

Breaking Jenkins to

BUILD IT BETTER



A Beginner's Guide to Jenkins Security Fundamentals

About CyberWarfare Labs :

CW Labs is a renowned Infosec company specializing in cybersecurity practical learning. They provide on-demand educational services. The company has 3 primary divisions :

- 1. Learning Management System (LMS) Platform**
- 2. CWL CyberSecurity Playground (CCSP) Platform**
- 3. Infinity Learning Platform**



INFINITE LEARNING EXPERIENCE

About Me

- > Security Intern at CW Labs
- > Part time hacker, full time disaster!



What we will cover in this session

1

**Introduction
to Jenkins**

2

**Common Jenkins
Vulns & Flaws**

3

Demos

4

**Jenkins
Security Best
Practices**

5

**Checklist for
common Jenkins
security flaws**

6

**Conclusion
and QNA**



Introduction to Jenkins

What is Jenkins?

- Java based program
- Used in software development to automate task
- Automates building, testing & deploying software
- Also helps automate CI/CD pipelines

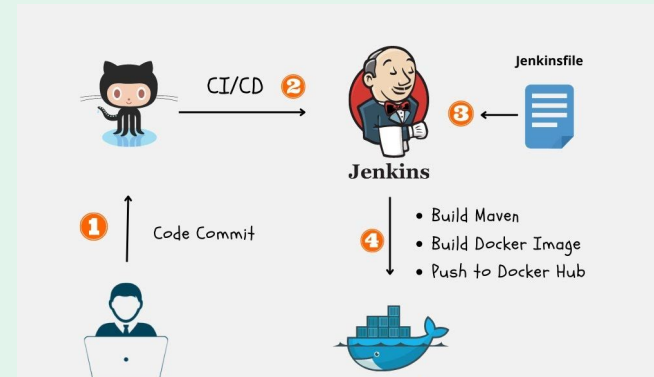


Image by Java Techie

Why Jenkins?

- Jenkins is not as popular as before, but it's still alive.
- Legacy systems continue to use Jenkins.
- Jenkins is everywhere, and "everywhere" means "everywhere attackers look!"
- Easy to learn and gets the job done. Also, who has time for security anyway? (It didn't end well.)

Why Should You Care About Jenkins Security?

- Your codebase
- Deployment credentials for AWS, Kubernetes etc
- Secrets like API keys and passwords
- The power to deploy (or destroy) production environments
- Many possible misconfigurations leave Jenkins servers vulnerable.





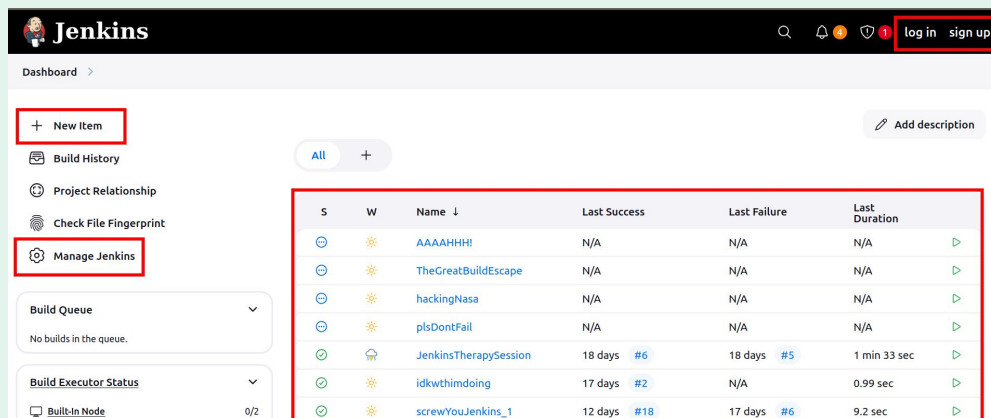
Common Jenkins Vulnerabilities and Security Flaws

1. The “Open For All” Jenkins Dashboard

- Sometimes Jenkins instances are left exposed with no authentication.
- Attackers could skip the login screen completely and walk right in and cause chaos



- Developers might think that the Jenkins server is safe behind a private network.
- An attacker could modify and mess around with the builds if left open without any authentication



2. Plugins: The Wolf in Sheep's Clothing

- Outdated or vulnerable plugins leave your system open to attacks.
- Installing plugins from untrusted sources jeopardizes your system's security.
- Avoid or use plugins like **"Script Security"** with extreme caution.



