The SaaS CTO Security Checklist

A PROJECT BY sqreen

This is a basic checklist that all SaaS CTOs (and anyone else) can use to harden their security. Security shouldn’t feel like a chore. Implement the rules adapted to your company stage to improve your security. This list is far from exhaustive, incomplete by nature since the security you need depends on your assets.

YOUR COMPANY 1/2

○ Ensure your domain names are secured

SEED / SERIES A / POST-SERIES A

Domain names should be renewed regularly. If you have bought one from a third party you should also make sure that the authoritative configured name server is your own.

Read more:
http://www.esecurityplanet.com/views/article.php/3928456/8-Tips-for-Protecting-Your-Domain-Names.htm

○ Be honest and transparent about any data you collect

SEED / SERIES A / POST-SERIES A

In the case of a breach, people will disclose any data they gather. Your customers need to be aware of what data you're storing.

Read more:
https://hbr.org/2015/05/customer-data-designing-for-transparency-and-trust

○ Make sure all your critical services are secured

SEED / SERIES A / POST-SERIES A

Many companies rely on Google Apps, Slack, Wordpress... These services all have defaults that should be improved to increase the security level. All these services should be updated on a regular basis when relevant.

Read more:
https://blog.trailofbits.com/2015/07/07/how-to-harden-your-google-apps/
https://codex.wordpress.org/Hardening_WordPress

○ Do not share Wifi

SEED / SERIES A / POST-SERIES A

Sharing Wifi networks with guests or neighbors may give them the opportunity to gather information on your network, and allow them to access resources protected by source IP. Use an isolated and dedicated guest Wifi network. Set up a calendar reminder to change the password every two months, since this password is shared.

○ Take special care of your non tech employees

SERIES A / POST-SERIES A

Non tech employees are less used to technical tricks and can be deceived more easily than others, opening the door to ransomware or confidentiality issues. They should be trained and empowered to be distrustful and to preserve the company's assets.

Read more:
http://www.zdnet.com/pictures/hacked-the-six-most-common-ways-non-tech-people-fall-victim

○ Have a public security policy

SERIES A / POST-SERIES A

This is a page on your corporate website describing how you plan to respond to external bug reports. You should advise you support responsible disclosure. Keep in mind that most of the reports that you receive probably won't be relevant.

Read more:
https://www.airbnb.com/security
https://www.apple.com/support/security/

YOUR COMPANY 2/2

○ Have an internal security policy

POST-SERIES A

This is a short document stating the security requirements in your company and defining who is responsible and who is concerned with all aspects of security.

Read more:
https://www.sans.org/reading-room/whitepapers/policyissues/creating-information-systems-security-policy-534

○ Set up a bug bounty program

POST-SERIES A

A bug bounty program will allow external hackers to report vulnerabilities. Most of the bug bounties program set rewards in place. You need security aware people inside your development teams to evaluate any reports you receive.

Places to start:
https://bountyfactory.io/
https://hackerone.com/
https://cobalt.io

○ Make an inventory of your company's assets

POST-SERIES A

An awareness of your company's assets enables you to monitor the points that need the most attention and vulnerabilities that need to be hardened.

Read more:
http://advisera.com/27001academy/knowledgebase/how-to-handle-asset-register-asset-inventory-according-to-iso-27001/

○ Have a security incident response plan

POST-SERIES A

This will allow whoever is in charge at the time of a breach to communicate accordingly about an incident and will allow the fastest response in technical / communication terms.

Read more:
https://zeltser.com/security-incident-response-program-tips/

YOUR EMPLOYEES 1/2

○ Accustom everyone to security practices

SEED / SERIES A / POST-SERIES A

Humans are often the weakest links in the chain of security. By explaining how an attacker could infiltrate your company, you will increase their awareness and thus minimize the chance of them falling for such a trap.

