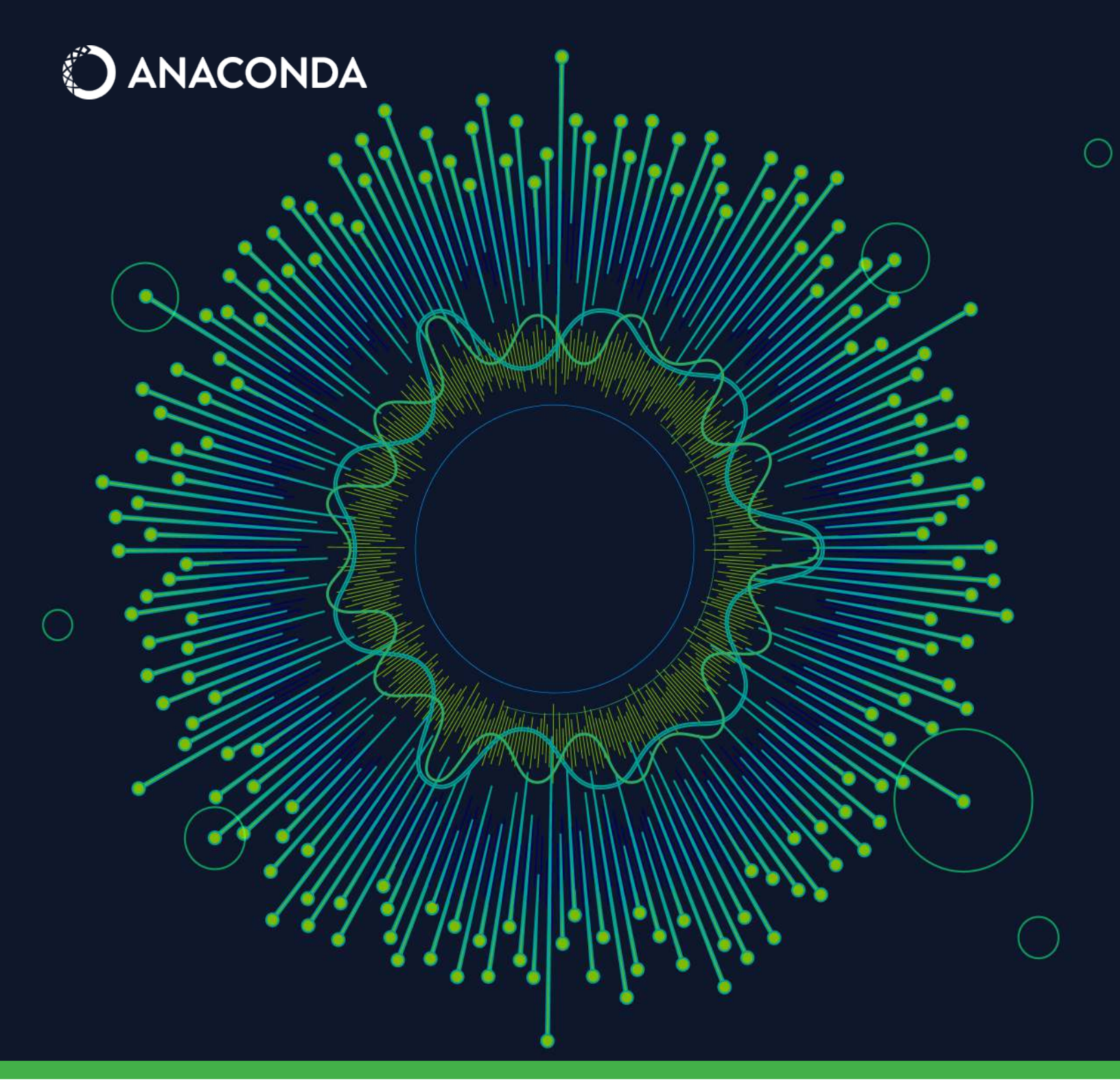
ANACONDA



An Enterprise Guide to a Secure Data Science Pipeline

ANACONDA.

