

PILAR
Batch Mode
7.5.2010

Changes

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

3.9.2009

- internal codes: dcode for safeguards

7.2.2010

- incremental risk: delta reports

7.5.2010

- vulnerabilities

Table of contents

| | | |
|------|--|----|
| 1 | Introduction..... | 4 |
| 2 | Set up | 4 |
| 2.1 | Header | 5 |
| 2.2 | <car> | 5 |
| 2.3 | <lic> | 5 |
| 2.4 | <log> | 6 |
| 2.5 | <bgr>..... | 6 |
| 2.6 | <ext> | 6 |
| 2.7 | <tsv> | 6 |
| 2.8 | Risk model | 6 |
| 3 | Execution | 8 |
| 3.1 | Valuation of domains | 8 |
| 3.2 | Valuation of assets | 9 |
| 3.3 | Valuation of threats..... | 10 |
| 3.4 | Valuation of safeguards | 10 |
| 3.5 | Set threat values | 12 |
| 3.6 | Apply tsv (threat standard values) | 13 |
| 3.7 | Search for vulnerabilities | 13 |
| 3.8 | Update vulnerabilities | 13 |
| 4 | XML Reporting..... | 14 |
| 4.1 | Assets | 14 |
| 4.2 | Threats..... | 15 |
| 4.3 | Valuation of domains | 16 |
| 4.4 | Valuation of assets | 17 |
| 4.5 | Valuation of threats..... | 18 |
| 4.6 | Valuation of safeguards | 19 |
| 4.7 | Accumulated impact and risk..... | 20 |
| 4.8 | Deflected impact and risk | 22 |
| 4.9 | Security profile (.evl)..... | 24 |
| 4.10 | Delta reports..... | 25 |
| 5 | Database reporting | 26 |
| 5.1 | Accumulated impact and risk..... | 26 |
| 5.2 | Deflected impact and risk | 26 |
| 5.3 | Security profile (.evl)..... | 27 |

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)

PILAR reads

1. a valid license (a ".lic" file)
2. a configuration file (a ".car" file)
3. a library (a ".bgr" file)
4. [optionally] one or more library extensions (".kb" and ".lle" files)
5. 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 XML format, 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"
  lib_dir="/Program Files/PILAR_4.1/bib_en"
  bcm="false"
  quantitative="false"
>
  <lic>/Program Files/PILAR_4.1/lics/L30000.lic</lic>
  <car>/Program Files/PILAR_4.1/STIC_en.car</car>
  <log>daily.log</log>
  <bgr>std_2007-08-27_en.bgr</bgr>
  <ext>threats.lle</ext>
  <mgr>dmz.mgr</mgr>
  <tsv>threats.tsv</tsv>

  <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> to select a library
- <ext> to set library extensions
- <tsv> to set threat values
- <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="..."

mandatory

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>

Mandatory.

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

Format:

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

2.3 <lic>

Mandatory.

Provides the license (the .lic file).

Format

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

2.4 <log>

Optional. Default is system console.

A file to record the batch activity.

Format:

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

2.5 <bgr>

Mandatory.

Specifies the library (the .bgr file).

Format:

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

2.6 <ext>

Optional: 0 or more.

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

Format:

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

2.7 <tsv>

Optional: 0 or more.

Loads threat standard values (.tsv files).

Format:

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

2.8 Risk model

It is mandatory to load a model.

Formats:

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

```
<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>
```

3 Execution

3.1 Valuation of domains

Format:

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

Pilar recognizes the following formats:

| standard risk | continuity |
|---|---|
| <pre>file ::= <pilar_domain_values> { value }0+ </pilar_domain_values> value ::= <set domain="code" dimension="code" [vl="level"] [vn="number"] > { why }0+ [comment] </set> why ::= <why> criterion </why> comment ::= <comment> comment </comment></pre> | <pre>file ::= <pilar_bcm_domain_values> { value }0+ </pilar_bcm_domain_values> value ::= <set domain="code" step="seconds" [vl="level"] [vn="number"] > { why }0+ [comment] </set> why ::= <why> criterion </why> comment ::= <comment> comment </comment></pre> |

Dimension code:

e.g. "C" for confidentiality

Please, note that code is language dependent.

Value level:

e.g. "8" for level 8

Value number:

e.g. "2400" for 2,400

Step:

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

Criterion

e.g. `<why>7.si</why>`

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

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

Maturity:

| examples | |
|----------|------|
| blank | "" |
| ...? | "?" |
| n.a. | "na" |
| L0 | "L0" |
| L1 | "L1" |
| L2 | "L2" |
| L3 | "L3" |
| L4 | "L4" |
| L5 | "L5" |

3.5 Set threat values

Format:

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

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" /> <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" /> <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” “3600s” “60m” “1h” | 1,5 hours | “5400” “5400s” “90m” “1h30m” |

3.6 Apply tsv (threat standard values)

Format:

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

Loads the file, and applies the matching values. The contents 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.

4 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” |

4.1 Assets

Format:

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

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

4.2 Threats

Format:

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

threats

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

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

4.3 Valuation of domains

Format:

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

| standard risk | continuity |
|--|--|
| <pre>file ::= <pilar_domain_values> { value }0+ </pilar_domain_values> value ::= <set domain="code" dimension="code" [vl="level"] [vn="value"] > { why }0+ [comment] </set> why ::= <why> criterion </why> comment ::= <comment> comment </comment></pre> | <pre>file ::= <pilar_bcm_domain_values> { value }0+ </pilar_bcm_domain_values> value ::= <set domain="code" step="seconds" [vl="level"] [vn="value"] > { why }0+ [comment] </set> why ::= <why> criterion </why> comment ::= <comment> comment </comment></pre> |

4.4 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> |

4.5 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> |

4.6 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" | "?"

4.7 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}").

4.8 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}").

4.9 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

```
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>
```

percent

An integer between 0 and 100.

4.10 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.

5 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 2.8)

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
```

5.1 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”.

5.2 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”.

5.3 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.