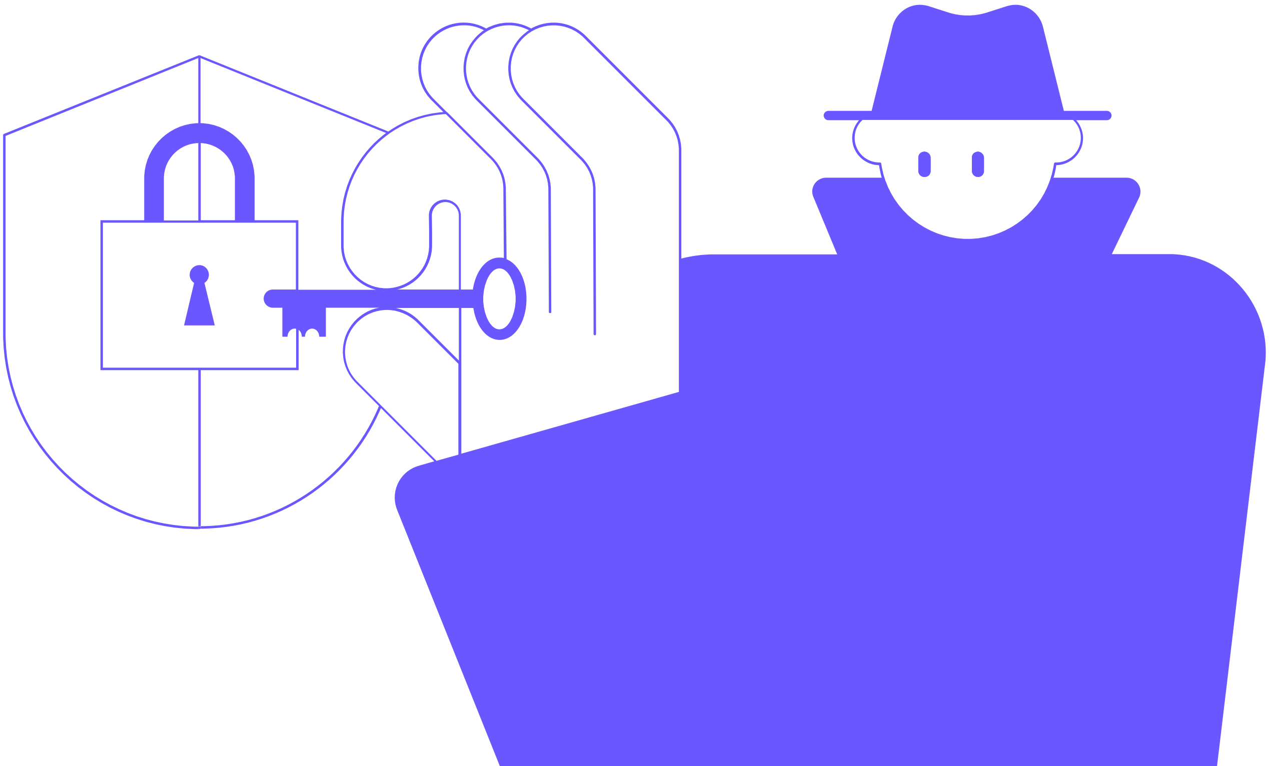
Avoid Becoming
a Data Breach Headline:

9 Ways to Prevent a Supply Chain Attack on Your CI/CD Server



Introduction

Why must your CI/CD server's security be a top priority?

CI/CD servers are becoming a prime target of attacks, since they are at the core of all critical development processes. A CI/CD server has access to source code, which is one of the most valuable assets any software company owns. The server produces build artifacts and can even deploy code to production environments, posing serious risks if not properly secured. **Exploiting just one weakness can give an attacker access to the supply chain** and, therefore, sensitive data, allowing them to inject malware and take control of the systems – something that has been occurring with increasing frequency. According to "The State of Software Supply Chain Security 2023" special report, enterprises have seen an exponential increase in supply chain attacks since 2020. A Forrester study states that **57% of organizations have suffered from a security incident related to exposures in the DevOps toolchain**.

To prevent data breaches and business disruptions that may result in huge financial losses, properly securing your CI/CD server should be a top priority today. Moreover, Google reveals in its "2022 Accelerate State of DevOps Report" that implementing appropriate security controls has a positive effect on software delivery performance and even brings additional benefits, such as reduced developer burnout.

What are the actual risks of a security breach?

- **Financial losses** – Depending on the nature of a breach, organizations may have to compensate affected customers, undertake expensive incident investigations and other response efforts, and pay fines for non-compliance. In addition, they may also lose significant business and see their share prices fall. According to the "Cost of a Data Breach 2022 Report", the average cost of a breach is now USD 4.35 million.
- **Business disruptions** – The investigation and recovery process can take a long time, and organizations often have to shut down some or even all of their operations during that period. Obviously, the longer operations are down, the more likely customers are to leave, which can result in additional revenue loss.
- **Reputation damage** – The risk of losing current and potential customers to competitors that are viewed as more secure is exceptionally high.
- **Legal ramifications** – Security breaches that involve personal information and target the organization's clients often result in class-action lawsuits, and authorities may even restrict companies from performing certain business activities until legal investigations are complete.

Introduction

How can you enhance the security of your CI/CD server and avoid a data breach?

CI/CD servers are at the heart of your software development processes. They check out, compile, test, and build your source code into deployable artifacts and often deploy them, which means potential access to sensitive information and critical systems. The number, frequency, and severity of incidents targeting vulnerabilities in the CI/CD ecosystem are increasing in the industry, as reported by the OWASP Foundation in their recent “Top 10 CI/CD Security Risks”. Security is not a one-off task, but rather a continuous process, so we’ve prepared some practical tips to help you improve the security of your CI/CD pipelines and protect your business from attackers, in alignment with the latest application security framework outlined in that document.

This whitepaper highlights 9 best practices for strengthening the security of both on-premises and cloud-based CI/CD solutions. However, since a significant number of companies worldwide use on-premises CI/CD servers, around 50% according to one of our recent studies, some of these tips are only applicable to on-premises setups.

Top 9 security tips to prevent an attack on your CI/CD server

1. Keep your CI/CD server up to date

First things first, we strongly recommended regularly updating your on-premises CI/CD server (and all related systems, such as build agents) to the latest version. You should also keep an eye out for security notifications, so you can be sure that your system and processes rely on a tool that has all the latest security improvements in place. Finally, don't forget to update your operating system and regularly install security patches.

When using a cloud-based CI/CD solution, this will generally be done for you automatically, though you may have to trigger major updates manually.

If you use TeamCity

TeamCity will automatically notify you via the UI when a new update becomes available. You can also manually check for new versions under Server Administration | Updates for TeamCity itself and under Server Administration | Plugins for any available plugin updates. You can also subscribe to the security notification service to receive the latest information about security issues that may affect TeamCity or any other JetBrains products.

TeamCity undergoes regular security assessments and updates to fix any potential security issues.

2. Keep your credentials secure

Use of stolen or compromised credentials remains the most common cause of data breaches. Breaches caused by stolen or compromised credentials have an average cost of USD 4.5 million. Moreover, such breaches usually have the longest lifecycle, requiring on average around 243 days to identify and another 84 days to contain.

- **Use strong credentials, and use them carefully**

Use strong credentials for your entire DevOps toolkit, including your CI/CD server, and make sure to keep them out of:

– Repositories, such as GitHub, GitLab, etc.
– Environment variables, as they're often logged or shared with third-party monitoring systems.
– The Build log, to ensure you don't randomly log sensitive information.

If you use TeamCity

If you're using Versioned Settings (in the Kotlin DSL or XML format), never store your credentials in configuration files. Use tokens instead.

- **Store secure data with the password parameter type**

Whenever you want to store passwords or other sensitive data in your CI server settings, use the password parameter type. This will ensure that sensitive values never appear in the web UI and that they will also be asterisked in the build log.

2. Keep your credentials secure

- Use a secrets management tool

In addition, you can also make use of a centralized secrets management solution that your CI/CD server will use to retrieve any secrets it needs at build runtime. This not only allows you to securely consolidate all your secrets in one place but also gives you the option of automatically rotating your secrets after certain time periods, which is a cumbersome process when done manually.

If you use TeamCity

Where it's the case, use one of our external authentication modules, ranging from LDAP and Windows Domain integration to authenticating via GitHub, GitLab, or others. You can then disable the built-in authentication of TeamCity so that it will no longer keep hashed passwords in the internal database.

- Use external authentication

If your company makes use of a centralized user management system, use an integration module between that and your CI/CD server if possible. That way, you will avoid storing sensitive user data in multiple different locations, and it will be a lot easier for you to streamline identity and access management.

If you use TeamCity

You may consider using a tool like HashiCorp Vault, which lets you manage and rotate all the sensitive credentials you'll be using in a build and which integrates well with TeamCity.

- Use 2-factor authentication

Consider enabling 2FA so that administrators, or at the very least, any user who can change the build pipeline configurations, needs to use 2 different methods of verifying their identity (like their username and password and a smartphone app).

- Use encrypted SSH keys

If an unauthorized user gets your SSH keys, they can also gain access to the respective systems. Therefore, you should protect your SSH keys by encrypting them with a secure passphrase when you upload them to your CI/CD server, making them useless to attackers.

3. Establish efficient Identity and Access Management

A malicious user who gains access to the build server can do massive damage to the build infrastructure, as well as the users or systems that use the builds produced in it. Suppose a CI/CD server is compromised by an attacker. The chances are high that the attacker can gradually grant more permissions, due to the privileges CI/CD servers usually contain.

- Use the principle of least privilege

To tighten access security, you should use the principle of least privilege, which means giving your staff only the permissions to jobs, pipelines, or projects that they absolutely require to perform their roles – and not to anything else. Your developers could, for example, have access only to the compilation part of your build pipeline, while DevOps engineers could access and run just the deployment part.

This will help you minimize the attack surface if a malicious actor gains access to one of the users.

If you use TeamCity

Learn more about per-project and role-based authorization and access control in TeamCity

- Use groups and roles

Avoid granting direct permissions to individual users, which creates more attack opportunities for hackers. Instead, create user groups that match your organizational structure and assign relevant roles to those groups. Then you can add your users to the respective groups, granting them just enough privileges for their business needs.

We strongly recommend creating new roles with dedicated permissions instead of immediately assigning the administrator role to anyone needing extra privileges.

If you use TeamCity

TeamCity allows you to configure custom roles with permission levels suited to a given user’s needs. It also offers several roles predefined out of the box:

System Admin
Project Admin
Project Developer
Project Viewer