Cloud Security for STARTUPS

Authored by CSA Israel, October 2017

cloud security alliance®

Acknowledgments

Creators

Moshe Ferber

Shahar Geiger Maor

Yael Nishry

Contributors

Marius Aharonovich

Ron Peled

Yuval Reut

Ofer Smadari

Omer Taran

Peer Reviewers

Rich Campagna

Govindasamy Chinnu

Scott Kennedy

Kyle McAuliffe

Alexandre Caramelo Pinto

Michael Roza

Gurpreet Sahota

Zeal Somani

James Stewart

Srinivas Tatipamula

Peter van Eijk

About the Cloud Security Alliance

The Cloud Security Alliance is a not-for-profit organization with a mission to promote the use of best practices for providing security assurance within Cloud Computing, and to provide education on the uses of Cloud Computing to help secure all other forms of computing. The Cloud Security Alliance is led by a broad coalition of industry practitioners, corporations, associations and other key stakeholders. For further information, visit us at www.cloudsecurityalliance.org and follow us on Twitter @cloudsa.

About the CSA Israeli Chapter

This document was created by the Israeli chapter of the Cloud Security Alliance (CSA). The Israeli chapter of the Cloud Security Alliance was founded by security professionals united in a desire to promote responsible cloud adoption in the Israeli market and and deliver useful knowledge and global best practices to the Israeli innovation scene.

Visit our Facebook group at www.facebook.com/groups/789522244477928 for more details.

Table of Contents

Introduction

Background

Information security is a complicated subject even for mature enterprises, so it's no wonder that startups find the area challenging. Planning, implementing and maintaining good-practice security are not only necessary, but can also serve as an important advantage that can be leveraged as a marketing differentiator. On the other hand, poor practices may result in dire consequences, ranging from difficulties in fundraising to bankruptcy[1].

The most crucial security challenge for young startups is to align security with their business growth so that security controls will match the risks at any point in time.

Purpose of this Document

The main purpose of this document is to help Software-as-a-Service Startups (SaaS-SUs) gain and maintain customers' trust, by building solid security foundations at an early stage of their product development process and aligning security controls with product development and investments rounds. In addition, these guidelines should help SaaS-SUs meet the most important security and privacy requirements presented by customers considering new services and products.

Target Audience

- This document is aimed at cloud-based startups, developing on public Infrastructure/ Platform-As-A-Service (IaaS/PaaS) who wish to understand their security roadmap.
- The target audience to benefit from these guidelines includes founders, CTOs, product managers and architects.
- SaaS-SU designated market sector, types of data collected and geographic locations might accelerate transition between stages.

1 The Code-Spaces case is one of the most eye-opening examples of poor security controls leading to a company going out of business.

Before You Start

Security Requirements for Early Stages of a Startup

Startups must plan their security posture according to the progress they make in funding and product development. To help startups evaluate necessary security requirements, we have outlined three phases of SAAS-SU maturity:

- **Phase 1: Inception.** From idea to first customers. In the phase between idea and the first customer, budget generally is limited, so startups should focus on laying building blocks for future potential security needs.
- **Phase 2: Prepare for Growth.** When the startup has paying customers.
- **Phase 3: Maturity.** When a startup has gained a strong, positive reputation and enough customers to create profit, it is time to advance to a more mature security posture.

When examining which security controls should be implemented for each phase, there is a difference between market sectors and the type of data your startups collect. As a general rule, if startup characteristics match any of the following, the company should prepare to move faster through phases of maturity discussed above.

- If a startup's target customers have become enterprises, the company can expect to be questioned about participation in the shared responsibility model, identity management and security policies.
- If the data a startup stores contains high volumes of PII or sensitive PII (e.g. health information or financial details).
- If a startup must comply with especially strict regulations and laws (e.g. HIPAA, GDPR, Privacy Act).
- If a startup's target sectors include representatives from the industries of health, government, financial or homeland security, the startup must then expect industry-

Tip

- The Cloud Security Alliance Cloud Controls Matrix (CCM) is an excellent tool for mapping the security requirements of various laws, regulations and standards, and for better understanding future challenges.

specific regulations and additional security needs regarding its location of services.

Why Pay Attention to Security Early in the Game?

- Implementing security measures early on can help a startup gain customer trust and meet the compliance requirements that will come later.
- Some startup's customers have internal IT security requirements that will need to be implemented by the startup.
- Inadequate attention to security risks early in the lifecycle of a startup may lead to "technical debt," which may be too expensive to resolve later.
- Adequate attention to IT security needs—especially to the startup's intellectual property (IP)—can significantly influence the startup's valuation and reduce risk to investors.

