Traefik TLS Documentation

- Configuration Example
- Configuration Options
- TLS Options and HostSNI
- Certificate Resolver
- Custom Domains

When a TCP router is configured to handle TLS traffic, include a `tls` field in its definition. This field tells Traefik that the router should process only TLS connections and ignore non-TLS traffic.

By default, a router with a TLS field will terminate the TLS connections, meaning that it will send decrypted data to the services.

Configuration Example

```yaml
tcp:
  routers:
    my-tls-router:
      rule: "HostSNI(`example.com`)"
      service: "my-tcp-service"
      tls:
        passthrough: true
        options: "my-tls-options"
        domains:
          - main: "example.com"
            sans:
              - "www.example.com"
              - "api.example.com"
        certResolver: "myresolver"
```

```toml
[tcp.routers.my-tls-router]
  rule = "HostSNI(`example.com`)"
```

```
  service = "my-tcp-service"

  [tcp.routers.my-tls-router.tls]
    passthrough = true
    options = "my-tls-options"
    certResolver = "myresolver"

    [[tcp.routers.my-tls-router.tls.domains]]
      main = "example.com"
      sans = ["www.example.com", "api.example.com"]
```

```
labels:
  - "traefik.tcp.routers.my-tls-router.tls=true"
  - "traefik.tcp.routers.my-tls-router.rule=HostSNI(`example.com`)"
  - "traefik.tcp.routers.my-tls-router.service=my-tcp-service"
  - "traefik.tcp.routers.my-tls-router.tls.passthrough=true"
  - "traefik.tcp.routers.my-tls-router.tls.options=my-tls-options"
  - "traefik.tcp.routers.my-tls-router.tls.certResolver=myresolver"
  - "traefik.tcp.routers.my-tls-router.tls.domains[0].main=example.com"
  - "traefik.tcp.routers.my-tls-router.tls.domains[0].sans=www.example.com,api.example.com"
```

```
{
  //...
  "Tags": [
    "traefik.tcp.routers.my-tls-router.tls=true",
    "traefik.tcp.routers.my-tls-router.rule=HostSNI(`example.com`)",
    "traefik.tcp.routers.my-tls-router.service=my-tcp-service",
    "traefik.tcp.routers.my-tls-router.tls.passthrough=true",
    "traefik.tcp.routers.my-tls-router.tls.options=my-tls-options",
    "traefik.tcp.routers.my-tls-router.tls.certResolver=myresolver",
    "traefik.tcp.routers.my-tls-router.tls.domains[0].main=example.com",
    "traefik.tcp.routers.my-tls-router.tls.domains[0].sans=www.example.com,api.example.com"
  ]
}
```

??? info "Postgres STARTTLS"

```
Traefik supports the Postgres STARTTLS protocol,
which allows TLS routing for Postgres connections.
```

```
To do so, Traefik reads the first bytes sent by a Postgres client,
identifies if they correspond to the message of a STARTTLS negotiation,
and, if so, acknowledges and signals the client that it can start the TLS handshake.

Please note/remember that there are subtleties inherent to STARTTLS in whether the connection
ends up being a TLS one or not.
These subtleties depend on the `sslmode` value in the client configuration (and on the server
authentication rules).
Therefore, it is recommended to use the `require` value for the `sslmode`.

Afterwards, the TLS handshake, and routing based on TLS, can proceed as expected.

!!! warning "Postgres STARTTLS with TCP TLS PassThrough routers"

    As mentioned above, the `sslmode` configuration parameter does have an impact on whether a
STARTTLS session will succeed.
    In particular in the context of TCP TLS PassThrough, some of the values (such as `allow`)
do not even make sense.
    Which is why, once more it is recommended to use the `require` value.
```

Configuration Options

Field	Description	Default	Required
`passthrough`	Defines whether the requests should be forwarded "as is", keeping all data encrypted.	false	No
`options`	enables fine-grained control of the TLS parameters. It refers to a TLS Options and will be applied only if a `HostSNI` rule is defined. Unlike HTTP routers, several TCP routers can serve the same `HostSNI` with different TLS options. See TLS Options and HostSNI for more details.	""	No

Field	Description	Default	Required
`certResolver`	The name of the certificate resolver to use for automatic certificate generation via ACME providers (such as Let's Encrypt). See the Certificate Resolver section for more details.	""	No
`domains`	List of domains and Subject Alternative Names (SANs) for explicit certificate domain specification. See the Custom Domains section for more details.	[]	No

TLS Options and HostSNI

Contrary to the HTTP routers, several TCP routers can serve the same `HostSNI` with different TLS options, and this is not a conflict.

A TCP router rule is not limited to the `HostSNI` matcher, and the TCP muxer is able to decide which route matches a connection before the TLS handshake happens. The TLS options are therefore selected per route, and not per host name, which makes a configuration such as the following one valid:

```
# Dynamic configuration

tcp:
  routers:
    routerfoo:
      rule: "HostSNI(`example.com`) && ClientIP(`192.168.0.11`)"
      tls:
        options: foo

    routerbar:
      rule: "HostSNI(`example.com`) && ClientIP(`192.168.0.12`)"
      tls:
        options: bar
```

```
# Dynamic configuration

[tcp.routers]
  [tcp.routers.routerfoo]
    rule = "HostSNI(`example.com`) && ClientIP(`192.168.0.11`)"
    [tcp.routers.routerfoo.tls]
      options = "foo"

  [tcp.routers.routerbar]
    rule = "HostSNI(`example.com`) && ClientIP(`192.168.0.12`)"
    [tcp.routers.routerbar.tls]
      options = "bar"
```

Certificate Resolver

The `tls.certResolver` option allows you to specify a certificate resolver for automatic certificate generation via ACME providers (such as Let's Encrypt).

When a certificate resolver is configured for a router, Traefik will automatically obtain and manage TLS certificates for the domains specified in the router's rule (in the `HostSNI` matcher) or in the `tls.domains` configuration (with `tls.domains` taking precedence).

!!! important "Prerequisites"

```
- Certificate resolvers must be defined in the [static configuration](../../install-
configuration/tls/certificate-resolvers/acme.md)
- The router must have `tls` enabled
- An ACME challenge type must be configured for the certificate resolver
```

Custom Domains

When using ACME certificate resolvers, domains are automatically extracted from router rules, but the `tls.domains` option allows you to explicitly specify the domains and Subject Alternative Names (SANs) for which certificates should be generated.

This provides fine-grained control over certificate generation and takes precedence over domains automatically extracted from router rules.

Every domain must have A/AAAA records pointing to Traefik.

{% include-markdown "includes/traefik-for-business-applications.md" %}

□□□ #2
□ □□□ □□□ 2026-10-31 14:05:51 UTC
□ □□□ □□□ 2026-10-31 14:05:52 UTC