# Assumed Secure
Security Controls Self-Assessment

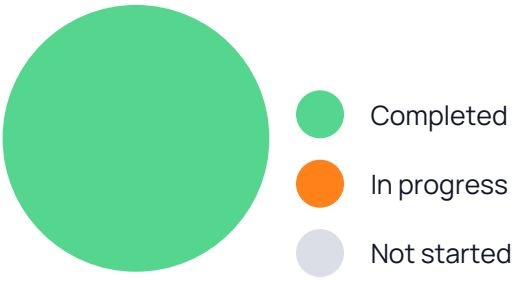
## Minimum Viable Secure Product
Assumed Secure uses the MVSP control checklist to guide users through a questionnaire that will help businesses implement the minimum level of security for their product, service or application.

Minimum Viable Secure Product is a minimalistic security checklist for B2B software and business process outsourcing suppliers. Designed with simplicity in mind, the checklist contains only those controls that must, at a minimum, be implemented to ensure a reasonable security posture.

**Learn more about MVSP here:** https://mvsp.dev/

## Compliance Graph

| Assessment Name | Completed By | Report Date |
|---|---|---|
| boberdoo Lead System | danc@boberdoo.com | 11/07/2023 |




| State | Count |
|---|---|
| Completed | 25 |
| In progress | 0 |
| Not started | 0 |

# Detailed Review of MVSP Compliance

| MVSP Control Requirements | Assessment Question | Status |
|---|---|---|
| **Data flow diagram**<br>• Maintain an up-to-date diagram indicating how sensitive data reaches your systems and where it ends up being stored | Do you maintain a current data flow diagram showing how sensitive data enters, leaves and is stored in the application? | Completed |
| **List of data**<br>• Maintain a list of sensitive data types that the application is expected to process | Do you maintain an inventory of sensitive data types that are processed by the application? | Completed |
| **Time to fix vulnerabilities**<br>• Produce and deploy patches to address application vulnerabilities that materially impact security within 90 days of discovery | Do you patch application vulnerabilities that materially impact security within 90 days? | Completed |
| **Vulnerability prevention**<br>• Train your developers and implement development guidelines to prevent at least the following vulnerabilities:<br>• Authorization bypass. Example:<br>Accessing other customers' data or admin features from a regular account.<br>• Insecure session ID. Examples:<br>Guessable token;<br>A token stored in an insecure location (e.g. cookie without secure and httpOnly flags set)<br>• Injections. Examples:<br>SQL injection, NoSQL injection, XXE, OS command injection<br>• Cross-site scripting. Examples:<br>Calling insecure JavaScript functions;<br>Performing insecure DOM manipulations;<br>Echoing back user input into HTML without escaping<br>• Cross-site request forgery. Example:<br>Accepting requests with an Origin header from a different domain<br>• Use of vulnerable libraries. Example:<br>Using server-side frameworks or JavaScript libraries with known vulnerabilities | Have your developers implemented controls to prevent vulnerabilities including authorization bypass, insecure session ID, injection attacks, cross-site scripting, cross-site request forgery and the use of vulnerable libraries? | Completed |
| **Build process**<br>• Build processes must be fully scripted/automated and generate provenance (SLSA Level 1) | Can you verify the origin of all code running in the production application? | Completed |

### Customer testing

- On request, enable your customers or their delegates to test the security of your application
- Test on a non-production environment if it closely resembles the production environment in functionality
- Ensure non-production environments do not contain production data

Do you allow customers or their delegates to test the security of your application upon request?

Completed

### Incident handling

- Notify your customers about a breach without undue delay, no later than 72 hours upon discovery
- Include the following information in the notification:
  - Relevant point of contact
  - Preliminary technical analysis of the breach
  - Remediation plan with reasonable timelines

Will you notify your customers about a security breach within 72 hours upon discovery?

Completed

### External testing

- Contract a security vendor to perform annual, comprehensive penetration tests on your systems

Do you use an independent security vendor to conduct penetration tests on your systems or application at least annually?

Completed

### Compliance

- Comply with all industry security standards relevant to your business such as PCI DSS, HITRUST, ISO27001, and SSAE 18
- Comply with local laws and regulations in jurisdictions applicable to your company and your customers, such as GDPR, Binding Corporate Rules, and Standard Contractual Clauses
- Ensure data localization requirements are implemented in line with local regulations and contractual obligations

Are you able to comply with all industry standards, regulations and data localization requirement that are applicable to your product?

