eRecruiter

Updated on:
30.07.2024

system.eRecruiter.pl application security

erecruiter.pl

Table of contents

1. Architecture requirements 3
2. Authentication requirements 4
3. Session management requirements 6
4. Access control requirements 7
5. Malicious input handling requirements 8
6. Cryptography at rest requirements 10
7. Error handling and logging requirements 11
8. Data protection requirements 12
9. Communications security requirements 13
10. Http configuration requirements 14
11. File and resource management requirements 15
12. Webservice requirements 16
13. Configuration process requirements 17

The security requirements below have been developed in accordance with the OWASP ASVS standard and best application security practices.

1. Architecture requirements

No.	Requirement description	Meets requirement	Comments
1.1	The application architecture is composed of minimum 3 layers: • data (database) • business logic (application) • display (user interface)	Yes	
1.2	Individual components of the application architecture are separated from each other via a defined security control, such as network segmentation, firewall rules or cloud-based security groups.	Yes	
1.3	Only justified ports and protocols designed for use in architecture component communication are used.	Yes	
1.4	All the components of the application, libraries, modules, frameworks and operating systems are regularly monitored for vulnerabilities and are free from known vulnerabilities.	Yes	
1.5	There is no sensitive business logic information, secret keys or other proprietary information in the client-side application code.	Yes	
1.6	Access to data is limited on the level of databases, not applications. The application does not communicate with the database under DBA privileges.	Yes	
1.7	The application does not use protocols generally considered unsafe (i.e. ftp, NFS).	Yes	
1.8	On the edge of the network, the application is secured by a perimeter firewall.	Yes	
1.9	System and application logs are monitored and stored in SIEM.	Yes	
1.10	The application regularly undergoes penetration tests, and the discovered vulnerabilities are immediately corrected. The last application penetration tests is fully documented, took place less than 1 year prior to signing the Agreement, and all discovered vulnerabilities were removed .	Yes	.

2. Authentication requirements

No.	Requirement description	Meets requirement	Comments
2.1	All pages and resources by default require authentication, except for those specifically intended to be public (principle of complete mediation).	Yes	
2.2	All user authentications are enforced on the server side.	Yes	
2.3	If authentication controls fail, they do so securely, ensuring that attackers cannot log in.	Yes	After 5 failed login attempts, the password is blocked for 30 minutes. Unsuccessful attempts are logged in the system.
2.4	The password field is suited to the use of long or highly complex passphrases, and does not prevent from using password managers.	Yes	
2.5	All account identity authentication operations (such as profile registration, profile update, forgot login/password functionality) that enable regaining access to the account are at least as secure as the primary authentication controls.	Yes	
2.6	The password change functionality requires entering the old password, the new password, and confirming the new password .	Yes	
2.7	All attempts of logging in are logged without storing sensitive session identifiers or passwords.	Yes	
2.8	Account passwords are one-way hashed with a salt, and they are complex enough to defeat brute force and password hash recovery attacks.	Yes	Password complexity requirements are configurable by your system administrator.
2.9	Credentials are transported using a suitable encrypted link and all pages/functions that require the user to enter credentials require using such an encrypted link.	Yes	
2.10	Forgotten password recovery functions do not reveal the current password and the new password is not sent to the user in plain text.	Yes	
2.11	Information enumeration is not possible via users' logins, password reset or the forgotten username functionality.	Yes	
2.12	The application or any components used by the application do not use default passwords (such as "admin/password").	Yes	
2.13	The application has anti-automation controls to prevent breached credential testing, brute force attacks and account lockout attacks.	Yes	After 5 failed login attempts, the password is blocked for 30 minutes. Unsuccessful attempts are logged in the system.
2.14	All authentication credentials for accessing services external to the application are encrypted and stored in a protected location (not in the source code).	Yes	
2.15	The application temporarily blocks access to the account after entering the incorrect password 5 times. The administrator may block users in specific cases.	Yes	
2.16	If the application uses secret questions, it does not violate privacy laws. Additionally, the questions are sufficiently strong to protect the application.	N/A	The application does not use secret questions.

