Bachelor thesis
Computing Science

Radboud University

Improving OpenCRE

Author:
Thomas Klein Breteler
s4068246

First supervisor/assessor:
dr. E. Poll (Erik)
erikpoll@cs.ru.nl

SIG supervisor:
R. van der Veer (Rob)
r.vanderveer@sig.eu

Second assessor:
dr. I. Buhan (Ileana)
ileana.buhan@ru.nl

June 14, 2022

Abstract

One of the goals of the OWASP Integration Standards project is to deliver a linking mechanism which connects any number of software security standards. This linking mechanism, OpenCRE, is currently in beta and while the core functionalities are working, much work must be done to improve the user experience and the linking between OpenCRE topics and standards.

In this thesis, we assess how openCRE can be improved. We performed a general assessment from the perspective of a newcomer to the project as well as through interviews with multiple stakeholders. Our initial findings are that the visualisation of the complex hierarchy of topics within the CRE clear, the page usage is inefficient, much unnecessary information is shown and the documentation is ineffective. We present several suggestions in the form of mockups for change based on our assessments and interviews. The accepted suggestions include a change in the hierarchy visualisation, removal of unnecessary text and renaming of many of the content blocks. The interviews yielded several interesting new insights which were worth considering. The most promising being the introduction of explanation tooltips and the possibility of adding a navigation sidebar.

A second requirement was to research the possibilities to improve the linking to standards. We provided a way to link directly to a location in an HTML document. For PDF and Markdown documents the possibilities are very limited.

Finally, we analysed how to improve the coverage of CWE in OpenCRE. CWE view 699 provides an overview of all CWE entries related to software development which is the most practical and relevant to OpenCRE. Other than manually linking and analysing CWE we could not think of an effective way of improving the linking in OpenCRE.

Contents

Chapter 1

Introduction

There are hundreds of known security threats to IT systems and just as many measures to stop these threats. To get an idea of what to implement to prevent known and unknown threats, standardisation is required. This is done through many different software security standards which cover all kinds of topics like security requirements, testing and good practice. Some leading makers of standards for software security are ISO (International Organisation for Standardisation), NIST (National standards institute of the USA) and OWASP (An open-source, community-led non-profit foundation). OWASP alone already has over 40 different standards ranging from detailed requirement lists like the ASVS (application security verification standard) [9] to broad explanation documents like the Cheat Sheet Series [6].

With over 200 relevant national and international standards, the standard landscape is very fractured. The 2019 ENISA (EU Agency for cybersecurity) report on the advancement of software security recommended to "Develop a common repository for shared security measures" [2], as there is much overlap between different standards. This is however easier said than done. Numerous attempts to visualise or link standards have been made however with little success. One example is `iotsecuritymapping.uk`, which is an initiative to map IoT security standards to a limited set of IoT-related topics as shown in figure 1.1. This overview is neither practical nor maintainable as updating it has to be done manually.

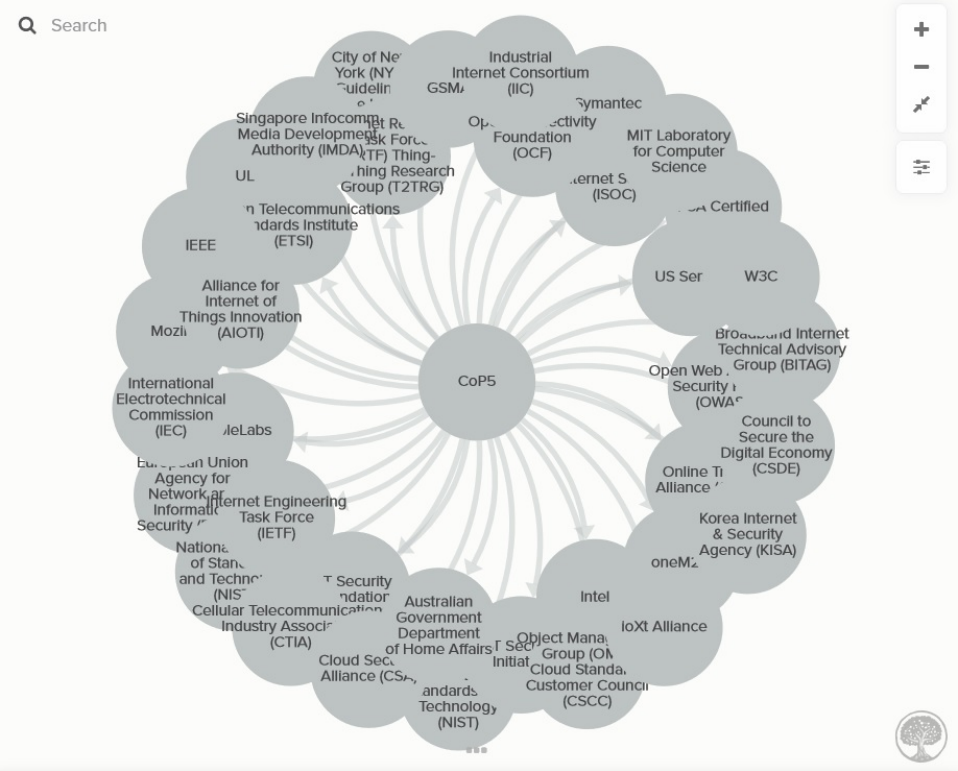

Figure 1.1: iotsecuritymapping.uk; code of practice 5: "communicate securely". A mapping of IOT standards to a shared topic.

OpenCRE is to be a solution to the fractured landscape. It is a mechanism designed to link specific sections of different security standards to common topics. CRE stands for Common Requirement Enumeration and OpenCRE consists of many of these requirements, referred to as topics. These topics are connected in the form of a hierarchy such that it can accommodate linking to both very specific standard entries as well as very broad explanation documents. Through cooperation with the makers of standards, OpenCRE can function in a maintainable fashion which is essential when having to incorporate data from a large number of sources into a single tool. We discuss the details of OpenCRE in chapter 2.

OpenCRE is currently in beta. The basic topic hierarchy is fully functional and links several standards, however fine-tuning is needed to make it a convenient and user-friendly tool. One of the goals of this thesis is to identify issues with the user experience and make recommendations on how to solve them. Furthermore, we will be looking at the

In Chapter 2 we go discuss the OWASP Integration Standards project of which OpenCRE is part and the technical details of OpenCRE itself. In

Chapter 3 we discuss the goal and the scope of this thesis. In chapter 4 we assess the user experience from the perspective of a newcomer to the project and we discuss the interviews we performed with different stakeholders of the project to get their views on how to improve OpenCRE. In section 4.4 we make concrete suggestions on how to improve the user experience of OpenCRE. Finally, in chapter 5 we discuss technical and strategic improvements to OpenCRE.

Chapter 2

Background

In Chapter 2.1 we discuss the OWASP Integration Standard project and in chapter 2.2 OpenCRE.

2.1 OWASP Integration Standard Project

The OWASP Integration Standards Project is an open initiative to promote technical interaction between software security initiatives in and outside of OWASP [11]. The goal is to reduce the fragmentation and complexity of the standard landscape. The project has 4 deliverables specified.

1. A report on the software development life cycle [1]
2. The security Wayfinder: an interactive overview of different OWASP projects. See figure 2.1.

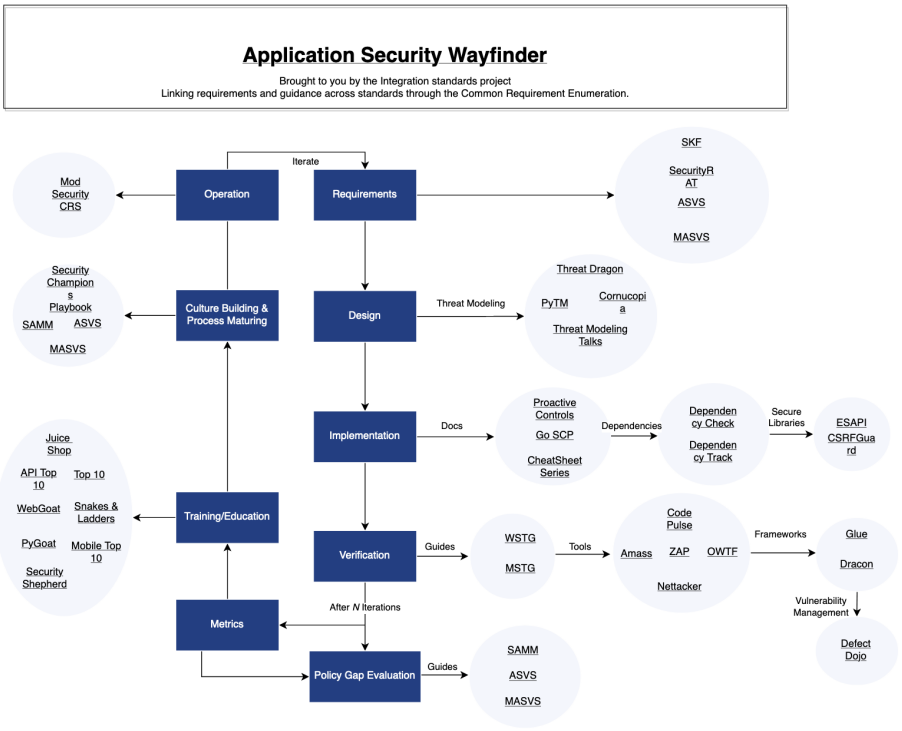

Figure 2.1: The OWASP application security Wayfinder

3. the common requirements enumeration (CRE): A mechanism which links content from different types of standards, bringing together requirements, testing strategies, countermeasures and repositories of weaknesses. The CRE (OpenCRE) is the focus of this thesis and we will provide more technical details in section 2.2.