Read more:
http://www.govtech.com/blogs/lohrmann-on-cybersecurity/Ten-Recommendations-for-Security-Awareness-Programs.html

○ Require 2FA in your services

SEED / SERIES A / POST-SERIES A

Your employees should all use 2-factor authentication. It means that if their password gets stolen, the attacker cannot use it without the second factor. As a CTO your role is to make sure everyone complies with this rule.

Read more:
https://en.wikipedia.org/wiki/Multi-factor_authentication
https://support.google.com/a/answer/184711
https://get.slack.help/hc/en-us/articles/212221668-Require-two-factor-authentication-for-your-team

○ Encrypt all employee laptops & phones

SEED / SERIES A / POST-SERIES A

By encrypting all laptops, you protect both your company's assets, and your employee's private files.

Read more:
https://support.apple.com/en-us/HT204837
https://wiki.archlinux.org/index.php/Dm-crypt
https://support.microsoft.com/en-us/instantanswers/e7d75dd2-29c2-16ac-f03d-20cfdf54202f/turn-on-device-encryption

Locking the employee's phone is the same, and will protect against both pranks and accidents (e.g. an employee's child accidentally wiping a mailbox).

○ Accustom your team to locking their machines while away

SEED / SERIES A / POST-SERIES A

Your office may be secured, but you will eventually have to receive external people for a party or a meeting. Locking all the machines is a great habit. If you get in the habit of locking your machine at the office, you'll be unlikely to forget to also do it in a Starbucks or at a meetup.

Read more:
https://www.cnet.com/how-to/7-ways-to-lock-your-macbook

○ Use a password manager to ensure you only use strong passwords

SEED / SERIES A / POST-SERIES A

Using a complex and unique password for every website is great advice, but it can be very difficult to remember all of them. Password managers are a great way to manage these, since they will remember everything for you with a master password.

Great password managers are:
https://www.dashlane.com
https://lastpass.com
https://support.apple.com/en-us/HT204085

YOUR EMPLOYEES 2/2

○ Follow an onboarding / offboarding checklist

SEED / SERIES A / POST-SERIES A

This checklist should contain a list of all the steps you need to enforce when an employee, contractor, intern, etc... joins your company. A similar list can also be used when the someone is leaving your team.

Great examples from Gitlab:
https://about.gitlab.com/handbook/general-onboarding/
https://about.gitlab.com/handbook/offboarding/

○ Do not share accounts

SERIES A / POST-SERIES A

Sharing a user account makes it hard to understand who is using the service or to identify who has performed a given action.

○ Use centralized account management

SERIES A / POST-SERIES A

A centralized place with all user authorizations is the best way not to forget anything once you need to update a user profile (e.g. if an internship came to its end). It is also great place to define standard account creation you need for a given user.

Configuring with Google Apps:
https://support.google.com/a/answer/6087519

YOUR INFRASTRUCTURE 1/2

○ Use SSL certificates to secure people using your website

SEED / SERIES A / POST-SERIES A

Encrypting communications is not only about privacy, but also about your users' safety, since it will prevent most attempts at tempering with what they receive.

Two free popular solutions are:
https://letsencrypt.org/
https://www.cloudflare.com/ssl/

You can also choose your own custom certificate (which may allow you to get a beautiful green bar if you pay for the extra "Extended Validation"):
https://www.digicert.com
https://www.rapidssl.com

○ Check your website's basic security

SEED / SERIES A / POST-SERIES A

Websites are vulnerable to many different classes of vulnerabilities, some may be prevented by the appropriate configuration on the server. Such headers include HSTS, X-Frame-Options, X-Content-Type-Options, etc… some of which will be very valuable for your user's protection. Static websites may expose your users to less risks.

Check your website configuration:
https://myheaders.sqreen.io
https://securityheaders.io
https://www.ssllabs.com/

○ Isolate assets at the network level

SEED / SERIES A / POST-SERIES A

Only your public APIs should be exposed to the Internet. You should isolate your networks to prevent any unauthorized accesses to your database. This will prevent attackers from connecting to it and attempting to crack the password - or exploit vulnerabilities.