No.	Requirement description	Meets requirement	Comments
2.17	The application disallows the use of the user's 5 previous passwords.	Yes	
2.18	The application allows the use of two-factor authentication.	No	On demand, an IP address filter may be switched on, which will allow only particular IP addresses to access the application. The Client is required to have one or more permanent IP addresses. Possibility to integrate Client's ADFS (also possibility to run SAML 2.0) with eRecruiter
2.19	The application blocks the use of common passwords and weak passphrases.	Yes	Password requirements: Must be at least 12 characters Contains lowercase letters Contains uppercase letters Contains numbers Contains special characters (e.g.: ?, !, @, #) Must change every 60 days Cannot be the same as the last 5 passwords
2.20	Keys and credentials are not included in the source code or application code repositories.	Yes	
2.21	The application's administrative interface is not accessible from the level of untrusted networks (i.e. from the Internet) or is secured by two-factor authentication.	Yes	The admin panel is accessible on the same principles as the rest of the system and using the same authentication methods as other users. The BackOffice/Extranet panel is accessible only to eRecruiter employees and is accessible only from our corporate network (via VPN or from the office).

3. Session management requirements

No.	Requirement description	Meets requirement	Comments
3.1	The application does not use a custom session manager.	Yes	
3.2	Sessions are invalidated when the user logs out.	Yes	
3.3	Sessions time out after a specified period of inactivity.	Yes	After 60 minutes.
3.4	All pages that require authentication have easy and visible access to the logout functionality.	Yes	
3.5	Session IDs are never disclosed in URLs, error messages or logs, and the application does not support URL rewriting of session cookies.	Yes	
3.6	Every successful authentication and re-authentication generates a new session and session ID.	Yes	
3.7	The application recognises the session IDs generated by the application framework as active.	Yes	
3.8	Session IDs are sufficiently long (min. 128 bytes), random and unique across the relevant active session base.	Yes	Session tokens are 32 bytes long
3.9	Session IDs stored in cookies have their path set to a value sufficiently restrictive for the application, and authentication session tokens additionally have the "HttpOnly" and "secure" attributes set.	Yes	
3.10	The application limits the number of concurrent active sessions.	Yes	
3.11	The list of active sessions is displayed for each user in the account profile or similar feature. The user is able to terminate any selected active session.	N/A	The application does not enable logging in to the same account from different devices. Re-logging in closes the existing previous active session.
3.12	The user is prompted with the option to terminate all other active sessions after a successful password change process.	Yes	

4. Access control requirements

No.	Requirement description	Meets requirement	Comments
4.1	The principle of least privilege is implemented – users are only able to access functions, files, URLs, services and other resources which they have authorisation for.	Yes	
4.2	Access to sensitive records is protected, so that only authorised objects or data are accessible to the user (e.g. protection against users tampering with parameters that enable seeing or altering another user's account).	Yes	
4.3	Directory listing is disabled, unless deliberately permitted. applications do not allow the discovery or disclosure of files or directory metadata, such as Thumbs.db, .DS_Store, .git or .svn folders.	Yes	
4.4	Access controls which fail, do so securely (e.g. sites with errors do not display any details).	Yes	
4.5	The access control rules applied in the display layer are also enforced on the server side.	Yes	
4.6	All user and data attributes, as well as access policy information used by access controls, cannot be manipulated by end-users, unless they are authorised to do so.	Yes	
4.7	All access control decisions can be logged and all failed decisions are logged.	Yes	
4.8	The application or framework generates strong, random anti-CSRF tokens or has other transaction protection controls.	Yes	
4.9	The system is protected against aggregate or continuous access to secured functions, resources or data, e.g. mechanisms preventing the entire database from being scraped by an individual user.	No	The system does not have such mechanisms, however, actions regarding the candidate, e.g. CV overview/download are saved in the application logs.
4.10	The application correctly enforces a context-sensitive authorisation, to disallow unauthorised manipulation by changing parameter values.	Yes	

