**1 Amazon Onboarding**

Welcome to Amazon!

**1.1 Action Items**

* Linked in it is the helpful [IT new hire guide](https://it.amazon.com/en/help/articles/i-am-a-new-hire) to get started (feel free to skip the soft phone though)
* Get your [Embark Launchplan](https://embark.corp.amazon.com/) and start going through the contents in there (especially the online AWS full day courses, [see links below](https://w.amazon.com/bin/view/STAR/Internal/Onboarding/#HOtherUsefulTools))
* Join the company-wide [Slack](https://amazon.enterprise.slack.com/) and ask a team member to get added to STAR specific channels
* Go through the [online SDE bootcamp](https://learn.a2z.com/app/course/amzn1.c3.v2.11d3a661-51bf-49bd-9147-c8b984b2a28d/) for basic SDE content
* After finishing the SDE bootcamp, go through [SDE Foundations](https://w.amazon.com/bin/view/EE/Learn/SDE_Foundations)
* Go through [BuilderHub's Getting Started guide](https://builderhub.corp.amazon.com/docs/getting-started.html)
* If you use a MAC, go through [this guide](https://builderhub.corp.amazon.com/docs/dev-setup/laptop-macos.html) to set it up
* If you use windows, go through [this guide](https://builderhub.corp.amazon.com/docs/dev-setup/laptop-windows.html) to set it up
* Other nice things:
  + [Transportation benefits](https://amazon-slu.luum.com/commute/content/resources/new-hire-guide) (get your ORCA card)
* Create your [cloud desktop](https://builderhub.corp.amazon.com/docs/cloud-desktop/user-guide/index.html).
  + You do not need AWS Security Identity Verification on this team.  If you are getting a message in the command line that you need it (it directs to this page [Identification Verification](https://w.amazon.com/bin/view/AWS_Security_Assurance/Digital_ID/IDVerify/#HVerificationProcess)) then you might have had govcloud access in your previous team and need to file a ticket to be removed following the FAQs in their wikis.  If you are in govcloud and are not seeing this warning, make sure you keep your FIPS token until you're 100% sure you are removed, because this can start showing later on.
* Learn how Amazon writes! Here is a [wiki](https://w.amazon.com/bin/view/WritingHub) of some resources.

**2 STAR Team Onboarding**

[STAR](https://w.amazon.com/bin/view/STAR) (Security Through Automatic Remediations) is a team that operates in the CDO security space, developing software solutions to bring security to Amazon scale in a transparent manner. We pioneer in the spaces of security visibility, unified security views, and continuous security assessment. We are in the Proactive Security/Security Products group within Core Consumer Security (CCS or C2S) under Director of Proactive Security and Compliance (ProSecCo) Magnus Nyström (magnusn@) and Director of Security Products Hudson Thrift (thrift@).

**2.1 Onboarding Action Items**

1. Update this wiki as you go! You are now the teams best expert on the onboarding process. Congrats!
2. Have your manager or someone on the team invite you to all regular STAR-team meetings listed [below](https://w.amazon.com/bin/view/STAR/Internal/Onboarding/#HRegularTeamMeetings)
3. Schedule with your manager a weekly 1 on 1 time to put in both your calendars
4. Request access to the following groups if you're not in them already
   1. [STAR](https://permissions.amazon.com/a/team/STAR)
   2. [RAP](https://permissions.amazon.com/a/team/RAP)
   3. [FENCE](https://permissions.amazon.com/a/team/Fence)
   4. [jbrew-org](https://permissions.amazon.com/a/team/jbrew-org)
5. Have yourself added to all the appropriate resolver groups
   1. <https://cti.amazon.com/group/STAR>
   2. <https://cti.amazon.com/group/fence>
   3. <https://cti.amazon.com/group/Review%20Automation%20Platform>
6. Get yourself added to all the appropriate slack channels
   1. star-team
   2. star-team-devs
   3. shepherd-team
   4. shepherd-team-devs
   5. shepherd-2-person
   6. shepherd-team-crs
   7. star-bots
   8. star-pipeline-status
   9. discovery-and-star
   10. security-products-iam-org-[austin or seattle], sim-org-[austin or seattle]
   11. security-products-tech
   12. shepherd-interest \*note: this channel is where 1k+ customers post quick questions once-in-a-while, and the Shepherd teams work together to try to answer them on a non-urgent basis.
   13. shepherd-ux-interest
   14. shepherd-announcements
   15. udd-discovery-shepherd-sheepdog-integration
   16. udd-interest
   17. women-of-ccs [if identifying]
7. Ensure you're part of the main email lists
   1. STAR, jbrew-org, c2s-org etc
      1. Compare your own [email-list subscriptions](https://email-list.corp.amazon.com/email-list/email-list.mhtml) to someone else's on the team for a general comparison check of email subscriptions
8. Go through the rest of this wiki!

**2.2 Regular Team Meetings**

If you haven't received meeting invites already, ask kaysim@ to extend you the meeting invites. These include

**Standups (daily)**

Quick daily team sync for progress on active tasks everyone is working on, and a place to surface items to bring attention to (AKA flags)

* [Your Sprint Boards (Contains Latest Star Sprint)](https://issues.amazon.com/sprints)
* [Old Star Kanban board](https://issues.amazon.com/sprints/%F0%9F%8C%9F%20STAR%20Kanban%20Board/312f2752-18e1-41af-a817-8328e6fd9720?mode=lanes)

**STAR Team Meeting (weekly)**

Every week, the whole team meets for 2 hours to discuss team topics, give execution updates, provide small project demos, give kudos to teammates and go over connection question results and manager updates

* [Team meeting agenda](https://w.amazon.com/bin/view/STAR/Internal/TeamMeeting/Agenda/)

**Execution Meeting (bi-weekly)**

Execution meetings are how we review project statuses, prioritize unassigned ops tasks and project work, and go over who is working on what roughly

* [Issues across all backlogs](https://issues.amazon.com/issues/search?q=status%3A(Open)+containingFolder%3A(18d5d320-eafc-483e-aae5-c50999e69244+OR+a67423f0-0aa9-45b7-9ae6-1f58c8b072f5+OR+29d564b7-0442-46df-ab4d-7bf7c0e32e8c+OR+3a1099ef-2cad-4563-bc8b-d96c436e524c+OR+e19358d2-fe36-40ca-a71c-1c3746e99e7b+OR+fc14aaeb-2dfe-485c-b6e0-512b625e90e6+OR+afb89466-fcd3-459a-aefb-ff4a475c4d9a)&sort=extensions.backlog.rank+asc)
* [User Story template (for Shepherd)](https://issues.amazon.com/issues/create?template=cf7d9a85-601c-46c4-a43c-aa21215a575c)
* [Execution Meeting wiki notes](https://w.amazon.com/bin/view/STAR/Internal/ExecutionMeeting/)

**Retrospectives (bi-weekly)**

We use retrospective meetings to reflect upon team processes, discuss what we want to start/stop/continue with these processes and new ways our team can do better or discuss

* [Retrospective Tracking](https://sim.amazon.com/sprints/STAR%20Retro%20Board/c6bb8647-3590-4ec3-b2da-8d852456b31b?mode=lanes)
* [Team Retrospective](https://retro.corp.amazon.com/#/retro/team/45e87d9b-40d8-440a-a868-b6ebc2b818af)
* [(OLD) team retro wiki](https://sim.amazon.com/sprints/STAR%20Retrospectives/f7fd9c6c-3903-47e9-8fe4-6e116de7f69a?mode=lanes)

**2.3 On-call Resources**

Some time in (usually ~3 months), you'll start to join the on-call rotation to handle customer queries, alarms, pipeline management and tickets. [Be sure to review the default ticket escalation timings for Sev2s!](https://w.amazon.com/bin/view/IssueManagement/TT/UserGuide/RemedyTicketEscalations/#Sev-2:_Escalations_and_Notifications)

Every Amazonian is required to do a few things before starting their first on-call. Download Pong Paging onto your phone. Configure your pager email settings at [this link](https://w.amazon.com/bin/view/Pagers/PeoplePortalPagerEmailSettings), and take the Tier 1 Resolver Training on KNET. ([Wiki link](https://w.amazon.com/bin/view/TechOps/Tier1_Resolver_Training))

As an on-call of the STAR team, you'll want to get familiar with a few services in particular that populates the majority of the tickets we receive. Take a look at the links in the table below and bookmark the dashboards and SOP links before your first on-call. You'll want to monitor them throughout the week. Sim tickets show up under [t.corp.amazon.com](https://t.corp.amazon.com/issues/all-my-groups) and will require most of your attention, and sims (feature requests etc) show up under [sim.amazon.com](https://sim.amazon.com/issues/search?q=status%3A(Open)+containingFolder%3A(13bf9a60-4463-4dbd-a82c-ec35fec3a9bd+OR+f1cb6daa-70cc-4494-9866-bbab6cc46532+OR+29d564b7-0442-46df-ab4d-7bf7c0e32e8c)&sort=lastUpdatedConversationDate+desc&selectedDocument=97f65007-7bc7-45f6-b0df-02fe23aff321). As on-call, it is also your duty to check [pipelines](https://pipelines.amazon.com/) to see that things are running smoothly and unblock blocked pipelines. This may often involve contacting those who worked recently on the pipeline to take a look.

|  |  |
| --- | --- |
|  | STAR |
| CTI (Category/Type/Item) | Consumer Cloud Security/STAR/\* |
| Product(s) Wiki | [Shepherd](https://w.amazon.com/bin/view/Shepherd/)  [Shepherd User Guide](https://w.amazon.com/bin/view/Shepherd/UserGuide/) |
| Dashboard | [Shepherd Dashboard](https://w.amazon.com/bin/view/STAR/Internal/Dashboards/) (this is our primary central dashboard we look at, with links to most service runbooks) |
| SOP(s) | [All Shepherd Runbooks](https://w.amazon.com/bin/view/Shepherd/Internal/Runbooks) |

**Incident Response**

There are two runbooks useful for handling incidents. There is a general [runbook](https://w.amazon.com/bin/view/STAR/Internal/IncidentResponse/) with an overview of general incident response guidelines. There is also a Tier-1 Incident [runbook](https://w.amazon.com/bin/view/STAR/Internal/IncidentResponse/TierOne) which contains useful information about the STAR, Discovery, SDM and IC2 teams including their on call schedules, pagers and manager information.

[**STAR Resolver Group**](https://t.corp.amazon.com/issues/?q=extensions.tt.status%3A%28Assigned%20OR%20Researching%20OR%20%22Work%20In%20Progress%22%20OR%20Pending%29%20AND%20extensions.tt.assignedGroup%3ASTAR) **Tickets Queue**

Most of the tickets in this ticket queue will be service autocut alarms ([example](https://t.corp.amazon.com/V480526252)) and some investing/issue fixing request tickets ([example](https://t.corp.amazon.com/P54334536)). If you see any customer request/bug report tickets, please reassign the ticket directly to [Shepherd Customer Support resolver group tickets](https://t.corp.amazon.com/issues/?q=extensions.tt.status%3A%28Assigned%20OR%20Researching%20OR%20%22Work%20In%20Progress%22%20OR%20Pending%29%20AND%20extensions.tt.assignedGroup%3A%22Shepherd%20Customer%20Support%22) queue, CTI: Information Security -> Shepherd -> Customer Help.

Service alarm autocut tickets usually are cut by the service automatically when the alarm of that service is triggered on CloudWatch in the service AWS account, the name normally will include the service name, stage name, region, and the alarm name. For these autocut tickets, you can follow the following steps to troubleshoot:

* Login to the correct service and stage AWS account, most of the case you can use the Conduit link attached in the ticket body to login in, otherwise, go to service pipeline and find the AWS account number for the correct stage and use it to login.
* Find the alarm metric and the timestamp the alarm was triggered on CloudWatch. The timestamp will also be attached in the ticket body, but it's always better to double check in the CloudWatch metrics. You can either search the alarm name in CloudWatch->Alarms->All alarms, or click the Console link after login to target to the alarm metric.
* After the time window of the alarm is narrowed down, go to CloudWatch->Logs-> Log Insight to find the corresponding service log in that period for troubleshooting.
* All the details of how to query log and troubleshoot for each specific alarms can be found in the service Runbook, which is also attached in the ticket body under "Alarm Details".
* Tips: Going through all the previous research/comment in the Correspondence and Worklog in the ticket Communication tab will also be very helpful for troubleshooting.

For other type of tickets, you might need to coordinate with the requester to understand the issue/request and troubleshoot for it.

[**Shepherd Customer Support Resolver Group**](https://t.corp.amazon.com/issues/?q=extensions.tt.status%3A%28Assigned%20OR%20Researching%20OR%20%22Work%20In%20Progress%22%20OR%20Pending%29%20AND%20extensions.tt.assignedGroup%3A%22Shepherd%20Customer%20Support%22) **Ticket Queue**

When addressing customer requests or question in this ticket queue, please refer to this [Shepherd Customer Support Tickets SOP](https://w.amazon.com/bin/view/Shepherd/Internal/Runbooks/ShepherdCustomerSupportTickets/) for help.

**Shepherd Onboarding Tickets Queue**

Currently this queue is mainly owned by IC2 team, but some times STAR team need offer support and help address campaign owner onboarding requests when the workload in this queue is too much. Oncall can refer to [Shepherd Onboarding Tickets SOP](https://w.amazon.com/bin/view/Shepherd/Internal/Runbooks/Onboardings/) and follow the steps there to work on the tickets.

**3 Shepherd & Work Onboarding**

As a team, we work on developing and operating on multiple services to improve security posture throughout the company. Among the services we own, Shepherd ([https://shepherd.a2z.com)](https://shepherd.a2z.com) acts as the envisioned customer facing, central platform that houses several more services and systems within or alongside it. Reading the Shepherd [User Guide](https://w.amazon.com/bin/view/Shepherd/UserGuide/) and playing around with the [beta version](https://beta.shepherd.a2z.com/) of the tool are some of the best ways to familiarize yourself with all our features.

Read the Shepherd Press Release for a high level overview of what Shepherd does internally <https://w.amazon.com/bin/view/Shepherd/PressRelease/>

**3.1 Shepherd and Dependencies Architecture**

At a very high level, in our envisioned state of things, this is what a more full system architecture would look like. Scanners such as UDD, ASAP and MC2S will perform scans on code packages and resources for any insecure configurations, code or more. These scanners will obtain **findings**, those of which are considered insecure are called **issues** and will be stored in a central findings repo called **Silo** and are eventually surfaced to the owner of these resources to remediate or deal with through **Shepherd**. *(4/29/2021: for a more up to date view of the larger ecosystem, view*<https://design-inspector.a2z.com/#IIssue%20Lifecycle>)

**3.2 Projects and Services we own**

You can find a brief list of the services and project the team currently owns here

<https://w.amazon.com/bin/view/STAR/Internal/Projects/>

In-depth knowledge sharing sessions in video format for our systems can be found on our [Broadcast channel](https://broadcast.amazon.com/channels/38312).

**3.3 Developer Tips, Resources and Help**

**3.3.1 Coding languages and IDEs**

From the inception of Shepherd, most of our applications we own and write are done in Java and [TypeScript](http://www.typescriptlang.org/), a typed superset of JavaScript. Developer setups are encouraged to do on a completely personal basis but my (anthshin@) personal recommendations are to work in [IntelliJ IDEA](https://www.jetbrains.com/idea/) for Java, and [Visual Studio Code](https://code.visualstudio.com/) for basically everything else. If you're working with IntelliJ IDEA, [this guide](https://w.amazon.com/bin/view/EE/Learn/Online/Intro_to_IntelliJ_IDE) will be helpful for setting it up with some configurations for working at Amazon (namely the [Black Caiman toolkit](https://builderhub.corp.amazon.com/docs/black-caiman/user-guide/index.html)) and basic tutorials. If you're on Mac, install [NinjaDevSync](https://w.amazon.com/index.php/NinjaDevSync) to synchronize content on your local machine with your cloud dev desktop.

Some tools and frameworks that our team commonly uses:

* [Typescript](https://www.typescriptlang.org/docs/)
  + For prettier formatting, use bb app fix
* [React.js](https://reactjs.org/docs/getting-started.html)
* [CDK](https://docs.aws.amazon.com/cdk/latest/guide/home.html)
  + [CDK workshop that covers end-to-end CDK development](https://cdkworkshop.com/)

**3.3.2 Getting Started with Brazil**

Like many teams at Amazon, we use [Brazil](https://builderhub.corp.amazon.com/tools/brazil/) as our primary build system. For those new to Amazon, it can be a little daunting to get started, but the main flow to work on packages locally is as follows:

Make sure you have setup your [workspace and the Brazil CLI](https://builderhub.corp.amazon.com/docs/brazil/cli-guide/getting-started.html) first.

cd ~/workplace

brazil ws create --root ShepherdFrontend # Doesn't need to be a specific name

cd ShepherdFrontend

brazil ws use --package ShepherdFrontendUI # The name of the package as found on code.amazon.com

# The above command should automatically detect the proper version set to use. But if you need to change it for any reason:

# brazil ws use --versionset ShepherdFrontend/development

cd src/ShepherdFrontendUI

brazil-build

Once you start working with multiple packages at once, doing builds can become problematic if you manually build each package separatedly, instead use brazil-recursive-cmd to execute a command in all packages while respecting dependency order.

brazil-recursive-cmd brazil-build release # Execute brazil-build release on package and all dependecies in the workspace

brazil-recursive-cmd --allPackages brazil-build release # Execute brazil-build release on all packages in the workspace

Here is a list of aliases of many useful Brazil commands that you can add to your shell profile:

alias bb='brazil-build'

alias br='brazil-runtime-exec'

alias bre='brazil-runtime-exec'

alias brc='brazil-recursive-cmd'

alias bws='brazil ws'

alias bwsuse='bws use --gitMode -p'

alias bwscreate='bws create -n'

alias bbr='bb release'

alias bbrec='brc brazil-build'

alias bbrr='brc brazil-build release'

alias bbt='bb test'

alias bbs='bb server'

alias bbc='bb clean'

**3.3.3 Committing Changes**

We use git as our version control system. These are some general steps to move your code to the code review stage of the development process

[Show](javascript:void(0))

For an elaborate tutorial on git, this interactive learning website is a great tool to learn or even refresh your understanding. <https://learngitbranching.js.org/?locale=en_US>.

Additional helpful links:

* [ShepherdFrontendUI Package page](https://code.amazon.com/packages/ShepherdFrontendUI/trees/mainline)
* [Setting Platform Support](https://builderhub.corp.amazon.com/docs/brazil/cli-guide/howto-platform-support.html)
* [Setting Up Ruby](https://w.amazon.com/index.php/BrazilCLI_2.0/Runtimes/Ruby)
* [Setting Up Node.js](https://w.amazon.com/index.php/BrazilCLI_2.0/Runtimes/NodeJS)

**3.3.4 Code Reviews**

Our team uses a Code Review Oncall rotation, where one person is assigned to look at the CRs submitted. Each CR request must list the CR oncall (tracked as the assignee via [this sim](https://issues.amazon.com/issues/8e597122-a731-418b-b1e8-3667e7c264c5)) plus a Subject Matter Expert (SME) who has extensive experience with that specific service or someone that has extensive experience on the project you are working on.  
More information can be found on the [Code Review wiki page](https://w.amazon.com/bin/view/STAR/Internal/CodeReview/).

**3.3.5 Browser add-ons/extensions**

Whether you prefer chrome, firefox or another browser, there's a number of extensions we recommend adding whether for development or general ease of life.

* Tampermonkey: Lets you allow scripts to run on certain web pages for helpful customizations ([firefox or chrome](https://www.tampermonkey.net/))
  + [IGraphHelper](https://w.amazon.com/bin/view/IGraphHelper/): adds helpful links for [iGraph](http://w.amazon.com/?IGraph)-created graphs discovered on Amazon webpages (icons at top right will appear when hovering over embedded iGraphs).
  + [Auto Relog for Conduit](https://axzile.corp.amazon.com/-/carthamus/download_script/auto-relog-for-conduit-1.user.js)Switch between conduit accounts without manually logging out first. When a conduit login fails because you have to log out first this script will request the logout and then re-attempt the login automatically
  + [CR Status](https://code.amazon.com/packages/GreasemonkeyScripts/blobs/mainline/--/Code%20Browser/CR%20Status/CR%20Status.user.js?download=1): Show the status of CRs in one glance: Approvals, comments, dry run failures.
  + [HyperBadge](https://hyperbadge.amazon.dev/): A hyper-realistic, pseudo-3D, interactive representation of an Amazonian’s physical badge.
* Containerize AWS Console: Lets you containerize individual AWS console sessions to separate tabs, allowing you to work in multiple AWS accounts simultaneously ([firefox](https://w.amazon.com/bin/view/AwsConsoleFirefoxAddon/) only)
  + (Haven't tested) If you use Chrome, [Bello](https://w.amazon.com/bin/view/BelloBrowserExtension) is an alternative
* React Developer Tools (if you'll work with React at all) ([firefox](https://addons.mozilla.org/en-US/firefox/addon/react-devtools/) or chrome)
* Redux DevTools (if you'll work with Redux at all)s ([firefox](https://addons.mozilla.org/en-US/firefox/addon/reduxdevtools/) or [chrome](https://chrome.google.com/webstore/detail/redux-devtools/lmhkpmbekcpmknklioeibfkpmmfibljd?hl=en))

**3.3.6 Other Useful Tools**

* [oh-my-zsh](https://ohmyz.sh): If you use zsh instead of bash for your terminal, this is an amazing power-up with some great default customizations.
  + There's a lot of customizable plugins and add-ons you can add to this. Some I'd recommend are zsh syntax highlighting, auto-suggestions and fuzzy finders all found on [this guide on medium](https://medium.com/@ivanaugustobd/your-terminal-can-be-much-much-more-productive-5256424658e8).
* [BuilderTools handy link](https://builderhub.corp.amazon.com/docs/dev-setup/laptop-macos-extra.html) for more dev tools for MacOS
* AWS Technical Essentials for CDO
  + <https://kiku.aws.training/Details/eLearning?id=10954> (virtual course without instructor)
  + <https://kiku.aws.training/SessionSearch?pageNumber=1&courseId=10012> (virtual course with instructor)
* Developing on AWS for CDO
  + <https://kiku.aws.training/Details/eLearning?id=74152> (virtual course without instructor)
  + <https://kiku.aws.training/SessionSearch?pageNumber=1&courseId=10017> (virtual course with instructor)
* Architecting on AWS for CDO
  + <https://kiku.aws.training/Details/eLearning?id=42594> (virtual course without instructor)
  + <https://kiku.aws.training/SessionSearch?pageNumber=1&courseId=10002> (virtual course with instructor)

**3.3.7 Other Useful Reading**

* Summary of some of the needs guiding Shepherd's creation.  An answer to "Why are Shepherd issues not Policy Engine issues? Why do both tools exist?" <https://sage.amazon.com/questions/1251728#1321564>
* Summary of some of tooling involved in building in AWS <https://w.amazon.com/bin/view/STAR/NativeTooling/> (also available in video form <https://broadcast.amazon.com/videos/125200)>

**3.3.8 Various Team Development Guides**

* [Git](https://w.amazon.com/bin/view/STAR/Internal/Development/Git/)
* [CodeReview](https://w.amazon.com/bin/view/STAR/Internal/CodeReview/)
* [CodeStyle](https://w.amazon.com/bin/view/STAR/Internal/CodeStyle/)
* [NativeTooling](https://w.amazon.com/bin/view/STAR/Internal/NativeTooling/)
* [DevOps](https://w.amazon.com/bin/view/STAR/Internal/DevOps/)
* [directory including git and more](https://w.amazon.com/bin/view/STAR/Internal/Development/)
* [EnvImprovementNinjas](https://w.amazon.com/bin/view/EnvImprovementNinjas/)
* [Builderhub page for CRUX CR templates. Our team doesn't use them now but potentially worth checking out later?](https://builderhub.corp.amazon.com/blog/cr-template/)

**3.4 Glossary**

* **Resource**: A resource is a generic entity owned and utilized internally (software package, AWS account, etc)
* **Application**: A logical grouping of resources used typically to denote the boundaries of an internal software service. In Shepherd, this is defined by bindles ("Bindle Application")
* **Issue**: An issue is a finding that indicates an actionable misconfiguration of a resource. Issues are what are surfaced to users as the things we want them to fix. Issues have a severity (e.x. Critical, High, Medium, Low) to better aid the user in prioritization.
* **Finding**: A finding is the output of the execution of a rule assigned to the resource against which the rule was evaluated. A finding is can either indicate a misconfiguration or a lack thereof. Findings that indicate an actionable misconfiguration are surfaced to the users as Issues. Findings that don’t indicate any misconfiguration are used for showing which rules the user is currently abiding by. With that, the term finding is never directly relayed to users and is primarily a backend construct.
* **Rule/Campaign**: A rule is logic used to evaluate the current status of a resource's configuration in comparison to a desired configuration. Rules are evaluated against resources and produce findings. Rules have additional metadata such as their rule category (e.x Security, Privacy), and other information needed to determine the context and implications of the misconfigurations detected by its logic. Rules produce findings.
* **Remediation/rollback**: A fix to a security issue that can be applied programmatically (automatically). In Shepherd, these are done by Sheepdog which is owned by IC2 team. Issues with a remediation will either auto-remediate or be run when a user clicks a "fix it for me" button in Shepherd.
  + **Rollback:** The corresponding "undo action" to a remediation. If it exists, this reverts a resource back to its original state from before a remediation was performed.
* **Task**: A task is a requested action from service owners. A task can be related back to an instance of an issue but doesn’t have to be. Given that tasks are not always related to specific issues and can be generic asks, tasks are assigned to applications. Tasks are tagged with risk categories. Tasks are able to be leveraged for accomplishing goals such as security certification
* **Health**: Health is the aggregation of issues and tasks used to denote the current status of an application’s or an organization’s resources. Health is primarily measured on a per-rule-category basis (such as Security) and surfaced using standardized metrics (such as Trust Scores).
* **Trust Score**: Trust Score is a deterministic numerical score that rates an application or organization’s security posture. This score is the primary means through which we encourage users to take action and drive change for the company. Included in the score is a list of primary factors that affect the score which functions as a prioritized list of issues users should address.
* **Exception**: Shepherd exceptions model long-term risk acceptance for security issues. An approved exception suppresses actions associated with the issues, including ticketing, reporting on the Shepherd home page, automatic remediation, SDC blocking, past-SLA remediation, and inclusion in reporting to leadership.
* **Risk Category**: Risk categories are groupings of negative implications for different types of rules or tasks. Risk categories are tagged with appropriate theme(s) and are utilized as tags applied to rules or tasks. Examples of risk categories include Data Exposure, DDoS, Credential Access, Discoverability, SOX Compliance, etc
* **Theme**: A theme is area of interest that relates risk back to business interests and goals. Examples themes would include Amazon protects critical data or Amazon abides by regulatory obligations
* **Resource Configuration**: Resource configuration is the basis of resource management and includes all configuration actions that can be taken on resources. This includes “intelligent” configuration options such as auto-remediation and 1-click-fixes.
* **Resource Management**: Resource management is the union of health tracking, task management, issue management, application definition, ownership, permissions, and resource configuration
* **UDD**: User Defined Detections [link](https://w.amazon.com/bin/view/C2S/Detections/UDD/)
* **FST**: Federated Security Team [link](https://w.amazon.com/bin/view/Dedicated-security/portfoliomanagement/list-of-fsts/)
* **DGI**: Data Governance Infra
* **KTLO**: Keep The Lights On (keep under maintenance, no plans to add features)
* **SIPP**: Security Intelligence Platform Portal [link](https://w.amazon.com/bin/view/Infosec/Private/SecInt/SIPP/QueryUI/)
* **SLA**: Service Level Agreement: [link](https://builderhub.corp.amazon.com/docs/sim-ticketing/user-guide/sla.html)
* **STAR**: Originally stood for Security Threat Awareness and Remediation, but we no longer own the Remediation part
* [[Link to BuilderHub glossary]](https://builderhub.corp.amazon.com/docs/glossary.html)
* [[Link to Anvil glossary]](https://w.amazon.com/bin/view/Anvil/CustomerWiki/Glossary/)
* [[Link to Amazon Acronym Central]](https://w.amazon.com/bin/view/Acronym_Central/)
* ...