# Writing a test report

## Tools

First of all, make sure you have the right tools installed. Check the tools manual for more info.

## Main structure

The report's main element is <pentest\_report>. It contains four major parts:

* Document information (metadata), in the element <meta>
* The index, in the element <generate\_index>
* A variable number of sections (main content), in several <section> elements
* A variable number of appendices (extra content), in one or more <appendix> elements

Additionally, the <pentest\_report> element has two attributes:

* findingCode, which is a three-letter prefix for the finding numbers, derivated from the client name (e.g. 'SID' for Sitting Duck BV, 'BIC' for Big International Company Ltd, etc.)
* findingNumberingBase, which can be set to 'Report' or 'Section' - this configures whether the numbering of findings in the report is report-based (i.e. starting with XXX-001 and continuing upwards) or section-based (i.e. findings in section 3 are numbered XXX-301 and up, findings in section 5 are numbered XXX-501 and up). Use 'Report' for smaller pentest reports and 'Section' for large ones.

## Document information / metadata

This is the part where we put all information that is *about the report* rather than about the pentest itself (hence the term metadata): who has been working on it, what is the document title, what versions has it gone through, etc.

In XML, this part is indicated by the <meta> element. It contains the following elements (mandatory and in the listed order):

* The document title, in the <title> element
* Client information, in the <client> element
* Targets listing, in the <targets> element
* (Optionally) Pentest-related information, in the <pentestinfo> element
* People who worked on the pentest and/or report, in the <collaborators> element
* Document classification, in the <classification> element
* The document's version history, in the <version\_history> element
* Your company contact information, in the <company> element

For more details, see the sections below.

### The document title

In the <title> element, put the document title (in text). This is something like 'Penetration Test Report', 'Security Audit Report', whatever fits the bill.

Example: <title>Penetration Test Report</title>

### Client information

The <client> element contains two other elements:

* <full\_name>, in which you should type the client's official name, e.g. 'Sitting Duck BV', or 'Big International Company Ltd'
* <short\_name>, in which you should type the client's shorter name, e.g. 'Sitting Duck' or 'Big International' (or, if there is no shorter name, just type the long name again)

Example:

<client>  
 <full\_name>Sitting Duck B.V.</full\_name>  
 <short\_name>Sitting Duck</short\_name>  
</client>

### Targets

The <targets> element contains one or more <target> elements, one for each target specified for the pentest. Put every target of the pentest in its own <target> element. If there is only one target, you'll end up with a <targets> element containing only one <target> element. This is ok.

Example:

<targets>  
 <target>fishinabarrel.sittingduck.com</target>  
 <target>hackthis.sittingduck.com</target>  
 <target>Sitting Duck's support staff</target>  
<targets>

### Pentest Info

The <pentestinfo> element contains some data about the pentest itself. This element is optional, but may be useful as you can refer to its content using placeholders, allowing e.g. for standard referrals to the tested application name, pentest type or pentest duration.

Example:

<pentestinfo>  
 <duration>10</duration><!-- duration of pentest, in working days -->  
 <test\_planning>January 1st until January 12th, 2015</test\_planning> <!-- date or date range in text, e.g. May 18th until May 25th, 2015 -->  
 <report\_writing>January 15th until January 20th, 2015</report\_writing> <!-- date or date range in text, e.g. May 18th until May 25th, 2015 -->  
 <report\_due>January 23rd, 2015</report\_due> <!-- date or date range in text, e.g. May 18th until May 25th, 2015 -->  
 <nature>time-boxed</nature>  
 <type>black-box</type><!-- please choose one of the following: black-box, grey-box, crystal-box -->  
 <target\_application>FishInABarrel</target\_application><!-- name of application to be tested (if any) -->  
 <target\_application\_producer>H4ckers 'R' Us</target\_application\_producer>  
 </pentestinfo>

### Collaborators

The <collaborators> element contains three other elements, mandatory and in the listed order:

* <reviewers>, containing one or more <reviewer> elements (same system as <targets>; put the name of each reviewer in its own <reviewer> element.)
* <approver>, containing only text. Here you put the name of the person who has approved the document for distribution to the client (usually this is Melanie)
* <pentesters>, containing one or more <pentester> elements (*not* the same system as targets, see the section on pentesters for more details)

Example:

<collaborators>  
 <reviewers>  
 <reviewer>Patricia Piolon</reviewer>  
 </reviewers>  
 <approver>Melanie Rieback</approver>  
 ...  
</collaborators>

### Pentesters

As said, the <pentesters> element contains one or more <pentester> elements.

