Information security training

for developers

2025

Table of Contents

1. IT security basics

1.1. The basics of information security

What is security?

Data security refers to data stored, transmitted or processed in a secure manner on information systems.

IT security is the favourable state of IT systems in which the confidentiality, integrity and availability of the data handled are ensured for the elements of the system (the so-called "CIA principle"):

- *confidentiality:* only those authorised have access to the information;
- *integrity:* the form and content of the information is as expected and the person who handles the information can be clearly identified;
- *availability:* the state when the services of an IT system are available to the authorised persons at a specified time, and the expected operation of the system is not temporarily or permanently obstructed.

Information security is a much broader concept than IT security, as it includes all forms of information (not just electronic), as well as the security of services.

The main objective of Act L of 2013 on the Electronic Information Security of State and Local Government Bodies (hereinafter referred to as the Information Security Act) is to provide a framework for the protection of public administration bodies, critical system elements, and local governments.

1.2. Legal basis

DORA Regulation

The Digital Operational Resilience Act (DORA) is an EU regulation that entered into force on 16 January 2023 and is applicable from 17 January 2025. It aims to strengthen the IT security of the financial sector to ensure that financial institutions remain resilient in the event of a major disruption.

The following pillars ensure the digital operational resilience of the financial sector:

1. **ICT risk management:** Financial institutions are required to develop a comprehensive framework for identifying, assessing, managing, and mitigating IT risks.
2. **Incident reporting**: Mechanisms must be put in place to report significant IT incidents to regulators in a timely manner, including detailed documentation and analysis of incidents.
3. **Digital operational resilience testing**: Regular testing must be carried out to ensure digital operational resilience, including basic and advanced tests.
4. **Third-party risk management**: Financial institutions need to manage the risks associated with third-party service providers, including key contractual provisions.
5. **Information sharing**: Exchange of information and intelligence on cyber threats between actors of the financial sector.

MNB Recommendation 1/2025 on IT system security

The aim of the Recommendation is to provide practical guidance to members of the financial intermediary system on how to design the protection of their IT systems in a risk-proportionate manner and to ensure a consistent interpretation of the application of the legal provisions on their protection. The Recommendation is applicable in conjunction with MNB Recommendation 2/2025 on the use of community and public cloud services, the Management Circular on written contracts and written declarations made electronically, MNB Recommendation 12/2022 (VIII. 11.) on the establishment and operation of internal security lines, on the management and control functions of financial institutions, and the MNB Recommendation on the use of external service providers.

1.3. General attack techniques based on the MITRE ATT&CK Framework

A good starting point for learning about the anatomy of information security and cybersecurity attack techniques is the MITRE ATT&CK framework.

The MITRE ATT&CK framework describes how attackers compromise IT networks and systems and how they carry out their activities by penetrating systems. The ATT&CK looks at attacks from the perspective of the attackers, what they are trying to achieve and what specific methods they use.

Like the original project, the current system is designed to better understand how attackers prepare for and execute attacks, which helps in planning the prevention and detection of attacks. The knowledge gathered in the ATT&CK framework can also be used to assess the protection capability, as well as to identify and strengthen weak points.

Tactics are the result that attackers hope to achieve through their actions. For corporate networks and mobile devices, the tactics and high-level targets used by attackers are similar.

A technique is a method of successfully carrying out a “tactic” – usually a single step in a series of actions taken by the attacker to achieve their mission. The ATT&CK contains, among many other things, descriptions, examples, references and suggestions for each technique to prevent and detect the particular attack method.

1.4. Most commonly used general attack techniques

Unauthorised access

Many factors can allow an attacker to gain unauthorised access. For example, a poorly configured web server or a website with poor code quality (which does not properly manage access privileges or does not perform some kind of input validation) can become the target of an attack.

To avoid this, it is important that configurations, web pages and other custom code are checked not only for functionality, but also for security. An adequate level of logging is also essential, either to detect or to decrypt a security incident.

Malware

Malware is a generic term that covers all malicious code designed to attack. Examples include viruses, worms, spyware, ransomware, and adware.

The most common types of malicious code today include:

- ***Infostealer software***

These software were designed to steal all kinds of passwords stored in programmes installed on the victim's computer. Infostealer software target email clients, web browsers, or even data transfer clients (such as FTP clients).