Choosing a Cloud Platform[2]

There are many parameters to consider when choosing an IaaS/PaaS provider. Many of these parameters are not directly related to cloud security, but the following are directly implicated.

- **Service location.** When targeting enterprises from a specific geographic jurisdiction, it is recommended to keep customers' data in the same geographic location[3]. Doing so can relieve compliance efforts and create a competitive advantage.
- **Regulations.** SAAS-SUs should strive to work with service providers who adhere to the same regulation regime and standards as their designated market[4].
- **Ecosystem.** A SAAS-SU usually strives to consume external software and services in order to reduce development hours. A large ecosystem of knowledge, tools and third-party software is an advantage for cloud providers.

Tips

- When targeting enterprises in the US, EU and/or APAC, consider deploying data storage into all of these regions to meet compliance.
- IaaS will provide better flexibility and control than PaaS, if you own your server's configuration. However, choosing a PaaS provider also establishes a responsibility to secure those servers.

2 For the purpose of this paper, "cloud platform" refers to the platform on which SUs develop their service or product. In a cloud environment, "cloud platform" refers to IAAS and PAAS providers.
3 Same geographic location could mean a large region such as the United States or the European Union, but may also refer to a country or other territory, depending on applicable laws and regulations.
4 E.g.: It will be impossible for SAAS-SUs to achieve ISO27001 certification without the SAAS-SUs provider's compliance with the same standard.

Application Security

Application security is an important pillar when planning security foundations. A lack of good application development and deployment methods can result in an inability to adhere to regulations and standards in addition to exposure to application attacks. Neglecting application security practices at the early stages of a startup makes it much harder and more expensive to correct later.

General Recommendations

Understand the difference aspect of application security and reflect them to your customers and employees:

- Be transparent. Offering your clients some useful security insights regarding the best ways to integrate your service into the client's environment may position you as a trusted advisor. This proactive approach may help you earn your client's appreciation when they go to evaluate your company's security risks. Consider providing clients with recommended guidance on security best practices that pertain to the services they seek.
- The CSA Guidance For Critical Areas In Cloud Computing provides information on all security domains of cloud computing. Use this guide to predict what your customers will want to know.

When your startup has reached maturity, your customers may ask for external verification of your security. It becomes time for the following actions.

- Test your application with external penetration testing.
- While your development team is growing, provide programmers and quality assurance

Tips

- Start small, but learn from big players: https://aws.amazon.com/security/, https://azure.microsoft.com/en-us/services/security-center/
- Each development environment has its own security implementation and best practice documents. Use them.
- Make sure to authenticate all services and validate all input.
- Explore security solutions for storing your application secrets (API keys, connection strings, etc.), such as AWS Parameter Store, GCE KMS or Azure Key Vault.
- Protect your source code. Make sure you map all access to source code and keep external backups.

representatives with security awareness training. All developers should be familiar with at least OWASP Top 10 Vulnerabilities.

Authentication and Authorization

SAAS-SU should be addressing Authentication and Authorization in two ways:

- Authentication (identity validation) and Authorization (granting permission) of SAAS-SU employees for cloud management components and internal servers (see the platform security chapter, below.)
- Authentication and Authorization of SAAS-SU customers to SAAS-SU services

Authentication to cloud apps is the first line of defense. Implementing advanced methods of user authentication and authorization will reduce risk of unauthorized access. The following applies to all SAAS-SU in all maturity phases:

- All customer accounts should be authenticated and authorized.
- Group your grants into roles as early as possible to reduce management.
- Establish a strong password policy so it is not the duty of customers to protect themselves.
- Use CAPTCHA to avoid authentication attacks. Some systems are able to offer a simple login and password form, and only switch to CAPTCHA after a first failed attempt.
- Validate your user's email address and phone numbers with email confirmations and SMS tokens.
- Audit login and logout activities whenever possible to help identify security incidents.
- Admin and privileged user accounts inside your application should support 2FA mechanisms.
- Do not attempt to recreate the wheel. Follow existing standards and Best Practices (e.g.: OAuth, SAML, OpenID) to help with future integrations. If possible, reuse existing authentication systems built on these standards and/or allow customers to use their own.

