COMMON REQUIREMENT ENUMERATION

For successful use and creation of security standards

Rob van der Veer,
representing the team behind the CRE initiative, including Spyros Gasteratos, Sylvan Rigal and Elie Saad

September 2021

CRE in a small nutshell

- Common Requirement Enumeration (**CRE**) is an initiative by independent security experts to link between standards and guidelines.
- The idea of CRE is to **link** each small section of a resource to a shared topic identifier(a *Common Requirement*), instead of linking to just 1 or 2 other resources. Through this shared link, all resources map to eachother.
- This 1) enables standard and guideline makers to work efficiently, 2) it enables users to find the information they need, and 3) it facilitates a shared understanding in the industry of what cyber security is. The key element is **self-maintainability**.
- Currently, CRE is **being implemented** as part of the OWASP *integration standards* project.

CRE in a nutshell

- It is **hard for people in software security to gain overview and find information** in today's cyber security standard and guidelines landscape. There is a tremendous amount of useful knowledge, yet it turns out to be fragmented, complex and confusing. The result: security weaknesses, incidents, unnecessary develop and test expenses, and a hard time for standard creators as well.
- Common Requirement Enumeration (**CRE**) is an initiative based on experience and research: experience form working closely with cyber security standardization organizations for many years and research for ENISA and work funded by the Dutch government on how to model software security.
- CRE builds on the **ENISA recommendation** to create a repository that brings standards and guidelines together, from a recent ENISA report on the security standard landscape.
- Currently, CRE is **being implemented** at OWASP, to provide for more consistency, more clarity and easier development and maintenance of the standards and guidelines at OWASP. The initiative is now embedded in the OWASP *integration standards* project.
- The idea of CRE is to **link** each small section of a resource to a shared topic identifier(a *Common Requirement*), instead of linking to just 1 or 2 other resources. Through this shared link, all resources map to eachother. This 1) enables standard and guideline makers to work efficiently, 2) it enables users to find the information they need, and 3) it facilitates a shared understanding in the industry of what cyber security is. The key element is **self-maintainability**: the CRE topic links in the standards and guidelines can be automatically parsed and used to map everything together. No more manual mapping.
- Additionally, the CRE repository of topics is a place to keep **metadata** on requirements, such as hierarchy, usage and discussions.
- The CRE initiative has **no commercial** intentions whatsoever. It is by the community, for the community. Furthermore, the CRE does NOT introduce a new standard, as the content is still maintained by the linked standards and guidelines.

Presenter introduction

Rob van der Veer, representing the CRE initiative with other co-leads: Spyros Gasteratos and Elie Saad