The Unprotected Data Science Pipeline

Open source is the backbone driving digital innovation(Gartner, 2019). It's crucial to many of today's leading-edge digital fields, including data science and machine learning. No single technology vendor can outmatch the pace of innovation the open-source data science community maintains. Thousands of open-source Python, R, and Conda packages provide data science practitioners with the building blocks they need to create models and applications using predictive analytics, natural language processing, robotics, and other cutting-edge tools.

These open-source tools are powerful, and they are essential for differentiation in a future where organizations must adopt AI to remain viable. But, there's one thing many enterprise data science teams are missing: security protocols. In many organizations, there simply are no security protocols or governance tools for open-source software (OSS) use in data science. A lack of security protocols exposes the organization to overlooked defects and vulnerabilities, not to mention potential licensing and intellectual property issues.

In some organizations, DevOps teams have already adopted security protocols related to their use of OSS. DevOps uses open-source building blocks to accelerate their workflows and build applications, but generally they do so within a framework of security and governance to protect their work and enterprise infrastructure. Enterprise data scientists also use OSS tools and packages all the time. But, they use OSS without this safety net, putting the organization and customer data at risk. In some cases, DevOps teams may catch vulnerabilities in data science models when they attempt to put them in production. But, this means valuable data science team effort was wasted building a model that will never see the light of day.

When data scientists don't monitor for potential threats, vulnerabilities inevitably creep into models over time. Data science leaders must step up and collaborate with IT and security leaders to take charge of their open-source data science and ML pipelines. Together, these leaders can increase the flow of innovative models to production while safeguarding against technical and legal risk.

"Open source" refers to a community-driven model through which large, diverse groups of developers and users collaborate on projects via the Internet. The innovation and stewardship of the OS community has driven advances across all kinds of fields, including data science and machine learning. In a very real sense, open source is the richest pathway and the fastest engine of innovation in data science today. A few commonly used OS tools include SciPy, NumPy, Jupyter, Bokeh, R, and pandas.

Just Like all Software, Open Source Carries Risk

Companies tend to choose OSS over proprietary software because it offers more choice, support flexibility, transparency, and unmatched innovation.

More Choice

The open-source community provides a veritable candy store of tools and libraries to work with — there's no need to get tied down to any single vendor. Try new tools, choose only the best of the best (or the ones that fit your needs best), with minimal hoops to jump through.

Support Flexibility

With proprietary software, support is generally bundled in by the vendor and available either through the original license or for an extra fee. The software vendor offers what it offers, take it or leave it. With OSS, you have multiple options among support providers — including community support, third-party vendors, and hiring in-house staff to support your open-source components.

Transparency

The source code of any OSS is viewable and fixable by anyone with the know-how to do so. Organizations using open-source software can verify its security themselves (or use an outside provider for verification). The source code in proprietary software, on the other hand, is usually only viewable and editable by a few internal people.

Unmatched Innovation

Data science and machine learning have a deep history with OSS, going back to the Apache Hadoop data-processing framework, which started a wave of open-source advances that's still going strong. The top ML libraries, deep learning tools, and visual processing tools all came out of the open-source community. No single proprietary vendor can match its depth and breadth of innovation. (See Anaconda's Guide to Open-Source Tools and Libraries for Data Science and Machine Learning for more.)

Vulnerabilities can creep into models over time.

OSS is routinely under-managed and flies under the radar.

OSS is copyright protected and licensed.

To reap the benefits of OSS with the lowest possible risk, data science teams must actively manage their organization's use of OSS to keep up with patches, updates, and vulnerabilities. Some of the most infamous data breaches have occurred due to vulnerabilities in open-source software, such as Apache Struts and OpenSSL. Just like all software components, open-source Python and R packages can contain vulnerabilities. If an organization is not actively monitoring for vulnerabilities, it is very likely they will creep into their models and applications over time.

Unfortunately, OSS is routinely undermanaged in organizations. Because OSS is freely available, it tends to fly under the radar. No one in procurement or the chain of command approves the addition of open-source artifacts to the environment, and no one incorporates them into the management workflow. Many organizations are unaware of the true amount of OSS in use in their environments — let alone the specific versions, vulnerabilities, compatibility issues, security patches, dependencies, and licensing requirements that need to be tracked, documented, and maintained.