3. Secrets in Plain Sight

- Credentials in plaintext in a pipeline script is a security flaw.
- Credentials should not be exposed in the build logs.
- Usernames, passwords, and API tokens, should be managed using Jenkins' credential storage feature.

- Take a look at this demo pipeline script. It's a great example of how not to pass credentials into a job!

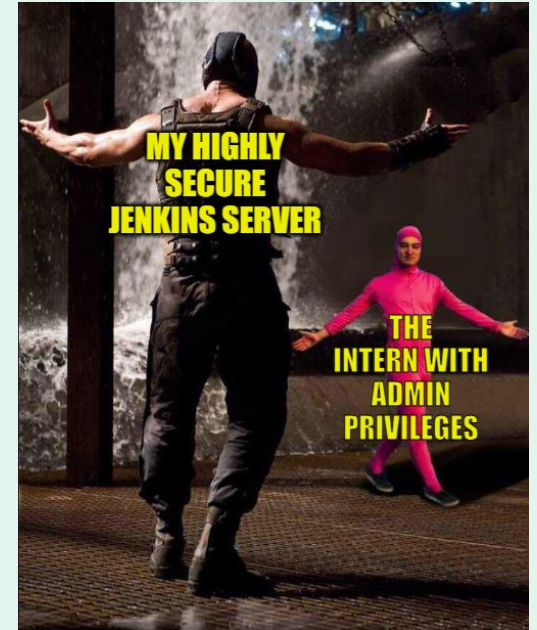
```

1 pipeline {
2   agent any
3   stages {
4     stage('Login to Jenkins UI') {
5       steps {
6         script {
7           sh '''
8             DEMO_URL="http://localhost:8080"
9             USER="enoch"
10            PASSWORD="testing@enoch"
11
12            RESPONSE=$(curl -s -u "$USER:$PASSWORD"
13            echo "Login Response: $RESPONSE"
14            '''
15         }

```

4. The Over Privileged “Admin”

- Assign appropriate permissions to users in Jenkins.
- Follow the principle of least privilege
- Give users only necessary permissions.



4. The Over Privileged “Admin”

- Use Matrix-based Security for flexible user access control.
- Implement Project-based Matrix Authorization Strategy for project-specific permissions.

Project-based Matrix Authorization Strategy

User/group	Overall	Credentials			Manage ownership	Agent					Job					Run													
	Administer	Read	Create	Delete	Manage Domains	Update	View	Jobs	Nodes	Build	Configure	Connect	Create	Delete	Disconnect	Provision	Build	Cancel	Configure	Create	Delete	Discover	Move	Read	Workspace	Delete	Replay	Update	
Anonymous	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Authenticated Users	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
developers	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Add user... Add group... ?

5. Allowing builds to run on the Built-In node

- Running builds on the "master" node (a.k.a. the Built-In Node) puts your systems at risks.
- Opens up more doors for threat actors
- Malicious actions can be performed on the system running jenkins.

5. Allowing builds to run on the Built-In node

- Attackers could execute commands, steal credentials, or access sensitive files.
- Could allow attackers to leave a backdoor into the system.





Demos

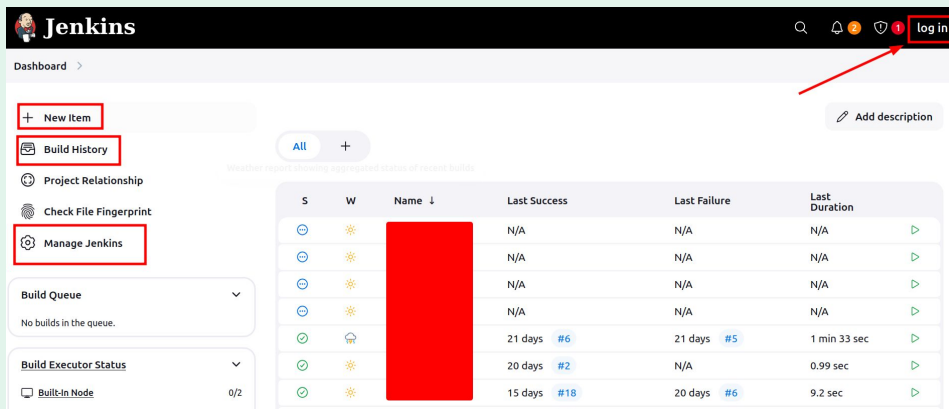
Demo 1: The Unauthenticated Dashboard

Demo 2: Groovy Script Injection

Demo 3: Exploiting Open Sign-Up and Unlimited
Access

Demo 1: The Unauthenticated Dashboard

- Anyone on the network can control Jenkins without restrictions.
- Attackers can find Jenkins using network scanning tools like Nmap.
- Unauthorized users can execute arbitrary scripts and commands.



The screenshot shows the Jenkins dashboard interface. In the top right corner, a 'log in' button is highlighted with a red arrow. On the left sidebar, the 'Manage Jenkins' button is highlighted with a red box. The main content area displays a table of build history with the following columns: S, W, Name, Last Success, Last Failure, and Last Duration. The 'Name' column is redacted with a black box.

S	W	Name ↓	Last Success	Last Failure	Last Duration
⊙	☀	[Redacted]	N/A	N/A	N/A
⊙	☀	[Redacted]	N/A	N/A	N/A
⊙	☀	[Redacted]	N/A	N/A	N/A
⊙	☀	[Redacted]	N/A	N/A	N/A
⊙	☀	[Redacted]	21 days #6	21 days #5	1 min 33 sec
⊙	☀	[Redacted]	20 days #2	N/A	0.99 sec
⊙	☀	[Redacted]	15 days #18	20 days #6	9.2 sec

Demo 2: Groovy Script Injection

- The Script Console allows admins to run Groovy code, typically in the **/script** directory.
- Security mechanisms are usually in place to protect it from unauthorized or unauthenticated users.

Script Console

Type in an arbitrary [Groovy script](#) and execute it on the server. Useful for trouble-shooting and diagnostics. Use the 'println' command to see the output (if you use Example:

```
println(Jenkins.instance.pluginManager.plugins)
```

All the classes from all the plugins are visible. jenkins.*, jenkins.model.*, hudson.*, and hudson.model.* are pre-imported.

```

1 // Specify the file path
2 def filePath = '/var/lib/jenkins/credentials.xml'
3
4 // Read the file contents
5 def file = new File(filePath)
6
7 if (file.exists()) {
8 // Print the file contents
9 println file.text
10 } else {
11 println "File not found: ${filePath}"
12 }
13

```

Demo 2: Groovy Script Injection

- Without security, unauthorized users could access the Script Console.
- Malicious users could exploit the console to run harmful Groovy code.

Result ⓘ

```
<?xml version="1.1" encoding="UTF-8"?>
<com.cloudbees.plugins.credentials.SystemCredentialsProvider plugin="credentials@1408.va_622a_b_f5b_1b_1">
  <domainCredentialsMap class="hudson.util.CopyOnWriteMap$Hash">
    <entry>
      <com.cloudbees.plugins.credentials.domains.Domain>
        <specifications/>
      </com.cloudbees.plugins.credentials.domains.Domain>
      <java.util.concurrent.CopyOnWriteArrayList>
        <org.jenkinsci.plugins.plaincredentials.impl.StringCredentialsImpl plugin="plain-credentials@183.va_de8f1dd5a_2b_">
          <scope>GLOBAL</scope>
          <id>dockerhubpwdcwl</id>
          <description></description>
          <secret>[REDACTED]</secret>
        </org.jenkinsci.plugins.plaincredentials.impl.StringCredentialsImpl>
        <com.cloudbees.plugins.credentials.impl.UsernamePasswordCredentialsImpl>
          <scope>GLOBAL</scope>
          <id>gitPAT</id>
          <description></description>
          <username>enoch-cwl</username>
          <password>[REDACTED]</password>
          <usernameSecret>false</usernameSecret>
        </com.cloudbees.plugins.credentials.impl.UsernamePasswordCredentialsImpl>
      </org.jenkinsci.plugins.plaincredentials.impl.StringCredentialsImpl>
    </scope>GLOBAL</scope>
```