The <pentester> element contains two other elements:

* <name>, containing the pentester's name (in text)
* <bio>, containing a paragraph about the pentester's l33tness :) - For many pentesters, you can get this bio from a previous pentest. If we're working with a new guy or girl, ask them for some info about themselves.

The names of the pentesters will appear on the document info page and their names and bios will be listed automatically in a table wherever you insert the <generate\_testteam/> element.

Example:

<collaborators>  
 ...  
 <pentesters>  
 <pentester>  
 <name>Melanie Rieback</name>  
 <bio>Melanie Rieback is a former Asst. Prof. of Computer Science   
 from the VU, who is also the co-founder/CEO of   
 Radically Open Security.</bio>  
 </pentester>  
 <pentester>  
 <name>William of Ockham</name>  
 <bio>English Franciscan friar and scholastic philosopher and theologian.   
 Considered to be one of the major figures of medieval thought.   
 At the centre of some major intellectual and political controversies.</bio>  
 </pentester>  
 </pentesters>  
</collaborators>

### Document Classification

The <classification> element contains information on the confidentiality level of the report. Usually this will be 'Confidential'.

The classification will appear in the header of each page.

Example:

<classification>Confidential</classification>

### Version History

The <version\_history> element contains one or more <version> elements, one for each version of the document you create. Whenever you start a new version, add a <version> element to the list.

The <version> element should contain the following:

* a date attribute with a date of your version as a value, in the format YYYY-MM-DDT00:00:00, e.g. 2015-04-18T00:00:00
* a number attribute with the version number as a value. This value can either be 'auto' or an actual version number, e.g. 1.0. If you use the 'auto' value, the system will automatically count it (starting with 0.1 for the first <version> element and going up from there: 0.2, 0.3, etc...).
* One or more <v\_author> elements, each containing the name of the person who worked on this version (that would be you at least, and perhaps a pentester or colleague who did significant work on it)
* A <v\_description> element with a (very short!) description of what has been done in this version, e.g. 'Added non-findings' or 'Revision'

Example:

<version\_history>  
 <version number="auto" date="2014-12-18T00:00:00">  
 <v\_author>Bob Goudriaan</v\_author>  
 <v\_description>Initial draft</v\_description>  
 </version>  
 <version date="2014-12-22T00:00:00" number="auto">  
 <v\_author>Bob Goudriaan</v\_author>  
 <v\_author>Patricia Piolon</v\_author>  
 <v\_description>Revision</v\_description>  
 </version>  
</version\_history>

### Contact information

This is the contact information in a basic XML format. It never changes, so it has been isolated in its own little xml file, which is referred to from the main document with an <xi:include> element:

<xi:include href="snippets/contact.xml"/>

If you need to edit the contact information, edit that file. But it's extremely likely that you won't need to.

## The index

The document index is generated at the location of the element <generate\_index/>. To make sure the index works (meaning that a reference page number is listed for each section), you will need to give a unique id attribute to all elements (sections, appendices, findings and non-findings) that need to be listed in the index.

Insert the <generate\_index/> element immediately after the <meta> element.

## Sections

The main bulk of the pentest report is made up of normal content. We divide our content into sections using the <section> element.

### id

Make sure that every <section> element, no matter where in the structure it is located, gets a unique id attribute. By 'unique' we mean that no other element in the report should have the same value for its id attribute. This is enforced by the schema, so you will get an error message if you have duplicate ids in your report.

The exact value of the id attribute doesn't really matter, it can be anything, but it is good practice to pick an id that has some kind of relation to the section subject. For example, if your section is titled 'Technical Summary', a good id value for this <section> element would be 'technicalsummary'. You can use dots (.), dashes (-), underscores (\_) and numbers and letters in the id. You cannot use spaces.

Example:

<section id="summary\_of\_findings">  
...  
</section>

### Section title

A section must always start with a <title> element, which should only contain text; after that you're free to do what you want. As explained in the previous section, it's a good idea to have the section id and the title be somewhat related.

