

Configuring Tomcat to use SSL

APACHE
HTTP SERVER

Apache Tomcat



Secure

https://

1. Create a self-assigned public key-certificate

- ▶ SSL-based servers use X509 certificates to validate to clients that they are who they claim to be. This prevents hackers from hacking DNS servers to redirect SSL request to their site. For real world use the certificate needs to be signed by a trusted authority. For testing purposes, a self-assigned certificate is sufficient. To generate one that will be valid for two years (730 days) execute the following at the DOS prompt:

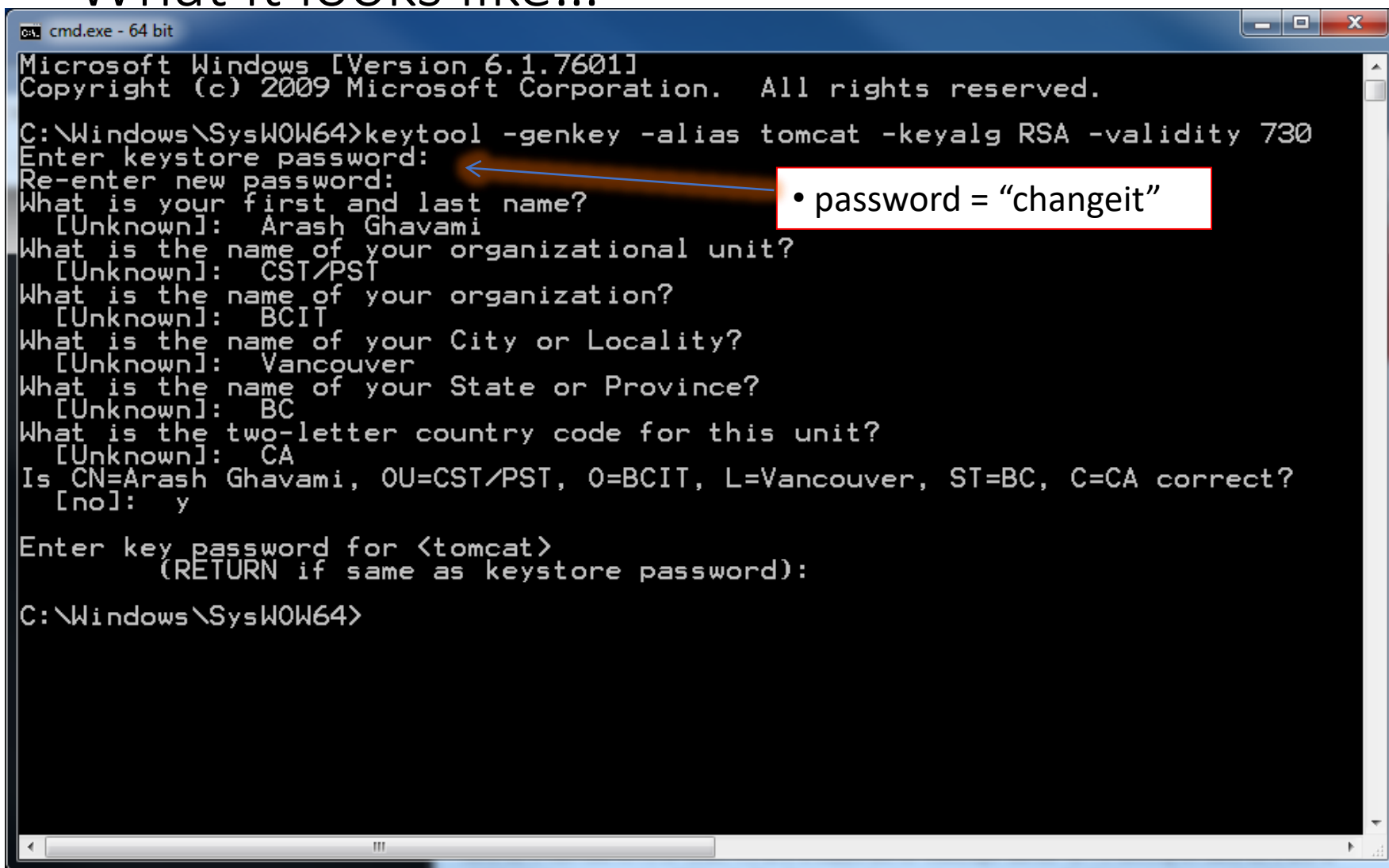
```
keytool -genkey -alias tomcat -keyalg  
RSA -validity 730
```

1. Create a self-assigned public key-certificate...

► The system will prompt for a variety of information:

- **First and last name** – this should be your server name, hostname, or IP address. Use localhost if you're running from the local machine.
- **Organization, and location** – enter any name for organization. I used my city of residence as the location.
- **Keystore password, and key password** – use “**changeit**” for both of these. I tried others that didn't work. The Tomcat documentation in server.xml instructs you to use “changeit”.
- The system will generate a file named **.keystore** and will save the file in `C:\Users\Users\<username>` on Windows machines, or `/home/username` on Unix.

