STITCH 1.0 English

English version of STITCH 1.0.

There is an increasing need for software and services to comply with security requirements. With many different lists of requirements, which one do you choose? SCIRT, the community for cybersecurity, has developed a simplified checklist called Security Technical IT CHecklist (STITCH).

The idea behind STITCH is simple: a baseline with a limited number of requirements that are easy to measure. The results enable Security Officers and other security experts to quickly assess whether a service or an application is safe.

The 8 STITCH items:

- STITCH-1) All data in transit is encrypted
- STITCH-2) The identity of users has been vetted and identities are managed federatively
- STITCH-3) Authorisation is based on separation of duties and least privilege
- STITCH-4) Secure session management is in place
- STITCH-5) All input and output data is normalised, validated, and limited
- STITCH-6) Avoid leaking of configuration information
- STITCH-7) Systems have sufficient logging and auditing capabilities
- STITCH-8) Continuous maintenance and patch management is in place

STITCH-1) All data in transit is encrypted

The confidentiality, integrity and non-repudiation of data and transactions must be ensured.

Risks:

Unauthorised access to services and view/change data.

Implications:

- Data must be transported using standard protocols and encrypted using up-to-date encryption standards.
- If the chosen transport protocol does not support encryption, the data itself must be encrypted.
- Data sent on physical media such as USB sticks must be encrypted using up-to-date encryption standards.
- Certificate management must be set-up and configured.

Testing:

The SSL Labs server test at https://www.ssllabs.com/ssltest/ must yield at least an A score. For non internet facing systems SSL Labs offers a standalone tool.

Map all data transfers and check the use of up-to-date encryption standards.

In case of IMAP, POP3, NNTP and LDAP check if the STARTTLS command is being sent after the initial unencrypted connection has been set up. For more information see: https://en.wikipedia.org/wiki/Opportunistic_TLS.

Check that Forward Secrecy is being used, ensuring continuing confidentiality in case of a key file breach. This can be checked with the SSL Labs server test as well.

References:

- SSD: 4
- OWASP: https://cheatsheetseries.owasp.org/cheatsheets/Transport_Layer_Protection_Cheat_Sheet.html
- ISO 27002:2013 13.1.1 13.2.1 14.1.2

STITCH-2) The identity of users has been vetted and identities are managed federatively

Institutions must check the identities of users, and applications shall work with federated identities. This ensures that actions can be traced back to individual persons.

Risks:

Abuse of identities or systems that cannot be traced back to individual persons. Consequences can include unauthorised information disclosure or modification, or falsely executing transactions on behalf of a valid user.

Implications:

- Web applications use the identity provider of an institution or SURFconext for authentication and identification purposes.
- Authentication happens federatively, ensuring passwords never leave the institution.
- A lock-out mechanism is put in place to prevent password brute-forcing.

Testing:

Ensure no default accounts using default passwords are present and no default SNMP community strings are in use.

Check that revoking authorisation in the central IDM system also means that the user no longer has access to the application.

Check that entering a wrong password leads to blocked access or blocked accounts after a limited number of attempts.

References

- SSD: 6
- ISO 27002:2013 18.1.1 9.2.4 9.2.1 9.2.6 9.4.2

STITCH-3) Authorisation is based on separation of duties and least privilege

Applications must apply separation of duties. Roles are defined based on tasks, responsibilities and privileges. Extra attention must be paid to accounts with the highest privileges.

Risks:

A lack of separation of duties can lead to fraud or misuse of organisational resources. If a user's authorisation level is higher than it needs to be for a given task, a compromise of the account can result in unnecessary damage.

Implications:

The application supports role based access management, which is also configured.

Testing:

Check if a newly created account has only basic authorisation. Add roles and check if the resulting authorisations on them are in line with the intended ones.

References:

- SSD: 7,8
- ISO 27002:2013: 6.1.2 9.1.2 9.2.2 9.4.1

STITCH-4) Secure session management is in place

Exploiting existing sessions should be countered by appropriate configuration of session management. This could be done by adding Security Headers in the case of HTTP. Other protocols such as SMTP, IMAP, POP3 and SSH require different ways of securing.

Risks:

Exploiting existing sessions, for example session hijacking and Man-In-The-Middle attacks.

Implications:

- Use HTTP headers, see https://securityheaders.com.
- Set up session management, see https://www.owasp.org/index.php/Session_Management_Cheat_Sheet
- Use session cookies with the 'HttpOnly' en 'Secure' flags set.

Testing:

Use the OWASP Session Management Cheat Sheet and the corresponding methods for testing. This document explains how to properly set up session management. Make sure to pay attention to topics such as transport security and session duration. See https://www.owasp.org/index.php/Testing_for_Session_Management_Schema_(OTG-SESS-001).

Use the online tools from https://securityheaders.com/ to check the HTTP security headers.

Make sure session cookies have the ‘HttpOnly’ and 'secure' flags.

Check that session termination is in effect.