r.vanderveer@sig.eu

@robvanderveer

+31 6 20437187

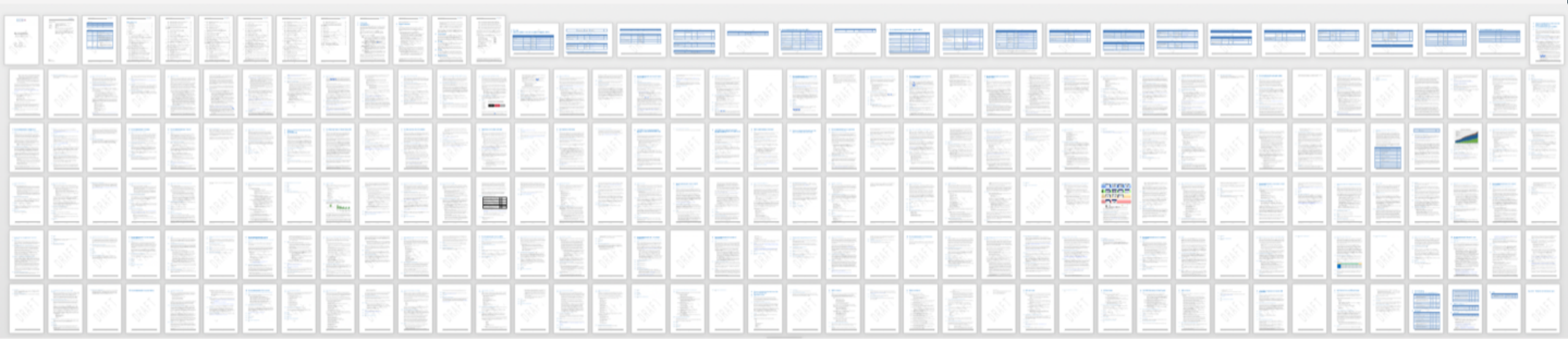
www.sig.eu/security

- Estabished and leads the security & privacy practice at Software Improvement Group
- Advisor to ENISA. Co-author of report ‘Advancing software security in the EU’
- Project leader of government-funded research on security requirements
- Project leader at OWASP (Co-lead of the *Integration standards* project)
- Contributor to various standardization initiatives: CIP (Grip on SSD), OWASP (SAMM), NCSC, IEEE, ISO/IEC

CRE

Challenge: the security standard landscape is a puzzle

The complexity of the security standard landscape is illustrated by the size of ECSO's *Overview of existing Cybersecurity standards* (2018) >200 pages:



The current standard and guideline landscape for security is **fragmented, complex and confusing** to many of its users. It is hard for engineers, testers and clients to select and apply appropriate standards, causing cyber security to stay behind.

Also, it is hard for **standard makers** to develop and to maintain their publications, link in a maintainabile way to other sources and to attain successful adoption.

Initiatives can benefit from **each other's content**, to save work in development and maintenance, but also to attain more consistency.

CRE

Security standards and guidelines are connected

OWASP ASVS 4.02

#	Description
6.1.1	Verify that regulated private data is stored encrypted while at rest, such as Personally Identifiable Information (PII), sensitive personal information, or data assessed likely to be subject to EU's GDPR.

More info on encryption

NIST SP800-53 rev.5, SC-12

More info on how to test this

OWASP Web Service Testing Guide 4.0, CRYP-04

CRE

Security standards and guidelines are connected

OWASP ASVS 4.02

#	Description
6.1.1	Verify that regulated private data is stored encrypted while at rest, such as Personally Identifiable Information (PII), sensitive personal information, or data assessed likely to be subject to EU's GDPR.

More info on encryption

NIST SP800-53 rev.5

SC-12 **CRYPTOGRAPHIC KEY ESTABLISHMENT AND MANAGEMENT**

Control: Establish and manage cryptographic keys when cryptography is employed within the system in accordance with the following key management requirements: [*Assignment: organization-defined requirements for key generation, distribution, storage, access, and destruction*].

Discussion: Cryptographic key management and establishment can be performed using manual procedures or automated mechanisms with supporting manual procedures. Organizations define key management requirements in accordance with applicable laws, executive orders, directives, regulations, policies, standards, and guidelines and specify appropriate options, parameters, and

.....
....
etc.

More info on how to test this

OWASP Web Service Testing Guide 4.0

Testing for Weak Encryption

ID
WSTG-CRYP-04

Summary

Incorrect uses of encryption algorithms may result in sensitive data exposure, key leakage, broken authentication, insecure session, and spoofing attacks. There are some encryption or hash algorithms known to be weak and are not suggested for use such as MD5 and RC4.

.....
....
etc.

The advantages and challenges of connections

Advantages:

Connections help **find more information** on a topic in other sources. Without them, documents need to cover that information themselves, typically briefly, typically without the right expertise and typically quickly outdated. This leads to inconsistencies, incompleteness and much work.

Connections help to attain **consensus** in the industry.

Challenges:

Connections get **outdated** quickly as other resources get restructured and updated.