Read more:
http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Subnets.html

○ Keep your OS up to date

SEED / SERIES A / POST-SERIES A

You should download all of your OS's security updates and regularly update your machines. For servers, you can delegate it to a PAAS provider (Heroku, AWS Beanstalk, etc…).
https://appcanary.com/

○ Backup

SEED / SERIES A / POST-SERIES A

Backup all your critical assets. Ensure that you attempt to restore your backups frequently so you can guarantee that they're working as intended. S3 is a very cheap and effective way to backup your assets:
https://aws.amazon.com/getting-started/backup-files-to-amazon-s3/

YOUR INFRASTRUCTURE 2/2

○ Restrict internal services by IP addresses (your company's ISP, VPNs, etc...)

SERIES A / POST-SERIES A

Everything non-public should only be accessible through a bounce host (e.g. no direct access to databases).

Read more:
https://aws.amazon.com/fr/blogs/security/securely-connect-to-linux-instances-running-in-a-private-amazon-vpc/

○ Centralize and archive your logs and make them meaningful

SERIES A / POST-SERIES A

Logs are necessary to trace what happened after an incident, find where the attacker came from, and possible even who they are. Many solutions exist to gather your logs. You need to take care about that the system time configured on each of your machines is in sync so that you can easily cross-correlate logs.

Read more:
https://en.wikipedia.org/wiki/Network_Time_Protocol
https://www.elastic.co/products

○ Protect your application from DDoS attacks

SERIES A / POST-SERIES A

A Distributed Denial-of-Service Attack (DDoS) can have devastating consequences on businesses. Basic DDoS protections can easily by integrated with a CDN such as:
https://www.cloudflare.com/
https://aws.amazon.com/fr/cloudfront

○ Protect your application from DDoS attacks

SERIES A / POST-SERIES A

This is built-in if you are using a cloud service and all your machines are registered / spawned through it. Otherwise, you will need to create and maintain a list of your assets (servers, network devices, etc...), and review it regularly to determine if you still need them, keep them up to date, and ensure that they benefit from your latest deployments.

○ Watch for unusual patterns in your metrics

SERIES A / POST-SERIES A

Takeovers will often be used to steal your data or setup your servers to be used as bouncers. These can be detected by watching for unusual patterns in metrics such as network bandwidth, CPU and memory consumption, and disk usage.

https://newrelic.com/server-monitoring
https://www.sysdig.com/

○ Know how t redeploy infrastructure from scratch

POST-SERIES A

This allows you to quickly spawn new infrastructure and populate it with data from your backups. This is the perfect use case for disaster recovery.

Read more:
https://aws.amazon.com/cloudformation/
https://cloud.google.com/deployment-manager/

YOUR CODE 1/2

○ Enforce a secure code review checklist

SEED / SERIES A / POST-SERIES A

Security should always be kept in mind while coding. Pull requests should be performed with security in mind as well. Depending on where the code is, the checks should be different. Dealing with user entry is one thing, dealing with business structures is another: the concerns are related to the context. In addition to common sense, keep in mind the typical security flaws. Security is also a good topic to ask about when interviewing a candidate.

Read more:
https://www.owasp.org/index.php/Top_10_2013-Top_10

○ Use a Static Security Code Analysis tools

SEED / SERIES A / POST-SERIES A

Static code analysis tools can quickly overwhelm you with a lot of meaningless false-positives. But switching on security-focused tools can help you discover vulnerabilities inside your code and most importantly increase the security awareness inside your team. Integrate these tools with your workflow to reduce friction. Post-commit checks that automatically comment where code reviews are performed are ideal.