5. Malicious input handling requirements

No.	Requirement description	Meets requirement	Comments
5.1	The application uses a single input validation for each type of data that is accepted.	Yes	
5.2	All SQL, HQL, OSQL and NOSQL queries, as well as stored procedures and stored procedure callings, are protected by using prepared statements or parameterised queries, and are thus not susceptible to an SQL injection.	Yes	
5.3	The application is not susceptible to an LDAP Injection and/or the security controls prevent an LDAP Injection.	N/A	The application does not use the Lightweight Directory Access protocol.
5.4	The runtime environment is not susceptible to an OS Command Injection and/or the security controls prevent an OS Command Injection.	Yes	
5.5	The application is not susceptible to a Remote File Inclusion (RFI) or a Local File Inclusion (LFI), when content constituting a path to a file is used.	Yes	
5.6	The application is not susceptible to XML injection attacks, XML External Entity attacks and XPath query tampering, and/or the security controls prevent such attacks.	Yes	
5.7	All strings of variables placed into HTML or into another web client code are either adequately contextually encoded manually, or utilise templates that automatically encode contextually to ensure the application is not susceptible to reflected, stored and DOM Cross-Site Scripting (XSS) attacks.	Yes	
5.8	If the application framework allows automatic mass parameter assignment (also called automatic variable binding) from the inbound request to a model, security sensitive fields such as "*accountBalance*", "*role*" or "*password*" are protected from malicious automatic binding.	Yes	
5.9	The system has defences against HTTP Parameter Pollution attacks, particularly if the application framework does not distinguish the source of the request parameters (GET, POST, headers, cookies, environment, etc.)	Yes	
5.10	Validation on the client side is used as a second line of defence, complementing validation on the server side.	Yes	
5.11	All input data is validated. This includes not only HTML form fields but all sources of input such as REST calls, query parameters, HTTP headers, cookies, batch files, RSS feeds, etc.	Yes	
5.12	Structured data is strongly defined and validated against: allowed characters, length and pattern compliance (e.g. credit card or telephone numbers, or validating that two related fields are reasonable, such as validating whether a given location and zip or postal code match).	Yes	
5.13	Unstructured data is sanitised and contains allowed characters and length; characters potentially harmful in a given context are removed (e.g. natural names with Unicode or apostrophes, such as ねこ or O'Hara).	Yes	

No.	Requirement description	Meets requirement	Comments
5.14	An untrusted HTML code from WYSIWYG or similar editors is sanitised and appropriately handled with regard to the input validation task and encoding task.	Yes	
5.15	For auto-escaping template technology, when UI escaping is disabled, appropriate HTML code sanitisation is enabled.	N/A	The application does not use auto-escaping. All templates are sanitised against potentially dangerous content.
5.16	Data transferred from one DOM context to another uses safe JavaScript methods, such as .innerText and .val.	Yes	
5.17	When parsing JSON in browsers, that JSON.parse is used to parse JSON on the client.	Yes	
5.18	Authenticated data is cleared from client storage, such as the DOM browser, after the session is terminated.	Yes	

6. Cryptography at rest requirements

No.	Requirement description	Meets requirement	Comments
6.1	All cryptographic modules that fail, do so securely, and errors are handled in a way that prevents oracle padding attacks.	Yes	
6.2	All random numbers, random file names, random GUIDs and random strings that are meant to be unguessable by an attacker are generated using the cryptographic module's approved random number generator.	Yes	
6.3	All cryptographic algorithms used by the application are compliant with FIPS 140-2 or an equivalent standard.	Yes	
6.4	There is an explicit policy for how cryptographic keys are managed (e.g. generated, distributed, revoked or how they expire).	Yes	
6.5	Stored Personally Identifiable Information (PII) is encrypted and communication is performed via protected channels.	Yes	
6.6	Sensitive passwords or keys maintained in the memory are overwritten with zeros as soon as they are no longer required, to mitigate memory dumping attacks.	Yes	Memory dumping requires access to the application server, which is restricted. Moreover, the memory is immediately overwritten with newer objects in the memory.
6.7	All keys and passwords are replaceable and are generated or replaced at installation time.	N/A	The application is distributed in the application-as-a-service model, hence, installation at the client's is not necessary.

