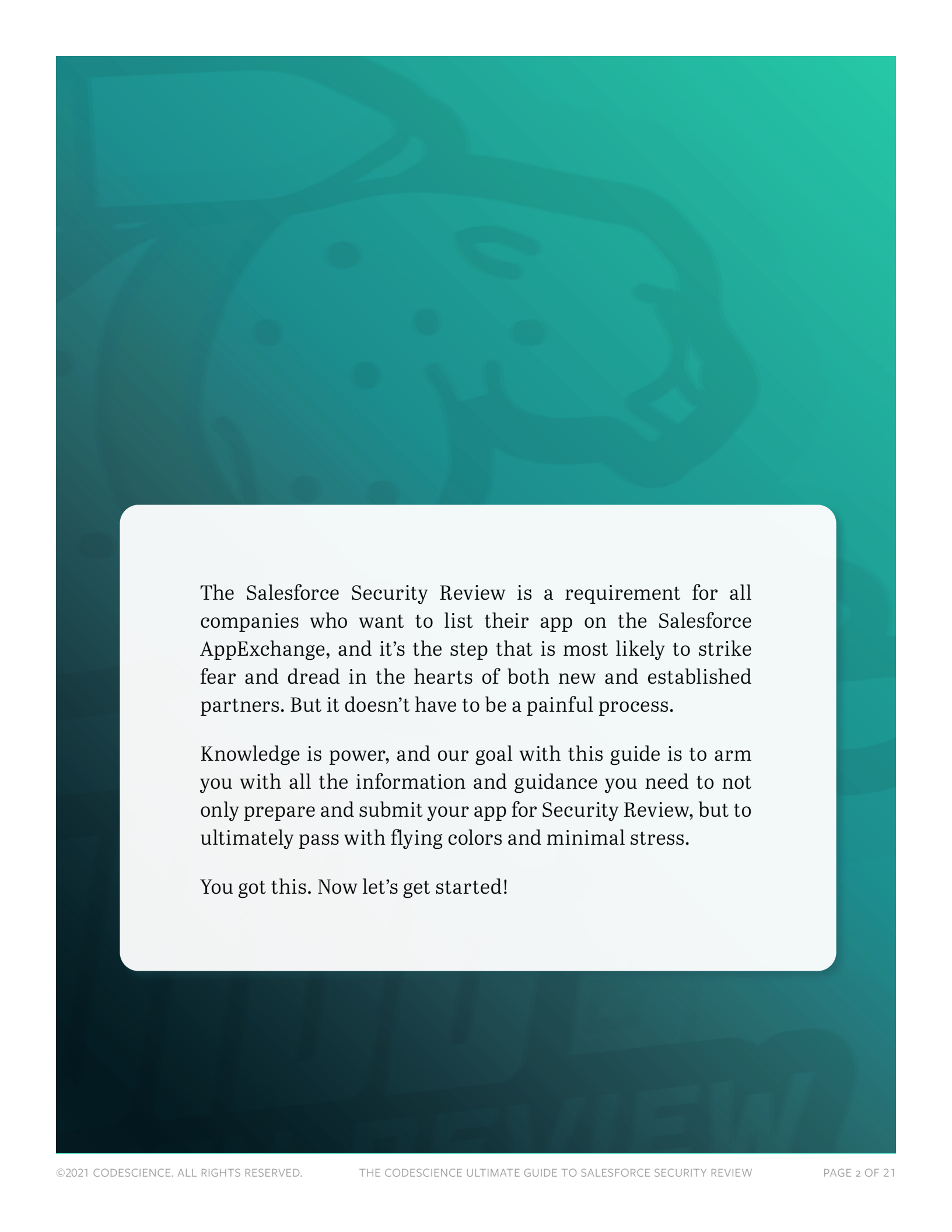


CODE|SCIENCE





The Salesforce Security Review is a requirement for all companies who want to list their app on the Salesforce AppExchange, and it's the step that is most likely to strike fear and dread in the hearts of both new and established partners. But it doesn't have to be a painful process.