Tips

- Deploy a good directory solution with identity federation support when creating identity management policies.
- Deploy an Identity Federation solution when your customers act as identity provider (IDP) and you are the solution provider.
- Store your customer's password according to Best Practices. Most development frameworks provide libraries for implementing hashing, salting and more.
- Consider ways to improve your customers' user experience, such as examining carefully all steps required for a password reset.
- There are now many ways to achieve multifactor authentication, such as Google Authenticator. Use these tools.

- Enforce Federation and IP-based restrictions, if possible, for privileged accounts.

APIs: Do Not Neglect Security

- Provide as many APIs for your clients as possible in order to support self-serve, reduce overhead and increase client satisfaction—at both application and infrastructure levels.
- For each API access, cover Authentication, Authorization and Audit capabilities. Make sure API access does not bypass any ACL defined.
- Align API development with product roadmap and company maturity.

Tip

- Privacy may bring many good examples for the use of self-serve APIs (e.g. online tracking of personal data being collected by the service, self-deletion/ update of client data, etc.).

Secure Software Security Development Lifecycle (SSDLC)

According to security best practices, optimal implementation of security should be incorporated into systems by design. SSDLC is a framework that defines the process of developing an application security, and is made from implementing tools, processes and methods, such as threat modeling, secure code review, code analysis, vulnerability testing and continuous training for employees. The goal of SSDLC is to make your system more secure.

Implementing full SSDLC on all development aspects is difficult even for mature organizations. SAAS-SUs should carefully implement the different aspects of SSDLC based on their own development maturity, budget and resources.

An important component of SSDLC is mapping and controlling the supply chain of software, including the provenance of open-source libraries, operating systems and other third-party software. (Heartbleed vulnerability in Open SSL is a good example.)

- In the first phase of an SAAS-SU's lifecycle, consider implementing threat modeling for major version upgrades. Later, with maturity, threat modeling will be used with every major feature. It is also important to separate your development and production environments.
- In the second phase of SAAS-SU lifecycle, use dynamic analysis tools and external penetration tests to test code for vulnerabilities. When development teams are growing, set ongoing security training for developers.

Tips

- To avoid becoming a bottleneck during phases of development, automate code analysis and security testing.
- Domain 10 of the CSA Guidance contains more information about application security in cloud environments.

- In **Phase 3: Maturity** of SAAS-SU lifecycle, add in more analysis tools, such as static analysis. Also change management processes should be mature by now with automating build and release mechanism.

The cloud platform is the IaaS/PaaS platform on which the SAAS-SU is developing. While securing the cloud platform is a responsibility of the cloud provider, securing the running instances and the management dashboard are a cloud consumer's responsibility.

Platform Security

The cloud platform is the IaaS/PaaS platform on which the SAAS-SU is developing. While securing the cloud platform is a responsibility of the cloud provider, securing the running instances and the management dashboard are a cloud consumer's responsibility.

General Recommendations

- Understand the shared responsibility model between you and your provider. Most mature providers have detailed documentation on this topic.
- Ensure you are using secure data backup strategy and processes. Always keep copies of critical backups outside the cloud environment and be sure to run through restore procedures to ensure the integrity of backups. Encrypt backups if they include sensitive data.
- Consider deploying to more than one region in your cloud provider platform in order to increase resilience to region-level failures.

Management (Client) Dashboard

The IaaS/PaaS management dashboard is a primary attack vector. Failure to protect it can result in an inability to access, manage or provide your services for good.

The following procedures should be implemented during Phase 1 of the SAAS-SU lifecycle:

- Follow your service provider's security Best Practices checklist.
- Avoid using the Master/Root account on your dashboard. Instead, create and use sub-accounts with relevant and least privileged roles.
- Protect your admins with 2FA. Revoke unused API keys.

Tips

- Activate management dashboard logging tools (e.g. AWS Cloud-Trail) from the first day of development.
- Create roles for operations admins, and separate account management to different roles. Check this GITHUB project for examples.
- Explore cross-account permission to limit blast radius in case of account hijacking.
- Use a designated email address for your master cloud account to protect against phishing.

- Protect your DNS with a trustable provider. Domain Name registrants and operators have become popular attack vectors.
- Store API keys and other secrets in a safe location.

Data Flows and Network Separation

During Phase 1, plan for the following:

- Resource separation. Separate production, test and development environments, and separation different services and roles.
- Visualize your assets and present a simple and clear diagram or chart to help clients gain a better understanding about the system in which they place their trust.
- Follow Best Practices such as using a three-tier architecture.
- Always use VPN to connect to your cloud data center.