7. Error handling and logging requirements

No.	Requirement description	Meets requirement	Comments
7.1	The application does not output error messages or stack traces, which contain sensitive data and which could assist an attacker; such data includes session ID, software/framework versions and PII.	Yes	
7.2	Error handling logic in security controls denies access to them by default.	Yes	
7.3	Security logging controls provide the possibility to log successful and failed events that are identified as security-relevant.	Yes	Attempts to access from an invalid IP address are not logged (in the case of a security configuration that specifies an allowed pool of IP addresses).
7.4	The log of each event includes necessary information that allows a detailed investigation of the timeline if an event occurs.	Yes	
7.5	Security logs are protected from unauthorised access and modification.	Yes	
7.6	The application does not log sensitive data defined under local privacy laws or regulations, organisational sensitive data as defined in a risk assessment, or sensitive authentication data that could assist an attacker, including user's session IDs, passwords, hashes or API tokens.	No	The IP address for the Candidate Card and the Application Review is logged, other sensitive data is not logged.
7.7	Non-printable symbols and field separators are properly encoded in log entries, in a way preventing log injection.	Yes	
7.8	Audit logs or similar registers allow the non-repudiation of key transactions.	Yes	
7.9	Time sources are synchronised to ensure that logs contain the correct time.	Yes	

8. Data protection requirements

No.	Requirement description	Meets requirement	Comments
8.1	All sensitive data is sent to the server in the HTTP message body or headers (i.e. URL parameters are never used to send sensitive data).	Yes	
8.2	All cached or temporary copies of sensitive data stored on the server are protected against unauthorised access or are sanitised/invalidated after an authorised user accesses the sensitive data.	Yes	
8.3	The application minimises the number of parameters in a request via i.a.: hidden fields, AJAX variables, cookies and header values.	Yes	
8.4	Data stored on the client side (e.g. locally stored HTML5 session information, stored sessions, IndexedDB, regular cookies or Flash cookies) does not contain sensitive data or PII.	Yes	
8.5	Access to sensitive data is logged, if the data is collected in accordance with relevant data protection regulations or where access logging is required.	Yes	

9. Communications security requirements

No.	Requirement description	Meets requirement	Comments
9.1	Valid SSL certificates from a trusted CA.	Yes	
9.2	A TLS certificate is used for all connections (external and back-end) that are authenticated or that involve sensitive data/functions, and does not fall back to insecure or unencrypted protocols. The strongest available encrypting algorithm is ensured.	Yes	
9.3	All connections to external systems that involve sensitive information or functions are authenticated.	Yes	
9.4	HTTP Strict Transport Security headers are included on all requests and for all subdomains, e.g. Strict-Transport-Security: max-age=15724800; includeSubdomains.	Yes	
9.5	Forward secrecy ciphers are in use to mitigate passive attackers recording traffic.	Yes	
9.6	Proper certification revocation mechanisms, such as Online Certificate Status Protocol (OCSP) Stapling, are enabled and properly configured.	Yes	
9.7	Only strong algorithms, ciphers and protocols are used through the whole certificate hierarchy, including root and intermediary certificates of the selected certifying authority.	Yes	
9.8	The TLS configuration is in line with current best practices, particularly as common configurations, ciphers and algorithms may prove insecure with time.	Yes	

