2ND EDITION

The SaaS CTO Security Checklist

sqreen

INTRODUCTION

This is a basic checklist that all SaaS CTOs (and anyone else responsible for security in their organization) can use to harden their security. Security shouldn't feel like a chore. Implement the rules adapted to your company size to improve your security. This list is not exhaustive, since the security you need depends on your specific assets and business, but it can provide a great foundation.

The security world evolves quickly. Since 2016, there have been new advancements and security issues that merit consideration. In the second edition, we've updated and expanded upon the original checklist to reflect the latest best practices for CTOs and security owners in SaaS startups and mid-market companies.

The Sqreen Team
@SqreenIO

YOUR COMPANY

✓ Ensure that your domain names are protected

Domain names should be renewed regularly. If you bought one from a third party, you should also make sure that the authoritative configured name server is your own. Take a few precautions when registering your domain to make it more difficult to hijack, including transfer locks and using an account owner email on a different domain.

Read more:
- https://www.icann.org/news/blog/do-you-have-a-domain-name-here-s-what-you-need-to-know-part-4
- https://www.eurodns.com/blog/domain-name-security-best-practices

---

✓ Protect against domain name phishing

Some attackers buy domain names that are similar to yours, by dropping letters or using homoglyphs. For instance phishng.com instead of phishing.com. Registering lookalike domain names will help you prevent against this. Also, monitoring Certificate Transparency can help in proactively detecting attacks.

✓ Be honest and transparent about any data you collect

Should you be breached, attackers may publicize the data that they gather. Your customers need to be aware of what data you're storing so they're not caught by surprise. Additionally, with GDPR now in place, you could face legal and financial repercussions if you collect data about your (European) customers and users that they haven't consented to give you. Ensure that you are clear about the data that you will collect from users that interact with you.

Read more:
- https://hbr.org/2015/05/customer-data-designing-for-transparency-and-trust
- https://www.enterpriseready.io/gdpr/preparing-for-gdpr/#
- https://github.com/privacyradius/gdpr-checklist

✓ Make sure all your critical services are secured

Many companies rely on 3rd-party services and platforms like Google Apps, Slack, and Wordpress. These services all have default settings that should be improved to increase their security level. All these services should be updated and checked on a regular basis, particularly when new versions come out.

Read more:
- https://blog.trailofbits.com/2015/07/07/how-to-harden-your-google-apps/
- https://support.google.com/a/answer/7587183?hl=en
- https://codex.wordpress.org/Hardening_WordPress
- https://medium.com/@longtermsec/more-tips-for-securing-your-g-suite-4d617bd04bc8
- https://get.slack.help/hc/en-us/articles/115004155306-Security-tips-to-protect-your-workspace

---

✓ Do not share your wifi network

Sharing your company wifi network 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 instead. Set up a calendar reminder to change the password every two months, since this password is shared among a potentially large number of people outside your organization.

✓ Have a public security policy

This is a page on your corporate website describing how you secure your users and their data, and how you plan to respond to external bug reports. You should advise that you support responsible disclosure. Keep in mind that you will likely receive reports of varying impact, so having a process for prioritizing them is important.

Read more:
- https://www.sqreen.com/resources/security-page
- https://www.airbnb.com/security https://www.apple.com/support/security/

✓ Have an internal security policy

This is a short document outlining the security requirements in your company for your employees and defining who is responsible and who they can turn to for all things security. Make this part of onboarding and ensure that it's easy to find.

Read more:
- https://hbr.org/2017/11/the-key-to-better-cybersecurity-keep-employee-rules-simple
- https://medium.com/starting-up-security/starting-up-security-policy-104261d5438a

---

✓ Set up a bug bounty program

A bug bounty program will allow external hackers to report vulnerabilities. Most of the bug bounty programs set rewards in place. You need security-aware people inside your development teams to evaluate any reports you receive, so make sure that you have the right internal resources before you set up such a program.

Read more:
- https://www.yeswehack.com/en/index.html
- https://hackerone.com/
- https://cobalt.io

✓ Create an inventory of your company's assets

A mapping of your company's assets enables you to monitor the points that need the most attention and vulnerabilities that need to be hardened. You can't understand your security if you don't know all the assets that should be secure.
For your servers, this is built-in if you are using a cloud service or a PaaS and all your machines are registered / spawned through it. Otherwise, you will need to review all your assets regularly to determine if you still need them, to keep them up to date, and to ensure that they benefit from your latest deployments.