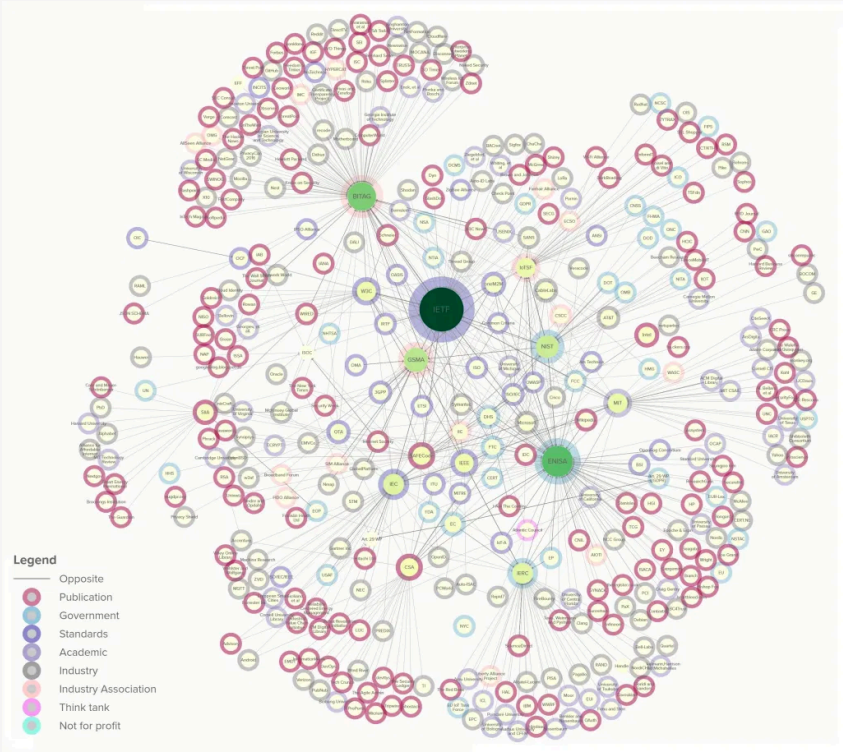
Finding the right connections is cumbersome, error-prone and takes up space, which is why connections are typicaly made with **only 1 or 2 sources**.

Wouldn’t it be great if all standards and guidelines were connected at the level of individual topics and automatically kept up to date?

Example of challenges 1: iotsecuritymapping.uk

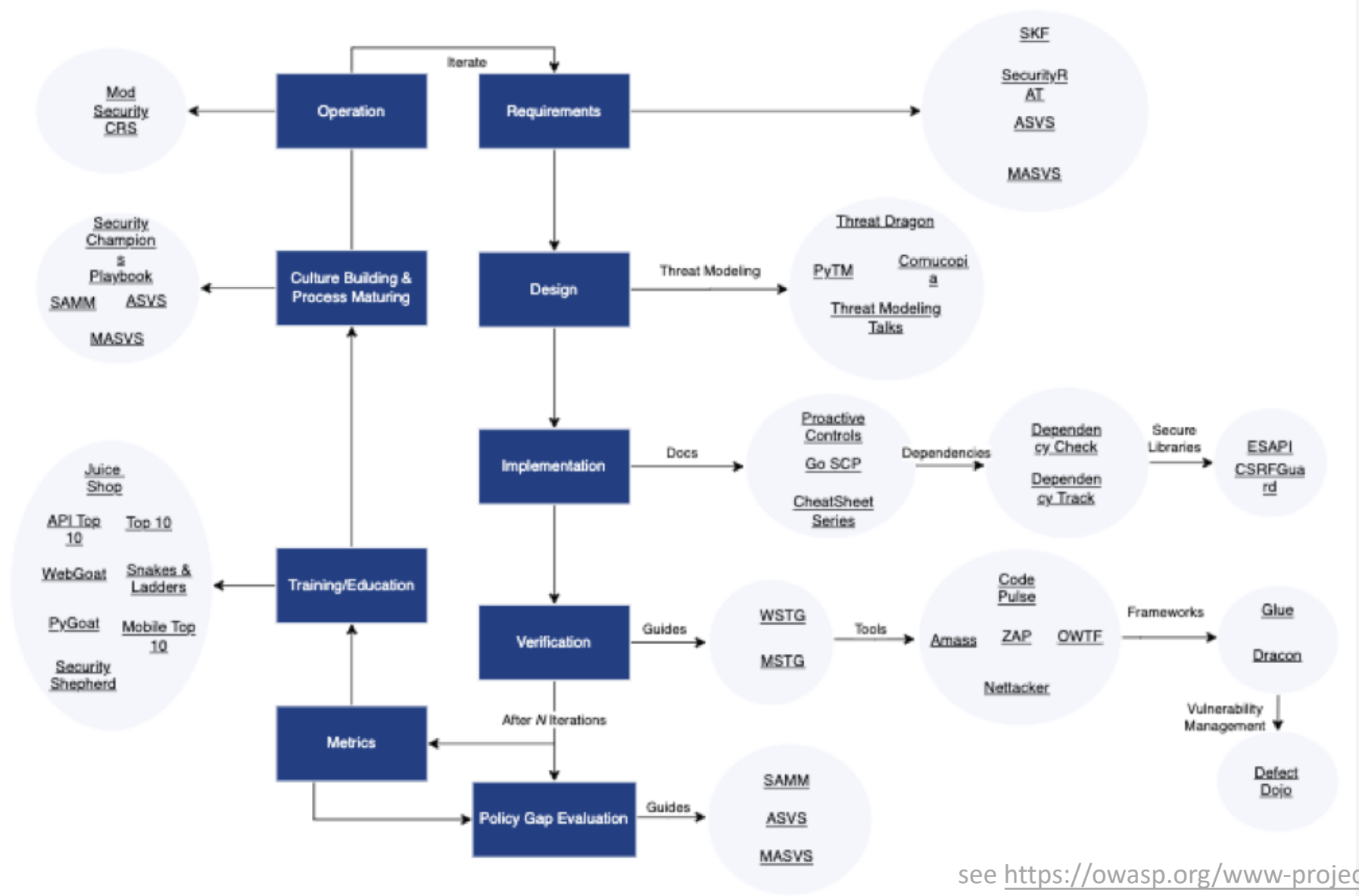
In an effort to make sense of IoT security recommendations and standards, a UK initiative has been manually mapping about a hundred sources. At the moment that one of these sources is updated, the mapping is outdated.

