

Configure GIT for TLS inspection

Bypassing TLS inspection or disabling SSL verification can expose your system and data to serious security risks, including man-in-the-middle attacks. This article is provided strictly for educational and debugging purposes in trusted, controlled environments. Do not use these settings on production systems or networks you do not fully control.

Bypass TLS Inspection for Git Using GIT CONFIG

In enterprise environments, it's not uncommon for outbound HTTPS traffic to be intercepted and inspected by network security tools. This process—known as TLS inspection or SSL interception—can cause tools like Git to fail when attempting to communicate with remote repositories over HTTPS. A common error you might see is:

```
fatal: unable to access 'https://github.com/user/repo.git': SSL certificate problem: unable to get local issuer certificate
```

This happens because the intercepted certificate doesn't match what Git expects from the server, especially when a custom or self-signed certificate is involved.

Quick and Dirty: Disable SSL Verification

The fastest way to bypass TLS inspection is to disable SSL verification altogether for Git using this configuration:

```
git config --global http.sslVerify "false"
```

This tells Git not to verify the SSL certificate when connecting to remote repositories over HTTPS.

What This Command Does

- `--global`: Applies the setting for all repositories for the current user.
- `http.sslVerify "false"`: Disables SSL certificate validation.

This bypasses the TLS validation errors caused by mismatched or untrusted certificates during deep packet inspection.

When to Use This

- **Controlled internal environments** where you trust the internal proxy or TLS inspection appliance.
- **Temporary workaround** for debugging connectivity issues.
- **CI/CD environments** behind secure corporate firewalls with certificate inspection.

A Safer Alternative: Add the Root CA to Git’s Trust Store

Instead of disabling verification, a better approach is to configure Git to trust the inspection proxy’s certificate. Here's how:

1. **Obtain the root certificate** used by the inspection tool (usually a `.cert` file).
2. **Tell Git to use it:**

```
git config --global http.sslCAInfo <pathtocorporate-root-ca>.cert
```

This method allows you to retain SSL verification while trusting your organization's inspection certificate.

Final Notes

- **Never disable SSL verification for external, public repositories unless absolutely necessary.**
- If you use the `http.sslVerify false` setting, consider using it with the `--local` or `--global` scope carefully. You can also override it per repository using:
`git config --local http.sslVerify false`
- Always revert to secure practices once the underlying issue (e.g., missing root CA) is resolved.

Method	Security Risk	Use Case
<code>git config --global http.sslVerify "false"</code>	High	Temporary debugging in trusted environments
Add custom CA via <code>http.sslCAInfo</code>	Low	Permanent, secure solution for TLS interception

When in doubt, favor secure configuration over convenience. If you're working behind a corporate firewall and unsure how to proceed, reach out to your IT department—they may already have a secure solution in place.

Revision #3

Created 10 June 2025 12:43:54 by Luca Noah Caprez

Updated 10 June 2025 13:32:42 by Luca Noah Caprez