4. A tool which helps integrate security initiatives into different stages of the software development lifecycle.

2.2 OpenCRE

OpenCRE is a mechanism which links common requirements to different software security standards. CRE stands for Common Requirement Enumeration and these common requirements form the topics on which the mechanism is built. The topics are interconnected and form a tree hierarchy, with a series of top-level topics which have multiple child topics. The result of this is a structure which can accommodate both broad topics such

as "authentication"[1] as well as very specific topics such as "Mutually authenticate application and credential service provider"[2].

OpenCRE has 2 focuses on use-cases:

1. The first use-case is for OpenCRE to enable a developer, tester or anyone involved in the software development process to quickly view what different standards have to say about a certain topic, navigate efficiently from one standard to another and get an overview of the standards relevant to the development process.

   For example, if a tester is looking through security requirements in the ASVS (the leading requirement standard) he/she can follow the link in the ASVS entry of interest to OpenCRE. In the OpenCRE he will find a list of standards linked to the same topic, among others the entry of WSTG (the leading OWASP testing guide). The tester can thus navigate from the same topics in the ASVS to WSTG without having to search for the corresponding coverage in the WSTG.

2. The second use case is to provide a comprehensive overview of a topic. This overview will provide different sources for the relevant topic as well as a comprehensive list of closely related topics.

   For example, if a developer wants to know more about "input and output verification"[3] the corresponding topic in OpenCRE offers an overview of related topics and standards.

OpenCRE can foster a better understanding of cybersecurity as a whole by helping standard makers link to other standards instead of having to cover everything themselves. It will also highlight security subjects which might be underrepresented in security standards, therefore contributing to the general understanding of the security field. The platform is in beta and is available on www.OpenCRE.org. It currently (February 2022) links 5 OWASP standards:

1. OWASP Top 10 [10], 10 most common security flaws. It provides a short description of the flaws and some general advice on how to fix them.

2. ASVS [9] (Application Security Verification Standard), One of the leading OWASP projects. It's a comprehensive list of requirements for developers.

3. OWASP Proactive controls [7] A top 10 requirement list of must-dos for architects and developers.

[1] https://www.opencre.org/cre/633-428
[2] https://www.opencre.org/cre/558-807
[3] https://www.OpenCRE.org/cre/503-455

4. OWASP Cheatsheets [6], Explanation documents on various security subjects, with a focus on how to securely implement them. E.g. Session Management Cheat Sheet[4] explains everything you need to know about session management from a security perspective.

5. WSTG (Web Security Testing Guide) [8], The leading OWASP security testing guide.

and 3 other important sources and standards:

1. Common weakness enumerations (CWE) [3], A repository of around 1000 known weaknesses maintained by MITRE. Although they refer to them as weaknesses, they are more accurately generalised vulnerabilities.

2. NIST-800-53 [4], NIST[5] standard on information systems in organisations.

3. NIST-800-63b [5] NIST standard on government system security.

OpenCRE is not the first attempt to link standards, multiple attempts have been made but with mixed results and limited usability. OpenCRE aims to solve 3 problems encountered in previous efforts [13].

- The first problem is that linking all standards to each other is too much work and unmaintainable. See figure 2.2

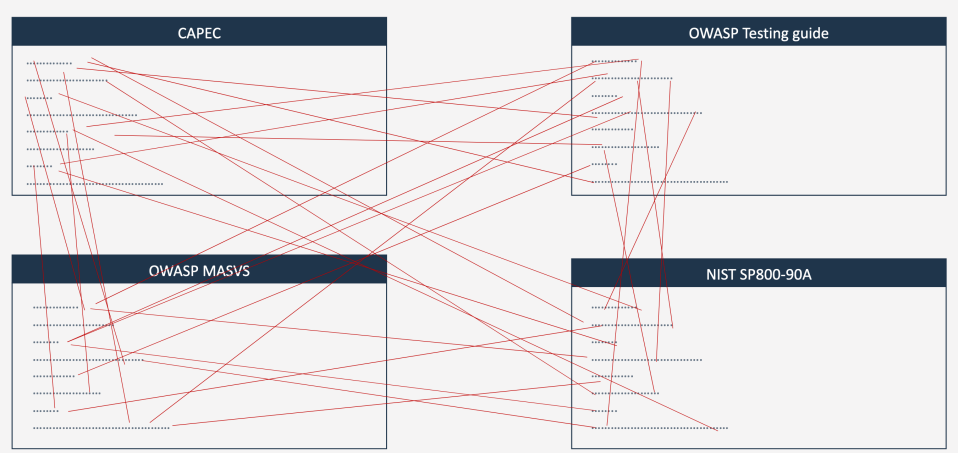

Figure 2.2: Linking every standard to an entry in another standard doesn't work

[4]https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html
[5]National Institute of Standards and Technology of the USA

The solution to this is to create shared topics and link the standards to those topics. This way users can view a certain topic and see what different standards have to say about it as if it were a single resource. These topics can be like "logging and error handling" [6], On this OpenCRE page a user can see all standards directly linked to this topic as well as other OpenCRE topics linked to it. See figure 2.3

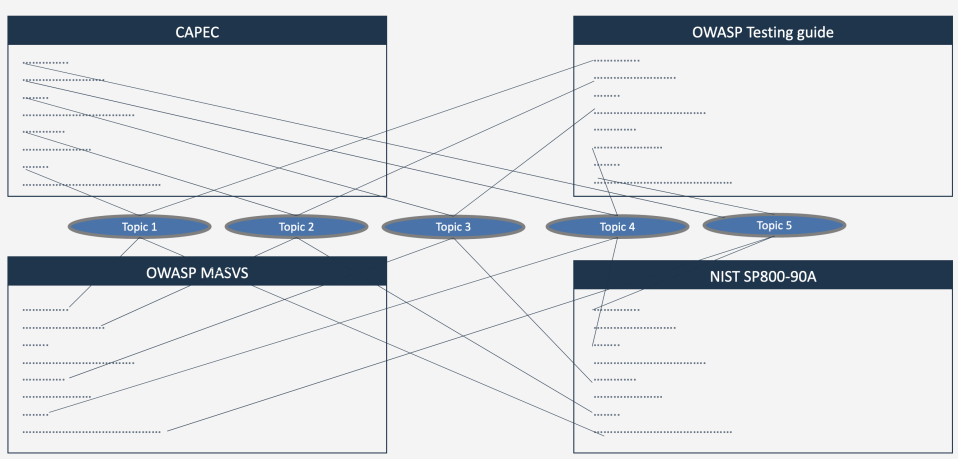

Figure 2.3: Shared topics linking standards [13]

- The second problem is finding a certain topic in the forest of subtopics is too much work for most users (see figure 2.4).

[6] https://www.OpenCRE.org/cre/842-876

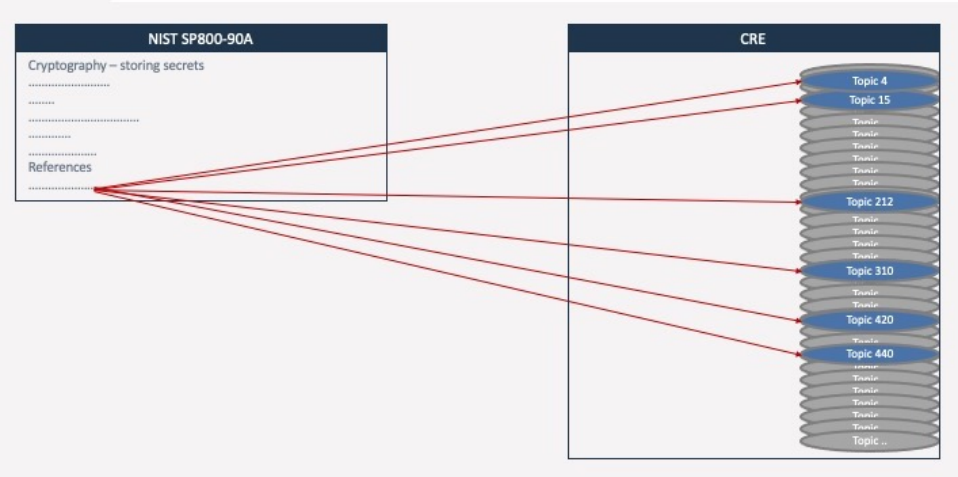

Figure 2.4: Multiple topics are related to certain standards

To remedy this, high-level topics are introduced. These high-level topics link broad standards which cover multiple subtopics at once. These high-level topics are connected to subtopics forming a hierarchy of related topics. (see figure 2.5)

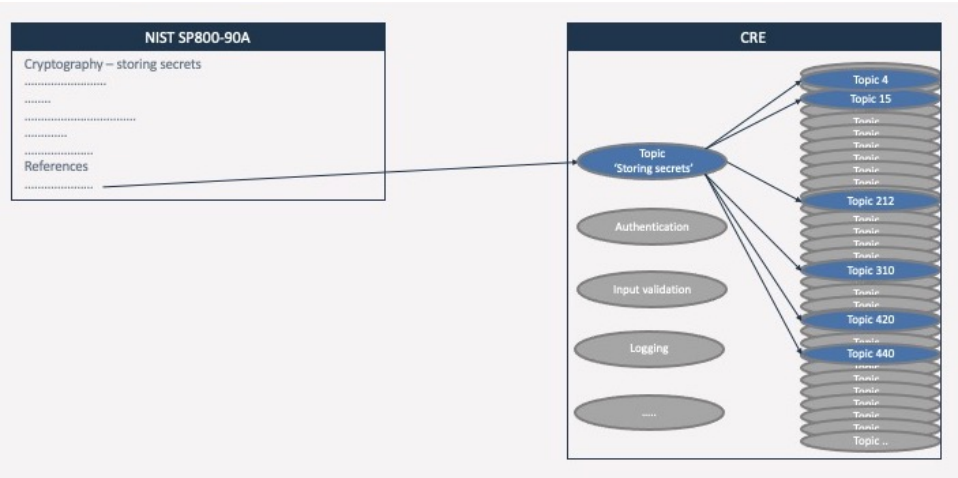

Figure 2.5: The topic hierarchy removes the mismatch in topic depth.