The mapping is stored in about a thousand pages of JSON specifications. It is a useful effort but **extremely hard to maintain** – let alone if the scope would need to be more general and extended beyond IOT.



Example of challenges 2: the great work of OWASP is a puzzle as well …

That's why we created the *Application security Wayfinder. Next: link them on detail level*



see https://owasp.org/www-project-integration-standards/

It is time to harmonize security standards

ENISA report:

“Requirements largely overlap, demonstrating that software security is mainly a generic problem and both Standards Developing Organizations (SDOs) and European Standards Organizations (ESOs) or good practice producers are often working without proper coordination and effective liaisons “

“DEVELOP A COMMON REPOSITORY FOR SHARED SECURITY MEASURES”

“Aligning on requirement commonalities across different schemes prevents proliferation and fragmentation, while also making drafting and maintaining a scheme more efficient in terms of mitigating the risks."

CRE

Enter: the Common Requirement Enumeration (CRE)

After extensive research and interviews with standard makers, procurement, industry, academia, engineers, testers and certification bodies, the idea for CRE was born, and it is now in the implementation phase.

Goal: enable alignment and cross-reference between security standards and guidelines, to:

- Attain shared understanding in market and industry on what security means (engineers, testers, and procurement).
- Make development and maintenance of standards and schemes easier
- Make it easier to find relevant work based on the context
- Achieve more consistency and less gaps between standards