Read more:
- https://resources.infosecinstitute.com/asset-management-guide-information-security-professionals/
- https://magoo.github.io/simple-risk/

✓ Have a security incident response plan

This will allow whoever is in charge at the time of a breach to communicate accordingly about an incident and will allow for the fastest response. Trying to make your plan up in the heat of the moment can make the impact of breaches much worse.

Read more:
- https://zeltser.com/security-incident-response-program-tips/
- https://github.com/meirwah/awesome-incident-response
- https://security.openstack.org/vmt-process.html
- https://medium.com/@magoo/incident-response-writing-a-playbook-773e7920f171
- https://www.amazon.com/How-Measure-Anything-Cybersecurity-Risk/dp/1536669741

✓ Build a security-friendly culture

Mistakes happen. People click on phishing emails, reuse passwords, or overlook vulnerabilities in their code. While you should focus on trying to prevent security breaches in the first place, it's also important to think about what needs to happen after a breach. From the culture side, the best thing you can do is try and minimize the time between a breach and you finding out about it. This means that your employees have to be trained to recognize potential security breaches, and that you have to build a culture that encourages them to report them. Everyone needs to understand that mistakes are possible and that if they fear that one has happened, they should report their doubt right away, rather than trying to hide it. This can only be achieved with a blameless attitude in the culture and a feeling of psychological safety. Work to instill those feelings.

Read more:
- https://securityculturefframework.net/
- https://i.blackhat.com/eu-18/Wed-Dec-5/eu-18-OBoyle-SDL-at-Scale-Growing-Security-Champions.pdf

---

✓ Work with compliance in mind

As you move into certain industries or upmarket to larger customers, your company might have to move to a formal security certification such as SOC2 or ISO/CEI 27001. Even without making a formal move in this direction, keeping in mind the security measures and the continuous improvement framework these measures recommend is a good inspiration. As you design your security practices, keep one eye on these frameworks and try to align where it makes sense for you.

Read more:
https://www.ispartnersllc.com/blog/4-areas-security-practice-soc-2/

✓ Prepare your security for scale

Scale comes to each company differently. In many startups, scale occurs on various levels, whether it is in the size of the sales team, a big growth in the engineering team, opening new offices, etc. Each of these scaling events brings specific challenges. For example, how are you gonna run your security onboarding in a new office abroad if the employees there don't speak the same language? It's important to evaluate your security policies and practices with each scaling event you have.
Read more:
https://ayeks.de/post/2018-06-11-automating_and_scaling_security/

✓ Leverage tools to prioritize your security

Early on, you'll want to focus on enhancing your security with smart internal practices. However, as you grow, it becomes more and more worth it to bring in some useful security tools. For instance, AWS offers AWS Trusted Advisor which, for a fraction of your billing, will provide you with actionable insights about your infrastructure security. Others can help you with different parts of your total security needs, from your application to your infrastructure.

Read more:
- sqreen.com
- https://aws.amazon.com/premiumsupport/technology/trusted-advisor/

YOUR EMPLOYEES

✓ Accustom everyone to good security practices

People are often the weakest links in any company's security. By holding trainings to explain how an attacker could infiltrate your company, you will increase their awareness and thus minimize the chance of them falling for common traps. Some things to cover include phishing emails, and the dangers of USB drives and email attachments.

Read more:
- https://www.secureworks.com/blog/cybersecurity-awareness-training-best-practices
- https://resources.infosecinstitute.com/top-10-security-awareness-training-topics-for-your-employees/
- https://sudo.pagerduty.com/

✓ Require 2FA wherever possible

Your employees should all use 2-factor authentication. By adding 2FA, you add an extra layer of security. Should your employee's password get stolen, the attacker would still be locked out unless they have access to the second factor (e.g. phone app or text) as well. As a CTO, your role is to make sure everyone complies with this rule. Phones are the most commonly used device for second factors, and thus have to be secured accordingly (e.g. with codes or biometry). Another option is to use purpose-built hardware-based 2FA, like Yubikeys.

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
- https://www.yubico.com/why-yubico/how-yubikey-works/

✓ Encrypt all employee laptops & phones

By encrypting all laptops, you protect both your company's assets, and your employee's private files. Encrypting your employee's phones is the same, and will protect their security in the case of either theft or accidents.

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

---

✓ Accustom your team to locking their computers

Your office may be secured, but you will eventually have to receive external people for a party or a meeting. Someone with physical access to an employee computer can do a lot of harm in a very short amount of time, so locking all computers 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.