- The final problem is that when OpenCRE links to a standard and if that standard changes anything, the OpenCRE link breaks or becomes incorrect. Take for example the differences between the OWASP Top 10 2017 and Top 10 2021 as visualised in figure 2.6.

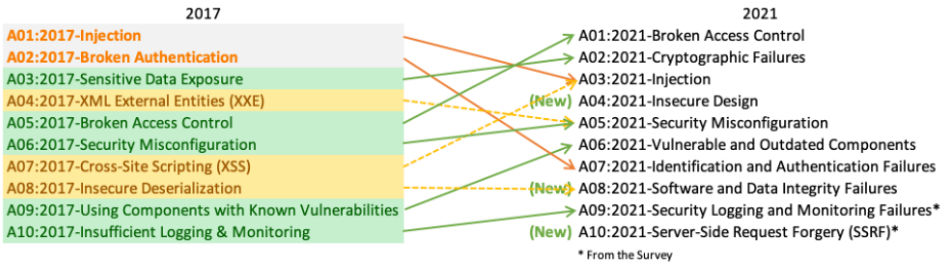

Figure 2.6: Mapping between OWASP Top 10 2017 and OWASP Top 2021 [10]

Mapping just the differences between 2 versions of the same standard is already a complex matter. Mapping between multiple standards makes it a lot harder and especially harder to maintain. Normally someone would have to manually change the links in OpenCRE to adopt the changes of OWASP top 10 2017 compared to OWASP top 10 2021. For a standard with just 10 entries which updates every few years, this might be doable (but still very undesirable) however, considering that if OpenCRE would incorporate 40 or so standards, maintaining the system would not be viable.

The solution to this is to make the standards link to the unique CRE code and map according to these codes. This way, when a standard changes, the mapping algorithm can automatically find the location of the relevant pages and will always display the latest version. This feature has yet to be implemented as standard makers have yet to update their standards to include the relevant CRE codes. In short, have the standards link to the correct topic OpenCRE rather than the other way around.

Figure 2.7 shows a schematic overview of OpenCRE. The topics are linked to each other in a hierarchy and form multiple trees which cover different subjects. Each topic is linked to several standards which cover the subject of the topic. Finally, some CRE topics are linked to a different CRE topic outside of the main hierarchy. These are "related to" relations as often 2 topics can have strong relevance to each other while not being part of the same hierarchy. For example: "Monitor unusual activities on the system" is part of the business logic hierarchy, however, this CRE is strongly related to "logging and error handling". The CRE can in this case refer to another topic to provide a more complete picture of the topic.

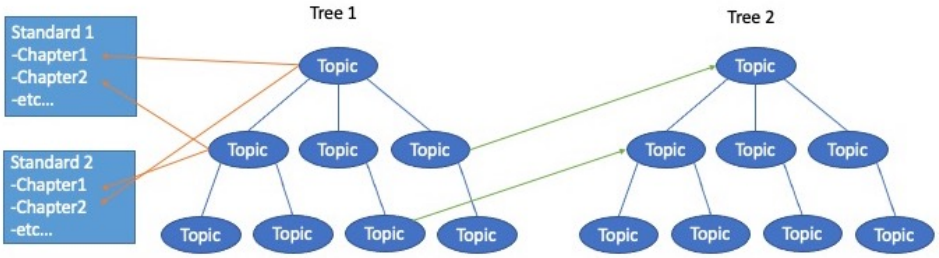

Figure 2.7: Schematic overview of OpenCRE

One of the future goals of OpenCRE is to enable an analysis of the standard landscape by looking at the overlap and gap between different standards. Another goal is to perform anonymised data analysis on the use of OpenCRE. The insights gained from this can help improve the software security field.

2.3 Human-Computer Interaction

Much of this thesis will involve redesigning the user interface of OpenCRE, it is worth taking a look at what is relevant in the field of human-computer interaction. Ben Shneiderman's introduced his 'eight golden rules of user interface design' [12] in 1986 which remains leading in the field up to this day. The principles are as follows:

1. Strive for consistency.
2. Enable frequent users to use shortcuts.
3. Offer informative feedback.
4. Design dialogue to yield closure.
5. Offer simple error handling.
6. Permit easy reversal of actions.
7. Support internal locus of control.
8. Reduce short-term memory load.

Of these rules, especially 2, 3 and 8 are relevant to this project as too much irrelevant information is shown and too little explanation of the topics is given.

Another useful lesson to be learned from Shneiderman's book is to consider the level of knowledge of the users. We can assume that all users will have a background in computing science and while not necessarily experts on security topics, they will have a basic understanding of computing science. This will give us some leeway in our use of language as we can use some technical terminology which would not be understood by someone outside of the field.

Chapter 3

How to improve OpenCRE?

The goal of this thesis is to maximise the success of OpenCRE. To do this we must first define what success entail, we discuss this in section 3.1.

3.1 Research approach

The goal of this thesis is to help **maximise the success of OpenCRE**. The problem is that success is a vague term which can be interpreted in many ways. Thus if we want to maximise the success of openCRE we must first determine what success is in this context.

OpenCRE is first and foremost a tool used to link the standard landscape. From this perspective, success can be seen as the adoption by standard makers. This means that as more standard makers link to the CRE, its value increases as it offers the bridge between standards.

There have been multiple initiatives that have linked standards with little success as these lists are generally incomprehensible due to the amount of different information it tries to group and link. This is the second key success point for OpenCRE. Not only should there be a link between many different standards, but the linking should also be clear such that users can conveniently navigate between standards or get an overview of the literature on a topic of choice.

To maximise the success of openCRE we will focus on making the tool more user-friendly and increasing the accuracy of the linking. By doing so standard makers will be more inclined to adopt CRE links into their standards.

Chapter 4

Assessing the user experience

In this chapter, we assess the user experience of OpenCRE and its documentation. We dive into OpenCRE from the perspective of a newcomer without receiving any substantial explanation to see what kind of first impression it gives in section 4.1. We assessed the documentation in section 4.2. In section 4.3 we discuss the interviews we performed with 3 stakeholders and some of the suggestions they gave. Finally, in section 4.4 we make suggestions on what and how to improve the user experience of OpenCRE.

4.1 Assessment of usability

The CRE project is currently in beta, most core functionalities have been implemented and are working. However, many improvements are necessary for it to become the product it was designed to be. In this section, we will assess different parts of OpenCRE to see what can be improved and make suggestions on how to improve these.

As part of the first assessment we covered 2 use cases:

1. Access OpenCRE through the link in an included standard such as ASVS 6.1 [1]
2. Use the text search or browse feature to find information on a certain topic.

In both cases, the focus was on the content, front-end, appearance and user experience. The initial finding was that the topic pages (i.e. https://www.opencre.org/cre/842-876) are hard to read and confusing. The reasons are as follows:

1. The inverted tree structure is confusing. Tree structures are normally top-down, having the parent node above the child node. In OpenCRE

[1] https://github.com/northdpole/ASVS/blob/04316f240bc1f7bad058394a40d183c34d14521f/4.0/en/0x14-V6-Cryptography.md

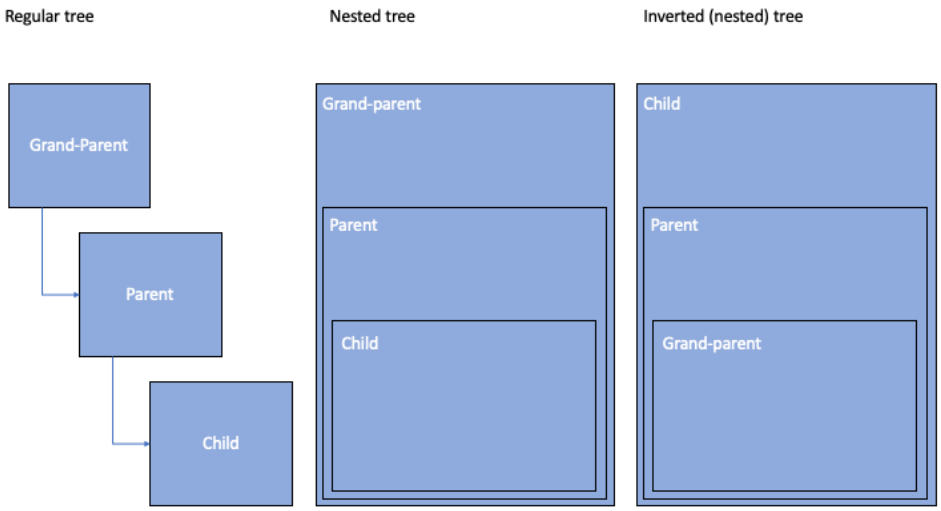

Figure 4.1: Different visualisations of a tree structure

the choice was made to visualise the tree recursively from the bottom up. This representation was chosen to show the most closely related topics first, however, the downside is that navigation becomes less clear. Figure 4.1 shows the difference between a regular tree and an 'inverted' tree.

2. Relations between topics and standards in OpenCRE are explained through a sentence like: "x is related to:" or "x is linked to". However, these textual relations are not explained anywhere and it is not always clear what they mean. Combining this with the previous point makes a topic page hard to understand for first-time users especially.

3. Every topic has a unique identification code connected to them. However, these codes make no real sense to users as they are semi-random. They only contribute to cluttering the screen with numbers.

4. Aside from the codes, there is a lot of unnecessary text and prefixes on the screen which makes the screen feel cluttered.

Additionally, there are several miscellaneous issues which can negatively affect the user experience:

1. To navigate from the page showing an overview of all topics linked to one standard (i.e. https://www.opencre.org/node/standard/ASVS/section/V5.3.8) to another standard linked to the same OpenCRE topic, the user has to go through 3 different pages. According

to the golden rules of Shneiderman, it is very desirable to allow users to have shortcuts, thus having a way to immediately access different standards would be ideal.