Knowledge is power, and our goal with this guide is to arm you with all the information and guidance you need to not only prepare and submit your app for Security Review, but to ultimately pass with flying colors and minimal stress.

You got this. Now let's get started!

Table of Contents

Introduction to Salesforce Security Review	4
Why is Security Review Necessary?	4
Pass/Fail Stats	4
A Little Perspective	5
Why You Should You Trust Us	5
Planning for Success	6
Include Security Review in Your Project Plan	6
Project Timing	6
Office Hours	7
Dreamforce Cutoff Date	7
ISV Partner Agreement	7
What Version to Submit?	7
Build with Security In Mind	8
Code Scans	8
Checkmarx	8
BURP, ZAP, Chimera	9
False Positives	9
Code Reviews	9
Preparing for Submission Day	10
The Submission Process	11
The Security Review Wizard	12
Submission Day	13
Step by Step in the Wizard	14
Post-Review Office Hours	15
What Happens If You Fail?	16
Perspective	16
Be Ready to Respond	17
Make a Plan and Stick To It	17
Addressing Security Vulnerabilities	18
Code Changes & Dev Ops	18
Update the Demo Environment	19
New Scans & False Positives	19
Take Two!	19
Summary & Further Assistance	20
Salesforce Resources	20
Additional Resources	20
Need a Little Help?	21

Introduction to Salesforce Security Review

In this section, we'll explore what the Security Review process is and how it helps maintain the trust relationship within the Salesforce ecosystem, along with some perspective that will hopefully help you maintain a positive attitude throughout the process. We'll also explain who we are and why you should trust us.

Why is Security Review Necessary?

Trust is Salesforce's #1 value. In their words, "Nothing is more important than the trust of our customers." Salesforce customers trust the platform because they know the company takes the security of customer data very seriously and is committed to setting the standard for the confidentiality, integrity, and availability of this data.

It is critical that these security standards extend to every single product on the AppExchange, because the ecosystem is only as secure as its weakest link. If any individual app was to compromise the integrity of a customer's data, the trust relationship would be irreparably damaged.

This is why it is necessary for all apps to pass Security Review, even if they are not going to be listed publicly. It's how Salesforce ensures that all customers can be confident that their data will remain secure and they will not be exposed to security breaches when using any app on the AppExchange.

Pass/Fail Stats

We'll be honest with you: a lot of ISVs fail Security Review the first time. How many? Well, Salesforce doesn't release exact stats, but it's safe to say...most of them. For many, it can take two or three more tries before they pass.

If you follow the advice in this guide, you'll certainly have a much better chance at passing Security Review on your first try. But no matter how well prepared you are, it is very possible that you'll fail the first time you submit your app. *And that's ok.*

A first-round fail is not a dead end; it's just a speed bump. First-round fails are quite common — so much so that we've dedicated a full chapter of this guide to explaining **what to do if you fail**. To paraphrase an old song, you'll pick yourself up and dust yourself off, but you won't need to start all over again! Be ready to pounce on any vulnerabilities the Salesforce security team has found, resubmit your app, and you'll be right back on track to pass and get your app listed on the AppExchange.

A Little Perspective

While many ISVs dread the Security Review process due to its stringent, pass/fail nature, Salesforce does want you to pass! Security Review isn't some kind of gate where they're trying to keep people out for some arbitrary reason. They want you to succeed, get to market, and bring more value to their customers, but they also need you to do your part to ensure the safety and security of the ecosystem.

If you do fail on your first try, that's not entirely a bad thing. Salesforce is giving you great feedback so you can make your app more secure, and that helps keep the entire ecosystem more secure, stronger, and more resilient.

Why You Should You Trust Us

CodeScience has built over 350 applications — that's about 10% of the AppExchange! — so we've brought a wide variety of solutions through the Security Review process. We know what it takes to get through Security Review swiftly, efficiently, and successfully.