Many organizations make it a game to catch fellow employees' computers unlocked. This is a good way to reinforce the habit of locking your computer for everyone. At Sqreen, for example, if someone catches another person's laptop unlocked while they're AFK, they can type "Cookies!" in that person's Slack. That person will then have to bring in cookies for the office!

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

Using a complex and unique password for every website is great advice, but it can be very difficult to remember all of them. Rather than reusing passwords or storing your passwords somewhere others could access, use a password manager. Password managers are a great way to manage multiple passwords across sites, since they will remember everything for you with a single master password, and can often generate unique strong passwords for you. Encourage your employees to do likewise, and purchase a business plan to a password manager if necessary.

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

✓ Follow an onboarding / offboarding checklist

Onboarding and offboarding are important security moments for your employees. You'll want to ensure that new employees enact the security measures needed and that your company follows the appropriate steps for employees who are leaving.

Your onboarding checklist should contain a list of all the steps you need to follow when an employee, contractor, or intern joins your company. A similar list can also be used when someone is leaving your team. Ensure that you deprovision all accounts they had access to.

Read more:
- https://github.com/92bondstreet/awesome-onboarding
- https://www.rippling.com/

✓ Do not share user accounts

Sharing a user account makes it hard to understand who is using the service or to identify who has performed a given action. This makes it much harder to recognize when an account has been taken over by an outside party. It also makes it harder to remove access to an account when employees leave the company, opening that account up to potential abuse.

✓ Use centralized account management

Having 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 a great place to define the standard account creation process you need for a given user. If you can, implement SSO to simplify and automate this process.

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

✓ Hire your first security engineer

As your company grows, you'll want to bring in a security expert and centralize your application security responsibilities on them. To determine if it's the right time to do so, ask yourself the following questions:
- Do we have a security roadmap?
- Do we manage to deliver on it?

If you don't, then it's the time to strategically consider what your security roadmap should look like, and to find out what kind of security engineer you need.

It's important to note that you shouldn't hire a specialized security person too early. Early on, security is something that needs to be baked into your engineering organization rather than immediately offloaded to someone else. Only when your engineering team is fully bought into security but is simply getting overwhelmed should you bring in a specialized security engineer.

Read more:
- https://medium.com/starting-up-security/hiring-the-cso-b737c30e098f
- https://www.darkreading.com/threat-intelligence/the--typical--security-engineer-hiring-myths-and-stereotypes/a/d-id/133334

✓ Monitor your user's computers

The more employees you have, the bigger the risk of them getting infected by malicious software, such as botnets. Using a HIPS system on employee hardware could help you get ahead of any problems via alerts and notifications.

Read more:
- https://www.stormshield.com/
- https://www.microsoft.com/en-us/windows/windows-defender/

YOUR INFRASTRUCTURE

✓ Use HTTPS to protect your users

Encrypting communications is not only about privacy, but also about your users' safety, since it will prevent most attempts at tampering with what they receive.
A free popular solution is:
https://letsencrypt.org/

Read more:
https://support.google.com/webmasters/answer/6073543?hl=en

✓ Check your website's basic security

Websites are exposed to many different classes of vulnerabilities, and some may be prevented by appropriately configuring the server. Best practices include adding headers such as HSTS, X-Frame-Options, X-Content-Type-Options, etc. Add in a Content Security Policy if possible.

Read more:
- https://www.sqreen.com/scanner
- https://securityheaders.com
- https://www.ssllabs.com/

✓ Isolate assets at the network level

Only your public APIs should be exposed to the Internet. You should isolate your networks to prevent any unauthorized access 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
- https://www.sqreen.com/resources/aws-security

✓ Keep your OS & Docker images up to date

You should download all of your OS's and Docker security updates and regularly update your machines and images. If you use a PAAS provider (Heroku, AWS Beanstalk, etc...), they will take care of this for you. If not, you will need to do it yourself. Ideally, automate this process if possible.

Read more:
- https://github.com/containrrr/watchtower
- https://spacewalkproject.github.io/

✓ Backup, test your backups, then backup again

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.

Read more:
https://aws.amazon.com/getting-started/backup-files-to-amazon-s3/

✓ Restrict internal services by IP addresses

Connections to your infrastructure and non-public properties (hosted CIs, admin interfaces, databases etc.) should only be accessible through a bounce host (in a VPC, behind a bastion host or VPN, etc.).