Tools:
https://www.codacy.com/
https://www.owasp.org/index.php/Source_Code_Analysis_Tools

○ Maintain a backlog of security concerns in your issue tracking tool

SEED / SERIES A / POST-SERIES A

Every developer should contribute to maintaining a list of security issues to be fixed in the future. Making them available to the rest of the team will increase the security awareness in the company.

○ Never do cryptography yourself

SEED / SERIES A / POST-SERIES A

Always rely on existing mechanisms, libraries and tools. Cryptography is an expertise. Building your implementations, or using flags and options you don't fully understand will expose you to major risks. Libraries such as na.cl (https://nacl.cr.yp.to/) expose few options and restrict you to the good choices.

○ Keep secrets away from code

SEED / SERIES A / POST-SERIES A

Never commit secrets in your code. They should be handled separately in order to prevent them accidentally being shared or exposed. This allows a clear separation between your environments (typically development, staging and production).

Read more:
https://12factor.net/

○ Perform security oriented test sessions

SERIES A / POST-SERIES A

Once in a while, the entire technical team should sit together and spend time targeting all parts of the application, looking for vulnerabilities. This is a great time to test for account isolation, token unicity, unauthenticated paths, etc... You will heavily rely on your browser’s web console, curl, and 3rd party tools such as Burp (https://portswigger.net/burp/).

Read more:
https://www.owasp.org/index.php/OWASP_Testing_Guide_v4_Table_of_Contents

YOUR CODE 2/2

○ **Use a secure development life cycle** POST-SERIES A

The secure development lifecycle is a process that helps tackle security issues at the beginning of a project. While rarely used as is, it provides good insights at all stages of the project, from the specification to the release. It will allow you to enforce good practices at every stage of the project life.

Read more:
https://en.wikipedia.org/wiki/Systems_development_life_cycle

YOUR APPLICATION

○ Run it unprivileged

SEED / SERIES A / POST-SERIES A

In case an attacker successfully attacks your application, having it running as a user with restricted privileges will make it harder for the attacker to take over the host and/or to bounce to other services. Privileged users are root on Unix systems, and Administrator or System on Windows systems.

○ Monitor your dependencies

SEED / SERIES A / POST-SERIES A

Applications are built using dozens of third party libraries. A single flaw in any of these libraries may put your entire application at risk. Some tools allow you to monitor your dependencies against vulnerabilities:

https://appcanary.com/
https://snyk.io/
https://gemnasium.com/

○ Use a real-time protection service

SERIES A / POST-SERIES A

These tools protect web applications from attacks at runtime. The protection logic is inserted into applications. They protect against all major vulnerabilities (SQL injections, XSS attacks, account takeovers, code injections, etc...) without false positives.

https://www.sqreen.io/
http://www8.hp.com/us/en/software-solutions/appdefender-application-self-protection/

○ Hire an external penetration testing team

POST-SERIES A

These take an external and naive point of view of your infrastructure and products. Pentesters will take nothing for granted and will check even the most basic assumptions, as well as all of your infrastructure. You can also ask them to start with a full, blind discovery of your infrastructure; which can help you remember about old assets.

Read more:
http://www.zdnet.com/article/10-things-you-need-to-know-before-hiring-penetration-testers/

YOUR PRODUCT USERS

○ **Enforce a password policy**

SEED / SERIES A / POST-SERIES A

Your user accounts will be way harder to steal if you require them to use complex passwords: mixed case, special characters, minimum length...

○ **Encourage your users to use 2FA**

SERIES A / POST-SERIES A

As you get higher profile customers, you will be required to implement stronger security practices. This includes offering them 2FA, role-based account management...

Read more:
https://auth0.com/
https://stormpath.com/

○ **Monitor your user's suspicious activities**

SERIES A / POST-SERIES A

Some users may behave suspiciously, trying to hack into your application, subvert your services or bother your other customers. By monitoring such users, you will be able to block or flag the illegitimate ones.

Great tools:
https://www.sqreen.io/
https://castle.io