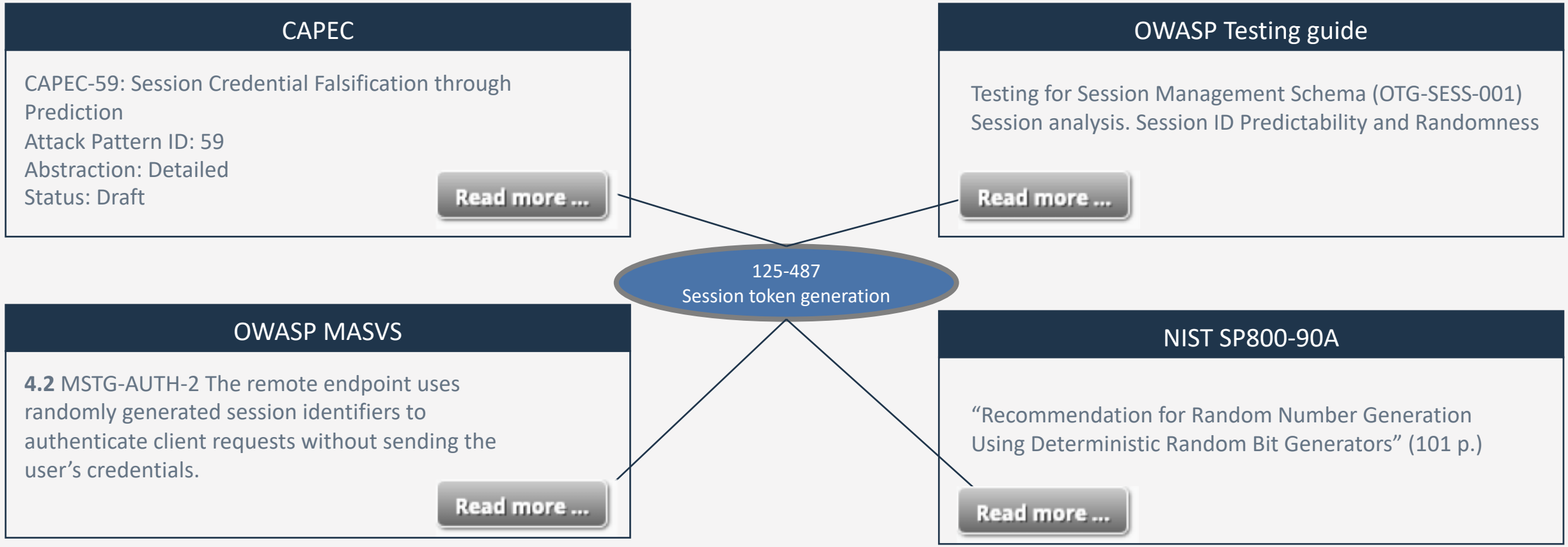
Method: provide a central repository of Common Requirement identifiers, to which standards can link their coverage of that topic, and by doing so, link to all the other relevant sources in other standards that refer to the same – and vice versa.

Deliverables:

- A r**epository** of technical requirement links in the form of an online service, offering just an identifier, metadata and links. The content is in the standards.
- Policy, guidelines and organization how the repository **changes over time**
- Tools and methods for standard makers to **link** using the repository

CRE

Common Requirement Enumeration example



Each topic in a standard links to the corresponding Common Requirement identifier and by doing so, standards link to each other. This allows readers to find all the information they need on a topic, as if they were using one single source – without links becoming outdated. For standard makers, maintenance, focus and consistency becomes easier, and a larger audience can be reached. For the industry, it becomes easier to define sets of requirements for different domains and types of systems, which allows procurement, engineering and testers to use one language in consensus when dealing with criteria for cyber security.

How are users referred?

OWASP MASVS

4.2 MSTG-AUTH-2 If stateful session management is used, the remote endpoint uses randomly generated session identifiers to authenticate client requests without sending the user's credentials. CRE

This section of a standard shows a CRE link.

When clicked, a pop-up menu is shown, or the user is taken to the CRE website.

CRE12459
Appsec-Session "Session management"
-Proactive Controls

CRE12452 -Appsec-Session-TokenGeneration
-ASVS 145, 146
-WSTG
-NIST
-CWE

CRE12451 -Appsec-Session-Removal
-ASVS 141
-WSTG
-NIST
-CWE

The CRE website shows the name of the requirement, links To relevant sources, and (through metadata) related requirements, including those on a higher level (in this case *session management*) with links to sources that cover that topic.

This allows multi-level browsing and exploration of requirements.

Standard users can find the right information and standard makers can provide access to their work, and they do not need to elaborate – just link.

CRE-linking is self-maintaining

CAPEC

CAPEC-59: Session Credential Falsification through Prediction
Attack Pattern ID: 59
Abstraction: Detailed
Status: Draft

125-487

OWASP Testing guide

Testing for Session Management Schema (OTG-SESS-001)
Session analysis. Session ID Predictability and Randomness

125-487

OWASP MASVS

4.2 MSTG-AUTH-2 If stateful session management is used, the remote endpoint uses randomly generated session identifiers to authenticate client requests without sending the user credentials.

125-487

NIST SP800-90A

"Recommendation for Random Number Generation Using Deterministic Random Bit Generators" (101 p.)

125-487

By storing the CRE-identifier IN the standards, the mapping become self-maintaining. It is updated constantly without new effort.

How does the self-maintenance work?

CRE supports two models of maintaining links:

1. A **mapping file** for a standard that contains the CRE identifiers that are covered with the hyperlinks to where they are covered. This can be maintained by a third party (e.g. the CRE team) and ideally by the standard maker.
2. **Embedded mapping**: the source files of the standard contain the CRE links which are scanned by the CRE parser to automatically create the mapping file. This approach makes things completely self-maintaining.

Example:

OWASP MASVS
** 4.2MSTG-AUTH-2 ** If stateful session management is used, the remote endpoint uses randomly generated session identifiers to authenticate client requests without sending the user's credentials. Read more

The CRE parser is configured to scan for CRE references (last line) and then register the CRE identifier and link that to the first section before that link with the class "ruletitle" and pick the corresponding ID (first line).

In this example this leads to the following entry in the mapping file:

125-487, http://www.owaspmasvs.org/rules.html#rule123

More advanced features

Reference flexibility:

- Using the CRE mechanism it is also possible to refer directly to a specific source, which is automatically kept up to date.
- Similarly the CRE link can control how the information is presented (e.g. OWASP sources first)

Furthermore, users can manage their own account or sessions on the CRE website and specify what sources they prefer to see.

Intel: The (anonymous) use of CRE leads to interesting insights into the use of standards and specific topics.

Search: while parsing all source files, an index may be built to allow ’federated’ search over all sources.

Map/gap analysis: using CRE it is easy to map standard X to standard Y to verify compliance across standards, and also to find gaps.

CRE roadmap

1. Create Application security wayfinder as a first service to the community
2. In progress: build first CRE system to harmonize the flagship standards at OWASP:
   https://owasp.org/www-project-integration-standards/
   Co-leaders: Elie-Saad, Spyros Gasteratos, Rob van der Veer
3. Take CRE implementation outside of OWASP and arrange independent governance
4. In progress: seek alignment with standard makers and other stakeholders
5. Process new ideas and learning points
6. Establish a consortium to govern CRE
7. Further connect more standards using CRE

Appendix: Introducing higher level concepts iN CRE

CRE 273-300 is titled “Encrypting personal data at rest” it was inspired by ASVS 6.1.1

ASVS 6.1.1 which is linked with link type “Same” to CRE 273-300 says: "Verify that regulated private data is stored encrypted while at rest, such as personally identifiable information (PII), sensitive personal information, or data assessed likely to be subject to EU's GDPR."