As Salesforce's only Master Product Development Outsourcer (PDO), we have the deep expertise and experience necessary to help you navigate the tricky parts of the Sec Rev process and avoid common mistakes and "gotchas."

Additionally, we maintain a very close relationship with the Salesforce security review team, so we're always up to speed on the latest updates and insider review team news. We've built this relationship of trust over many years by bringing hundreds of apps through Security Review and even helping the review team refine the process along the way. When we submit a CodeScience-built app for Security Review, their team knows they can expect a well-designed, highly secure solution.



Planning for Success

This chapter will illustrate how best to plan for Security Review so you're not waiting until the day you want to submit to think about security-minded topics. Making sure to bake security review prep into your development process is the best way to avoid dealing with a mountain of technical debt when you'd rather be submitting to the review team.

Include Security Review in Your Project Plan

How well you plan for Security Review will greatly determine your success, and timing is (nearly) everything!

Project Timing

Successful teams plan for Security Review from day one of development, taking into account the time necessary to pass the review, not just submitting your app. It's possible that you may have to re-submit once or twice and, because there is a waiting period after submission, you definitely don't want to submit on the day you plan to launch. In fact, you may want to consider a pre-MVP, "security review-only" version of your solution that can be submitted before your actual MVP is planned to be Dev Complete.

The typical turnaround for Salesforce's security review team to get through their review of your app can be up to six to eight weeks, though it's usually not quite that long. It all depends on the current backlog and time of year. Keep in mind that there's always a rush up to Dreamforce each year since a lot of ISVs hope to launch their product during the event. If you plan to submit your app in the lead up to Dreamforce, it would be wise to build in some extra time.

Pro Tip

Salesforce does offer a limited number of priority reviews. If you're able to score one of these priority slots, your app will be up for testing by the review team right away when you submit. This will typically cut the time it takes for your review in half and you'll likely get your results within three to four weeks. Your PAM is the best person to reach out to for more details or to request a priority review.



Office Hours

Be sure to book **Security Review office hours** early on in your planning so you've got them later when you'll likely have questions. Office hours give you direct, one-on-one access to Salesforce security review team members. It's a great time to ask questions about the submission process or troubleshoot issues. Office hours get booked up fast, so it's smart to block some time in advance, otherwise you might not be able to book them when you really need them.

When you book your office hours, you'll see that there's an intake form where you can explain your issue and upload relevant attachments. The more context you can give the person who's going to be staffing your office hours, the more prepared they can be to help.

Dreamforce Cutoff Date

If you plan to have your solution available for customer consumption by Dreamforce then you need to know that there is a cutoff date each year, several months prior to the event, by which all packages must be submitted in order to be guaranteed *one pass only* through Security Review, no more. With that said, it's very common for ISVs who fail at first to get approved in time for the big show, so don't worry too much about that "one pass only" guarantee.

ISV Partner Agreement

While the Security Review process is primarily technical, don't neglect the business side of things. As your solution is getting built out, getting your ISV partner agreement fully executed and signed is critical. Until your partner agreement is fully signed, the button to submit for security review will not be active, so you won't be able to submit your app even if you have all of your other documentation ready.

What Version to Submit?

As long as what you submit to Security Review is "representative of the ultimate solution," then you can submit a pared down version of the app for review while your team continues to work on the MVP.

One important note here is that you should not package or promote code to the packaging org before you pass Security Review, as the review team may require you to make changes to your package and re-submit. If you have included other components in the package since the version you submitted, all of those changes will be considered in scope for the review, and may cause delays. You are better off creating a side branch in your code repo for new development and then merging that branch back to master once you pass.

Build with Security In Mind

This section could really be its own book, and there are plenty of great resources from Salesforce on this already, so we're not going to dive too deeply into it here. You should definitely check out Salesforce's recommendations on **how to build a secure app**. We also recommend checking out the **Secure Coding Guide** and the **AppExchange Security Requirements Checklist** as you're getting started.

