GUIDE

How to Implement an OSS Governance Program

for Data Science, Artificial Intelligence, and Machine Learning

ANACONDA.

The open-source ecosystem is crucial to many of today's cutting-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 more.

These open-source tools are powerful, and they are essential for differentiation in a future where organizations must adopt Artificial Intelligence (AI) to remain viable. But, there's one thing many 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, there are teams dedicated to adopted security protocols related to their use of OSS. DevOps teams, for example, often use open-source building blocks to accelerate workflows and build applications—but they generally do so within a framework of security and governance to protect their work and infrastructure. However, data science teams sometimes use OSS tools and packages outside of their organization's security framework. Without this safety net, they are inadvertently putting their organizations and customer data at risk and making room for vulnerabilities to creep into their models over time.

Data science leaders must collaborate with IT and security leaders to take charge of their opensource data science and machine learning (ML) pipelines. Together, these leaders can increase the flow of innovative models through 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 open-source 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 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 because it offers:

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 dataprocessing 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.

JUST LIKE ALL SOFTWARE, OPEN SOURCE CARRIES RISK

To reap these benefits with the lowest possible risk, data science and security teams must actively manage their organization's use of OSS. Some of the most infamous data breaches have occurred due to vulnerabilities in open-source software, such as Log4j. 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 those vulnerabilities will begin to plague their models and applications.

Unfortunately, OSS is often undermanaged in organizations. Because OSS is freely available, it tends to fly under the radar. Without an OSS governance model in place, 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 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.

What's Needed Is a Formalized OSS Governance Program for Data Science

There's a very real gap between having an OSS security policy and actually instituting a program to ensure best practices and compliance with that policy. Moreover, it's rare for basic policies to account for the prevalence of OSS and its 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 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, but it's also important to be educated on the nuances of CVE scores and what they truly mean.

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.

Risk thresholds vary by industry. An apparel company, for example, may be more likely to tolerate higher CVE scores than a finance or healthcare organization. Determine your risk threshold and include this in your OSS security policy for downloading any packages (explained in the following section). Beyond understanding the severity of a vulnerability, 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 ML

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 open-source 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 maintained it will be.
- **License obligations and requirements**
  Make sure a piece of software has a license type that permits your specific use case.

4

Establish Procedures for Monitoring Vulnerabilities and Addressing Risks

Create processes for:

- Conducting a metadata analysis for CVE data and evaluating a package's vulnerability
- Staying up to date with patches and updates for OSS your team is currently using
- Monitoring and controlling access to the packages your team is using
- Remediation
  - › Many 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.
- OSS support
  - › 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 a particular skillset.

5

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

Enforcing and maintaining the governance program's standards and guidelines is a 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" OSS artifacts
- Automated scanning for vulnerabilities and licensing requirements
- Package management to continually evaluate and update open-source artifacts
- Automated patch notification
- Admin controls that allow the data science manager to include, exclude, and block packages according to your internal standards
- Systemic controls that only allow the data science team to download packages from the managed repository

6

Evangelize Good OSS Governance and Create a Security-Aware Environment

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.

Enforcing Security Standards Prepares Data Science for Production

Data science and IT leaders should consider enforcing security standards with 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 efficient, reliable, and can go a long way in maintaining a solid data science governance program with package management processes. With a solid OSS governance program in place, your company's data scientists will no longer waste time building models that cannot reach production due to vulnerabilities, licensing issues, or other security concerns. By working together to resolve security discrepancies within the data science/ML pipeline, teams can set projects up for success.

Support Your OSS Security Initiatives with Anaconda

With over 35 million users, Anaconda is the world's most popular platform to develop and deploy secure Python solutions, faster. We champion a vibrant open-source community and empower enterprises to innovate with confidence.

Anaconda's distribution is **curated and built from source on our secure network**. Packages are verified upon installation to ensure that they are tamper-free. With a secured supply chain for open source software, you can spend less time managing risk and more time on execution.

Anaconda **identifies the security vulnerabilities (CVEs)** that packages and their dependencies are exposed to. We curate and enhance the accuracy of CVE data so that you can block unsafe packages with precision and make well-informed decisions.

Anaconda **generates a Software Bill of Materials (SBOM)** for customers in accordance with evolving security standards and best practices around the use of open source in sensitive environments. SBOMs are important because they provide visibility into the components of your software, facilitating awareness of potential risk factors and quicker reaction times should an issue arise.

To learn more about Anaconda's governance, security, and compliance capabilities and to schedule a demo, visit anaconda.com/security-compliance.

With more than 35 million users, Anaconda is the world's most popular platform to develop and deploy secure Python solutions, faster. We pioneered the use of Python for data science, champion its vibrant community, and steward the open-source projects behind tomorrow's artificial intelligence (AI) and machine learning (ML) breakthroughs. Our solutions enable practitioners and institutions around the world to securely harness the power of open source for competitive advantage and groundbreaking discoveries. Visit anaconda.com to learn more.

ANACONDA.