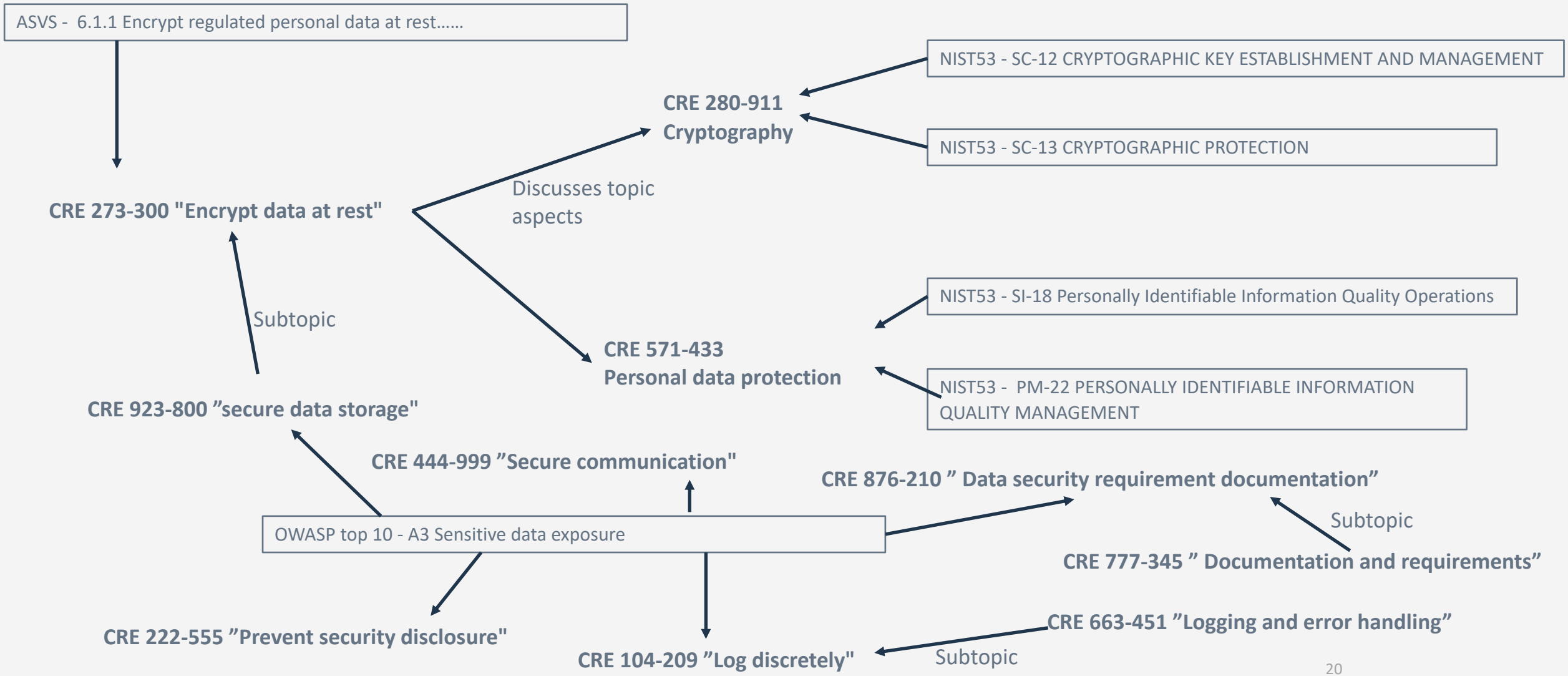
Related segments of NIST SP800-53, are:

-SI-18 Personally Identifiable Information Quality Operations,
-PM-22 PERSONALLY IDENTIFIABLE INFORMATION QUALITY MANAGEMENT,
-SC-12 CRYPTOGRAPHIC KEY ESTABLISHMENT AND MANAGEMENT,
-SC-13 CRYPTOGRAPHIC PROTECTION

It is unworkable if we would specify these relations by mapping directly from these NIST items to 273-300. For example SP800-53 SC-13 would then have to contain a link to CRE 273-300 and numerous other CRE's that are related to cryptography - also beyond OWASP. That would be unmaintainable, and a task too difficult for most standard makers.

This shows that we need higher-level abstract CRE topics that we can link these NIST topics to, and within the CRE administration define relations between those CRE's.

How higher level concepts connect topics



How we deal with higher level concepts in mapping and use cases

What would be needed for the above example is:

- A new CRE ID 571-433 for the topic Personal data protection
- A new CRE ID 280-911 for topic of Encryption
- Linking CRE 273-300 to both 280-911 and 571-433 with a link type 'discusses topic aspects'
- Linking 571-433 to NIST SI-18 and PM-22 with link type: 'discusses topic aspects"
- Linking 280-911 to NIST SC-12 and SC-13: 'discusses topic aspects"

Resulting in the following CRE reference pages for the user:

CRE 273-300 "Encrypt personal data at rest"

Same
-ASVS V6.1.1

Similar:
-CWE-311: (Missing Encryption of Sensitive Data)

Is part of:
-OWASP Top 10 A3 (Sensitive data exposure)

Discusses topic aspects:
-OWASP WSTG – CRYP-04 (Testing for weak encryption)
-CRE Encryption
-CRE Personal data protection

CRE 280-911 "Encryption"

Discusses topic aspects:
-NIST SC-12
-NISC SC-13

CRE 571-433 ”Personal data protection"

Discusses topic aspects:
-NIST SI-18
-NISC PM-22

CRE

Example CRE links for OWASP Top 10 A3-Sensitive Data Exposure

* Store passwords using strong adaptive and salted hashing functions with a work factor (delay factor), such as Argon2, scrypt, bcrypt or PBKDF2.
* Verify independently the effectiveness of configuration and settings.