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

Logs are very useful for understanding what happened after an incident occurs, finding where an attacker came from, and possibly even who they are. Many solutions exist to gather and organize logs.

Don't forget, you need to take care that the system time configured on each of your machines is in sync so that you can easily cross-correlate logs. You'll have a much harder time if they're not (no pun intended).

Read more:
- https://en.wikipedia.org/wiki/Network_Time_Protocol
- https://www.loggly.com/
- https://www.elastic.co/products/kibana

✓ Protect your application from DDoS attacks

A Distributed Denial-of-Service Attack (DDoS) can have a real impact on your bottom line and customer experience. Basic DDoS protections can easily be integrated with a CDN, but there are purpose-built DDoS protection tools available as well.

Read more:
- https://www.cloudflare.com/
- https://aws.amazon.com/cloudfront/

✓ Monitor exposed services

Your developers constantly deploy new services. Step one is to ensure that you keep track of them, but you also want to ensure that they don't expose sensitive services to the outside world, (for instance, a database accessible from the Internet without network filtering). Using a network scanner will help you ensure that no unexpected services are exposed, and will tell you when new services are vulnerable and should be updated.

Check this cloud friendly tool:
https://intruder.io/

✓ Monitor internal services

It's a fairly common attitude to not focus on the security of your internal services as much as your external services. However, as you get bigger, you will lose visibility on the services used internally. When you start to lose track of internal services, they become a vector through which viruses or worms could spread. Additionally, more people (like contractors) will have access to your internal network. If it's not secured, this puts it at risk.

Read more:
https://www.tenable.com/downloads/nessus

✓ Watch for unusual patterns in your metrics

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 key metrics, such as network bandwidth, CPU and memory consumption, and disk usage.

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

✓ Know how to redeploy your infrastructure from scratch

Hopefully you never need to, but in the case of a disaster, 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

✓ Enforce a secure code review checklist

Security should always be kept in mind while coding. Pull request reviews 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 typical security flaws. For example, many code snippets from places like StackOverflow have not been written with security in mind. If your team pulls code snippets from the Internet, make sure they double check them for security before deploying them.

Security competency is also a good topic to ask about when interviewing a candidate.

Read more:
https://www.owasp.org/index.php/Top_10-2017_Top_10

✓ Use a pre-production analysis tool

Pre-production analysis tools like static code analysis (SAST) can help identify some of your low-hanging security fruits. They also improve the overall security awareness of your team when the checks are automatically integrated into the code review process. But keep in mind that these tools generate a lot of false positives that can quickly overwhelm you with meaningless alerts. The best practice is to make them part of your process, but not too rely too heavily on them.

Tools:
https://www.owasp.org/index.php/Source_Code_Analysis_Tools
Findbugs (Java)
Brakeman (Ruby)

✓ Add security bugs to your incident tracking tool

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

Treat security bugs like any other type of bug – determine their priority based on whether or not they are exploitable and the damage that could be done. Additionally, hold post-mortems for serious security bugs with the team to ensure that everyone gets visibility and learns from them.

✓ Never do cryptography yourself

Always rely on existing mechanisms, libraries, and tools. Cryptography is an expertise. Building your own 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 only a few options and restrict you to the good choices.

✓ Keep secrets away from code

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

Read more:
- https://www.infosecurity-magazine.com/opinions/comment-tips-for-private-key-management/
- https://www.digitalocean.com/community/tutorials/an-introduction-to-managing-secrets-safely-with-version-control-systems
- https://www.vaultproject.io/

✓ Perform security-oriented test sessions

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 Zap (https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project).

The benefit of doing these test sessions yourselves is that your team has the best understanding of your application, and likely where the weak points are. Showing that they can be exploited (or not) is valuable feedback for the team.

These sessions complement external pentests quite well.

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

---

✓ Use a secure development life cycle

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
- https://www.owasp.org/images/7/76/Jim_Manico_(Hamburg)_-_Securing_the_SDLC.pdf

---

✓ Onboard your software engineers with a security training

Secure applications start with secure developers. Your software engineers need to be aware of security best practices in order to write secure code and to perform security-minded code reviews. Since security is usually not something hiring managers consider during recruitment, an initial training at onboarding will help your devs reach a minimum level of security.

Also, consider checking for security competency during the hiring process. This will help you better shape your training.