In addition, infostealer codes can also perform other functions, such as capturing keystrokes and screenshots, or even stealing bitcoin wallets.

- ***Trojans or RATs (Remote Access Trojans)***

  They are usually disguised as legitimate programmes, lying dormant after infection until instructions are received from the attacker.

  - Generally, they are capable of executing the following actions:
  - Establishing connection to a botnet (e.g. for spamming or Bitcoin mining)
  - Electronic money theft
  - Data theft
  - Installation of other (harmful) software
  - File operations (create, modify, delete)
  - Screen monitoring, capturing screenshots
  - Monitoring, controlling, recording and taking photos with a webcam
  - Remote control
  - Running a proxy server
  - Banking trojans (Their main target is to obtain bank passwords and credit card details)

- ***Ransomware***

The purpose of these software is to obtain financial gain in the form of a “ransom”. They encrypt the files on the victim's computer and then ask for money to unlock them, usually through untraceable means such as cryptocurrency.

- ***Macro viruses***

These are malicious codes written in macro language that run when executed in other programmes. Most often, macro viruses arrive on the victim's device embedded in MS Office documents. They are often used as an element of a complex infection chain. For example, an MS Word document infected with a macro virus is opened from a phishing email, which downloads an infostealer or ransomware code from a remote device controlled by the attacker.

Social Engineering

Social engineering (manipulation) is an attack where an authorised user provides an unauthorised person with confidential information or the opportunity to gain access to the system due to the deceptive behaviour of the attacker. The most common forms of social engineering include:

- ***Phishing***

  In this case, the attacker poses as a trustworthy partner to obtain confidential information by e-mail or on a website. The attacker may target a variety of information, such as usernames, passwords, credit card numbers, bank account details, etc. The message invites the user to log on to a website that looks very similar to a legitimate website (PayPal, eBay, a well-known bank, etc.) and provide confidential information through that website, which is, in fact, run by the attacker.

- ***Vishing***

  The essence of vishing is that the attacker contacts the victim via some form of voice transmission technology (phone, web voice call, etc.). The caller is usually uses this method to obtain credit card information or other information suitable for identity theft. Vishing can be done by contacting the user by email and asking them to call the phone number provided.

- ***Pharming***

  Pharming involves an attacker redirecting users to fake sites that appear to be real sites through redirecting network traffic. As a result, when the user types in a web address believed to be genuine, the system redirects to a fake site. The attackers then either try to gain access to personal and financial data or infect the device with additional malicious code.

- ***Business e-mail compromise (BEC):***

  In this type of attack, attackers send emails to request electronic transfers or other information. These attackers often gain access to an executive's email or impersonate executives via spoofed email. Attackers scout potential targets and organisations in advance, and mask their identity from employees through emails that look like were sent by CEOs or other executives.

Misuse

Unintentional human activity that does not comply with a company's internal policies or regulations, for example:

- unauthorised file copying (copying trade secret data to an external mobile storage device);
- downloading and installing unapproved software;
- unauthorised use of music or other media;
- P2P file sharing (Torrent);
- use of remote access applications (e.g. AnyDesk, Teamviewer);
- use of a company e-mail account for private purposes;
- personal web browsing from a corporate network.

2. Preventing hacker attacks

To prevent hacker attacks, organisations have a number of process and technology controls in place. In this chapter, we are going to discuss the most important techniques.

Access management

Access management refers to a set of security mechanisms that determine what users can do on the system, i.e. the resources they can access and the operations they can perform.

If a user account has been accessed by attackers, the damage can be minimised through access management.

The main principles used in access control:

- Separation/Segregation of Duties

  The goal is to assign different people to different steps in a process. This requires planning to ensure that no single person can control or manipulate the entire process.

- Least Privilege

  By respecting this principle, the system restricts access to resources for users and applications to only those that are necessary.

Network separation

The design of the network must ensure that each device can communicate only with those devices that are absolutely necessary for its functioning. Firewalls are one technological way to achieve segregation, while IDS and IPS systems are used to identify and possibly intervene in network traffic.

Firewall

A firewall is a network security system that monitors and controls incoming and outgoing network traffic based on predefined security rules. Typically, it forms a barrier between a trusted internal network and an untrusted external network.

