# Security Policy

For SmartCloud Connect by InvisibleSolutions

Security is of utmost importance to protect customers' valuable assets, be it data, processes or availability of systems and services, from misuse or damage. To prevent such damage and their corresponding liability risks, loss of reputation and revenue, SmartCloud Connect was developed and provisioned accordingly to the state-of-the-art security standards.

This includes strict adherence to Web Application Security Policy, constant monitoring of production environments for vulnerabilities or non-standard activities, and implementing of important qualities and features supporting secure operation of solution. See description below for more details.

## OWASP A1. No Injection

See https://www.owasp.org/index.php/Top_10_2013-A1-Injection

### No SQL Injections

Software disallows manipulating SQL or other data query and manipulation statements through input parameters so that no malicious statements can be run on the database by attackers.

### No Path Traversal vulnerabilities.

Input parameters are thoroughly checked in case path names are submitted. No services, data or data files are accessible that are not provided for this purpose or require different privileges, both via absolute and relative path names.

See http://www.owasp.org/index.php/Path_Traversal

### No operating system command injection vulnerabilities.

Software ensures that user input cannot be misused to execute unintended operating system commands or to access data without proper authorization.

See http://www.owasp.org/index.php/Command_Injection

### No memory corruption vulnerabilities

SmartCloud Connect is built using .NET Framework. In .NET buffer overflows are prevented by the framework itself.

# OWASP A2. No Broken Authentication and Session Management

See https://www.owasp.org/index.php/Top_10_2013-A2-Broken_Authentication_and_Session_Management

- Solution is based on industry standard technologies (ASP.NET Identity Framework, etc.)
- User authentication credentials are stored in encrypted and hashed form
- Strict password complexity policies prevent attackers from guessing passwords.
- Periodic password rotation prevents attackers from using leaked passwords
- Account lockout prevents brute-force attacks
- Authenticated and encrypted communication channels are enforced. All encrypted communication implies preliminary authentication to server.

## Sessions are managed securely

- It is impossible to guess or predict a valid security session identifier.
- Session identifiers change whenever the authentication status changes.
- A user´s security session is locked after a configurable period of inactivity.
- A server accepts only security session identifiers created by the server itself.
- Session identifiers are never written to a log or trace file, or displayed anywhere, not even to an administrator.
- Session IDs are never exposed in URL
- No session fixation attacks are possible
- Passwords/session IDs are sent over encrypted connections only
- See https://www.owasp.org/index.php/Session_Management_Cheat_Sheet

## Secure authentication is enforced.

At least one secure authentication mechanism (e.g., UserID/ Password, Single Sign-On) is provided.

Administrative functions (e.g., configuration, user management) and administrative tools require admin user authentication.

All services (e.g., URLs, transactions, reports) are controlled by proper authentication.

IP addresses are not exclusively used for authentication purposes. The IP address can only be an additional authentication attribute for authentication.

When password-based authentication is provided, passwords are not stored in plaintext, but iterated random-salted hash values are used for verification using a cryptographically strong hash function.

Passwords or other authentication information is not written to error, trace or log files or displayed anywhere.

### Authorization of users are enforced that are reduced to the minimum necessary level as required by their business-related function (applies for end users, technical users, administrators, developers)

Authorizations (roles) for different types of users (e.g., administrative and non-administrative users) are separated. Users only need the permissions that are required for their work (least privilege principle is enforced) and permission assignment supports segregation of duties. All external callable services (e.g., URLs, transactions, reports) are controlled by proper authorization checks. Search results may contain sensitive data and thus are displayed based on the access permissions. The software uses or sets the minimal necessary privileges on operating system and on file system level, if applicable.

### Audit relevant data can be logged and archived.

Security relevant events (e.g., failed login attempts, failed authorization checks, start of critical transactions, changes to user data and authorizations, changes to security configuration, etc.) are logged and available in read-only mode for internal security audits.

## OWASP A3. No Cross-Site Scripting (XSS) vulnerabilities

See https://www.owasp.org/index.php/Top_10_2013-A3-Cross-Site_Scripting_(XSS) and *http://www.owasp.org/index.php/Cross-site_Scripting_(XSS)*

All user-supplied input is properly escaped and verified via input validation. Use of development frameworks (AngularJS) guarantees state-of-art escaping on the client side.

### No clickjacking vulnerabilities

User interfaces, in particular web-based ones, shall be composed in a way so that they disallow malicious third-parties to misuse user interactions.

See: https://www.owasp.org/index.php/Clickjacking

### Secure input validation and output encoding are enforced

All input is validated against expected format, type, size, schema. All output is encoded according to the target component/UI.

See: https://www.owasp.org/index.php/Data_Validation

## OWASP A4. No insecure direct object references

See https://www.owasp.org/index.php/Top_10_2013-A4-Insecure_Direct_Object_References

SmartCloud Connect security model is based on Access Control Lists security where access to each resource is guarded by set of security rules. This guarantees that users are not allowed to see restricted resources, neither directly nor indirectly.

All requested data is checked against user authorization. This includes protection against Insecure Direct Object References and Parameter Tampering. Input validation includes also checks of incoming XML files and uploaded files if they can be manipulated by end users.

## OWASP A5. No Security Misconfiguration

See https://www.owasp.org/index.php/Top_10_2013-A5-Security_Misconfiguration

- SmartCloud Connect runs as managed service on Windows Azure platform and gets automatic updates for all discovered OS/Webserver/DB vulnerabilities.
- SmartCloud Connect application libraries are periodically reviewed for available security patches.
- SmartCloud Connect exposes minimal ports/services to outer worlds and does that in a secure way.
- No default accounts/password present in system. Test and production systems uses same security configuration (differ only by security credentials)
- Security settings in ASP.NET framework were reviewed to have proper security values
- Scans and audits are run periodically against SmartCloud Connect, ranging from built-in Windows Azure capabilities (like Security Center) to external audits like Salesforce security review for AppExchange apps, Microsoft Office Store publish review etc.