References

OWASP

* OWASP Proactive Controls: Protect Data Everywhere
* OWASP Application Security Verification Standard (V7, 9, 10)
* OWASP Cheat Sheet: Transport Layer Protection
* OWASP Cheat Sheet: User Privacy Protection
* OWASP Cheat Sheet: Password and Cryptographic Storage
* OWASP Cheat Sheet: HSTS
* OWASP Testing Guide: Testing for weak cryptography

External

* CWE-202: Exposure of sens. information through data queries
* CWE-310: Cryptographic Issues
* CWE-311: Missing Encryption
* CWE-312: Cleartext Storage of Sensitive Informatio
* CWE-319: Cleartext Transmission of Sensitive Infor
* CWE-326: Weak Encryption
* CWE-327: Broken/Risky Crypto
* CWE-359: Exposure of Private Information (Privacy

* Store passwords using strong adaptive and salted hashing functions with a work factor (delay factor), such as Argon2, scrypt, bcrypt or PBKDF2.
* Verify independently the effectiveness of configuration and settings.

References

CRE

- Encrypt data at rest
- Catch exceptions safely and completely
- Cryptographic fail securely

Data protection – encrypt data at rest

- Pro-active controls C8: Protect Data Everywhere

- NIST SP800-53 SC-12 CRYPTOGRAPHIC KEY ESTABLISHMENT AND MANAGEMENT
- NIST SP800-53 SC-13 CRYPTOGRAPHIC PROTECTION
- NISTSP800-53 SI-18 Personally Identifiable Information Quality Operations
- NISTSP800-53 PM-22 PERSONALLY IDENTIFIABLE INFORMATION QUALITY MANAGEMENT,

- ASVS V6.2.2 Verify that industry proven or government approved cryptographic algorithms, modes, and libraries are used, instead of custom coded cryptography. (C8)
- ASVS V6.2.3 Verify that encryption initialization vector, cipher configuration, and block modes are configured securely using the latest advice.
- ASVS V6.2.4 Verify that random number, encryption or hashing algorithms, key lengths, rounds, ciphers or modes, can be reconfigured, upgraded, or swapped at any time, to protect against cryptographic breaks. (C8)
- ASVS V6.2.5 Verify that known insecure block modes (i.e. ECB, etc.), padding modes (i.e. PKCS#1 v1.5, etc.), ciphers with small block sizes (i.e. Triple-DES, Blowfish, etc.), and weak hashing algorithms (i.e. MD5, SHA1, etc.) are not used unless required for backwards compatibility.
- ASVS V6.2.6 Verify that nonces, initialization vectors, and other single use numbers must not be used more than once with a given encryption key. The method of generation must be appropriate for the algorithm being used.
- ASVS V6.2.7 Verify that encrypted data is authenticated via signatures, authenticated cipher modes, or HMAC to ensure that ciphertext is not altered by an unauthorized party.
- ASVS V6.2.8 Verify that all cryptographic operations are constant-time, with no 'short-circuit' operations in comparisons, calculations, or returns, to avoid leaking information.
- ASVS V6.3.1 Verify that all random numbers, random file names, random GUIDs, and random strings are generated using the cryptographic module's approved cryptographically secure random number generator when these random values are intended to be not guessable by an attacker.
- ASVS V6.3.2 Verify that random GUIDs are created using the GUID v4 algorithm, and a cryptographically-secure pseudo-random number generator (CSPRNG). GUIDs created using other pseudo-random number generators may be predictable.
- ASVS V6.3.3 Verify that random numbers are created with proper entropy even when the application is under heavy load, or that the application degrades gracefully in such circumstances.
- ASVS V6.1.1 Verify that regulated private data is stored encrypted while at rest, such as personally identifiable information (PII), sensitive personal information, or data assessed likely to be subject to EU's GDPR.
- ASVS V6.1.2 Verify that regulated health data is stored encrypted while at rest, such as medical records, medical device details, or de-anonymized research records.
- ASVS V6.1.3 Verify that regulated financial data is stored encrypted while at rest, such as financial accounts, defaults or credit history, tax records, pay history, beneficiaries, or de-anonymized market or research records.

Part of:

- OWASP Top 10 – A3 Sensitive data exposure