As part of traffic control, firewalls can perform a number of functions, such as packet filtering, stateful filtering, application layer firewalling, and content filtering.

IDS

IDS is short for “Intrusion Detection System”. The main purpose and function of IDS is to identify suspicious or malicious activities on the network.

It has two types:

- Host-based IDS (HIDS), which is used to monitor the activity of a standalone system. Examples include a mail system, a web server, or an end-user computer. It only deals with its owner and has no contact with its environment.

- A network-based IDS (NIDS) that monitors the activity of a computer network. It usually monitors the traffic passing through the network switches. In a typical case, it does not know what is happening within the units belonging to the system.

IPS

IPS is short for “Intrusion Prevention System”. It works very similar to IDS, the difference being that IDS only detects the attack, while IPS also tries to block it.

(D)DOS attack protection

Denial of Service (DoS), also known as overload attack or Distributed Denial of Service (DDoS), is the complete or partial paralysis or diversion of an IT service from its correct mode of operation.

Firewalls can be used to protect against small-scale attacks, but they can be very expensive to use against wide bandwidth attacks, and therefore it is worth choosing an external service provider to prepare for this.

Antivirus software

Antivirus software are designed to detect malware and prevent them from doing harm.

Basically, there are two kinds of solution:

- Signature-based: These solutions represent older technology, and work by identifying malware based on their signature. They have the advantage of being less resource-intensive. The disadvantage is that it is relatively easy to modify a piece of malware so that it is not detected.
- Behaviour-based: This model represents a new, more advanced detection mechanism. Their mechanism of action is based on recognising and blocking viruses by monitoring for common, suspicious characteristics, such as certain sequences of API calls, creation of registry entries or new files in the file system, etc. It has the advantage of detecting more suspicious samples with fewer false alarms.

Security incident and event management systems

Security Incident and Event Management (SIEM) refers to the identification, monitoring, recording and analysis of security incidents or events in a real-time IT environment.

SIEM performs monitoring and detection by analysing the activity and behaviour of security-relevant log entries, network traffic, and user accounts.

Hardening

Hardening refers to procedures used to reduce system vulnerability. The following methods are commonly used to reduce the attack surface:

- changing default passwords;
- removing software that are not necessary for the functionality of the system;
- removing unnecessary accounts;
- disabling or removing unnecessary services, closing unnecessary open ports.

Hardening guides can be found here:

- NIST:
  - https://nvd.nist.gov/ncp/repository
  - https://csrc.nist.gov/publications/detail/sp/800-123/final

- CIS:
  - http://benchmarks.cisecurity.org

Configuration management

The purpose of configuration management is to regularly audit the configuration of services to detect misconfiguration.

Updating system elements

One of the most important tasks related to the security of operations is eliminating vulnerabilities in electronic information systems through security updates.

Penetration testing

Auditing one or more elements of an IT system (in part by manual means) to determine whether it has a known vulnerability that can be exploited in practice.

Data Leak Prevention (DLP)

DLP systems are designed to detect and prevent the unauthorised use and transmission of confidential information, both at the endpoints, the network, and the data storage devices.

Backup

Perhaps the most important measure is to back up critical data, especially data that cannot be accessed from the network in any form (cold backup). This can be useful, among other things, in the case of successful ransomware attacks.

3. Vulnerabilities

Vulnerability management aims to identify system vulnerabilities and reduce the associated risks by eliminating the vulnerabilities or applying compensatory controls.

3.1. Vulnerability analysis

Information about newly disclosed vulnerabilities are available from a variety of sources, such as CVE or NVD databases. CVE provides basic information, classified into categories and uniquely identified, about each vulnerability. Information from the CVE is used to build the NVD database, which enriches data about particular vulnerabilities with known patches, as well as information about severity and potential impact.

