Suricata - Feature #1612

Sphinx for docs

12/02/2015 05:17 AM - Victor Julien

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td>OISF Dev</td>
</tr>
<tr>
<td>Category:</td>
<td></td>
</tr>
<tr>
<td>Target version:</td>
<td>Documentation</td>
</tr>
<tr>
<td>Effort:</td>
<td></td>
</tr>
<tr>
<td>Difficulty:</td>
<td></td>
</tr>
</tbody>
</table>

**Description**

Move version dependent docs, like keyword docs to sphinx.

**Subtasks:**

- Feature # 1613: document sphinx switch
- Feature # 1621: Docs: Convert Section 6. Command Line Options
- Feature # 1622: Docs: Convert Section 9. Rule Management with Oinkmaster
- Feature # 1623: Docs: Conversion Section 10. Performance
- Feature # 1624: Docs: Convert Section 11. Configuration
- Feature # 1625: Docs: Convert Section 12. Reputation
- Feature # 1626: Docs: Convert Section 13. Init Scripts
- Feature # 1627: Docs: Convert Section 14: Setting up IPS/Inline for Linux
- Feature # 1628: Docs: Convert Section 15. Output
- Feature # 1629: Docs: Convert Section 16. File Extraction
- Feature # 1630: Docs: Convert Section 18. Public Data Sets
- Feature # 1631: Docs: Convert Section 20. Using Capture Hardware

**History**

#1 - 12/02/2015 07:22 AM - Victor Julien

We should probably create various sub-tickets with this one as parent to divide up the effort.

#2 - 12/02/2015 07:28 AM - Jason Ish


I've done a few more items on "8. Suricata Rules" so it might make sense for me to keep going down that section.

Some are very fast to convert, others, particularly where tables are involved are a little more tedious.

Does Redmine support real "sub-issues"?

Related: I'm not sure if we should bother with the development docs in Sphinx. Thoughts? Developers should be developing against bleeding age, and I wouldn't want someone developing against old developer docs. So perhaps those are best kept on the wiki, at least for now. Concentrate this effort on the user guide?

#3 - 12/02/2015 07:32 AM - Victor Julien

Agreed on limiting this to the User docs. Also we can exclude a bunch of things, like 'misc guides'.

Wrt sub tickets, the procedure is that you create a regular ticket and then set it's parent field to this ticket.

#4 - 12/03/2015 05:37 PM - Andreas Herz

Jason Ish wrote:

Related: I'm not sure if we should bother with the development docs in Sphinx. Thoughts? Developers should be developing against bleeding age, and I wouldn't want someone developing against old developer docs. So perhaps those are best kept on the wiki, at least for now. Concentrate this effort on the user guide?
I agree on concentrating on user guide first, but unless i missed something, i would also go on with developer docs, too.

I see no gain in splitting, especially when keeping the docs up to date it might be easier if it's in the .rst format for sphinx. So instead of editing the wiki pages in redmine manually, you could automate it better within the sphinx scenario.

#5 - 12/03/2015 06:03 PM - Jason Ish

Andreas Herz wrote:

    Jason Ish wrote:

    Related: I'm not sure if we should bother with the development docs in Sphinx. Thoughts? Developers should be developing against bleeding age, and I wouldn't want someone developing against old developer docs. So perhaps those are best kept on the wiki, at least for now. Concentrate this effort on the user guide?

    I agree on concentrating on user guide first, but unless i missed something, i would also go on with developer docs, too.

    I see no gain in splitting, especially when keeping the docs up to date it might be easier if it's in the .rst format for sphinx. So instead of editing the wiki pages in redmine manually, you could automate it better within the sphinx scenario.

Maybe I'm overthinking it, but with the user guide it makes sense to have the current state of the documentation snapshotted with that release, and staying that way. We don't have to have "if this version, do this.. If that version do that". Its specific to the version its included in.

What I worry about with developer documentation, if we do it the same way is some young dev is will follow the developers guide in the documentation with the release they are working from, then perhaps submit it to us. Practices could have changed, APIs will have changed, etc. Which is why I lean to keeping the developer guide out of the releases at least.

Now, at some point we may have 2 types of developer guides. We may have the one for developers working on Suricata itself (this is what I refer to above), and we may need a guide for those developing with Suricata - Lua plugins, Lua scripting, etc - now that should get versioned with the release.

#6 - 12/04/2015 02:38 AM - Victor Julien

Lua docs are part of user docs in my thinking. Wrt dev docs, lets revisit that after we've done the user docs. Just converting the user docs is a big effort.

#7 - 12/04/2015 02:40 PM - Andreas Herz

Jason Ish wrote:

    Maybe I'm overthinking it, but with the user guide it makes sense to have the current state of the documentation snapshotted with that release, and staying that way. We don't have to have "if this version, do this.. If that version do that". Its specific to the version its included in.

That's what i missed, i didn't see the release specific doc and with that in mind i understand why you want to manage it differently. So let's see how it works for the user part and talk again how we can improve the developer guides as well. I just think it's better to work on files within version control like .rst format instead of the normal wiki, but that's just my personal view :)