The more important document to review, however, is the Open Web Application Security Project (OWASP) **Top Ten Project**. OWASP is a nonprofit foundation that works to improve the security of software across the web, and their Top Ten Project is where they list the most critical security risks that appear in web apps. The list is regularly updated, since security issues are constantly evolving as new threats appear.

Being proactive about the security of your app by regularly reviewing and scanning your code as you build will help you minimize delays and avoid last-minute scrambles when it's time to submit for Security Review.

Code Scans

Several of the documents that ISVs are required to submit with their packaged application are the code scan reports from approved vendors. Salesforce makes several of these available to ISVs free of charge.

Checkmarx

For applications with an on-platform presence, like 99% of the apps on the AppExchange, you are required to run a code scanner called **Checkmarx** on your packaged metadata. This scanner produces a report which will let you know where you have security vulnerabilities and of what severity they are. Anything above "Information" should be addressed in the code and a new scan run to show that it no longer exists.

Pro Tip

It's important to keep in mind that the code scanner results will not identify every single instance of the vulnerability! It's up to you to ensure that your code has no evidence of any particular issue.



BURP, ZAP, Chimera

For those solutions that require an off-platform integration, i.e. a composite app, the review team requires that you run a **BURP**, **ZAP**, or **Chimera** scan on your endpoints and web UI. Salesforce is very alert to lateral attacks, which would be malicious data or code entering the platform via a partner integration and doing harm to customers. These scanners will scan possibly wider than you're expecting for just this reason.

At CodeScience, we always emphasize to our clients that the biggest risk to the Security Review timeline is the off-platform scanner and its results. If the BURP scanner does return some legitimate results, they will need to be fixed and re-scanned. For some development organizations, this is not a trivial issue given that they already have an existing roadmap and backlog lined up, and may not have the capacity to take on this additional work. Unfortunately it has to be done, so the sooner you scan your own systems, the better, so you can start planning for any remediation work.

False Positives

Some of the issues that are reported by the code scanners will not sound like a real vulnerability to you, and that's ok. If there's an issue that's a legitimate false positive and is not a threat, it can be documented as a false positive. It can't just be an issue that might be hard to fit into your current backlog, it has to be an actual false positive.

Pro Tip

As we mentioned earlier, it's a good idea to book **Security Review office hours** early in your project so that you can have them ready later on to validate your false positives with the review team.



Code Reviews

In addition to the code scanners that you'll run on the packaged and off-platform code, it's a good idea to have regular code reviews, ideally on every check-in to the repo. When we're building apps for our clients, CodeScience Technical Architects and Lead Devs review all code before it gets merged to our integration branch so that they can catch missed CRUD/FLS enforcement, among other issues. We also run scripts to scan for certain issues so that we don't have to rely on human eyeballs.

Preparing for Submission Day

In addition to the scans listed above, there are a few more documents to complete and set up to do in advance to be fully prepared to submit your app:

Detailed description of your app: While this is a straightforward requirement, you may not be ready to just write it off the top of your head. Spend some time beforehand to get this ready so you can just copy and paste the text into the Security Review Wizard when you're ready to submit. (We'll cover the **Security Review Wizard** in the next chapter).

Solution Architecture: This is a recent requirement that is basically asking for an overview of your application. There's no real set format for this, but it needs to include what you think the Salesforce review team should know about your solution and architecture. Check out the "pro tip" on this page for a lot more information on this one.

Use case document: This is another relatively recent requirement. At CodeScience, we include only the primary use cases, not how to use every last thing in the application. Put yourself in the security reviewer's shoes: they've never seen your app before, so they really don't know where to get started. Do you get into the solution through a custom button on a detail page, or does it have its own full page UI? That's the sort of question you'll be answering here.

Your use case document is a great way to put all the relevant information the reviewer needs right in front of them so they don't waste any time figuring out how your app works and can quickly get started examining the app, looking for attack factors and so on.