2. Search results are unsorted and unstructured. i.e. https://www.opencre.org/search/asvs shows an unsorted list of ASVS pages.

3. The function of the "related" category is unclear in the current layout. The "related to" blocks tend to be scattered about the page without clear intent.

4. No explanation of the topics and standards. The user is expected to know what all the terms mean and what the different standards are for.

5. Long lists of standards reduce the readability. OpenCRE shows all standards connected to the topic you are viewing, standards connected to "related" topics and standards connected to topics in the parent topics. This can result in large lists of sometimes up to 10 different entries of the same standard [2]. Grouping and hiding these lists will decrease the amount of text shown and greatly increase the readability of the page.

Figure 4.2 shows a page which features most of the above-mentioned issues.

[2]See for example https://www.opencre.org/cre/153-513

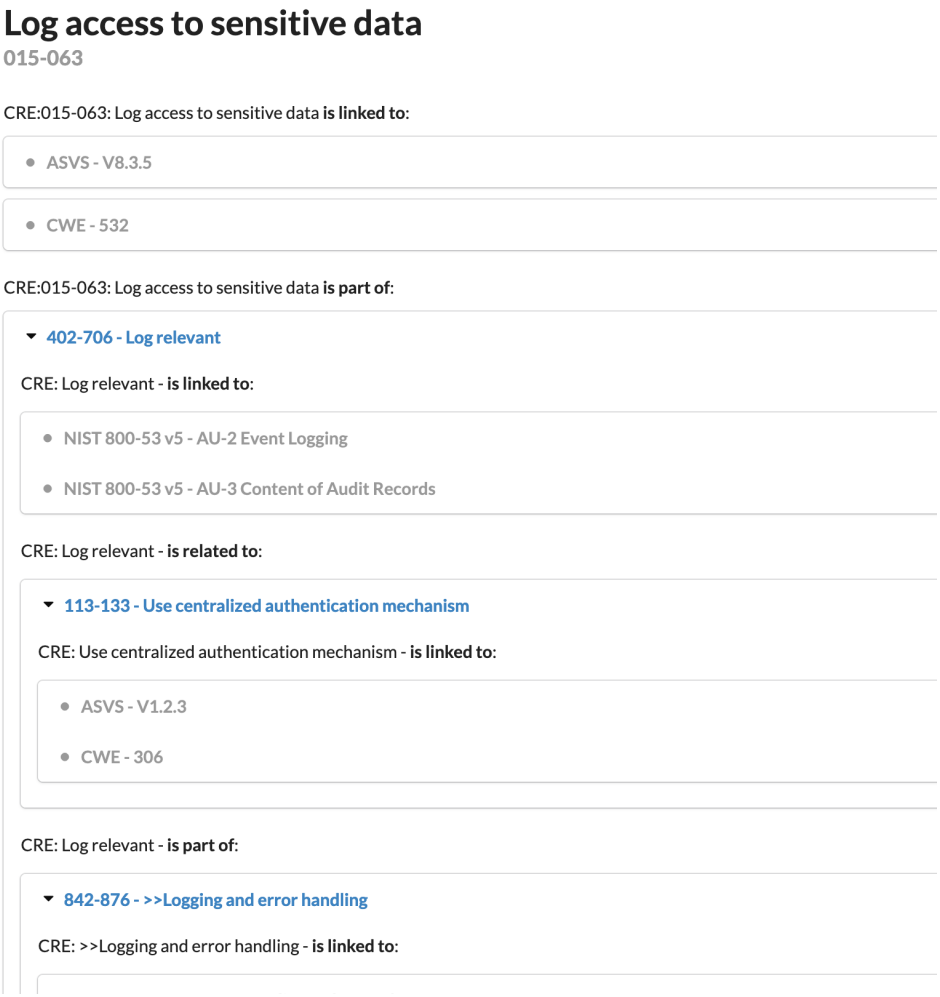

Figure 4.2: An example of an OpenCRE topic page. (cropped to be made readable).

In section 4.4 we discuss ways to solve these issues. To solve the issues found we tried multiple alternative layouts however the results were unsatisfactory. The topic pages contain a lot of information and the options to properly visualise them without losing the oversight it is to create are limited. The most obvious solution is to make the tree into a regular top-down tree, this takes away the unintuitive aspect of it.

Once the tree can be visualised intuitively, the need for a textual description of the relations is no longer needed. This allows us to reduce the amount of text on the page. The CRE codes can also simply be removed as they add no value.

4.2 Documentation of OpenCRE

To promote the adoption of OpenCRE, its usage should be clear to new users and more importantly, standard makers. A clear and accessible front page and documentation are essential to achieve this.

Currently, the front page is pretty much an about page, which is fine as users will likely not access or bookmark the front page if they use OpenCRE. The explanation video and document presented on the front page are more of an introductory presentation than a dedicated explanation. Making both shorter and more to the point would make things a lot more accessible to newcomers.

Especially the explanation of and solution to "problem 2" (section 2.2 on page 10) was very confusing to me. The explanation document [3] provides argumentation for the design choices, however, the function of the hierarchy was not elaborated upon. The document was updated to reflect on the design choices. Figures 2.4 and 2.5 are the product of this finding.

4.3 Interviews about the user experience

To get a broader understanding of how to improve the user experience of OpenCRE we interviewed 3 employees of SIG (Software Improvement Group) with various backgrounds about their general impression of OpenCRE and how they think it could be improved. The first interviewee has worked on OpenCRE before and is well versed in the security field, the second is an expert in the security field but is only familiar with OpenCRE on a conceptual level and the final interviewee is neither a security expert nor familiar with OpenCRE. We performed a semi-structured interview which lasted 30-45 minutes and provided several links to OpenCRE and mockups to discuss. In appendix A we provide the general questions of the interview and in section 4.3.1 we discuss the various ideas the interviewees come up with.

4.3.1 Summary of feedback points

1. More explanation for the topics: A tooltip (hover-over pop-up window) which shows 1 or 2 lines of description to explain what the topic is about without having to go into it. Especially to someone who is not a security expert, some terms used might not be clear to these users which will make using OpenCRE hard for them. Descriptions can be provided by a hover-over, an icon/button which displays it or an explanation line behind the topic name. Figure 4.3 shows how

[3] https://raw.githubusercontent.com/OWASP/www-project-integration-standards/master/writeups/CRE-Explained6.pdf

CWE pages utilise tooltips to provide much information unobtrusively.

Figure 4.3: CWE pages utilise tooltips extensively

2. Textual relations are confusing. What do "part of" and "related to" mean? This is the same issue which was identified in section 4.1 and the fact that the interviewees pointed this out as well underlines the need to change this. An obvious way to implement this is to adopt a parent/child naming scheme. Since the stakeholders are predominantly people with a technical background, it is safe to assume they will be familiar with tree structures and how the naming of these works. As an added upside, it will be immediately clear that the CRE topic hierarchies are trees, making the internal structure clear to the user.

3. Hiding long lists of standards behind a collapsible button/box would be useful.

4. On the architecture page there is a long list of topics. This could use some guidance to help the user figure out where to start. A solution could be to group topics which are related. The general problem is that a new user would not know where to start and providing natural groupings would reduce the number of options a user would have to choose between.

5. Completely separate all relations. Make blocks for child, parent, grandparent, linked and related topics/standards.

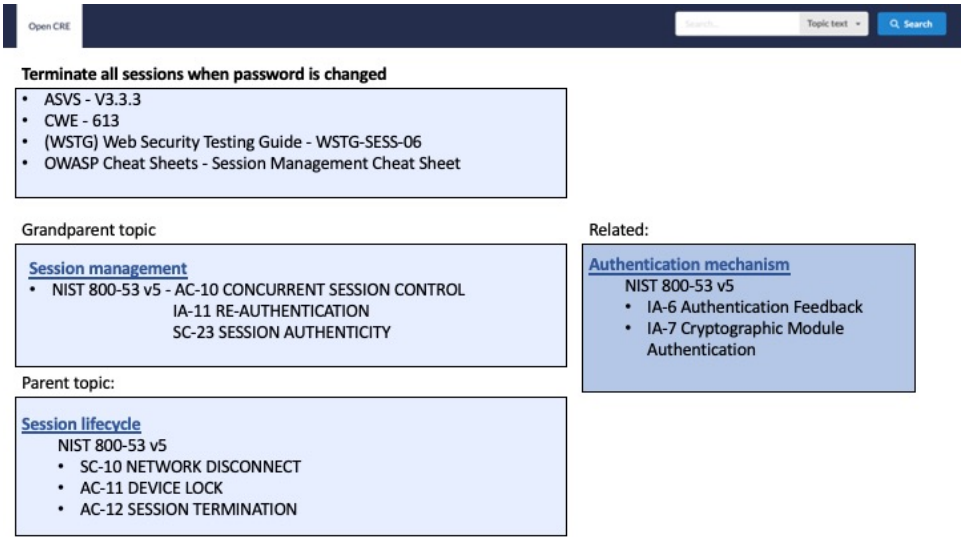

Figure 4.4: Mockup showcasing the possibility of separating all blocks

6. Make it clear what the related topics relate to by including them in the block it belongs to. In some of the early mockups used during the interviews, the related topics were moved outside of the main structure. Some interviewees preferred the related blocks to be kept in the blocks of the topics they belong to.