Completed

### Training

- Implement role-specific security training for your personnel that is relevant to their business function

Do you provide your employees with security training that is relevant to their role in your organization?

Completed

### Data handling

- Ensure media sanitization processes based on NIST SP 800-88 or equivalent are implemented

Do you properly sanitize and dispose of customer data stored on physical media?

Completed

### Self-assessment

- Perform annual (at a minimum) security self-assessments using this tool

Do you perform annual security self-assessments?

Completed

### Vulnerability reports

- Publish the point of contact for security reports on your website
- Respond to security reports within a reasonable time frame

Is the point of contact for security reports published on your website?

Completed

### Single Sign-On

- Implement single sign-on using modern and industry standard protocols

Does the application support single sign-on (SSO)?

Completed

### Password policy

- Do not limit the permitted characters that can be used
- Do not limit the length of the password to anything below 64 characters
- Do not use secret questions as a sole password reset requirement
- Require email verification of a password change request
- Require the current password in addition to the new password during password change
- Store passwords in a hashed and salted format using a memory-hard or CPU-hard one-way hash function
- Enforce appropriate account lockout and brute-force protection on account access

Does your product use secure password authentication mechanisms?

Completed

### Security libraries

- Use frameworks, template languages, or libraries that systemically address implementation weaknesses by escaping the outputs and sanitizing the inputs
  • Example: ORM for database access, UI framework for rendering DOM

Does your application address implementation weaknesses by escaping the outputs and sanitizing the inputs?

Completed

### Security Headers

- Apply appropriate security headers to reduce the application attack surface and limit post exploitation:
  • Set a minimally permissive Content Security Policy
  • Limit the ability to iframe sensitive application content where appropriate

Does your product or application apply appropriate security measures to reduce the application attack surface and limit exploitation?

Completed

### HTTPS-only

- Redirect traffic from HTTP protocol (port 80) to HTTPS (port 443)
- This does not apply to secure protocols designed to run on top of unencrypted connections, such as OCSP
- Scan and address issues using freely available modern TLS scanning tools
- Include the Strict-Transport-Security header on all pages with the includeSubdomains directive

Does the application only allow secure traffic (HTTPS)?

Completed

### Logging

- Keep logs of:
  • Users logging in and out
  • Read, write, delete operations on application and system users and objects
  • Security settings changes (including disabling logging)
  • Application owner access to customer data (access transparency)
  • Logs must include user ID, IP address, valid timestamp, type of action performed, and object of this action. Logs must be stored for at least 30 days, and should not contain sensitive data or payloads.

Do you log user activity?

Completed

### Encryption

- Use available means of encryption to protect sensitive data in transit between systems and at rest in online data storages and backups

Is application data encrypted in transit and at rest?

Completed

### Dependency Patching

- Apply security patches with a severity score of "medium" or higher, or ensure equivalent mitigations are available for all components of the application stack within one month of the patch release

Do you apply security patches for medium to high severities in a timely manner?

Completed

### Backup and Disaster recovery

- Securely back up all data to a different location than where the application is running
- Maintain and periodically test disaster recovery plans
- Periodically test backup restoration

Does your application have backup and disaster recovery procedures in place?

Completed

### Subprocessors

- Publish a list of third-party companies with access to customer data on your website
- Assess third-party companies annually against this baseline

Do you maintain a list of all third-party companies with access to customer data?

Completed

### Physical access

- Validate the physical security of relevant facilities by ensuring the following controls are in place:
  - Layered perimeter controls and interior barriers
  - Managed access to keys
  - Entry and exit logs
  - Appropriate response plan for intruder alerts

Are physical security measures in place in your data-centers, hosting providers, and offices?

Completed

### Logical access

- Limit sensitive data access exclusively to users with a legitimate need. The data owner must authorize such access
- Deactivate redundant accounts and expired access grants in a timely manner
- Perform regular reviews of access to validate need to know
- Ensure remote access to customer data or production systems requires the use of Multi-Factor Authentication

Are secure logical access controls implemented to protect access to customer data?

Completed