What's more, OSS is always copyright protected and licensed, even though the total acquisition costs are basically nil. Some open-source projects have "copyleft" licensing, which allows the software to be modified by the user but requires that the modified software use the same license as the original. Others have "permissive" licenses that allow users to modify and redistribute OSS as part of their new creation. Within these broad general categories, OSS licenses are actually quite varied — with a wide range of potential ramifications when it comes to compliance. Many organizations are essentially blind to their risk of litigation for OSS licensure non-compliance.

Enterprises Need an OSS Governance Program for Data Science

According to Gartner research, 60% of IT organizations have an open-source software policy of some kind. However, there's a very real gap between having a policy and actually instituting a program to ensure best practices and compliance with that policy. At least 75% of current policies are deemed to be ineffective. It's also quite rare for basic OSS policies to account for its prevalence and unique usage in data science. A formalized OSS governance program for data science should include both policies and tools designed to ensure proper usage of the artifacts your data scientists need — and are probably already using — to do their jobs effectively. This will ensure access to OSS innovation and choice, while minimizing your risk exposure.

Before you begin: Understand your organization's risk tolerance for OSS. This may be different for data science than it is in other departments, depending on the objectives of your DS program. An important part of determining your risk threshold is understanding CVE scores and determining an acceptable range for your organization.

What Are CVEs and Why Do They Matter?

CVEs are Common Vulnerabilities and Exposures found in software components. Due to the complexity of modern software with its many layers, interdependencies, data input, and libraries, vulnerabilities tend to emerge over time. Ignoring a high CVE score can result in security breaches and unstable applications.

When someone finds a CVE, they report it to a CVE Numbering Authority (CNA). CNAs assign identification numbers to CVEs and list them in publicly accessible databases. Many IT and software development teams refer to the National Institute of Security and Technology's database (NIST) for updates. There are thousands of new vulnerabilities reported each year. Each vulnerability listed in a CVE database has a score from .1 to 10, 10 being the highest risk level. These scores are based on exploitability, impact, remediation level, report confidence, and other qualities. To better understand how a CVE score is derived, read this documentation from FIRST that describes the scoring system in detail.

Your DevOps team may have already determined what range of scores are acceptable for your company. Talk to your CISO or DevOps manager to see if a threshold has already been set and if the same range should apply to the data science team. Risk thresholds vary by industry. For example, finance and healthcare organizations are likely to tolerate a lower range of scores than an apparel company. Determine your risk threshold and include this in your policy for downloading any packages (explained in the following section). CVE scores will also help you determine how you want to go about managing threats (remediation) and how to prioritize releases.

How to Implement an OSS Governance Program that Works for Data Science

1. **Obtain Buy-in From Executive Leadership, Such as a CISO or CDO**

Start by lining up a champion, such as the Chief Information Security Officer, Chief Data Officer, or another member of the executive team. Their authority and credibility will go a long way in ensuring the program's adoption and long-term success.

2. **Form a Program Committee of IT and Data Science Leaders**

This group should consist of at least one senior executive leader and management-level stakeholders from data science, IT, and security. A DevOps leader may also be a great contributor, if they have experience implementing an open-source security plan. Once established, this committee will develop core guidelines and then roll out, enforce, and maintain the program.

3. **Develop OSS Security and Governance Policies Specific to Data Science and Machine Learning**

Begin with the development of best practices for using OSS and procedures for monitoring and supporting it. These should include guidelines for evaluating the stability, reliability, and security of OS tools and packages before use based on:

- **Code activity and release history**
  A steady cadence indicates a project is updated and maintained regularly —and therefore more likely to be stable and reliable.
- **An acceptable range of CVE scores**
  Any packages with CVE scores above your risk threshold should be blocked or blacklisted. Risk thresholds will vary by industry.
- **The availability of community support and documentation**
  Look for active forum discussion and bug fixes in the project issue tracker.
- **The number of active project contributors**
  The more developers and engineers updating and maintaining the software, the better.
- **License obligations and requirements**
  Make sure a piece of software has a license type that permits your specific use case.

4. Develop Processes and Procedures for Monitoring and Remediation

Develop the following processes or procedures for monitoring vulnerabilities and minimizing and addressing risks. There are a few automation tools on the market that can make some of these processes easier and faster.