Pro Tip

Salesforce doesn't provide a whole lot of guidance on how to create your Solution Architecture documentation, so we developed our own template and decided to share it with you for free! Our template has been given the thumbs up from the Salesforce security team, though it's still up to you to make sure you include the correct info about your app. You can learn more and **download this free resource here**.



Demo org: Your demo org needs to be fully configured, so if it has an off-platform element, that needs to be fully authenticated and you need to have a test account on both sides. Basically, all setup should be done; you shouldn't expect the reviewer to be doing any kind of setup in your solution.

Ideally, your demo org should be a standalone developer org — and definitely not your packaging org — with nothing else in it. With that said, it's really helpful to add in some sample data to flesh out the app and give the reviewer a better picture of how it works.



The Submission Process

If you've been following the advice laid out in this guide so far, you've built your app with solid security baked in, you've run your scans, and you've made a plan for your review. The submission process itself should be relatively smooth sailing, but we still have plenty of tips to help you avoid any issues or missteps along the way!

Pro Tip

Your marketing team can begin filling out the content for your listing (including white papers, videos, screen shots, etc.) while you're preparing to submit your app for Sec Rev — there's no need to wait until after you pass. ISVs can get their app to market much faster by following this tip. If you've got your listing ready and your app passes Security Review, you can go public with your listing that very day.



The Security Review Wizard

The Security Review Wizard is the step-by-step wizard-based UI inside the Salesforce Partner Community where you'll submit your app for review. You don't need to wait until you're ready to submit your app to look into the wizard; as soon as you have a managed package uploaded, you can preview the wizard and walk through each of the steps — and you should!

Remember, it's never too early to begin preparing your answers and deliverables. Start early and take the time to get familiar with each section so you don't have any surprises on the day you submit. You can flip back and forth between the steps and check the wizard out as much as you need without actually hitting "submit" yet. It's a great way to get a preview of what you need and minimize your stress on submission day.

The screenshot shows the Salesforce Partner Community interface for the Security Review Wizard. The top navigation bar includes the Salesforce Partner Community logo, a search bar, and user information (2 Certifications, 150 Badges). The main navigation menu includes Home, Collaboration, Learn, News & Events, Support, Business, Manage (New), Publishing, and COVID-19. The breadcrumb trail shows Listings > New Security Review Wizard. The wizard has tabs for BUSINESS PLAN, APP, SECURITY REVIEW (selected), TEXT, MEDIA, TRIALS, LEADS, and PRICING. The 'Get Started with Your Security Review' section for BuildScience includes a brief introduction and five steps: 1. Trailhead AppExchange Sec... (Devise a solution security strategy and prepare for the security review), 2. Create a Secure Solution (Develop a solution that resists common security threats), 3. Pass the AppExchange Secu... (Prepare for a security review), 4. Complete the Security Revie... (Learn how to submit your solution for review), and 5. Review Fees (When you first submit a paid solution, you pay review and listing fees). A note at the bottom states: 'Start your security review submission. You can save your work at any time and come back to finish later.'

Submission Day

Again, if you've been following the recommendations in this guide, you've already prepared the deliverables you need to submit your app for Security Review and you did not wait until your submission day to gather all the information you need and get familiar with the Security Review Wizard!

There are a couple more bits of information you'll need to have on hand to get your app submitted:

Contact info: At CodeScience, we make it a rule to always use a distribution list for at least one of your contact options. You never want your messages from the Salesforce security team to go to only one person, because if you get a failure notice and that person is out that day (or week!), the rest of the team won't be aware and won't be able to start responding. Always set up a distribution list for your team and include it in your contact info.

If you use Gmail like we do at CodeScience, when you set up a Google Group for your security contact distribution list, you need to change the default settings so that it can accept messages from the public, not just your organization. Otherwise, you won't get any messages from Salesforce and you'll be wondering what's happening!

Payment: There is a \$2,700 fee for Security Review and you must use a credit card for this payment (Salesforce is unable to invoice for Security Review at this time, nor can you use PayPal, Venmo, etc.), so you'll want to have a corporate card on hand if possible. Of course you can always use a personal card, but keep in mind the card on file will be charged an annual \$150 fee in subsequent years. And make sure you get reimbursed!

Also be aware that if you're submitting several packages all together, you need to make sure your corporate card has a high enough limit — and available credit — for all the different fees that will hit all at once. We've had more than one client run into issues with this, so it's more common than you'd think.

ProTip

Salesforce does sometimes offer discounts for the \$2,700 Security Review fee. To learn more about these discount code vouchers, reach out to your PAM who can give you more information.



Step by Step in the Wizard

After all that planning and preparation, now you're ready to submit your app! Head on over to the Salesforce Partner Community and log in to get to the Security Review Wizard. You'll find the wizard on a new tab (aptly titled "Security Review") in the Listing section of your Partner Community.

Note that if this is the first time you're getting your app reviewed, you'll need to link your uploaded package to a listing first, which happens on the App tab. There are a few other basic questions you'll answer on the App tab, then you can start your submission.

Pro Tip

If you'd like more detail about how to complete each step of the Security Review Wizard, or would like to see it in action, check out our on-demand webinar that includes a full demo of the wizard, **Salesforce Security Review Wizard: Recent Updates & Tips for ISVs**.



Here are the steps you'll need to complete in the wizard:

Contact Information: This is where you'll define who will be contacted for more information or for a failure or pass notification. Again, make sure one of your contact options is a distribution list!

Compliance: Here you'll submit information regarding your company's security readiness. There are questions regarding PCI, credit card information if you integrate with a credit card gateway, HIPAA, and so on. If your app deals with any of these, the Compliance section is where you'll share this information. If none of these items apply to your app, you can just check "N/A" and move on.

Questionnaire: In this step, you'll describe your packaged solution's architecture along with any third party services it integrates with. You're basically explaining the general architecture environment that your packaged application requires. So, does it have a client or desktop app? What sorts of languages and frameworks does it use? What operating systems does it support? These are the types of questions you'll answer in the Questionnaire step.

Docs: This is where you'll upload the code scans, false positives, solution architecture, and use case documentation you've prepared. We covered these scans and documentation earlier in the **Planning for Success** chapter.

Test Environments: Here you'll submit credentials for your demo environment along with any integrated off-platform service, client, mobile, or desktop app. You must submit a fully configured representative demo environment of your packaged solution. Any integrations must be pre-authenticated and completely set up. Don't expect the reviewer to do any kind of app setup or authentication — that's not what you really want them spending their time on.

Pro Tip

There are a few extra hoops you need to jump through with your demo org now that multifactor authentication is being enforced. You'll need to whitelist several IP addresses for Salesforce and check a couple permissions to allow the reviewer to successfully log in as your users. Your PAM can share more information on how to spin up your demo org so that it has all those IP addresses whitelisted already.



Summary: Take the time to thoroughly review all of your information and selections — you're almost done!

Payment: In this final step, you'll submit your credit card info for the \$2,700 Security Review fee as well as the recurring annual fee.

As mentioned previously, getting your ISV partner agreement fully executed and signed is critical. Until this agreement is executed, the button to submit for security review will not be active and you will not be able to submit your app.

Post-Review Office Hours

We've talked about office hours a couple times in this guide already, but just in case you don't pass Security Review on the first round, we recommend that you pre-book additional office hours with the Salesforce review team around the time you submit your app. The goal here is to have a session locked in right after your app has been reviewed, and again, you'll need to schedule this well in advance.

A good rule of thumb is to book your post-review office hours for around 10 weeks after you submit your app. This way, you can discuss your review and go over any issues with the reviewers who actually tested your app, plus get help with how to resolve them.

What Happens If You Fail?

If you pass on your first attempt at Security Review, congratulations! You must have done a great job of following all our recommendations in this guide, and you can stop reading now...

But if you receive a “Your app did not pass” email from the Salesforce security team, don’t despair! As we said in the intro, a “fail” is a speed bump, not a dead end.

Perspective

First off, let’s get into the right mindset:



Don't worry so much about the terminology of “failing” Security Review.

Salesforce *wants* you to pass the review and go on to do great things on the AppExchange, but they do also need to safeguard the trust of their customers who might end up using your solution. So you may end up with a few iterations of your package before you finally pass.

Don’t feel like failing the review for the first time is a judgment of your app by Salesforce, because it’s not. They are not judging the quality or functionality of the app and it’s very common for ISVs to fail security review the first time. There are partners who have gone through three or four reviews. And in fact, after failing for the fourth time, the review team will schedule office hours with you themselves to get you the support you need to succeed on your next attempt. They really do want you to pass!

Ultimately, the Security Review process is really a security hardening sprint. The issues that you get back from Salesforce are helping you make your app more secure. Even if it takes you a couple of times to get through, don’t get discouraged. There is a path through.

Be Ready to Respond

The good thing about seeing a failure notice is that it's kind of a good sign. It means that the reviewer has gone through your app and you're in the queue. If you can quickly resolve the issues the review team has found, you can kick your app back into the queue and you'll probably pass in the next 24 to 48 hours.

How do you get back into the queue quickly? Be ready! Don't get caught off-guard if the review team comes back and needs some fixes from you. Be prepared to pounce on the issues (we cover the most common ones below) and stay focused until they're resolved.

Make a Plan and Stick To It

First off, you want to do some triage. The Salesforce review team will provide you with a report listing all of the security vulnerabilities they found during their review. Your job is to address each issue thoroughly, then re-submit your updated version and pass!

At CodeScience, we track every issue that comes back from the review team as a new bug. We do some investigation to understand, "is this in the package or is this something happening off-platform?" Any issues that happen to be off-platform will tend to pose a slightly higher risk because there's usually other development going on that you now need to slot this into. For us, if there are issues on the package side, we aim to fix those in the first 24 to 48 hours if possible. The goal is to get the app right back into the queue and push it through to completion.

Pro Tip

Make sure you're assigning owners and level of effort (LOE) required for each issue so you can get your team working on resolving them as soon as possible. It's up to your team as to whether there's an urgency requiring a "hero effort" of nights & weekend work — for example, if you're racing to get your app approved in time for Dreamforce. If not, you can take a more comfortable pace, but you still want to make a plan, set some deadlines, and make sure you're knocking out the issues methodically.



Addressing Security Vulnerabilities

The number one most commonly reported security vulnerability from **this list** by Salesforce is the lack of enforcement of CRUD/FLS at the code level. Because all code runs as an admin, it's imperative that the developer checks the permissions of the running user every time the code wants to perform any action on an object and its data.

There are instances where a business process or system process requires that a field or object be modified by a class which was triggered by a user who does not have that permission. An example would be a counter field that logs the number of emails sent but shouldn't normally be editable by that user. Those use cases simply need to be documented as false positives, both in your documentation and in the code itself.

Here are a few of the other vulnerabilities most frequently flagged by the Salesforce security team, and how to address them:

Insecure Software Version: Avoid using third party software with documented security vulnerabilities. To check if your app is using software with known vulnerabilities, review the **Common Vulnerabilities and Exposures (CVE)** database.

Violating sharing rules in Apex: Avoid exposing sensitive data by actively managing profile-based permissions, field-level security, sharing rules, and org defaults in your Apex code.

Insecure storage of sensitive data: Always follow **enterprise security standards** when you move sensitive data in or out of the Salesforce platform.

Cross-site request forgery (CSRF): To protect your solution against CSRF attacks, use `confirmationTokenRequired` or trigger state changes with user actions.

Code Changes & Dev Ops

Depending on what you hear back from Salesforce in your "failure" email, you'll fix the issues on the platform or off-platform. One very important item to note (we mentioned this earlier, but it bears repeating): if you continue to develop in your packaging org after you've submitted your app for review and then release new packages, you're shooting yourself in the foot from a Security Review standpoint.

Why? If the security team wants you to make changes in the package, well, you've now introduced new code. You've increased the scope of your Security Review, and possibly introduced new security vulnerabilities.

Pro Tip

What should you do if you need to keep building your app while you wait for your review?

Keep all new code off to the side. At CodeScience, as soon as we submit an app, we make a new branch in our code repo so that we can keep working, release to integration, release to QA, and test new work. We don't put anything into the packaging org and we don't upload any new packages.

We cannot stress enough that you should keep any new code separate from your packaging org to avoid potentially lengthy delays in your review. Once you pass, you can merge it back and you're all set.



Update the Demo Environment

Make sure not to submit the packaging org for your re-submission as well. We've seen a lot of partners submit their packaging org which has the version installed for their review. The review team will only accept the developer demo org that you've set up.

New Scans & False Positives

Be sure to take new code scans. You want those code scans to be fresh because ideally you're showing that your scans do not show the issues they reported. The same thing goes for your false positives. If anything they pointed out is actually a false positive, you just need to add that to your false positives documentation.

Take Two!

When you're ready to submit a new package version, you're going to go through the wizard again. All your answers and deliverables should still be there. You're just going to upload new forms — all your scans and your false positives — and submit just like you did the previous version.

Once you re-submit, it's a good idea to log a case to let the review team know that you've done so. The form will ask for the package ID and a few other details. It also can't hurt to alert your PAM as well. Err on the side of over-communicating

here. You want to make sure that everybody on the Salesforce side that *should* know about it *does* know about it. The last thing you want is for your resubmission to somehow get missed and sit there, not getting looked at.

If you've been diligent about addressing all instances of all vulnerabilities noted by the review team during your first review and have not introduced any new code or vulnerabilities in the meantime (did you **miss that bit of advice?**) you should soon receive that coveted "Congrats! Your Application Passed Security Review" email!

Summary & Further Assistance

Salesforce has plenty of helpful resources that you should take advantage of as you're preparing for Security Review. These are referenced throughout this guide, but we've collected them here for ease of reference.

Salesforce Resources

[Build Secure Apps Trailhead](#)

[Create a Secure Solution](#)

[Secure Coding Guidelines](#)

[Prevent Common Violations of Secure Coding Guidelines](#)

[Partner Security Portal](#) – this is the login page where you can go to schedule Office Hours with the review team and access various Scanners

Additional Resources

[OWASP Top 10 Security Issues list](#)

[Common Vulnerabilities and Exposures \(CVE\) database](#)

[CodeScience Solution Architecture Template](#)

[\[Webinar\] Salesforce Security Review Wizard: Recent Updates & Tips for ISVs](#)

[\[Webinar\] 10 Tips to Pass Salesforce Security Review](#)

Need a Little Help?

Look: we get it. Security Review is not a walk in the park. While our goal with this guide is to give you all the information you need to succeed in the Security Review process, ultimately, we know it's tough. As an ISV, you've got a million other things to do, and Security Review feels like a dreaded but unavoidable burden and the last thing you want to deal with.

We're here to help. Whether you just want a quick security assessment before you submit or you've reached a point where you'd rather hand the whole thing over to a team you can trust to get you over the finish line quickly and efficiently, CodeScience has the experience and expertise to guarantee your success.

For ISVs who could use a little help creating their plan for Security Review or need a fast but thorough assessment of their app submission, our **Pre-Security Review Service** has got you covered.

And if you're just plain stuck and you need more comprehensive assistance, don't hesitate to **reach out to us**. We have been through this process many, many times and we can get you through too, so you can get your solution listed and focus on bigger and better things!

CODE:SCIENCE

Better Designed • Better Architected Solutions for Salesforce ISVs

Headquartered in Chattanooga,
located across the US.

1401 Chestnut Street
Chattanooga, TN 37402

423.954.7400
www.codescience.com

Get help navigating Security Review now 