Tip

- When developing in PaaS, usually there is no control of the underlying OS. In these cases, put your security emphasis on other compensating controls, such as application security.

In the second phase, consider implementing a cost-effective Distributed Denial of Service (DDOS) protection. DDOS protection services can later be upgraded to include advanced services, such as Web Application Firewall (WAF).

Physical Security

- Physical security of your servers is probably managed by your IaaS/PaaS provider. Collecting your data center's SOC 2 reports in advance can be useful, as you will likely be asked for them by your prospects and customers. Creating an established process for sharing your data center's SOC 2 audit reports with your clients is a proactive move that can build trust.
- Don't forget that physical access to your SAAS-SU offices is a consumer responsibility. Implement good physical security practices. If using a shared-office location, for example, boost logical access controls (such as VPN/2FA) and boost passwords to workstations.

Tip

- Make sure to proactively set a process for sharing your data center's SOC 2 audit reports with your clients.

Protecting Your Instances

When developing on IaaS, it is the cloud consumer's responsibility to make sure virtual machines are fully patched, hardened and scanned periodically. Due to the dynamic nature of cloud services, we recommend automation of these features.

Tip

- Use already hardened images (e.g. CIS hardened images in AWS marketplace).

Encryption and Key Management

Encryption of data in transit is mandatory for complying with many regulations and standards, and should begin with Phase 1. Some laws or regulations (HIPAA, PCI) require encryption of data at rest, which is challenging. Data at rest can be encrypted at various levels (e.g. hardware, storage, OS, DB, File), and proper analysis is necessary to choose the best solution for your requirements.

Tips

- Encrypt internal traffic between servers to satisfy compliance efforts.
- Encrypt backups, including server snapshots.
- Let's encrypt is a free, automated service for generating SSL certificates.

- Use standard encryption protocols and algorithms (IPSEC, TLS) for all data in transit, including customer's data, APIs traffic, and remote access connections.
- Use cloud provider tools to encrypt data at rest. Most tools support encrypting volumes and DBs with provider-managed keys. These solutions are simple and relatively inexpensive.
- Once matured (Phase 3), consider moving your keys to a Hardware Security Module (HSM) or key management service. For highly regulated industries, this strategy can become a marketing advantage.
- Ensure appropriate rotation of keys.
- Segregate duties of managing and encrypting keys.
- Log all activities pertaining to lifecycle of encryption key management: key generation, pre-activation, activation, expiration, post-activation, escrow, and destruction.

Security Management

Tips

- Policies should be clear, concise and friendly so all employees can understand and follow them.
- Focus on what you think is most important in order to address your specific security risks, and avoid using cut-and-paste policies. Consulting with other companies can be beneficial, as well.
- Consider implementing an open status page for services (i.e. Google status page).
- Refer to Domains 7 and 8 of CSA Guidance for more information about Disaster Recovery and Business Continuity in cloud environments.

Tips

- Identify the most effective standard with which your company may comply, according to your industry and prospective clients. Do not over-qualify your organization with an unnecessary array of expensive accreditations.
- If your customers are enterprises, expect to have a mandatory request to comply with either SSAE16 SOC 2 or the IS27001 certification.
- If you are hosting protected health information (PHI) of U.S. citizens, you must comply with the HIPAA security and privacy rules.
- If you are hosting Personally Identifiable Information (PII) in the EU or about EU citizens, you will need to comply with EU GDPR laws.
- Consider getting accredited with advanced standards (ISO 27018/ ISO 27017/CSA STAR) if you want to differentiate your services as highly secure.

Transparency

Transparency is an important cornerstone when attempting to win customers' trust. Without true transparency about SAAS-SU operations, it will also be difficult to demonstrate company maturity. To be transparent, document relevant security policies and share them with your customers.

Starting in Phase 2, SAAS-SUs should be able to confirm security for customers with the following documents:

- Security policy and incident management.
- Disaster Recovery and Business Continuity Policy with clear Recovery Point/Time Objectives (RPO and RTO).
- Privacy Policy (may have been required in Phase 1)

Industry Standards

Aligning with industry standards makes it easier for your clients' audit/compliance departments to attest they have properly vetted their vendors. Vendors should strive to comply at least with one of the major IT Security and Management standards (i.e. ISO-27001, CoBIT or SSAE 16 SOC2/ISAE3402). In addition, vendors should adhere to industry-specific standards

(i.e. ISO 22307 for financial services). Mapping relevant standards to build a roadmap for compliance is part of security management. The compliance roadmap should be aligned with your product roadmap.