3.2. Vulnerability scan

Vulnerability scan is a technical way of identifying vulnerabilities affecting an organisation and can, therefore, be used as a complementary technique to vulnerability analysis. In large corporate environments, using vulnerability scans can be difficult for a number of reasons, such as:

- before the scan itself takes place, careful planning is needed to determine the appropriate timing of the test to minimise potential business impacts, as even passive scans can negatively affect the operation of systems,
- in a large corporate environment, it can take a long time (weeks or even months) to run a full scan due to the large size of the infrastructure. If the vulnerability scans are not complemented by vulnerability analysis, many vulnerabilities will go unnoticed for a long time between two scan cycles.
- In certain environments using critical or unique technologies, design and implementation requires additional care.

3.3. Prioritisation and classification of vulnerabilities

If a vulnerability is found to affect an information asset of the organisation, the related risks must be determined to allow the prioritisation of response and the implementation of compensating controls.

The purpose of vulnerability triage is to determine the timeframe for response.

Prioritisation should not be based solely on the severity of the vulnerability, but also take into account the criticality of the asset within the organisation. Therefore, the way the organisation prioritises vulnerability should take both into account.

So-called 0-day vulnerabilities, for which no patches exist yet, but have already been targeted by malicious codes (exploits), require special prioritisation because they sometimes receive too much attention compared to their real impact and risk exposure.

3.4. Reducing the impact of vulnerabilities

If an information asset is confirmed to be vulnerable, and the vulnerability is exploitable, it must be determined how to mitigate the impact of the vulnerability.

If a security update is available, the patch should be managed in accordance with the established priority. Otherwise, some form of compensatory control must be defined. If there is publicly available information on vulnerability mitigation (e.g. from the manufacturer of the vulnerable system), this can be used as a basis, but its applicability within the organisation must be checked in all cses.

Even when there is a security update available for the vulnerability, it may be necessary to define a compensating control. In a large corporate environment, patch management can present many challenges, such as:

- how a particular security update should be prioritised over other patches that have not yet been applied;
- what resources are involved in checking for the security update;
- how to find an approved time window to apply the security update;

- how to manage diverse environments, portable devices, virtualised environments, environments with non-standard IT components (e.g. appliance devices where per-component upgrades are not supported by the vendor).

4. Security issues related to encodings

Over the last few years, the threats to software have grown exponentially. Software and application vulnerabilities have caused a lot of damage to companies and people.

Secure coding standards and best practices enable developers to create more secure software. These standards ensure that developers write codes that are free from vulnerabilities that can be exploited by attackers.

Although there are many ways to develop secure applications, the Open Web Application Security Project (OWASP) has created a comprehensive checklist for secure coding. This list is mainly focused on web applications, but can be used as a security protocol in any software development cycle or software deployment platform to minimise the risks from bad coding practices.

The OWASP has compiled the following list of secure coding requirements, which includes a number of preventive techniques to minimise and manage the damage from different types of attacks.

Input control

Verification of inputs and data entered by the user or the application.

The check can be used to filter out parameters from the inputs that would cause the application to behave abnormally. Methods to check the input:

- Verification must be done on a reliable system.
- Check by data type, possible value and length, and comparing incoming input against allowed characters.
- Examine request headers and check that they contain only ASCII characters.
- Check all data coming from client/user, including URLs, HTTP headers, embedded code, etc.

Output encoding

Output encoding is the process of converting user input/unsafe data into a form that is safe from being executed by a browser or programme when the input is displayed. The use of output encoding prevents data from an unreliable source that is not understood by the interpreter from being executed.

Cross-site scripting can be prevented by using output encoding.

Techniques included in OWASP recommendations:

- Verification must be done on a reliable system.
- Encode all characters unless they are presumed safe for the interpreter.
- Clean up untrusted data with OS commands.

Authentication and password management

Authentication is the process of identifying the user. Password management is the enforcement of a set of rules and techniques for storing passwords. By following these rules, users can gain confidential access to certain critical assets. OWASP's recommended techniques for this include:

- Require authentication for any device, website, where confidentiality, integrity, availability matter.
- Refrain from reusing passwords.
- Notify users of password resets.
- Indicate the number of failed login attempts to the user on the next successful login.
- Set short expiry dates for temporary passwords and change them upon next login.
- Impose and enforce password complexity

Session Management

In services or web applications, the secure management of requests from multiple users is called session management. Related advice on OWASP:

- Complete termination of sessions and connections on exit.
- Allow only one login with one user ID.
- Taking into account the risks and business objectives, the inactivity time limit for a session should be as short as possible.