10. Http configuration requirements

No.	Requirement description	Meets requirement	Comments
10.1	The application only accepts a defined set of required HTTP request methods (e.g. GET, POST, PUT, DELETE), and explicitly blocks unused methods (e.g. TRACE).	Yes	
10.2	Every HTTP response contains a content type header specifying the safe character set (e.g. UTF-8, ISO 8859-1).	Yes	
10.3	HTTP headers added by a trusted proxy or SSO devices, such as a bearer-token, are authenticated by the application.	N/A	The application enforces https encoding, hence, it is impossible to add HTTP headers to messages sent.
10.4	A suitable X-FRAME-OPTIONS header is in use for sites whose content should not be viewed in a 3rd-party X-Frame.	Yes	
10.5	HTTP headers or any part of the HTTP response do not disclose detailed information of system component versions.	Yes	
10.6	All API responses contain information on the content type in the header.	Yes	
10.7	The content security policy v2 (CSPv2) mechanism is in place, in order to mitigate DOM, XSS, JSON and JavaScript injection vulnerabilities.	Yes	
10.8	The X-XSS-Protection: 1; mode=block header is in place to enable browser reflected XSS filters.	No	This approach is not in line with current guidelines: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-XSS-Protection

11. File and resource management requirements

No.	Requirement description	Meets requirement	Comments
11.1	URL redirects and forwards are only allowed for whitelist sites and warn the user when redirecting to potentially untrusted content.	Yes	
11.2	Untrusted file data submitted to the application is not used directly with file I/O commands, particularly to protect against path traversal, local file inclusion and OS command injection vulnerabilities.	Yes	
11.3	Files obtained from untrusted sources are validated for expected file type and scanned by antivirus scanners to prevent uploading malicious content.	Partly	The application does not interfere with data sent by candidates via application forms. Such files are validated for file type (executable files, such as .exe, are blocked), but they are not scanned with an antivirus scanner. File recognition is performed both by the extension and by the content of the first bytes of the file.
11.4	Untrusted data is not directly used in mechanisms of inclusion, class loader or reflection, in order to prevent remote/local file inclusion attacks.	Yes	
11.5	Untrusted data is not used within cross-origin resource sharing (CORS) to protect against external potentially malicious content.	Yes	
11.6	Files obtained from untrusted sources are stored outside the webroot, with minimum permissions, preferably with strong validation.	Yes	
11.7	The web or application server is configured by default to deny access to remote resources or systems outside the web or application server.	Yes	
11.8	The application code does not execute uploaded data obtained from untrusted sources.	Yes	
11.9	Flash, Active-X, Silverlight, NACL and client-side Java applets, which are not supported natively via W3C browser standards, are not used.	Yes	
11.10	The provider is able to precisely determine the countries in which the data is processed.	Yes	The data centers used are located in the EEA (Netherlands, Ireland and Poland).

12. Webservice requirements

No.	Requirement description	Meets requirement	Comments
12.1	The same encoding style is used by the client and the server.	Yes	
12.2	Access to administration and management functions within the application is limited to service administrators.	Yes	
12.3	The XML or JSON scheme is in place and is verified before accepting input.	Yes	
12.4	All input is limited to an appropriate size.	Yes	
12.5	SOAP-based web services are compliant at least with the Web Services-Interoperability (WS-I) Basic Profile. This refers to TLS encryption in particular.	Yes	
12.6	Session-based authentication and authorisation are used. Static API keys and similar solutions are avoided.	Yes	
12.7	REST services are protected against Cross-Site Request Forgery attacks by using at least one of the following mechanisms: ORIGIN checks, double submission of cookie patterns, CSRF nonces and/or referrer checks.	Yes	
12.8	REST services explicitly check whether the incoming Content-Type is the expected one, e.g. application/xml or application/json.	Yes	
12.9	The message payload is signed to ensure safe transmission between the client and service, using JSON Web Signing or WS-Security for SOAP requests.	Yes	
12.10	No alternative or less secure access paths exist.	Yes	

13. Configuration process requirements

No.	Requirement description	Meets requirement	Comments
13.1	All components are up to date, have proper security configurations and versions, including the removal of unneeded configurations and folders, such as sample applications, framework documentation and default or sample users.	Yes	
13.2	The communication between components, such as between the application server and the database server, is encrypted, in particular when the components are in different containers or different systems.	Yes	
13.3	The communication between components, such as between the application server and the database server is authenticated using the account with the least necessary privileges.	Yes	
13.4	The deployed applications are adequately sandboxed, containerised or isolated to delay or deter attackers from accessing other applications.	Yes	
13.5	The building and deployment process for the application is performed in a secure manner.	Yes	