- **A process for conducting a metadata analysis for CVE data**
- **A process for evaluating package reliability**
- **A process for staying up to date on patches and updates for OSS your team is currently using**
- **A process for monitoring the packages your team is using and controlling user access to these packages**
- **A process for remediation**
  Reliable OSS projects provide a system for handling pull requests. Train your data science team to submit pull requests to project maintainers whenever they find a vulnerability.
- **A support plan for open-source tools**
  Third-party commercial support is available, but can be pricey.
  Community support may not be able to respond on demand.
  In-house support requires bandwidth and skills that may not be available.

5. Automate Your Standards with a Best-in-Class Governance Solution

The committee should be responsible for enforcing and maintaining the governance program's standards and guidelines. This is a highly complex job that can be extremely time-consuming if undertaken manually — especially if you have any plans to scale your data science efforts. Instead, look for an enterprise-class governance solution that's designed for data science and machine learning, with features such as:

- **A managed repository of "known good" OS artifacts**
- **Automated scanning for vulnerabilities and licensing requirements**
- **Package management to continually evaluate and update open-source packages**
- **Automated reporting**
- **Admin controls that allow the data science manager to blacklist, whitelist, and block packages according to internal standards**
- **Systemic controls that prevent the data science team from downloading packages that do not meet internal standards**

6. Evangelize Good OSS Governance and Create a Security-Aware Environment

It's a good idea to develop a communications plan ahead of program rollout, so that everyone on your data science team understands the reasons for the new policies and is adequately trained in the use of any governance tools. Consider planning a training session to explain the significance of software vulnerabilities, CVEs, and the guidelines your committee developed to evaluate OSS. Plan another session to train the team on how to use any new tools you've adopted.

7. Bolster OS Principles, Skills, and Experience Across Your Data Science Team

Data science professionals come from many different backgrounds and disciplines, and not all of them have experience with open-source software. As OSS usage will continue to be commonplace in data science and across the enterprise, it's a good idea to promulgate an understanding of the open-source community, it's importance in data science, and how to engage with it in a meaningful way.

Foster a culture that encourages openness, collaboration, and collective knowledge, with learning as a primary goal. Build skills such as version control and dependency management, as well as the usage of a variety of open-source libraries tools. Data science and machine learning teams commonly use pandas, SciPy, NumPy, scikit-learn, TensorFlow, PyTorch, and Dask; but there are thousands of open-source projects available for their use.

An OS-friendly culture should also include participation in the broader open-source community. Encourage your team's contribution to the development of data science tools. If necessary, have an experienced colleague train others on how to collaborate on and champion interesting OS projects, as well as contribute to open-source libraries.

Enforcing Security Standards Prepares Data Science for Production

Many DevOps and DevSecOps teams have adopted a CI/CD workflow (continuous integration, continuous delivery). With continuous integration, new code is continuously merged into the codebase. When this is done, testing automatically ensues. If everything checks out, then the continuous delivery process begins, and changes are automatically deployed into production. This software development process is important for data scientists to understand because it is efficient and reliable. The data science team should strive to maintain their models in a similar fashion. Establishing a solid data science governance program with package management processes will facilitate the team's adoption of a CI/CD workflow.

With a solid OSS governance program in place, your data scientists will no longer lose time building models that DevOps has to reject due to vulnerabilities, licensing, or reliability issues. By resolving discrepancies within the ML/DS pipeline, their projects will be better prepared to go to production.

Govern Your Open-Source Data Science Pipeline with Anaconda Team Edition

No provider knows open source like Anaconda does. We originated the use of Python for data science – and today, our team builds and maintains many of the data science packages that millions use every day. We understand the challenges of balancing innovation against security and governance all too well — so we turned our experience into Anaconda Team Edition.

Team Edition is an enterprise-class mirrored repository with access to more than 7,500 open-source packages, built by data scientists for data scientists. With Team Edition, you have a central repository for conda, PyPI, CRAN, and custom-built packages with security and governance in place by default. You also get access to support from open source experts. Control which packages are available to the team according to risk levels, manage user access, maintain control over the chain of custody, and much more.

Learn more about Team Edition at anaconda.com/repository.