Example:

<section id="summary\_of\_findings">  
 <title>Summary of Findings</title>  
 ...  
</section>

### Section content

As said, after the title, anything goes (well, almost):

* A section can be subdivided into smaller sections (section 1 can be subdivided into 1.1, 1.2, etc.)
* A section can contain generic content, that is to say any number and order of:
  + paragraphs (<p>)
  + lists (ordered <ol> or unordered <ul>)
  + tables (<table>)
  + command input/output boxes (<pre>)
  + div containers (<div>)
* A section can contain any number of findings (<finding>)
* A section can contain any number of non-findings (<non-finding>)
* A section can contain any number of finding or recommendation summary tables (<generate\_findings>, <generate\_recommendations>)
* A section can contain a listing of targets, taken from the <targets> element in the meta section (<generate\_targets>)

All of these elements are described elsewhere in this document; see the appropriate sections for details.

## Appendices

Appendices (using the <appendix> element) work the same as sections, they just come last in the report. Like sections, they must have a unique id, and must start with a title. Also like sections, the rest of their content is free-form.

You will need at least one appendix, for the pentester listing (name and bio). This is generated from the info you provided in the <meta> section in the beginning of the report, so all you need to do is insert a <generate\_testteam/> element.

Example:

<appendix id="pentesters">  
 <title>Pentesters</title>  
 <generate\_testteam/>  
</appendix>

## Findings

Findings are special sections with a specific structure. Findings are written by the pentesters. It is the job of a report writer to copy them into the report (or reference them using an xi:include) and edit/elaborate.

A finding consists of a <finding> element with the following attributes:

* id - to uniquely identify the finding in the document
* threatLevel - which can be set to 'N/A', 'Low', 'Moderate', 'Elevated', 'High', or 'Extreme'
* type - the finding type (free text, but keep it short)

Furthermore, the <finding> is made up of several sub-elements:

* <title>, a title for the finding
* <description>, a short, general description of the finding
* <description\_summary>, an *optional* shorter description for use in the summary tables
* <technicaldescription>, a technical elaboration on what the problem entails
* <impact>, the finding's impact on the target's security
* <recommendation>, instructions or advice on how to improve security
* <recommendation\_summary>, an *optional* shorter recommendation for use in the summary tables

For more details, see the sections below.

### Note to pentesters

**PENTESTERS** should only use the <finding> element containing:

* the threatLevel attribute
* the type attribute
* the <title> element
* the <description> element
* the <technicaldescription> element
* the <impact> element, and
* the <recommendation> element

The contents of these elements is free - write whatever you like. The report writer will mark up your text.

Please create one file per finding.

There is no need to number your findings, as they will be numbered automatically in the report. There is also no need to add an id since the report writer is better positioned to do that.

Finding template:

<finding threatLevel="SelectFromList" type="ThreatTypeText">  
 <title>Finding Title</title>  
 <description>General Description of the problem</description>  
 <technicaldescription>Technical description of the problem.</technicaldescription>  
 <impact>Impact of the finding</impact>  
 <recommendation>Advice/tips/instructions on how to solve the problem</recommendation>  
</finding>

### Note to report writers

**REPORT WRITERS** should: - Add any necessary xml to the main structure elements (e.g. add paragraphs or <pre> text to the technical description) - Edit the text so that it is in correct english and informative/helpful to the client (or, when in doubt, ask pentesters to make it more informative/helpful) - Add a <description\_summary> and/or <recommendation\_summary> element if necessary.

### Description

This is a general intro to the problem. It can be left as-is, in which case the contents of this element will be treated as a paragraph, or it can be marked up as generic text (using the Generic content elements listed elsewhere in this document - with paragraphs, lists, tables, images).

The contents of the <description> element will be used verbatim in the finding summary table **unless** the finding *also* contains a <description\_summary> element, in which case that element will be used.

Example:

<finding id="xmlsignatureexclusion" threatLevel="Low" type="Signature Exclusion">  
 <title>XML signature exclusion</title>  
 <description>  
 <p>This is a reasonably short general description. It can easily fit in the summary table.</p>  
 </description>  
 ...  
</finding>

### Description for Summary Table

If the general finding description is too long or uses elements like images or tables, it will not be usable to put in the finding summary table. In this case you can add a <description\_summary> element right after the <description> element. The system will then use this description for the summary table instead. The contents of the <description\_summary> element will **only be used in the summary table**. This means that it **will not be visible in the finding text**.

