

[New issue](#)[Jump to bottom](#)

默认的 HostnameVerifier 为什么选择 TrustAnyHostnameVerifier 呢? #2042

✓ Closed

shansing opened this issue on Dec 26, 2021 · 1 comment

Labels

question

shansing commented on Dec 26, 2021 • edited ▾

版本情况

JDK版本: corretto_8_312

hutool版本: 5.7.18 (请确保最新尝试是否还有问题)

问题描述 (包括截图)

注意到 hutool HttpUtil 中 HttpURLConnection#setHttpsInfo 默认传入的 HostnameVerifier 是 DefaultSSLInfo.TRUST_ANY_HOSTNAME_VERIFIER, 即:

```
package cn.hutool.http.ssl;

import javax.net.ssl.HostnameVerifier;
import javax.net.ssl.SSLSession;

public class TrustAnyHostnameVerifier implements HostnameVerifier {
    public TrustAnyHostnameVerifier() {
    }

    public boolean verify(String hostname, SSLSession session) {
        return true;
    }
}
```

查看 javax.net.ssl.HttpURLConnection.DefaultHostnameVerifier 代码如下:

```
/**
 * <code>HostnameVerifier</code> provides a callback mechanism so that
```

```

    * implementers of this interface can supply a policy for
    * handling the case where the host to connect to and
    * the server name from the certificate mismatch.
    * <p>
    * The default implementation will deny such connections.
    */
    private static HostnameVerifier defaultHostnameVerifier =
        new DefaultHostnameVerifier();

    /*
    * The initial default <code>HostnameVerifier</code>. Should be
    * updated for another other type of <code>HostnameVerifier</code>
    * that are created.
    */
    private static class DefaultHostnameVerifier
        implements HostnameVerifier {
        @Override
        public boolean verify(String hostname, SSLSession session) {
            return false;
        }
    }

    /**
    * The <code>hostnameVerifier</code> for this object.
    */
    protected HostnameVerifier hostnameVerifier = defaultHostnameVerifier;

```

从 doc 看到 HostnameVerifier 是一种 fallback 机制，当要连接的 host 与服务端证书 server name 不匹配时才有作用。并且 javax.net.ssl.HttpURLConnection 的默认机制是这时候拒绝此类请求。而 hutool 则默认放行所有这类请求。

如果调用 cn.hutool.http.HttpRequest 时手动传入上面的
javax.net.ssl.HttpURLConnection.DefaultHostnameVerifier:

```

String result1 = HttpRequest.get("https://www.baidu.com")
    .setHostnameVerifier(HttpURLConnection.getDefaultHostnameVerifier())
    .execute().body();
System.out.println(result1);

```

也会请求成功。按我理解这样才是合理的，如果服务端证书 server name 与 host 不匹配，应该中断请求才安全。请问我的理解是否正确？是不是其实 javax.net.ssl.HttpURLConnection 的默认行为会有坑，hutool 才另外指定了 HostnameVerifier 呢？

谢谢。

looly commented on Dec 27, 2021

Member

之前Hutool的处理逻辑是按照JDK默认行为来的，但是大量的issue说他们的服务器或者URL无法请求，最后被当作bug修复的。

我理解的是服务端的证书一致与否对于浏览器来说比较重要，但是对于请求客户端来说并不重要，大量的内部接口证书十分混乱，错误问题不可避免，如果因为这种错误导致接口无法请求，我想不是大家所希望的。

Hutool的理解是，希望用户请求到给定地址的内容，至于内容可信与否并不重要，这个交给使用者来完成判断。



 looly closed this as completed on Dec 27, 2021

  looly added the `question` label on Dec 27, 2021

Assignees

No one assigned

Labels

`question`

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants

