Week 3 - Networking, Multiplayer & Compliance

Potential Tool Security, Networking or Compliance Risks With Jenkins

Whilst Jenkins offers significant productivity benefits, it also has several security, networking and compliance risks that need to be discussed to ensure safe use. Understanding these risks and implementing appropriate mitigations is essential for anyone using Jenkins.

A primary risk associated with Jenkins is data transmission over a network. Jenkins pipelines frequently fetch code, dependencies or triggers builds from GitHub repositories. Plugins may also call external APIs. This data being sent often contains sensitive data including environment variables, credentials or build artifacts. If transmitted without proper protections, this data could be intercepted and accessed by unauthorised parties. As well as this, Jenkins also stores credentials for repositories, databases and APIs, increasing the amount of data accessed and that could possibly be leaked.

There are also many ways that Jenkins can be exploited. Pipeline scripts and plugins can introduce vulnerabilities, including remote code execution. Hackers who gain access to Jenkins through scripts or plugins could hijack builds, inject malware or steal data. Jenkins becoming compromised can therefore result in corruption of projects and builds, access to private repositories and deployment of dangerous software. These threats can also arise from security issues including untrusted developers with excessive privileges to sensitive data, weak credentials on accounts, repository tokens with too many unnecessary permissions enabled and unpatched instances of the project that are exposed to the internet. There are many methods of mitigating these security and data issues that using Jenkins risks occurring. One of these methods is making sure that all network traffic uses HTTPS with TLS 1.2 or higher, making sure data is protected during transit. To avoid unauthorised parties accessing secrets and data, the Jenkins credentials plugin can be used to store credentials that other plugins and builds use to access secrets without having to embed them in pipeline code. Jenkins provides plugins to help with tasks but not all plugins are safe as they aren't all checked for safety. To avoid malicious plugins, make sure to research what plugins you are downloading and using them. Projects will most of the time have many people working on or using a pipeline and this can risk people leaking sensitive data, this can be prevented by using role based actions to stop certain people accessing specific data or performing certain actions. Another thing that helps with a team of people is the audit trail plugin for Jenkins which keeps a log of who performed particular actions within Jenkins, so that you can easily see if someone did something they shouldn't have.

In conclusion, Jenkins does provide great automation benefits but also introduces a range of security and compliance challenges. Risks can include data exposure, exploitation and internal or external misuse, all of which can lead to serious consequences. Implementing strong encryption, access control and logging can mitigate these risks and protect any projects that Jenkins automates.

Bibliography

- Vulnerabilities and Scoring (s.d.) At: https://www.jenkins.io/security/vulnerabilities/https:/www.jenkins.io/security/vulnerabilities/ (Accessed 23/02/2026).

- Managing Security (s.d.) At: https://www.jenkins.io/doc/book/security/managing-security/https:/www.jenkins.io/doc/book/security/managing-security/ (Accessed 23/02/2026).
- credentials-plugin/docs/user.adoc at master · jenkinsci/credentials-plugin (s.d.) At: https://github.com/jenkinsci/credentials-plugin/blob/master/docs/user.adoc (Accessed 23/02/2026).
- Audit Trail (2025) At: https://plugins.jenkins.io/audit-trail (Accessed 23/02/2026).

AI Usage Declaration

- Chatgpt was used for paragraph structure and spelling and grammar checking.
- ChatGPT (s.d.) At: https://chatgpt.com/ (Accessed 09/02/2026).