Example:

<finding id="xmlsignatureexclusion" threatLevel="Low" type="Signature Exclusion">  
 <title>XML signature exclusion</title>  
 <description>  
 <p>This is a more elaborate general description:</p>  
 <img src="../graphics/screenshot.png"/>  
 <p>In this case, the screenshot above makes this description a bad candidate for inclusion in the summary table.</p>  
 <p>Additionally, the description is very very very very very very very very very very very very   
 very very very very very very very very very very very very very very very very very very very   
 very very very very very very very very very very very very very very very very very very very very   
 very very very very very very very very very very very very very very very very very very very   
 very very very very very very very very very very very very very very very very very very very very very   
 very very very very very very very very very very very very very very very very very very very   
 very very very very very very very long.</p>  
 </description>  
 <description\_summary>  
 <p>This is the alternative summary.</p>  
 </description\_summary>  
 ...  
</finding>

### Technical Description

This is a technical description of the problem. It can be left as-is, in which case the contents of this element will be treated as a paragraph, or it can be marked up as generic text (using the Generic content elements listed elsewhere in this document - with paragraphs, lists, tables, images).

Example:

<finding id="xmlsignatureexclusion" threatLevel="Low" type="Signature Exclusion">  
 <title>XML signature exclusion</title>  
 <description>  
 <p>This is a reasonably short general description. It can easily fit in the summary table.</p>  
 </description>  
 <technicaldescription>  
 <p>This is a very detailed and technical description of the finding using a screenshot:</p>  
 <img src="../graphics/screenshot1.png"/>  
 <p>And another screenshot:</p>  
 <img src="../graphics/screenshot2.png"/>  
 </technicaldescription>  
 ...  
</finding>

### Impact

This describes the impact of the problem. It can be left as-is, in which case the contents of this element will be treated as a paragraph, or it can be marked up as generic text (using the Generic content elements listed elsewhere in this document - with paragraphs, lists, tables, images).

Example:

<finding id="xmlsignatureexclusion" threatLevel="Low" type="Signature Exclusion">  
 <title>XML signature exclusion</title>  
 ...  
 <impact>This finding is not a big threat because most people don't even know what a  
 signature inclusion is. Do you?</impact>  
</finding>

### Recommendation

This element contains tips/advice/instructions to deal with the problem. It can be left as-is, in which case the contents of this element will be treated as a paragraph, or it can be marked up as generic text (using the Generic content elements listed elsewhere in this document - with paragraphs, lists, tables, images).

The contents of the <recommendation> element will be used verbatim in the recommendation summary table **unless** the finding *also* contains a <recommendation\_summary> element, in which case that element will be used.

Example:

<finding id="xmlsignatureexclusion" threatLevel="Low" type="Signature Exclusion">  
 <title>XML signature exclusion</title>  
 ...  
 <recommendation>Advise all users to change their passwords. That's always a good idea.</recommendation>  
</finding>

### Recommendation for Summary Table

If the recommendation is too long or uses elements like images or tables, it will not be usable to put in the recommendation table. In this case you can add a <recommendation\_summary> element right after the <recommendation> element. The system will then use this recommendation for the summary table instead. The contents of the <recommendation\_summary> element will **only be used in the summary table**. This means that it **will not be visible in the finding text**.

Example:

<finding id="xmlsignatureexclusion" threatLevel="Low" type="Signature Exclusion">  
 <title>XML signature exclusion</title>  
 ...  
 <recommendation>  
 <p>Advise all users to:</p>  
 <ol>  
 <li>Stand on their head</li>  
 <li>Restart their computers</li>  
 <li>Change their passwords</li>  
 </ol>  
 <img src="../graphics/pictureofauserchangingtheirpassword.png"/>  
 </recommendation>  
 <recommendation\_summary>  
 <p>Advise all users to change their passwords.</p>  
 </recommendation\_summary>  
</finding>

## Non-findings

Non-findings are much more freeform than findings. They consist of a <non-finding> element with: - a <title> - generic elements such as paragraphs, images, etc.

<non-finding> elements must have an id attribute.

(Basically, they are a <section> with a special name.)

Example:

<non-finding id="nf\_xss">  
 <title>Mail Server</title>  
 <p>  
 The server was running MailServer ABC for SMTP, POP3 and IMAP. This is   
 the most recent version of this particular piece of software.   
 No relevant vulnerabilities or exploits were found.  
 </p>  
 <p>  
 Note that this does not mean it's a good idea to use it, since the company  
 making MailServer ABC is notoriously terrible at pushing out secure software.  
</p>  
</non-finding>

## Summary tables

Every pentest report should include summary tables for all findings and recommendations. This is easy, however, as these tables are generated automatically by the software. You just need to indicate where, by using the <generate\_findings/> and <generate\_recommendations/> elements.