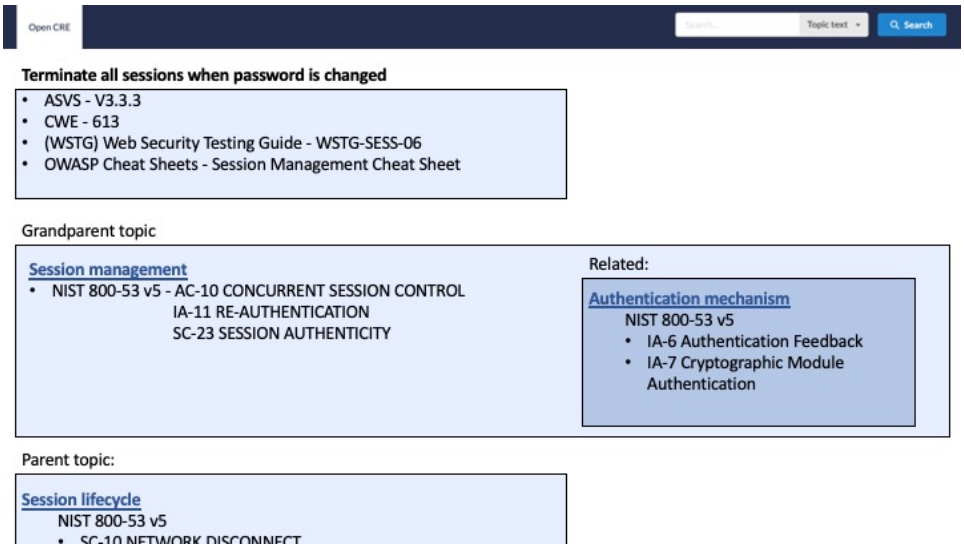

Figure 4.5: Mockup separating the topics of the main hierarchy but including "related" in the connected topic

7. more colour coding to make clear which section is which. No reading would be required, especially for people familiar with the system. For example, someone would know or learn to recognise the red block as the linked standards, green as the related pages, blue as the leaves, and yellow as the parents. This can also be done with icons like on CWE pages.

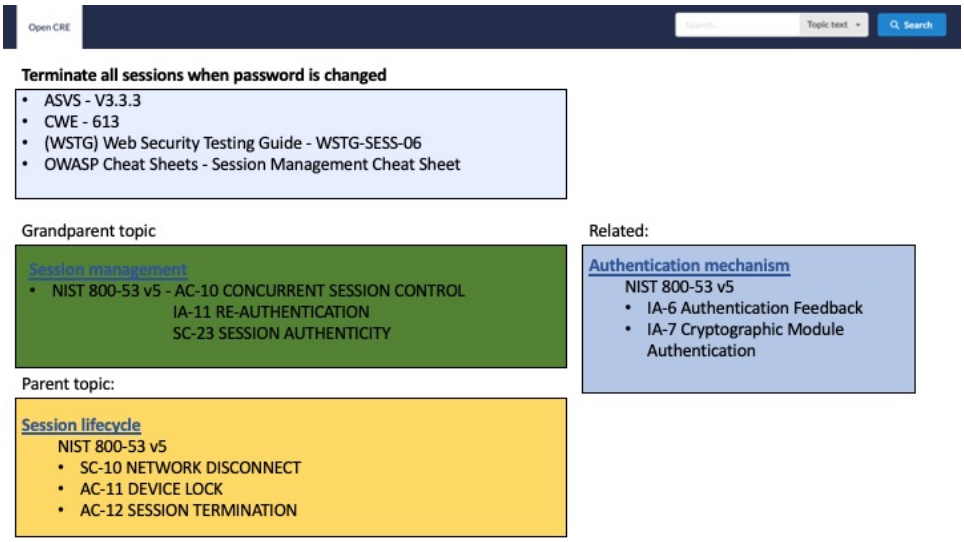

Figure 4.6: Mockup adding more different colours to blocks

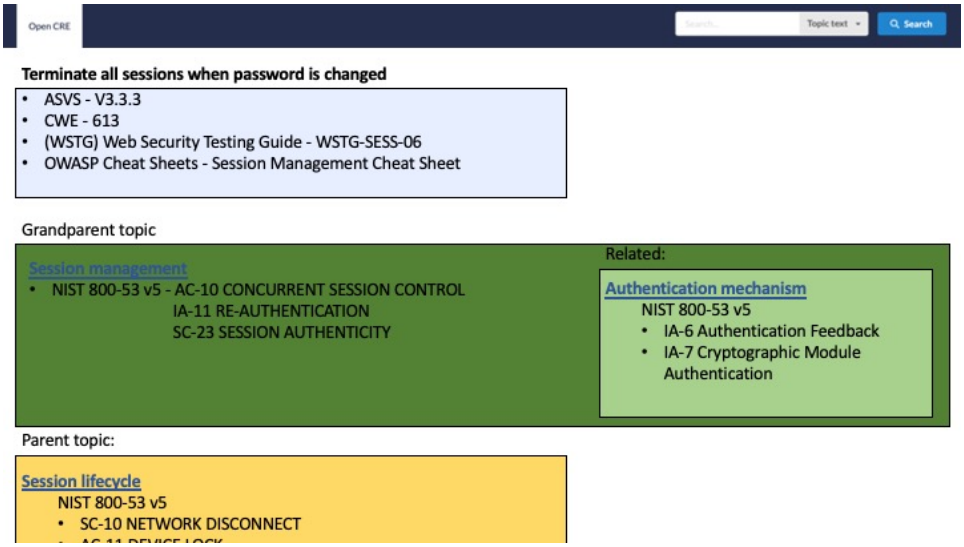

Figure 4.7: Alternative view of coloured blocks

8. Reconsider the order of blocks. Maybe move the "containing" (child) block up as users will be more likely to be interested in the underlying topics rather than the overarching topics.

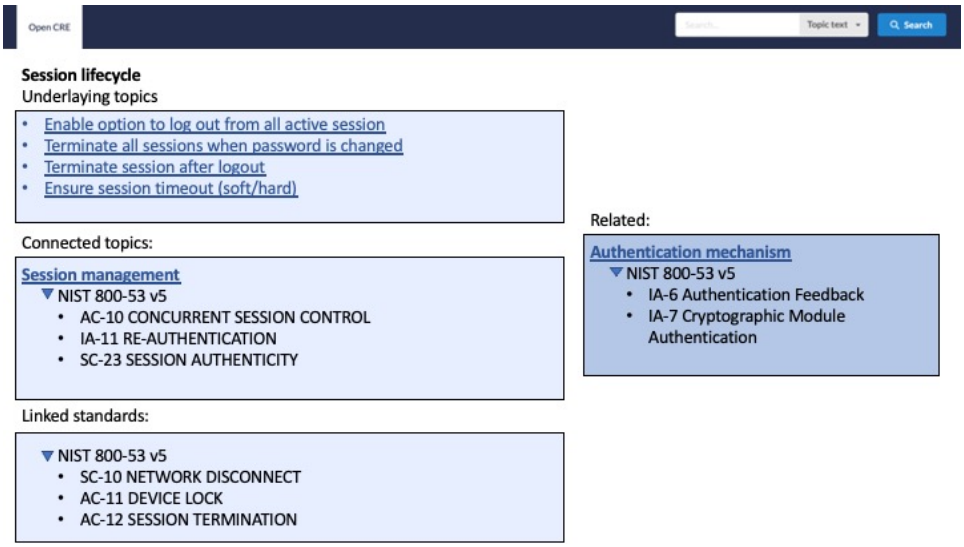

Figure 4.8: Mockup of CRE page showing an alternate ordering of blocks

9. A sidebar like in "readthedocs" might give more oversight and the possibility to quickly access the top-level topics.

Figure 4.9: "readthedocs" has an extensive sidebar used to navigate through topics

4.4 Suggested changes

We have identified several issues in sections 4.1 and 4.3, in this section we will now provide suggestions on how to improve OpenCRE to solve these issues. The goal is to provide sufficient context for the suggested changes to be implemented by the developers.

4.4.1 Cleaning up the topic pages

All unnecessary text and codes should be left out as more information for users to process make a page harder to read. See figure 4.10

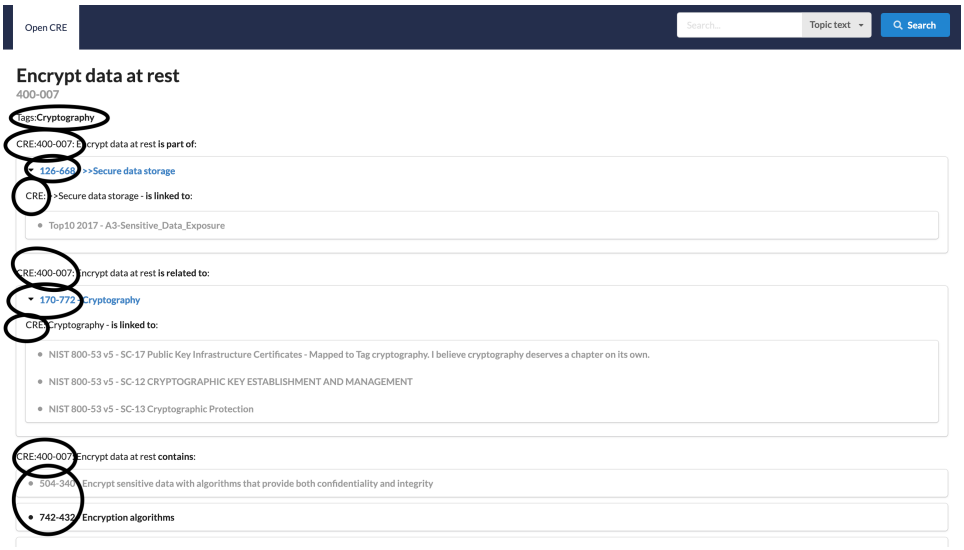

Figure 4.10: Unnecessary codes and text highlighted on the topic page of "encrypt data at rest"

Suggested changes are:

1. The unique CRE topic code "xxx-xxx" have no value to users and can be removed without losing any functionality.
2. "Tag:cryptography" can be removed as these tag labels also appear as a "related to" category. Thus the line adds no value.
3. "CRE:" prefix before codes or topic names can be removed as this is considered jargon and does not add value to the user.