Some good security training options:
- https://safecode.org
- https://sudo.pagerduty.com/

✓ Automate security within your SDLC

If your security practices impact your development velocity, they will be looked at as more of a burden than a valuable step. The best practices today are to take lessons from DevOps and find ways to bring security closer to developers. Leverage tools that can automate security checks and monitoring. Implementing automated SAST/DAST tools, vulnerability dependency scanning, and others will help you catch the obvious flaws before they get into production. Just beware that you'll have to sift through false positives and that these tools have limited scope.

Read more:
- https://en.wikipedia.org/wiki/Systems_development_life_cycle
- https://github.com/devsecops/awesome-devsecops

YOUR APPLICATION

✓ Run it unprivileged

In the case that an attacker does successfully attack 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.

✓ Keep track of your dependencies

Applications are built using dozens of third party libraries. A single flaw in any of these libraries may put your entire application at risk. According to OWASP, one of the most common application security risks is using dependencies with known vulnerabilities. Some tools allow you to check your dependencies for vulnerabilities and ensure that they are up-to-date.

Read more:
- Github
- https://sqreen.com/
- https://snyk.io/

✓ Use a real-time protection service, like a RASP

These days, WAFs are pretty outdated. It's better to use services that sit closer to your application. These tools protect web applications from attacks at runtime. An Application Security Management (ASM) tool can do for security in your application what APM tools do for performance. They can monitor and protect against all major vulnerabilities (SQL injections, XSS attacks, account takeovers, code injections, etc...) without false positives.

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

✓ Hire an external penetration testing team

Pentesters take an external and naive point of view of your infrastructure and products. They will take nothing for granted and will check even the most basic assumptions, as well as all of your infrastructure. The experience can help focus your security efforts and mindset.

Read more:
- https://www.softwaretestinghelp.com/penetration-testing-guide/
- https://blog.sqreen.com/leverage-pentest/
- https://www.sqreen.com/checklists/pentest-checklist

✓ Automate security once your app is in production

Several tools offer ways to automate custom security protection in production. Wherever possible, leverage your business information and logic to automate monitoring and protection of security situations systemic to your particular business. The more you can automate, the easier you'll be able to scale your security.

Read more:
https://docs.sqreen.com/security-automation/introduction-playbooks/

✓ Don't forget about your FaaS security

If you're using FaaS in your company, you should ensure that it's not a weak point for security. Make sure:
- Your code is centralized - either in a FaaS-specific repository or within the applications that the function depends upon
- Deployment is centralized in the CI. With FaaS abstracting things for you, it can be easy to forget about the different functions!
- Privileges used by the function are minimalist (and distinct from the privileges used to deploy it)

On top of that, FaaS should follow all the security criteria that you apply to your applications - from specifications, to development, to operating in production.

Read more:
https://techbeacon.com/enterprise-it/how-lock-down-your-serverless-apps-five-steps

YOUR PRODUCT USERS

✓ Enforce a password policy

Your users' accounts will be much harder to steal if you require them to use strong passwords. Ideally, stick with common strong password policy requirements, to prevent your users from getting frustrated at not remembering some arcane rule.

Read more:
https://www.digicert.com/blog/creating-password-policy-best-practices/

✓ Encourage your users to use 2FA and uplevel your authentication security

As you get higher profile customers, you will be required to implement stronger security practices. This includes offering them 2FA, role-based account management, SSO, etc. Often times, these features are entry level requirements for more enterprise deals.

Read more:
- https://auth0.com/
- https://www.okta.com

✓ Monitor your user's suspicious activities

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

Read more:
- https://www.sqreen.com/
- https://castle.io

✓ Double down on user privacy

Many successful attacks happen through social engineering. This means that access to your users' data has to be a big deal to you. Require a user's explicit consent before allowing support / sales to access their data. This access should also be audited. Good security hygiene here can make social engineering attempts to get into your users' data harder.

Read more:
https://resources.infosecinstitute.com/5-best-practices-for-ensuring-data-privacy/

---

sqreen

Trusted by security teams, loved by developers.

Monitoring and protection platform made to be incredibly powerful yet very easy to use.

Unmatched security insights: Access to more detailed security analytics than ever, including app-level incidents you can act on immediately.

Instant Protection: Out-of-the-box modules protect apps against a broad array of threats. Setup takes minutes, no config required.

Easily meet enterprise compliance needs: Get access to the best controls without hiring expensive security teams or consultants.

Start your free trial at sqreen.com

Want this handbook as a PDF?
Scan the QR-code, or go to:
https://www.sqreen.com/checklists/saas-cto-security-checklist