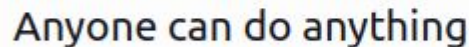
Demo 3: Exploiting Open Sign-Up and Unlimited Access

- The user sign-up option should never be enabled.
- Its disadvantages are more the advantages.

 A screenshot of a user interface element showing a toggle switch. The switch is currently turned on, indicated by a blue square with a white checkmark. To the right of the switch is the text "Allow users to sign up" followed by a question mark icon in a grey circle.

Allow users to sign up ?

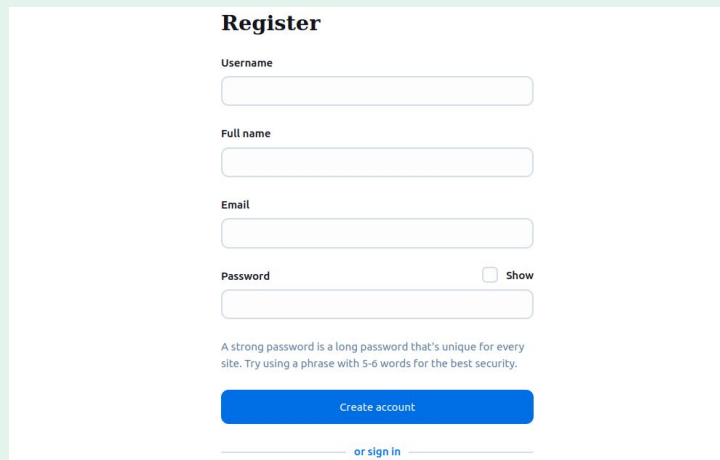
Authorization

 A screenshot of a user interface element showing a rounded rectangular box with a blue border. Inside the box, the text "Anyone can do anything" is displayed in a dark grey font.

Anyone can do anything

Demo 3: Exploiting Open Sign-Up and Unlimited Access

- Users are prompted to sign up once enabled.
- The system allows authenticated users full access.
- This can let attackers sign up and act maliciously.



Register

Username

Full name

Email

Password Show

A strong password is a long password that's unique for every site. Try using a phrase with 5-6 words for the best security.

[Create account](#)

[or sign in](#)



Jenkins Security Best Practices

Secure the Dashboard

- Never expose Jenkins over HTTP. Implement a reverse proxy (eg: Nginx, Traefik).
- Keep Jenkins in a private subnet, not letting the public access it.
- Use a strong password and implement additional verification strategies like MFA

Plugins: Less Is More

- Avoid experimenting with plugins in production and delete any unused plugins.
- Update plugins religiously. Jenkins plugin manager is your friend.

Hudson SCP publisher plugin 1.8

This plugin uploads build artifacts to repository sites using SCP (SSH) pro

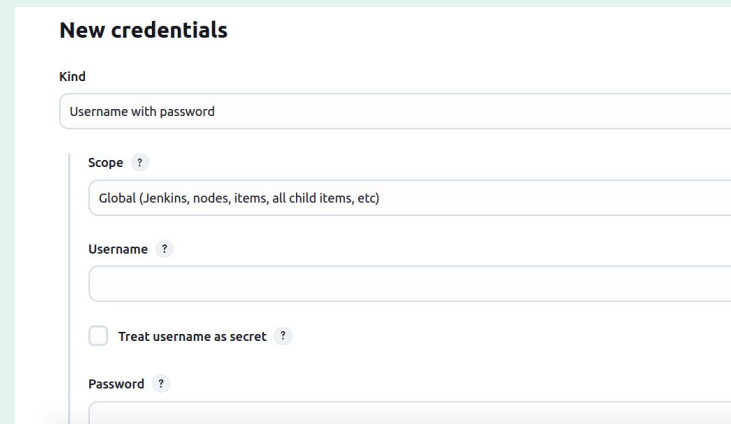
[Report an issue with this plugin](#)

