situation when the mark was established.

6 Database reporting

Results may be exported to a database. In order to write into a database, the script must specify a JDBC connector and a driver class (see section 2.9)

Database tables are reported in a separate document

DB_structures

The following tables may be generated from a batch plan:

Risk analysis: impact and risk

```
riskdown1
riskdown2
riskup1
riskup2
```

Business continuity: impact and risk

```
bcmdown1
bcmdown2
bcmup1
bcmup2
```

Security profiles

```
EVL_apps
EVL.value
```

6.1 Accumulated impact and risk

Format:

```
<report what="risk_down"
  format="sql" user="..." password="..."
  [ domain="comma-separated list of domain codes" ] >
  [ phase="comma-separated list of phase codes" ] >
  jdbc url
</report>
```

If no “domain” is specified, all the domains are used.

If no “phase” is specified, all the phases are used. The potential values are reported under phase “null”.

6.2 Deflected impact and risk

Format:

```
<report what="risk_up"
  format="sql" user="..." password="..."
  [ domain="comma-separated list of domain codes" ] >
  [ phase="comma-separated list of phase codes" ] >
  jdbc url
</report>
```

If no “domain” is specified, all the domains are used.

If no “phase” is specified, all the phases are used. The potential values are reported under phase “null”.

6.3 Security profile (.evl)

Format:

```
<report evl="code"  
  format="sql" user="..." password="..."  
  [ domain="comma-separated list of domain codes" ] >  
  [ phase="comma-separated list of phase codes" ] >  
  jdbc url  
</report>
```

If no “domain” is specified, all the domains are used.

If no “phase” is specified, all the phases are used.

7 Changes

3.9.2009

- internal codes: dcode for safeguards

16.11.2009

- add security profiles
- add reporting on database
- change “c” attributes into “code” attributes
 - risks_down: phase code
 - risks_up: phase code
- revise directories for relative file names
- degradation is always an integer between 0 and 100

7.2.2010

- incremental risk: delta reports

7.5.2010

- vulnerabilities

18.7.2011

- revise application of TSV files
- copy from external model
- remove phases

5.3.2012

- load library extensions from a database

13.1.2014

- report generation based on template