<generate\_findings/> and <generate\_recommendations/> will generate findings/recommendation summary tables for the complete report. If you only want to generate a table for findings in a specific section, add a Ref attribute and enter the id of the section you want to reference as its value.

Example:

<section id="xmlxamlsummary">  
 <title>Summary</title>  
 <generate\_findings Ref="section2"/>  
</section>

## Generic content

Generic content is modeled on very basic HTML.

### Paragraphs

Paragraphs ('

') go in sections or in the various sub-elements of findings and non-findings. They are the basic way of displaying text.

Example:

<p>This is a paragraph</p>

### Lists

Lists can be ordered (<ol>, for '**o**rdered **l**ist') or unordered (<ul>, for **u**nordered **l**ist). Regardless of whether a list is ordered or unordered, it contains one or more list items (<li>, for **l**ist **i**tem).

**Unordered lists**

Example:

<ul>  
 <li>Some item</li>  
 <li>Some other item</li>  
</ul>

**Ordered lists**

Ordered lists are numbered by default. You can configure a different ordering system by setting its type attribute to one of the following values:

|  |  |
| --- | --- |
| type | ordering |
| a | lowercase alphabetic |
| A | uppercase alphabetic |
| i | lowercase roman |
| I | uppercase roman |

Example:

<ol type="i">  
 <li>Some item</li>  
 <li>Some other item</li>  
</ol>

### Code/Input/Output Blocks

Whenever you need to display some command line input/output or code, use the <pre> element. It will conserve any whitespace you leave, so you can format the contents of this element in a pleasant/readable way. Use spaces for indents. Note that text in the <pre> element *will not wrap*.

Example:

<section id="nmap">  
 <title>nmap</title>  
 <p>Command:</p>  
 <pre>$ nmap -vvvv -oA fishinabarrel.sittingduck.com\_complete -sV -sC -A -p1-65535 -T5 fishinabarrel.sittingduck.com</pre>  
 <p>Outcome:</p>  
 <pre>Nmap scan report for fishinabarrel.sittingduck.com (10.10.10.1)  
In several lines  
 Some indented  
and some not  
  
This is not a haiku.</pre>  
</section>

#### Help! The code in my pre element contains < characters and it messes with my xml!

You can escape the < character by replacing it with its entity &lt;.

### Div containers

#### What does <div> do?

Nothing. <div> just *is*.

#### Sigh. Ok, why *is* <div>?

You can use <div> as a container for other block elements. This is basically only (but very) useful for snippets, as snippets need to be well-formed XML documentlets and can therefore only have one root element. If the snippet is a complete section, this is not a problem. If the snippet is a bunch of paragraphs or something, you're out of luck. Or rather, you used to be out of luck, because there was no <div>. But now there is <div>. So your snippet can be <div> (root element), containing everything you want. Well, everything that's allowed, anyway.

#### So what's allowed in <div>?

All block elements: <p>, <ul>, <ol>, <table>, <img>, <pre>, <code>

#### And what elements can *contain* <div>?

<section> and <annex>.

### Tables

**Rows**

Tables consist of a <table> element containing one or more rows (<tr>).

Example:

<table>  
 <tr>...</tr>  
 <tr>...</tr>  
</table>

**Cells**

A table row consists of one or more cells (<td>).

Example:

<table>  
 <tr>  
 <td>Cell 1 in row 1</td>  
 <td>Cell 2 in row 1</td>  
 </tr>  
 <tr>  
 <td>Cell 1 in row 2</td>  
 <td>Cell 2 in row 2</td>  
 </tr>  
</table>

Columns are implicit: each cell in a row corresponds to a column.

**Header Cells**

Instead of normal cells, you can also use header cells (<th>) for a table header.

Example:

<table>  
 <tr>  
 <th>Header cell 1 in row 1</th>  
 <th>Header cell 2 in row 1</th>  
 </tr>  
 <tr>  
 <td>Cell 1 in row 2</td>  
 <td>Cell 2 in row 2</td>  
 </tr>  
</table>

**Borders**

To turn on borders for your table, set its border attribute to '1'.

Example:

<table border="1">  
...  
</table>

You can also turn borders on or off (border="0") on lower levels (on the row level, for example) for finer-tuned border control.

**Setting column width**

To set the width for your columns, add a number for each column to the cols element. This number is in millimeters (you can either type 200mm or just 200; don't use cm or pt or px or other measures though). The total width between the margins is 17cm, so 170mm.

