Graphical user interface, text, application, chat or text message

Description automatically generated

**Description**

Sensitive Information Disclosure (also known as Sensitive Data Exposure) happens when an application does not adequately protect sensitive information that may wind up being disclosed to parties that are not supposed to have access to it.

Sensitive data can include application-related information, such as session tokens, file names, stack traces, or confidential information, such as passwords, credit card data, sensitive health data, private communications, intellectual property, metadata, the product's source code, etc.

Whichever security flaw is causing the information to be disclosed, all aspects of all kinds of services are potentially at risk. Sensitive Information Disclosure can arise in databases, operating systems, and network devices. It is particularly occurrent in web applications, as highlighted in [OWASP's Top 10](https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A3-Sensitive_Data_Exposure), which lists Sensitive Information Disclosure as the third most critical web application security risk of which to be aware.

Necessary privacy and security protection legislation and regulations are created and reworked to try to ensure that organizations holding sensitive data meet their obligations to safeguard such data. The European General Data Protection Regulation (GDPR) is one such law; the Payment Card Industry Data Security Standard (PCI DSS) is an example of regulation.

**Impact**

The scale of impact from a Sensitive Information Disclosure event is limited only by the type of sensitive information disclosed and a malicious actor's ability to leverage it.

For example, the fallout could be as minor as a local pathname being disclosed in a stack trace, allowing a malicious actor to improve their knowledge of the target's implementation details, right through to a full-blown data leak involving millions of customers' confidential data.

**Scenarios**

One typical example is to permit an end user to receive the *default* error pages of the application server. This can expose the location on the file system of the file that caused the issue along with the precise version of the server itself, and the third-party components. Attackers can use this knowledge in a variety of ways, for example, to target well-known exploits in one particular version of a component.

A more severe scenario involves a web page rendering an error message from a SQL server for a failed query. If some parameter is in control of the attacker, a malicious actor could exploit this exposure to exfiltrate arbitrary data from the database by sending specially crafted queries.

There are countless technologies sat under the IT umbrella susceptible to this comprehensive vulnerability class; basically, anything not properly tied down containing even minimal information may become the prey of a determined malicious actor.

**Prevention**

Sensitive Information Disclosure is a symptom of poor security-control implementation in web applications. Preventing it requires developers to adhere to numerous, necessary industry best-practices in line with current regulations to increase the difficulty for the attacker.

* Developers must first identify which data are sensitive according to the system architecture and regulatory requirements.
* Developers must ensure data in transit or storage are encrypted.
* Developers should remove debugging and test functionality from production applications and systems.
* Developers should review the listed items to determine if a justifiable business need exists for possessing each item present. Any items deemed unnecessary should be removed.
* Defined application/system build procedures should include steps to remove the files and features that are unnecessary for a production deployment, and internal security processes and controls should confirm this has occurred prior to production release.

**Testing**

Ensure that data's confidentiality is protected from unauthorized observation or disclosure.

* **OWASP ASVS**: [8](https://github.com/OWASP/ASVS/releases/download/v4.0.2_release/OWASP.Application.Security.Verification.Standard.4.0.2-en.pdf)
* **OWASP Testing Guide**: [Review Webserver Metafiles for Information Leakage](https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Application_Security_Testing/01-Information_Gathering/03-Review_Webserver_Metafiles_for_Information_Leakage.html), [Review Webpage Content for Information Leakage](https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Application_Security_Testing/01-Information_Gathering/05-Review_Webpage_Content_for_Information_Leakage.html), [Test File Extensions Handling for Sensitive Information](https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Application_Security_Testing/02-Configuration_and_Deployment_Management_Testing/03-Test_File_Extensions_Handling_for_Sensitive_Information.html), [Review Old Backup and Unreferenced Files for Sensitive Information](https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Application_Security_Testing/02-Configuration_and_Deployment_Management_Testing/04-Review_Old_Backup_and_Unreferenced_Files_for_Sensitive_Information.html)

**References**

[OWASP - Top 10 2017 A3 - Sensitive Data Exposure](https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A3-Sensitive_Data_Exposure)

[Infosec Institute - 2017 OWASP A3](https://resources.infosecinstitute.com/2017-owasp-a3-update-sensitive-data-exposure/)

[CWE-200: Exposure of Sensitive Information to an Unauthorized Actor](https://cwe.mitre.org/data/definitions/200.html)