4.4.2 Restructuring of the tree hierarchy

The current version of OpenCRE uses an inverted tree structure (As shown in figure 4.1). This way of visualising the hierarchy is counter-intuitive and

we suggest using a regular tree structure instead. The difference with the original layout is that "session management" and "session lifecycle" have been swapped. (figure 4.11)

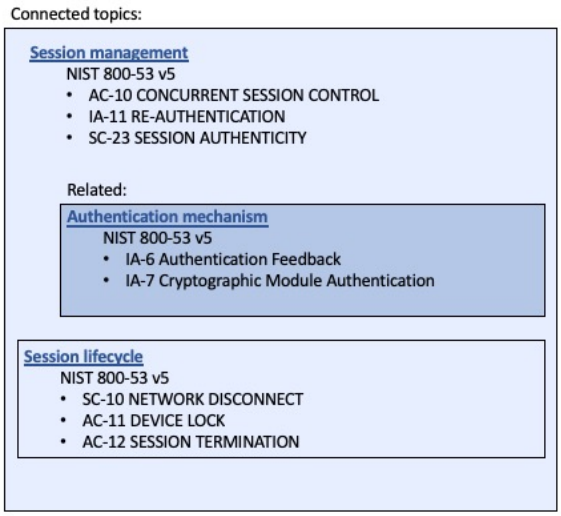

Figure 4.11: Normalised tree and added basic colour distinction

Alternatively, we can completely separate the tree nodes and explain the hierarchy by use of common terminologies such as parent, grandparent and child. Adopting the naming scheme of a tree structure will immediately make the hierarchy clear to anyone familiar with tree structures and as the users will likely have a technical background, it can be assumed they will be familiar with this. As a bonus, this change will also eliminate the vagueness of the terms "is part of" and "contains" which are currently used to describe child and parent nodes. (figure 4.12)

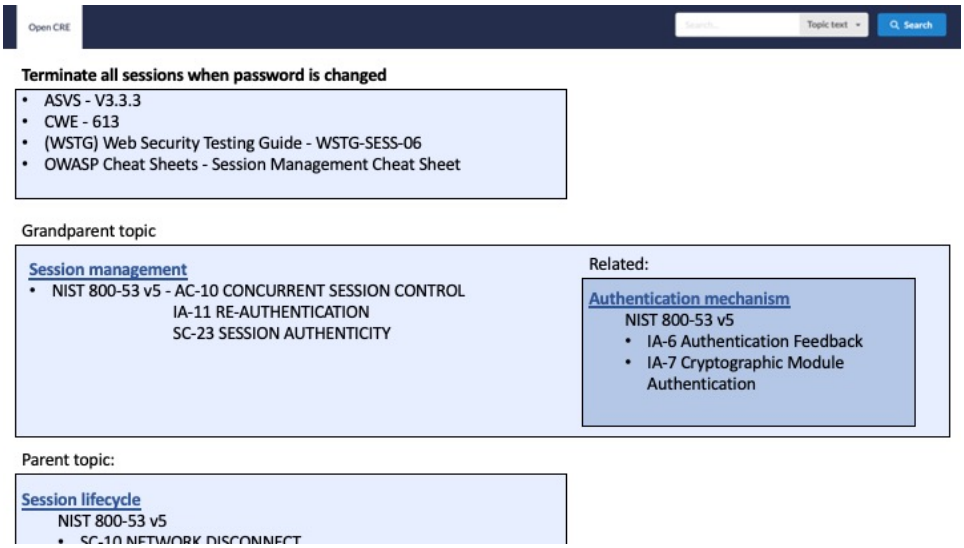

Figure 4.12: Tree nodes have been separated

Adopting this way of visualising eliminates the problems with the ordering of the topics as we now use common terminology rather than visual tricks to explain the structure. Using the style as shown in figure 4.12 is preferred as it is completely clear what the relations are, while figure 4.11 still has some ambiguity.

4.4.3 Tooltips and explanations

The topics used in OpenCRE have a great variety in depth. Some are very specific requirements while others are broad concepts. Take for example "session management", "session lifecycle" and "terminate all sessions when password is changed". For the latter, it is clear what is meant by the topic just by reading its name. "Session management" and "Session lifecycle" are quite broad and vague. This would require a user to click on the topic and infer based on the standards linked to it and topics connected to it to see what it is actually about. This vagueness can be easily countered by introducing a single line of explanation. We propose adding a small explanation line to each topic which explains what the topic entails. One way this can be done is by using tooltips and icons like they are being used in CWE pages (see Figure 4.3). These tooltips can be shown when hovering the cursor over the topic name or by adding a small icon after the topic. This way explanation is shown to those who want it and it will not clutter the screen with more text.

4.4.4 Topic browsing through a sidebar

At the moment there is no easy way to access the list of top-level topics or to freely navigate between topics which are not directly connected within the tree hierarchy. As suggested by an interviewee we can implement this in a way similar to `www.readthedocs.org` (figure 4.9). See figure 4.13 for a mockup.

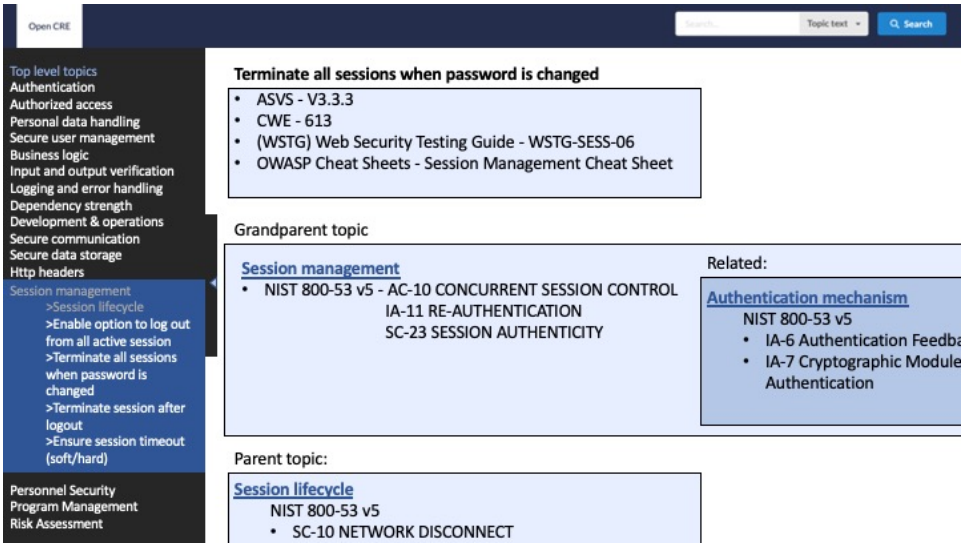

Figure 4.13: Proposal for an OpenCRE page with a navigation sidebar

As the use of the sidebar is purely situational, we suggest adding the possibility to hide it (or to hide it by default). See figure 4.14

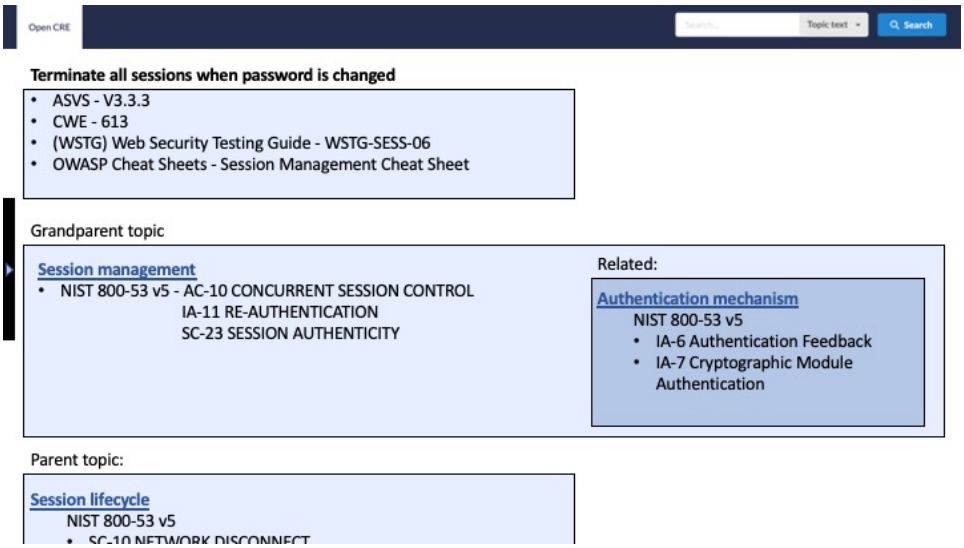

Figure 4.14: Navigation sidebar hidden

4.4.5 Miscellaneous changes

Aside from the bigger changes mentioned in previous sections, there are several smaller changes which would improve OpenCRE but are not big enough for a separate section:

1. Add a hyperlink icon to standards to enable users to immediately navigate to the standard without having to go through the OpenCRE page for the standard. Users will likely be more interested in going immediately to the standard than to view a page which shows which topics are connected to a specific OpenCRE topic. An icon is unobtrusive and will provide this functionality without having to change much in the system.

2. Rename ”Is linked to” to ”refers to” when referring to the standards connected to a CRE topic. This more accurately describes the function of the block as topics refer to certain standards. ”linking” is jargon used by the developers of OpenCRE which is not immediately clear to new users. See figure 4.15

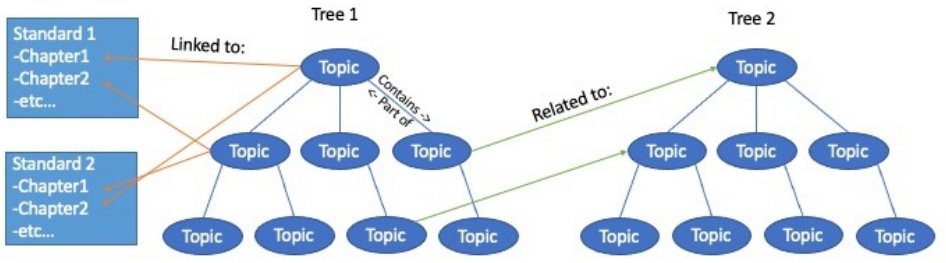

Figure 4.15: OpenCRE refers to standards as "linked to" which doesn't explain its function accurately

3. Collapse "related to" blocks by default. Currently, everything is expanded which results in the topic pages being big and chaotic. Collapsing the related topics clean up the page considerably.