Vulnerability Assessments

All SAAS-SU should undergo periodic vulnerability assessment and penetration tests on their environment. The goal is to provide unbiased evidence of the security of your service's infrastructure/system. Scanning should include front-end and back-end components, as well as third-party components integrated into your SAAS-SU application (e.g.: Heartbleed in OpenSSL).

Tips

- Provide SAAS-SU customers with a summary of your security findings and relevant mitigation plans from the latest vulnerability assessment. Some clients will settle for a formal attestation letter by a professional third-party auditor.
- Scanners integrated into cloud platform management can provide useful automation advantages, such as scanning newly added servers upon launch. Consider implementing one of these tools.

Tips

- Index logs and events in a way that makes it easier to isolate logs per customers.
- Automate log collection, correlation and alerting. Focus on actionable alerts.
- Consider buying cyber insurance, which is becoming mandatory.
- If you are hosting PII, you may be required by law to perform breach notification procedures for security breaches. Encrypt PII to lower risk exposure.

Incident Response

SAAS-SU customers expect vendors to be able to contain any security event that might impact customers, and to notify them with full transparency. The goal of Incident Response is not to create an all-encompassing methodology to tackle every potential use case, but rather devising basic procedures to identify security incidents and demonstrate some proven level of due-care when handling them. Some aspects of SAAS-SU services are also relevant to security incidents (i.e. loss of availability). Align SLA with incident response procedures. Incident response policies should be created during Phase 2.

Log Management

Collecting, indexing and analyzing logs and audit trails are mandatory procedures to comply with regulations, and are the root enabler of most other security controls.

- Plan ahead. Think of how you can easily index, search, access, aggregate and filter logs.
- In Phase 1, activate logging services in your application and IaaS/PaaS dashboard. (You may not have the capacity to analyze them at this point.)
- In Phase 2, consider logging to external specialized logging providers. Most vendors have logging available for 90 to 180 days.

Tips

- Provider's tools, such as AWS Cloudtrail/ Cloudwatch-logs and GCE StackDriver logging, are efficient tools to start your log collection processes.
- When your startup is still small, simply collect and store logs. When you begin to grow, add more real-time detection and analysis capabilities.
- A reliable source for detecting security events are the SAAS-SU applications logs. Use them and make sure they contain all relevant information.

- Where possible, implement log validation controls in order to verify log integrity for the purpose of investigations. To meet some standards, log validation controls are mandatory.
- If your logs may collect or include sensitive data, ensure they are encrypted in transit and at rest, and that data is masked.

Tips

- Familiarize yourself with the EU GDPR (European Union General Data Protection Regulation) and other applicable privacy laws and regulations.
- Consult with privacy law professionals. If you host EU PII in the United States, consider becoming certified with Privacy Shield. For a more generic option, use the Standard Model Clauses.
- Implement customer data purging and data portability solutions, preferably via self-service.
- Use the Cloud Security Alliance Privacy Level Agreement for mapping PII collections, and complying with regional privacy laws and regulations.

Data Processing Limitations

Local privacy laws may require strict personal data processing limitations from cloud service providers.

- Ensure your platform can support data-processing, regional-based limitations, as well as prompt deletion of data, in cases of contract termination.

Appendix I

Security Essentials Maturity Model
Applying security controls as you grow.

In order to effectively apply security controls as your startup grows, consider at what point you will need to implement various controls. The following graphic highlights important milestones in each maturity phase. *Note that this list does not replace your need to build security strategy for SAAS-SU, in accordance with business-use cases.*

	1st Phase: From idea to first customers	2nd Phase: Growth and addition of customers	3rd Phase: Continued maturity and growth
SECURITY MANAGEMENT	Choose providers wisely	Transparency	Industry standards
	Collect logs	Data processing limitations	Incident response
	Security policy foundation	Industry regulations	Analyze logs
APPLICATION SECURITY	Threat modeling	Application security awareness training	Static analysis
	Protect application secrets	Dynamic analysis	Implement APIs
	Authentication mechanism maturity	External penetration tests	Mature SDLC
PLATFORM SECURITY	Protect management dashboard with 2FA and roles	Vulnerability and patch management	Physical security
	Segmentation of services	Encrypt data at rest	DDOS as a service
	Encrypt data at motion	Protected and tested backups	Use HSM

*Some industries may require compliance with standards as a prerequisite.