Cryptographic practices

Cryptographic operations are usually used to preserve confidentiality to ensure that only those users can access and modify sensitive data for whom this is absolutely essential. Advice on cryptographic practices:

- Cryptographic operations should be used on a trusted system to ensure the confidentiality of sensitive data.
- Use an approved random number generator to generate random numbers, file names, and GUIDs (Global Unique Identifier).
- Cryptographic key management should be applied by developing and following guidelines and processes.
- Master keys should be protected from unauthorised access.

Troubleshooting and logging

Error handling refers to procedures related to cases where the application/system gives unexpected output, usually as the result of abnormal input. Logging makes it possible to track changes to the software or application. Related OWASP advice:

- When troubleshooting, do not return information about the operation of the system.
- In the event of errors, memory must be released accordingly.
- Logs should not store sensitive information about the system.
- Logs related to failed input validation, authentication attempts, access and exception management and security configuration changes need to be maintained and audited.

Data protection

Data protection is the process that protects data from being altered, compromised or lost. While maintaining confidentiality, integrity and availability, data can be protected using the following techniques:

- Follow the principle of least privilege. Limit users' rights and access to data, systems and functions, so that they only have access to those that are essential for performing their tasks.
- Exclude sensitive data from HTTP GET requests.
- Protect server-side code and prevent it from being accessible to regular users. Access to critical and sensitive data must be controlled.
- Turn off autocomplete when entering data into forms in applications or websites.

Communication security

All sensitive information must be protected by encryption. To protect connections, TLS (Transport Layer Security) can be used in conjunction with other encryption algorithms.

- If TLS is used, unsuccessful connections should not switch to a less secure protocol.
- Use TLS to protect sensitive data from external sources.

System configuration

OWASP recommends using the following list when configuring systems:

- Ensure that you are running the latest versions of systems, frameworks, and system components that include patches.
- Apply the principle of least privilege to web servers, service accounts, and processes.
- HTTP response headers should only contain strictly necessary information. Information about OS, web server, and software framework should not be displayed.
- Test and development environments should be isolated from the live environment.

Database security

Advice on preventing threats to databases:

- The application should use the lowest possible privilege level to access databases.
- Instantly change default passwords.
- Use multifactor identification where possible.
- Disable default users that are not strictly necessary.

File management

Steps recommended by OWASP:

- Authenticate all file uploads to the server.
- Check files uploaded to a server by looking at their header.
- Disable run permission in folders where files can be uploaded.
- Never send the absolute path to the user.

Memory management

Memory management is a particularly important aspect of protection, because many attacks are memory-related. The following steps must be followed for proper memory management:

- Avoid using vulnerable functions (e.g. print, strcat, strcpy, etc.).
- Test the buffer size against overflows.
- Proper truncation of input strings is required before using copy or concatenation functions.

5. OWASP vulnerabilities

5.1. OWASP and the OWASP Top 10

The Open Web Application Security Project (OWASP) is an open community determined to promote the development, deployment, and maintenance of trusted applications and APIs.

OWASP Top10

The OWASP Top 10 focuses on raising the security level of web applications. The top 10 most critical risks to web applications have been identified with the help of application security companies and experts. The list was compiled taking into account the prevalence of the threat, the exploitability of the vulnerability and the difficulty of detection, as well as the impact that the successful exploitation of the vulnerability, i.e. a successful cyber attack, would have on a company's finances, systems and reputation.

OWASP's Top 10 list is a good basis for developers to learn about the most pressing security issues they should, and hopefully want, to avoid. At a higher level, the list helps to prioritise the vulnerabilities identified, so that companies can use it as a guide for the issues they should dedicate resources to.

5.2. Practical application of OWASP

While the OWASP Top 10 is primarily an awareness-raising document, many organisations use it as a de facto application security standard. It should be noted, however, that the OWASP Top 10 is nothing more than the first step towards achieving a coding a testing standard.

What is ASVS (Application Security Verification Standard)?

OWASP advises anyone looking to implement an application security standard to use their Application Security Verification Standard (ASVS), which is designed to be verifiable and testable, and can be used at any point in the secure application development cycle.