4. Sorting search results alphabetically is something very easy and will greatly improve the readability of search results. Another improvement to search results is introducing the grouping of entries from the same source. For example, searching for something which yields 13 NIST entries and 5 other sources can be shown as having results from 6 different sources where 1 (NIST) has 13 entries.

Chapter 5

Improved deep linking

OpenCRE is a linking platform, it links specific entries of security standards to common topics. However, there is no standardisation in how security standards are published, which results in standards being published in many different formats like HTML pages, PDF documents and markdown pages. In this chapter, we will discuss what deep linking is and how it is done for the previously mentioned document types.

5.1 Deep linking

Deeplinking is a way of linking to a specific page on a website or a specific location on a page. www.opencre.org links to the OpenCRE homepage, while https://www.opencre.org/cre/402-133 deep links to a specific OpenCRE topic. One of the main features of OpenCRE is that it links to specific pages of standards to take away the need to look through an entire document to find what you are looking for. Not all standards, however, have a format which allows for this conveniently. Some standards have a plain HTML page and others are only available as PDF. We searched for ways to facilitate deep linking to specific points in both HTML and PDF documents. In sections 5.1.1, 5.1.2 and 5.1.3 we present our findings for these different document formats.

5.1.1 HTML

Deep linking in a specific location in an HTML page is fairly easy as every section in an HTML is marked by an 'id' and adding #id to the link makes the page load at the location of the id in the page. The id can be found in the source code of the page. For example: to link to "NIST 800 63b section 5.1.9.1" we take the general link to the NIST 800 63b page https://pages.nist.gov/800-63-3/sp800-63b.htmland add the id of the specific section we want to go to https://pages.nist.gov/800-63-3/sp800-63b.html#

5191-multi-factor-cryptographic-device-authenticators.

5.1.2 PDF

After much searching, we concluded that it is not possible to link to a specific section in a PDF document. While there exist some features which allow users to jump to a specific bookmark or search term through the URL, these only seem to work reliably on some browsers and are thus not a viable solution. The closest solution we could find was the option to link to a certain page in the pdf. This is done by adding #page=10 to jump to page 10 upon opening the link. This feature seemed to work on all browsers. For example, https://owasp.org/www-pdf-archive/OWASP_Application_Security_Verification_Standard_4.0-en.pdf#page=10 brings you to page 10 of the ASVS. All in all, this is not very useful for OpenCRE and this remains a problem.

5.1.3 Markdown

Markdown is a lightweight markup language which can add formatting to plaintext. It is fairly easy to use, however, deep linking is impossible unless the author makes specific links in the markdown code. Take for example https://github.com/OWASP/ASVS/blob/v4.0.2/4.0/en/0x10-V1-Architecture.md#v12-authentication-architectural-requirements. This link goes to chapter 1.2 of the ASVS but this only works because there is a pre-existing link made by the authors. Another way Markdown documents are displayed is through gitbooks, this is an even more stylised version of a markdown document, however, the same restrictions apply. There is no way to refer to a more specific spot if the writer has not linked to it as there is no 'id' or similar point of reference to link to as with HTML.

Chapter 6

Improved linking to CWE

The Common Weakness Enumeration (CWE) is a large resource of great importance to OpenCRE as the requirements of OpenCRE need to protect against the weaknesses documented in CWE. Improving the linking to CWE is hard however as there are almost 1000 CWE entries of which only a part is relevant in the scope of CRE.

To get a grasp of how to improve the linking between OpenCRE and CWE we analysed the OpenCRE coverage in section 6.1, we researched the coverage and structure of CWE in section 6.2 and looked into the discrepancies found in the mapping between OpenCRE, CWE and OWASP ZAP in section 6.3. Finally, we present our conclusion on this subject in section 6.4.

6.1 Charting CRE

To get a better understanding of the content-wise coverage of OpenCRE we performed an analysis of the top-level topics. What we found is that the CRE covers 3 main categories:

1. Web application security. ranging from detecting whether communications are automated or not, to authentication, to crypto. This category is well established, covering a wide range of logically connected topics. Figure 6.1 shows a visualisation of the coverage of OpenCRE regarding Web application security.

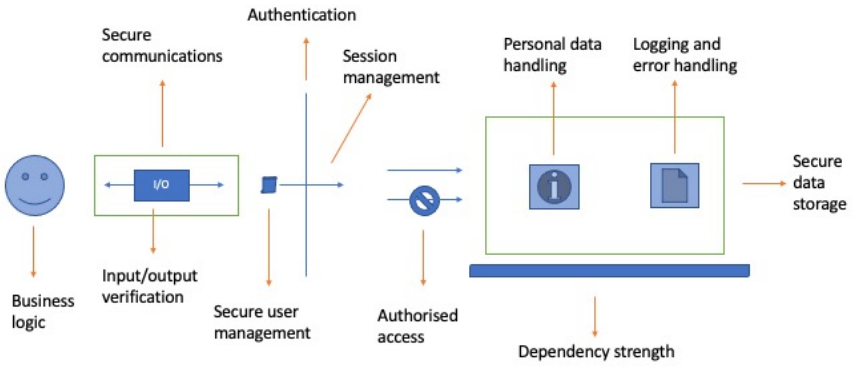

Figure 6.1: Web-app coverage of OpenCRE

2. Organisational. Requirements revolving around for example hiring personnel and risk assessments. This has 4 top-level CREs but can be expanded to include more topics such as physical security and others mentioned in ISO 27k and NIST.

3. Deployment. One top-level CRE covers deployment and operations. This CRE covers aspects of the SDLC.

Most of the top-level topics cover web application security and the deployment and organisational topics are entirely based on NIST standards.

6.2 Charting CWE

Not all CWEs are relevant. However, figuring out which are relevant is not a straightforward task. Hardware CWEs are obviously out of the scope of the CRE and variant CWEs might be too specific to include and would not contribute much. For example, CWE 41 "improper resolution of path equivalence" could be relevant but the variant children such as CWE 42 "Path Equivalence: 'filename.' (Trailing Dot)" would not add much, especially if its parent is already included.

MITRE offers several CWE 'views' which might be the key to this problem. 2 views stand out as they categorise and reduce the amount of CWEs to a more usable selection. The simplified view (CWE 1003) is a selection of 127 weaknesses with some categorisation, which had some potential. The focus of this view is on open source weaknesses however and thus is too

limited for OpenCRE. More notably the software development view (CWE 699) is likely most relevant as this is a set which has already been filtered and sorted into categories relevant to software development. This set has 419 entries spread over 40 categories, making it a lot more manageable than the complete set.

6.3 ZAP-CWE-CRE discrepancies

OWASP ZAP is a new resource which is integrated into OpenCRE during the span of this thesis. ZAP is an automated code analysis tool which checks the code against predefined rules. The ZAP rules are now linked to a relevant OpenCRE topic but also have a link to a CWE which covers the weakness detected. The linking between OWASP ZAP, OpenCRE and CWE is not perfect, however. There is a set of CWEs which are linked to a ZAP rule but are not in OpenCRE. Figure 6.2 elaborates the relations between the 3 resources.

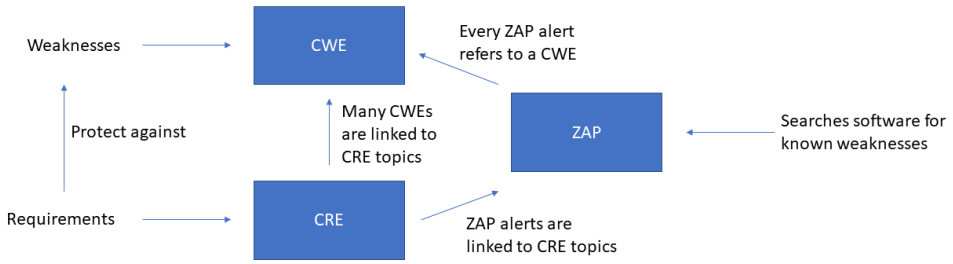

Figure 6.2: Relations between CWE, CRE and ZAP

We were provided with a list of CWEs which were connected to OWASP ZAP rules but not in the CRE. This list was compiled before the start of this thesis as part of an effort to verify the existing mappings of CWE and CRE. The data consists of CWEs, not in openCRE which are connected to ZAP rules which ARE in OpenCRE. We manually cross-referenced all the CWEs and ZAP rules to make sense of the what is going on and found that these zap rule-CWE couples fall into 7 groups based on the similarity of their subject which can be further generalised into 2 categories:

1. Specific ZAP rule with just as specific CWE. These CWEs and ZAP alerts can be linked to a CRE topic which covers the broad categories. The reason these were not already in the CRE is likely because they can be considered variants of a more specific weakness which is in OpenCRE. Adding these will not add any value to OpenCRE as they are too specific to make for requirements.

2. Specific ZAP rule with very broad CWE.This category is ZAP rules which do not have a proper matching CWE entry. The problem here lies mostly with MITRE as these are weaknesses which are simply not covered in the CWE. The broad CWEs are not useful to include in the CRE as they provide very little information, possibly a child CWE of the broad entry can be added to provide more coverage but this should be assessed case by case. The ZAP rules can be added to an appropriate CRE. For instance "ZAP Rule: "Source Code Disclosure - Git " is about being able to access source code without proper authorisation and can be added to a relevant CRE ("data access control"?).