What it looks like...



```
cmd.exe - 64 bit
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\SysWOW64>keytool -genkey -alias tomcat -keyalg RSA -validity 730
Enter keystore password:
Re-enter new password:
What is your first and last name?
[Unknown]: Arash Ghavami
What is the name of your organizational unit?
[Unknown]: CST/PST
What is the name of your organization?
[Unknown]: BCIT
What is the name of your City or Locality?
[Unknown]: Vancouver
What is the name of your State or Province?
[Unknown]: BC
What is the two-letter country code for this unit?
[Unknown]: CA
Is CN=Arash Ghavami, OU=CST/PST, O=BCIT, L=Vancouver, ST=BC, C=CA correct?
[no]: y

Enter key password for <tomcat>
(RETURN if same as keystore password):

C:\Windows\SysWOW64>
```

• password = "changeit"

2. Deploy the .keystore file

- Copy the .keystore file into the tomcat home directory
 - (eg. C:\\%tomcat_home%\\)

3. Edit server.xml file

- Uncomment the SSL connector entry in tomcat_home/conf/server.xml (somewhere around line 84).
- Change code:

```
<Connector port="8443"  
  protocol="org.apache.coyote.http11.Http11NioPr  
  otocol"maxThreads="150" SSLEnabled="true"  
  scheme="https" secure="true"  
  clientAuth="false" sslProtocol="TLS"  
  keystoreFile=".keystore"  
  keystorePass="changeit"/>
```

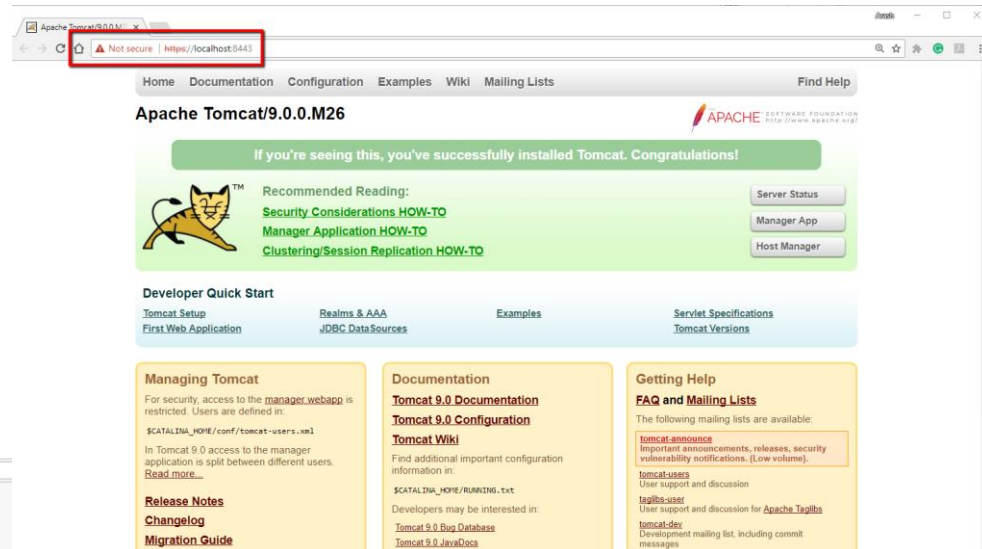
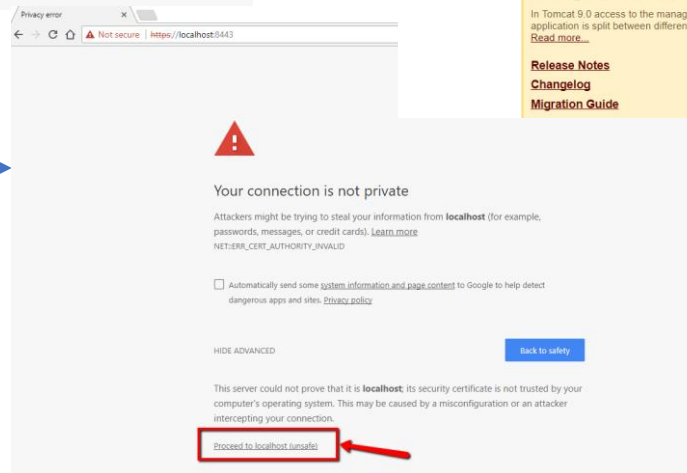
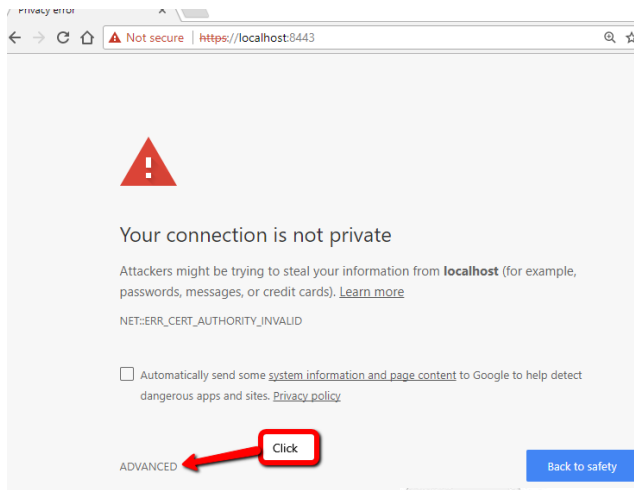
Watch for typos!

Add **bolded**
code

5. Restart the web server.

- Start Tomcat and browse to <https://localhost:8443/>.
- You won't see the page right away. Security warnings will be displayed in popup windows. In most browsers you will have to follow the prompts to install the certificate you created. Once you have done this, the Tomcat homepage should be visible. From here you are all set to employ SSL in your applications.

In Google Chrome



Enabling SSL for your application

- Annotate a Servlet class with @ServletSecurity

```
@WebServlet(urlPatterns="/http2")  
@ServletSecurity(@HttpConstraint(transportGuarantee = TransportGuarantee.CONFIDENTIAL))  
public class Http2Servlet extends HttpServlet {
```

OR

- In the deployment descriptor file (web.xml) define security constraints

```
<security-constraint>  
    <web-resource-collection>  
        <web-resource-name>HTTP2</web-resource-name>  
        <url-pattern>/http2/*</url-pattern>  
    </web-resource-collection>  
    <user-data-constraint>  
        <transport-guarantee>CONFIDENTIAL</transport-guarantee>  
    </user-data-constraint>  
</security-constraint>
```