The ASVS is a collection of application security requirements and tests for designers, developers, testers, security experts, device manufacturers, and customers to design, develop, test, and verify secure applications.

6. NIST guidelines for developers

6.1. Preparing the organisation

The organisation needs to ensure that its people, processes and technology are ready to develop secure software.

Defining security requirements for system development

Ensure that the security requirements for software development are always known to the organisation. In the first step, gather and share these requirements with the relevant people. Requirements from internal sources (e.g. the organisation's policies, business objectives and risk management strategy) and external sources (e.g. relevant laws and regulations) should also be taken into account.

1. All security requirements for the organisation's software development infrastructures and processes should be defined and documented.
2. All security requirements for software developed by the organisation must be defined and documented to meet the expectations.
3. The requirements must be communicated to all third parties who provide the organisation with software components.

Definition of roles and responsibilities

It is important to ensure that everyone involved in the software development lifecycle is aware of their role and responsibilities, as well as the security requirements.

1. The roles and their associated obligations and responsibilities should be defined in such a way that the roles cover the entire software development cycle. These roles should be regularly reviewed, and their tasks and responsibilities updated as necessary.
2. Training should be provided for the responsible persons in each role to ensure secure development.
3. Senior management must be committed to secure development and must communicate this commitment to all stakeholders in the secure software development process.

Defining and using software security controls

Since the product generated during the software development cycle must meet the organisation's requirements, conditions must be defined and applied to control the security of the software.

1. For software security auditing, criteria should be defined for the entire development lifecycle.
2. Use processes and solutions that collect and store the necessary information to ensure that the software meets the criteria.

Implementing and maintaining a secure environment for development

Ensure that all processes relating to software development (e.g. development, testing, operation) are protected against internal and external threats.

1. Separate and protect the different environments involved in software development.
2. Development-related activities carried out by development endpoints (e.g. designers, developers, testers) must be made secure.

6.2. Software protection

The organisation must protect all components of the software from code tampering and unauthorised access.

Protecting code from unauthorised access and manipulation

Prevent careless or unauthorised modifications to code that could circumvent the intended security specifications of the software.

Code that is not publicly available helps prevent software theft and makes it more difficult and time-consuming for attackers to discover and exploit potential vulnerabilities.

The code must be stored according to the principle of least privilege, so that only authorised persons have access to the information they need.

Providing a mechanism to verify the integrity of software releases

Buyers and users of the software should be given the opportunity to verify that the product is legitimate and not tampered with, and the authentication information should be made available to them.

Archiving and protecting all software versions

Different software releases (versions) should be stored (archived) securely, so that the discovered vulnerabilities can be detected, analysed and patched later. The credentials and origin data should also be archived for each version, together with the required files and other data.

6.3. Secure software development

The organisation should produce well-protected software, with as few vulnerabilities and security issues as possible.

Designing software to meet security requirements and reduce security risks

Different risk modelling approaches can be used to assess the security risks of software, such as threat modelling or attack modelling. The security requirements of the software, the risks involved and the related design decisions must be documented.

Overview of the draft software to check compliance with security requirements and risk information

It must be ensured that the software complies with the security requirements and properly addresses the identified risk factors. A qualified person who has not been involved in the design should be engaged to check that the software meets all safety requirements, using appropriate automated processes.

Reusing existing, well-protected software, rather than duplicating functionality

By reusing some of the software modules and services (whose security has already been verified), the cost of software development can be reduced and the development process accelerated. If all third-party tools fail to meet the organisation's development and security needs, it must create a proprietary tool that meets the security and operational requirements.

The origin and integrity of all components from internal and external sources should be checked before reuse.

Creating source code by following safe coding practices

The source code should be written in consideration of the criteria set by the organisation. Safe coding practices of the programming languages and development environments used during development must be followed (e.g.: checking inputs, coding outputs, avoiding the use of unsafe functions and function calls, proper error handling, logging, change tracking, etc.).

Configuring the integrated development environment and compilation processes to improve security

Use source code compilation, interpretation and assembly tools that have features for improving the security of the resulting executable software. It is necessary to define which features of these tools should be used, and to configure the tools accordingly (e.g., during compilation, allow the compiler to send a message if it finds badly protected code).