Example:

<table cols="50 50 70">  
 <tr>  
 <td>cell 1</td><td>cell 2</td><td>cell 3</td>  
 </tr>  
 <tr>  
 <td>cell 4</td><td>cell 5</td><td>cell 6</td>  
 </tr>  
</table>

This will give the first column a width of 50mm (5cm), the second as well, and the third a width of 70mm (7cm).

**Spanning multiple rows/columns**

To make a cell span multiple columns, set its colspan attribute to the number of columns you want to span.

Example:

<tr>  
 <td colspan="2">This cell spans the two cells in the row below.</td>  
</tr>  
<tr>  
 <td>Cell 1 in row 2</td>  
 <td>Cell 2 in row 2</td>  
</tr>

To make a cell span multiple rows, set its rowspan attribute to the number of rows you want to span.

Example:

<tr>  
 <td rowspan="2">This cell spans the two cells in the second column.</td>  
 <td>Cell 2 in row 1</td>  
</tr>  
<tr>  
 <td>Cell 2 in row 2</td>  
</tr>

**Alignment**

Set the align attribute of any cell, row or table to one of the following values to change the text alignment in that cell/row/table:

|  |  |
| --- | --- |
| align | result |
| right | right alignment |
| center | centered |
| justify | justified |

## Images

To insert an image, use the <img> element. In its src attribute, enter the relative path to the image file you want to reference.

To set the height or width, use *either* the height or width attribute. Any numerical value you enter will be interpreted as centimeters.

If you set both, only the width will be interpreted.

If you do not set any height or width, the image will be displayed at full page width (i.e. 17 cm wide)

Example: <img src="../graphics/xmlsignatureexclusion.png" width="5"/>

Optionally, you can set an image caption by adding some text in the title attribute.

Example: <img src="../graphics/xmlsignatureexclusion.png" width="5" title="This is a funny picture LOL"/>

### Inline elements

Inline elements are elements that modify the text inside e.g. a paragraph or a list item, for styling or linking purposes. You have the following options available to you:

**Bold**

To make text bold, wrap it in <b> tags.

Example:

<p><b>This text is bold</b> and this text is not.</p>

**Italic**

To make text italic, wrap it in <i> tags.

Example:

<p><i>This text is italic</i> and this text is not.</p>

**Underline**

To make text underlined, wrap it in <u> tags.

Example:

<p><u>This text is underlined</u> and this text is not.</p>

**Monospace**

To have inline text in a monospace font, wrap it in <monospace> tags.

Example:

<p><monospace>This text is monospace</monospace> and this text is not.</p>

**Superscript**

To have inline text in superscript, wrap it in <sup> tags.

Example:

<p><sup>This text is in superscript</sup> and this text is not.</p>

**Subscript**

To have inline text in subscript, wrap it in <sub> tags.

Example:

<p><sub>This text is in subscript</sub> and this text is not.</p>

**Links**

Link to internal (in the report) or external (on the web) pages using the <a> element. For internal destinations, you can either use an empty <a/> (recommended, see example 1) or 'normal' linking (see example 2).

In the href attribute of the <a> element, type:

# + the id of the section you're linking to (when linking to a section

* in the report), or
* the url of the website you're linking to (when linking to a website)

Example 1 - linking with an empty element:

<p>Please refer to <a href="#xss\_finding"/>.</p>

(Note that in this case, we would need to have an element with id "xss\_finding" in the report, otherwise the link wouldn't resolve.)

This will auto-generate the linked text: 'Please refer to SID-004 (page 4).', or 'Please refer to section 2 (page 13).'

Example 2:

<p>Please refer to <a href="#xss\_finding">our finding on insecure mailservers</a>.</p>

(Again, we would need to have an element with id "xss\_finding" in the report, otherwise the link wouldn't resolve.)

Example 3:

<p>Please refer to <a href="http://www.radicallyopensecurity.com">our amazing website</a>.</p>

## Manual breaks

### Line breaks

Mostly text is broken automatically (between paragraphs etc.) but in some rare cases you may need to insert a manual line break. To do so, use the <br/> element.

Example:

<p>This is my haiku<br/>  
my line is broken, but still<br/>  
the paragraph flows</p>

### Page breaks

To force a page break before or after a section, set its break attribute to 'before' or 'after'.

Note: breaks are inserted automatcally before every appendix and before/after the index.

Example:

<section id="technicalfindings" break="before">  
<title>Technical Findings</title>  
...  
</section>