Warning: The currently installed plugin version may not be safe to use.

- [CSRF vulnerability and missing permission check](#)
- [Insecure credential storage and transmission](#)

Utilize Secret Management

- Use Jenkins' Credentials Binding Plugin to store secrets.
- Never print secrets in logs
- Make sure that sensitive information is not displayed in the output console.
- Consider adding set +x in scripts to disable command echoing.



The screenshot shows the 'New credentials' configuration page in Jenkins. The 'Kind' is set to 'Username with password'. The 'Scope' is set to 'Global (Jenkins, nodes, items, all child items, etc)'. There are empty input fields for 'Username' and 'Password'. A checkbox for 'Treat username as secret' is present and unchecked.





```

1 stage('Docker Push') {
2   agent any
3   steps {
4     withCredentials([usernamePassword(credentialsId: 'dockerHub', passwordV
5       sh "docker login -u ${env.dockerHubUser} -p ${env.dockerHubPassword}"
6       sh "docker push shanen/spring-petclinic:latest"
7     ]
8   }
9 }
10 }
11 }

```

Role-Based Access Control

- Create roles (e.g., “developer”, “tester”) with the Role-Based Strategy Plugin.
- Follow the principle of least privilege.

User/group	Credentials			Manage ownership	Job						Run	SCM									
	Create	Delete	ManageDomains	Update	View	Jobs	Build	Cancel	Configure	Delete	Discover	Move	Read	Workspace	Delete	Replay	Update	Tag	<input type="checkbox"/>	<input type="checkbox"/>	
 Anonymous	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
 Authenticated Users	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
 developer	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	





Audit 🙌 Audit 🙌 Audit 🙌 Everything

- Check Manage **Jenkins** > **log** for suspicious activities.
- Plugins like “**Probely Security Scanner**” can help you perform vulnerability scans.
- Always keep your Jenkins server updated





Checklist for Common Jenkins Security Flaws

-  Is Jenkins exposed to the internet? Use Shodan to check if your Jenkins server is exposed.
-  Is the Jenkins UI left without any authentication
-  Are the plugins up to date? Manage Jenkins > Plugin Manager tells you all.
-  Can non admin users create jobs or run scripts?

- Are secrets (e.g., AWS keys) exposed in logs/Jenkinsfiles?
- Are sensitive data like usernames and passwords in plaintext in pipeline scripts?
- Are you logging too much?
- Is the **/script** directory accessible without authentication in Jenkins?
- Are you still using 'admin/admin' for authentication?

Here are some resources for future learning:

- [Official documentation on Securing Jenkins](#)
- [Hacking Jenkins!](#) ~ By Orange Tsai
- [Continuous Intrusion: Why CI Tools Are An Attackers Best Friend](#)





Conclusion and QNA



5TH
YEAR

ANNIVERSARY

SALE - 2025

UPTO **90%** OFF!

THE BIGGEST SALE OF THE YEAR IS

LIVE

ENROLL NOW



ENTERPRISE
RED TEAM
OPERATIONS
BUNDLE

CYBER DEFENSE
BUNDLE

CLOUD
RED TEAM
BUNDLE

EVASION &
EXPLOITATION
TACTICS BUNDLE

360
CYBER SECURITY
STARTER
PACK

Thank you for making it to the end! 🙌🙌

If you have any questions, now's your chance to ask

Feel free to connect with me on LinkedIn

<https://www.linkedin.com/in/enoch-benjamin-4835191a9/>