The raw data can be found in Appendix A.2

6.4 How can the CWE-CRE linking be improved?

Linking CWE and CRE is hard as it's like comparing apples and oranges. While there is much overlap, fitting them together is a complex puzzle which ultimately boils down to manually going through CWEs to figure out whether there are requirements connected to them in the CRE or if new requirements should be added.

The sheer number of CWEs makes it impossible to add everything and most of the 'views' provided by MITRE are not very useful. The best approach for improving CWE linking in the CRE is through the CWE view 699 [1]. This provides 40 software development categories which can serve as a basis for finding gaps in the CRE and creating new mappings from CRE topics to CWEs.

When comparing CRE and CWE the most notable gap is the lack of best practices, code quality and documentation entries in the OpenCRE. CWE view 699 offers several categories which cover these topics and can thus be used as a basis for adding or expanding OpenCRE. These categories should be carefully assessed by an expert to make suitable requirements for these uncovered subjects.

[1] https://cwe.mitre.org/data/definitions/699.html

Chapter 7

Future work

1. In chapter 4 we identified several changes which can greatly improve the usability of OpenCRE. The implementation however falls outside of the scope of this project.

2. In section 6.4 concluded that there is room for improvement in the CRE-CWE linking in the area of code quality, documentation and best practices. However, we did not have the time to make concrete recommendations for linking these topics or categories nor did we have the expertise to create new topics.

3. In section 6.1 we analysed the content-wise coverage of OpenCRE and found that the majority of the topics cover application security. The deployment and organisational security categories were limited in comparison and only linked to NIST. There is much room for expansion in these areas.

Chapter 8

Conclusions

In this thesis, we worked to identify ways to improve OpenCRE. To do this we performed assessments of its current functionality. The first step was a general assessment from a newcomer's point of view. The second step of the assessment was through interviews with multiple stakeholders of the project, including developers and security experts. We conclude that OpenCRE in the way it looks during its current beta phase can be confusing even to someone with a broad understanding of the security field. The main issues regarding user experience were:

1. The reverse tree structure shown in figure 4.1 is counterintuitive from a user's perspective, causing unnecessary pause on an overview which is designed to provide quick and easy access to numerous sources.

2. There is a large amount of information shown. There was a lot of text used to explain relations and codes used which are mostly meaningless to users.

3. Not all relations on the topic pages are clear. These should be intuitive and recognisable to ensure users can easily navigate to where they want to go.

4. Lack of explanation of topics and standards. Topic names can be cryptic for example, "Encode user input before logging". Having a line of explanation or a tooltip which explains this would enhance the user experience.

5. Poor page usage. Everything is ordered vertically making OpenCRE pages longer than needed while leaving the right side empty.

We made several suggestions to improve these issues in section 4.4, most notably:

1. Remove unnecessary codes and other jargon to reduce short-term memory load.

2. Restructure or break up the tree hierarchy to make the layout more intuitive.

3. Adding explanations of topics in the form of hover-over tooltips to enable more informative feedback.

4. A topic browsing sidebar to enable easy reversal of action and shortcuts.

5. Changing the way topics refer to each other to be more natural and informative.

In chapter 5 we discussed ways to deep link HTML, PDF and Markdown documents in OpenCRE. Finally, in section 6 we explored the gap between CWE and CRE and made recommendations on how to improve the linking of CWE in OpenCRE.

Bibliography

[1] Rob van der Veer Elie Saad, Spyros Gasteratos. Software development lifecycle report. https://owasp.org/www-project-integration-standards/writeups/owasp_in_sdlc/.

[2] ENISA. Enisa report on security standards and certification, 2019. https://www.enisa.europa.eu/publications/advancing-software-security-through-the-eu-certification-framework.

[3] Mitre. Common weakness enumeration. https://cwe.mitre.org/.

[4] NIST. nist-800-53, rev. 5 September 2020. https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/final.

[5] NIST. nist-800-63b, rev. 5 September 2020. https://pages.nist.gov/800-63-3/sp800-63b.html.

[6] OWASP. Cheatsheet series. https://cheatsheetseries.owasp.org/.

[7] OWASP. Proactive-controls, 2018. https://owasp.org/www-project-proactive-controls/.

[8] OWASP. Wstg (web security tesing guide), WSTG 4.1 2020. https://owasp.org/www-project-web-security-testing-guide/v41/.

[9] OWASP. Asvs (application security verification guide), ASVS 4.0.3 2021. https://owasp.org/www-project-application-security-verification-standard/.

[10] OWASP. Owasp top 10, Top 10 2021 2021. https://owasp.org/www-project-top-ten/.

[11] OWASP. Owasp project integration standard homepage, 2022. https://owasp.org/www-project-integration-standards/.

[12] Ben Shneiderman, Catherine Plaisant, Maxine S. Cohen, Steven Jacobs, Niklas Elmqvist, and Nicholas Diakopoulos. *Designing the user interface: strategies for effective human-computer interaction*. Pearson, 2016.

[13] Rob van der Veer. Initial explanation document of the opencre tool. https://github.com/OWASP/www-project-integration-standards/raw/master/writeups/CRE-Explained6.pdf.

Appendix A

Appendix

This is the rough layout of the questions posed during the semi-structured interviews.

A.1 Interview

1. Work history
   (a) Education
   (b) past functions
   (c) current function
2. Standards and security?
   (a) Experience with security
   (b) Experience with security standards.
3. OpenCRE
   (a) What does the interviewee know about openCRE?
   (b) Case with CRE: wiki pagina die naar cre linked
   (c) What does the interviewee miss contentwise in openCRE? For the various usecases that might be relevant to the interviewee.
   (d) What does the interviewee miss UX in openCRE?
4. Design changes
   (a) Show mockups and ask which would be preferable?

A.2 ZAP-CWE-CRE analysis raw data

Output of a script which analysed which CWEs are not in CRE but are referenced in ZAP rules. The colours indicate similarities in subject.

opencre.org does not know of CWE 113, it is linked to by zap alert: ZAP Rule: "CRLF Injection"
opencre.org does not know of CWE 472, it is linked to by zap alert: ZAP Rule: "Parameter Tamperin
opencre.org does not know of CWE 776, it is linked to by zap alert: ZAP Rule: "Exponential Entity Ex
opencre.org does not know of CWE 91, it is linked to by zap alert: ZAP Rule: "XSLT Injection"
opencre.org does not know of CWE 917, it is linked to by zap alert: ZAP Rule: "Expression Language
opencre.org does not know of CWE 943, it is linked to by zap alert: ZAP Rule: "NoSQL Injection - Mo
opencre.org does not know of CWE 97, it is linked to by zap alert: ZAP Rule: "Server Side Include"
opencre.org does not know of CWE 119, it is linked to by zap alert: ZAP Rule: "Heartbleed OpenSSL
opencre.org does not know of CWE 1275, it is linked to by zap alert: ZAP Rule: "Cookie without Sam
opencre.org does not know of CWE 215, it is linked to by zap alert: ZAP Rule: ".env Information Leal
opencre.org does not know of CWE 215, it is linked to by zap alert: ZAP Rule: "Spring Actuator Infor
opencre.org does not know of CWE 215, it is linked to by zap alert: ZAP Rule: "Trace.axd Informatio
opencre.org does not know of CWE 264, it is linked to by zap alert: ZAP Rule: "Cross-Domain Miscor
opencre.org does not know of CWE 530, it is linked to by zap alert: ZAP Rule: "Backup File Disclosur
opencre.org does not know of CWE 538, it is linked to by zap alert: ZAP Rule: "Hidden File Finder"
opencre.org does not know of CWE 541, it is linked to by zap alert: ZAP Rule: "Source Code Disclosu
opencre.org does not know of CWE 541, it is linked to by zap alert: ZAP Rule: "Source Code Disclosu
opencre.org does not know of CWE 541, it is linked to by zap alert: ZAP Rule: "Source Code Disclosu
opencre.org does not know of CWE 541, it is linked to by zap alert: ZAP Rule: "Source Code Disclosu
opencre.org does not know of CWE 565, it is linked to by zap alert: ZAP Rule: "Loosely Scoped Cooki
opencre.org does not know of CWE 642, it is linked to by zap alert: ZAP Rule: "Emails Found in the
opencre.org does not know of CWE 642, it is linked to by zap alert: ZAP Rule: "Insecure JSF ViewSta
opencre.org does not know of CWE 642, it is linked to by zap alert: ZAP Rule: "Old Asp.Net Version
opencre.org does not know of CWE 642, it is linked to by zap alert: ZAP Rule: "Potential IP Addresse
opencre.org does not know of CWE 642, it is linked to by zap alert: ZAP Rule: "Split Viewstate in Us
opencre.org does not know of CWE 642, it is linked to by zap alert: ZAP Rule: "Viewstate without M
opencre.org does not know of CWE 642, it is linked to by zap alert: ZAP Rule: "Viewstate without M
opencre.org does not know of CWE 693, it is linked to by zap alert: ZAP Rule: "CSP"
opencre.org does not know of CWE 693, it is linked to by zap alert: ZAP Rule: "Insufficient Site Isola
opencre.org does not know of CWE 693, it is linked to by zap alert: ZAP Rule: "Insufficient Site Isola
opencre.org does not know of CWE 693, it is linked to by zap alert: ZAP Rule: "Insufficient Site Isola
opencre.org does not know of CWE 693, it is linked to by zap alert: ZAP Rule: "X-Content-Type-Optic
opencre.org does not know of CWE 74, it is linked to by zap alert: ZAP Rule: "Server Side Template
opencre.org does not know of CWE 933, it is linked to by zap alert: ZAP Rule: "X-AspNet-Version Re
opencre.org does not know of CWE 942, it is linked to by zap alert: ZAP Rule: "CORS Header"
opencre.org does not know of CWE 942, it is linked to by zap alert: ZAP Rule: "CORS Misconfiguratio