Human-readable code interpretation and analysis to identify vulnerabilities and verify compliance with security requirements

Potential vulnerabilities should be identified through code analysis and code review so that they can be fixed before the software is released. This process can be made more efficient through automated procedures.

Executable code testing to identify vulnerabilities and verify compliance with security requirements

Testing should also aim to detect, analyse and document vulnerabilities and security holes in all components of the software so that the development team can fix them before the software is released. Testing should be planned and then all inputs, events, results and problems should be documented during the actual testing.

Configure default software security settings

Security requirements should also be kept in mind when installing software to minimise the likelihood of installing the product with incorrect security settings. These settings must be documented and the documentation must be provided to the system operators.

7. Conditions for implementing Google Analytics code from an IT security perspective

7.1. Patch management of third-party frontend components

Risks from third-party components

When developing web applications, because they are publicly available, particular attention must be paid to security. Developers should ensure that these applications are adequately protected against the major known security risks.

Efforts to develop secure web applications can easily be undone by vulnerabilities in third-party components, such as source code, scripts or CSS files.

Best practices for risk reduction

In order to minimise the risks associated with the use of third-party components, the following steps should be incorporated into the development process.

- Inventory of components
- Testing for addictions
- Risk assessment
- Risk reduction

Inventory of components

The first step is to find out the dependent components that the application uses. This is essential to reduce the security risks associated with the project. The process also helps to consider whether it is really necessary to use certain third-party components.

Testing for addictions

Once the inventory of dependent components is complete, the next step is to look for vulnerabilities. If the component is open source, the code can be easily tested. If the component is not open source, it can be checked in the manufacturer's security reports or searched in specific databases. Evidently, it is not advisable to manually check all the dependent components – it is better to use dedicted automated tools. Regardless of the method and instrument, the results of the study should include a list of vulnerabilities of the dependent components, also indicating their severity.

Risk reduction

If there is a solution to the vulnerability and the update does not cause any issues, it should be applied. If there is no solution available for the vulnerability of the third-party dependent component of the application, there are a couple of options:

- Working with the code writer/manufacturer to improve vulnerability.
- Finding a way around the vulnerability.

Keeping risk reduction under control

Reaching an acceptable level of risk does not mean that the problem will not occur again. Even if the proprietary code stays the same, third-party components can change at any time, and a simple rebuild can introduce unknown code into the application.

Client-side component locking – checking the integrity of external resources

Sub-resource integrity is a security feature that allows the browser to check whether the components it handles (i.e. collected from a Content Delivery Network) arrive without modification. It works based on a cryptographic hash value that the handled component must match.

The use of Content Delivery Networks (CDNs) for scripts and style files that are shared across multiple sites can make the site more efficient and save bandwidth. However, the use of CDNs carries the risk that, if an attacker takes control of a CDN, they may be able to add malicious code or even rewrite the entire code. This allows an attacker to attack all sites that use files from a given CDN.

External resource integrity checks reduce the risk of such attacks by ensuring that files from an external source of a web application or website arrive without having been modified or having been added by a third party.

8. Declaration on the learning outcomes of the course

Please send a declaration that you have successfully completed this course by e-mail to dora_oktatas@otpbank.hu with the following text:

“The training material developed by OTP Bank Plc. for the Contracting Partners who are ICT service providers in accordance with the requirements of the DORA Regulation has been read and understood by, and is accepted as binding on, all persons employed by [name of Partner company] in relation to the service provided to OTP Bank Plc. [Name of Partner company] undertakes to ensure that its employees and subcontractors involved in the provision of the service are familiarised with the training material throughout the life cycle of the service and to expect its employees and subcontractors to comply with the terms of the training material.”

9. Declaration on the learning outcomes of the course for sole traders

If you are a sole trader, please complete the following form and send it to dora_oktatas@otpbank.hu: “By completing this declaration, I declare that I have read and accept the contents of the General Privacy Notice (https://www.otpbank.hu/portal/hu/adatvedelem) of OTP Plc.”

Please also be informed that OTP Bank, as the controller, processes the name and e-mail address of, and the fact of completion of the training by, the sole trader in the context of the declaration in order to comply with the requirements of the DORA Regulation on the basis of a legitimate interest pursuant to Article 6(1)(f) of the General Data Protection Regulation.