### No system internal information is disclosed.

System internal information is not shown to unauthorized users. This includes information in URLs, hidden html fields, configuration settings, error messages, stack traces and log files.

### No intentionally hidden and non-documented features that circumvent security measures (backdoor).

Backdoors always pose a great security risk, since the customer is not aware of them and thus does not enforce any protection mechanisms.

See also http://en.wikipedia.org/wiki/Backdoor_(computing)

### Configuration is secure by default

The default configuration of the software keeps the customer's risk at a minimum.

This includes:

- No users have default passwords

- No hard-coded passwords are used, passwords for technical users are configurable

- Open ports are reduced to a minimum

- File system is appropriately protected (e.g., no world readable/writeable files, exceptions documented)

- Functions/services rarely used (such as demonstrations/examples, optional functionality) are not activated

- Display of debug information is switched off

- Security configuration is reversible, re-editable, and viewable.

- Configuration data is secured against unauthorized access.

## OWASP A6. No Sensitive Data Exposure

See https://www.owasp.org/index.php/Top_10_2013-A6-Sensitive_Data_Exposure

Sensitive data includes authentication data, security-critical data, personal data and confidential business data.

- Sensitive data is never stored or transmitted in clear text
- Only state of art cryptography is used (standard algorithms, recommended key lengths, use of secure key stores instead of hard-coded keys)
- Key management and rotation policies are in place
- Proper browser security directives and headers are provided with resources which may contain sensitive data

### HTTP request and response handling is implemented in a secure way.

- Insecure HTTP protocol is NOT used for communications
- Application state is only changed via the POST, PUT and DELETE methods

- HTTP TRACE is switched off by default, and there are no HTTP Verb Tampering, HTTP Parameter Pollution and HTTP Response Splitting vulnerabilities in the software.
- See http://capec.mitre.org/data/definitions/274.html, http://capec.mitre.org/data/definitions/460.html

### Sensitive data stored on user computers is protected.

No sensitive data is stored unprotected at file system level on the backend server, the frontend client, or on any intermediary computers or mobile devices. Sensitive data includes authentication data, security-critical data, personal data and confidential business data. Protection includes access control, strong encryption for confidentiality and message integrity/authentication codes or digital signatures for integrity and authenticity protection.

When sensitive data is stored on user environments (Exchange Mailbox, or Exchange clients) it uses the protection services offered by the user's security configuration.

### Access to sensitive data is logged

The software ensures that when personal is accessed, the action and the actor can be verified later.

### If data is transferred to other service providers this is made transparent and configurable.

In case data is forwarded to additional service providers, consent from users is required. The usage of additional services or service providers is configurable and documented.

## OWASP A7. No Missing Function Level Access Control

See https://www.owasp.org/index.php/Top_10_2013-A7-Missing_Function_Level_Access_Control

- No security decisions are made on UI level
- Backend verifies each user request in regards to his authorizations
- Authentication is based on standard protocols and reliable data sources

## OWASP A8. No Cross-Site-Request-Forgery (XSRF) vulnerabilities.

See https://www.owasp.org/index.php/Top_10_2013-A8-Cross-Site_Request_Forgery_(CSRF) and https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF)

Software prevents attacks forcing an authenticated user to perform unwanted actions by accident.

For web applications, state changing HTTP requests are protected with a random short-time and user specific XSRF token.

## OWASP A9. Not Using Components with Known Vulnerabilities

See https://www.owasp.org/index.php/Top_10_2013-A9-Using_Components_with_Known_Vulnerabilities

SmartCloud Connect uses NuGet as modern package management solution, and presence of new versions of components is constantly monitored, and update paths are planned.

## OWASP A10. No Unvalidated Redirects and Forwards

See https://www.owasp.org/index.php/Top_10_2013-A10-Unvalidated_Redirects_and_Forwards

SmartCloud Connect does not redirect/forward to dynamically constructed URLs using information provided by caller.

## Privacy policy

InvisibleSolutions privacy policy: https://invisiblesolutions.com/s/privacy-policy

SmartCloud Connect provides privacy by design and privacy by default.

### Privacy by Design

The software is developed in such a way that personally identifiable data which could be collected and used is understood and drives the development of the privacy practices. Collecting personally identifiable data not needed for service's basic functionality is avoided or limited. A privacy policy is developed that is clear, accurate, and conspicuously accessible to users and potential users. Short privacy statements and privacy controls are used – to draw users' attention to data practices that may be unexpected and to enable them to make meaningful choices.

Application privacy policies are accessible from the app platform so that they may be reviewed before a user downloads an app.

### Privacy by Default

An easily understood privacy policy is provided that tells the user, who to contact for the application, personal data collected and processed, what purpose the data is collected for, any disclosure of the data

to others, withdrawal of consent mechanisms. Before accessing data, the product shall ask for consent which is made granular in nature. Categories that should be included but not limited to are: Location, Contacts, Unique Device Identifier, Identity of the data subject, Identity of the phone, Credit card and payment data, Telephony SMS, browsing history, Email, Social networks credentials and Biometric history. The user is allowed to opt-in or opt-out of capabilities at any time. Special notices are used to bring to the attention of the user what has changed and allow them to opt-in to the new change. If data is going to be shared, even anonymized, this is made obvious to the user and the user is allowed to opt-in.