References:

- SSD: 12
- ISO 27002:2013: 11.2.8 9.4.2
- W3C: https://www.w3.org/TR/CSP3/

STITCH-5) All input and output data is normalised, validated, and limited

Data integrity must be ensured. Many attacks are the result of incorrect or forged data put into applications.

Risks:

Not normalising, validating, or limiting input and output data will increase the chance of vulnerabilities getting successfully exploited. This can then lead to a breach in data integrity and confidentiality.

Spreading of malware by input and output of infected content.

Implications:

Normalisation, validation, and limitation should be applied during input and output.

If necessary, content should be converted into a format that can not cause damage.

Testing:

Use a vulnerability scanner to check for vulnerabilities such as SQL injection and Cross Site Scripting (XSS). Use existing information, for instance previous pentesting reports, or conduct a pentest.

Examples of vulnerabilities are:

- Input fields that allow injection attacks
- Including files from remote sites that can be manipulated by third parties
- Files that might contain malicious functionality such as macros
- Web servers that offer open directory listings

References:

- SSD: 18, 19, 20, 21, 22, 23, 29
- OWASP: https://www.owasp.org/index.php/Data_Validation
- ISO 27002:2013 14.2.1

STITCH-6) Avoid leaking of configuration information

Leaking of configuration information in headers, banners, and error pages must be avoided.

Risks:

Such information can be used to find out server and software versions, application configuration details, etc.

Implications:

- Banners and headers should not leak configuration information.
- Error pages should not leak configuration information (such as stack traces, debugging output, etc).
- Comments in code should not be accessible by end users.

Testing:

- Use vulnerability scanner tools to do banner grabbing and finger printing.
- Check the content of error pages.
- Check for database dumps, backups, etc. that are inadvertently available publicly.
- Check client side code for comments.
- Make sure files are not directly accessible if they are not supposed to - either by hand or through tools such as URL fuzzers.

References:

- SSD: 2, 24, 25, 26, 27
- RFC7762, RFC7508

STITCH-7) Systems have sufficient logging and auditing capabilities

Systems must have sufficient logging and auditing capabilities.

System logging is necessary for day-to-day administration and monitoring. This includes the logging of error conditions.

System auditability, through audit logging, is needed to ensure integrity and confidentiality of data. This is also needed to trace back any activity in the application, when it was done, and by whom.

Risks:

Regarding logging: application defects and vulnerabilities can not be detected, and fixes for them can not be applied in time.

Regarding auditing: security incidents can not be dealt with, and there is a lack of evidence.

Implications:

- Logging and auditing data are stored.
- Logging and auditing is secured against unauthorised access.
- Logging and auditing information contains sufficient details to trace back incidents to natural persons.

Testing:

Check logging and auditing for the following items:

- There is sufficient information for administration tasks and fault finding.
- Auditing can be traced back to a person.
- Log data is being stored to local files, or to a central logging server.

References:

- SSD: 9, 13, 27, 30
- ISO 27002:2013: 12.4.1.12.4.3

STITCH-8) Continuous maintenance and patch management is in place

Vulnerabilities will be discovered in applications. To ensure safe use of applications they must be maintained continuously, and patches must be developed and applied.

Risks:

Applications that are not actively maintained will have no patches being developed for them. Applications that are behind on their patching schedule may contain vulnerabilities.

Implications:

- Patch management must be in place.
- New and relevant vulnerabilities (for instance identified by their corresponding CVE numbers) must be addressed in time by the developers.
- End-of-life applications must not be used.

Testing:

- Ensure that the application is actively being developed, by checking for recent patches, documentation, or the existence of an active community.
- Ensure patches have recently been released and applied, for instance by using a vulnerability scanner.

References:

- ISO 27002:2013: 12.5.1 12.6.1 14.2.2
- SSD 1

Sources

- ISO: International Organization for Standardization - https://www.iso.org
- SSD: Grip op Secure Software Development - https://www.cip-overheid.nl/wp-content/uploads/2018/01/Grip-op-SSD-Beveiligingseisen-v2_0.pdf
- OWASP: Open Web Application Security Project - https://www.owasp.org
- RFC: Request For Comments - https://www.ietf.org/standards/rfcs/
- W3C: World Wide Web Consortium (W3C) - https://www.w3.org
- SSL Labs: SSL Labs server scan: - https://www.ssllabs.com
- Securityheaders.io: Security headers scan - https://www.securityheaders.io

Prerequisites

The STITCH requirements assume a number of prerequisites to be met in areas of information security and privacy protection. These include:

Legal:

- Compliance with GDPR
- Use of standards that are on the "Comply or explain" list of the Dutch Standardisation Forum.

Organisational:

- The organisation's architectural design principles should be taken into account.
- All data should be classified according to the existing guidelines.
- The organisation's information security policy should be applied.
- Shadow-IT, islands of automation, and ad-hoc solutions should be avoided. Use existing services that are available in the organisation.
- Adhere to existing conventions for DNS names, system names, etc.