Running Jenkins behind IIS

In situations where you have existing web sites on your server, you may find it useful to run Jenkins (or the servlet container that Jenkins runs in) behind Nginx, so that you can bind Jenkins to the part of a bigger website that you may have. This document discusses some of the approaches for doing this.

Make sure that you change the Jenkins httpListenAddress from its default of 0.0.0.0 to 127.0.0.1 or configure the firewall to block request on the port Jenkins is bound to, otherwise any IIS-level restrictions can be easily bypassed by hitting the Jenkins port directly.

Requirements

- IIS 7.0 or greater.
- IIS 8.5 or greater if you want Certificate Rebind.
- URL Rewrite 2.1 or greater.
- As the announcement explains, it introduces a feature flag to turn off the default non-compliant-RFC3986 behavior. Which is what we want.
- Application Request Routing 3.0 or greater.
- Server access

Example use case

I have a dedicated Jenkins installation on a Windows Server 2012 R2 server with a Common Name of VRTJENKINS01 in the Active Directory domain acme.example and is reachable by the Fully Qualified Domain Name vrtjenkins01.acme.example. Additionally Jenkins runs on port 8080 and already listens to 127.0.0.1 instead of 0.0.0.0 and the server has additional DNS names: jenkins and jenkins.acme.example.

I want to have an IIS installation which acts as a TLS/SSL terminating reverse proxy. In combination with our in-house Active Directory Certificate Services (ADCS, Microsoft's Certificate Authority software) this should make certificate management a lot easier since Windows can be configured to automatically renew certificates, and the IIS 8.5+ Certificate Rebind feature can listen to renewal events (which contain the fingerprints of both the old and new certificate) and update the relevant bind(s) to use the fresh certificate. This would ensure that after the initial manual request it would only be necessary to manually change TLS/SSL related settings when the set of Alternate Subject Names on the certificate IIS presents should change.

IIS will only have to act as 1) a reverse proxy for Jenkins 2) redirect non-canonical URLs to the canonical URL: https://jenkins.acme.example/

I have installed the IIS (8.5) role using the Add Roles and Features Wizard with the all the default and also the following non-default features:

- HTTP Redirection (Under Common HTTP Features, to redirect http(s)://jenkins/, etc. to https://jenkins.acme.example/)
- WebSocket Protocol (Under Application Development, because I felt like it)

Then I installed URL Rewrite and Application Request Routing.

Configuration Time

Enabling Reverse Proxy functionality

1. In the Internet Information Services (IIS) Manager click on the VRTJENKINS01 server.
2. Go to Application Request Routing Cache.
3. In the Actions panel click on Server Proxy Settings...
4. Enable the proxy
5. Disable the Reverse rewrite host in response header
   a. Don't worry, it will work, just follow the rest of the instructions
6. Set the Response buffer threshold (KB) to 0.
   a. This helps to prevent HTTP 502 errors on Jenkins' Replay pages.
7. Apply (the Actions panel again)

Configuring TLS/SSL

Out of scope, there are enough tutorials on the rest of the interwebs for this part. The rest of this tutorial will assume it has been configured with a certificate trusted by your browser of choice.

Configuring rules for response rewriting

1. Go to the Default Web Site
2. Go to URL Rewrite
3. In the Actions panel click View Server Variables...
4. Add the following is not already define on the server level:
   a. Name: HTTP_FORWARDED
5. Click on Back to Rules...
6. Click on Add Rule(s)...
7. Select Reverse Proxy and click on OK
8. Enter jenkins.acme.example and click on OK
9. Open the rule you just created
10. Under **Conditions** add:
   a. Condition input: `{CACHE_URL}`
   b. Pattern: `^({http|ws}s://)`

11. Under **Server Variables** add:
   a. Name: `HTTP_FORWARDED`, Value: `for=[REMOTE_ADDR];by=[LOCAL_ADDR];host="[HTTP_HOST]";proto="https"`, Replace: yes
   i. Jenkins runs under Jetty, Jetty supports RFC7239, so all should be well.

12. Under **Action change**:
   a. Rewrite URL to `C:1://jenkins.acme.example:8080{UNENCODED_URL}`
   i. Note that there is no slash between the port number and the opening curly bracket
   b. Remove the check from the **Append query string** checkbox

13. Apply the changes.

14. Edit `C:\Windows\System32\drivers\etc\hosts` so that `jenkins.acme.example` points to 127.0.0.1
   a. When resolving names Windows will check if the name is its own name before consulting the hosts file. Meaning that adding `vrtnjenkins01.acme.example` to the hosts file won't have any effect.
   i. The hosts file will however be consulted before consulting the DNS infrastructure

**Experiencing the dreaded "It appears that your reverse proxy set up is broken." error for yourself**

1. https://jenkins.acme.example/configure
2. Configure the **Jenkins URL** to be **https://jenkins.acme.example/** and Save the change
3. Go to **Configure Global Security and enable Enable proxy compatibility** if you have already enabled. Prevent Cross Site Request Forgery exploits
4. Go to https://jenkins.acme.example/manage
5. You will still experience the "It appears that your reverse proxy set up is broken." as expected
   a. If you do not get that at this point, then that is very weird... Continue anyway.
6. Right click the **Configure System** link and choose to inspect the element.
   a. Make sure you are still on the Manage page as you will want it as your referrer
7. Change the value of the `href` attribute to `administrativeMonitor/hudson.diagnosis.ReverseProxySetupMonitor/test` and turn on **Allow doube escaping**
8. Open the link you just changed in a new tab.
   a. Keep this tab open
9. Observe the "https://jenkins.acme.example/manage vs http: error and bask in its glory
   a. a white page served with HTTP status code is 200 indicates all is well.
   i. If you do get that at this point, then that is very weird... Continue anyway.

**Fixing the errors**

1. In IIS Manager got to **Application Pools** then edit **DefaultAppPool** so that the **.NET CLR version** is **No Managed Code**
   a. You might find that this is not necessary (at far as you can tell) for your setup, since IIS will only act as a TLS/SSL offloading reverse proxy, we don't need it. IIS on Windows 7 does appear to need this to be turned off for some urls (https://jenkins.acme.example/*)
2. Then go to **Sites Default Web Site Request Filtering** and in the **Actions** panel choose **Edit Feature Settings...** and turn on **Allow doube escaping**
   a. This is so IIS forwards URLs like `https://jenkins.acme.example/%2525` to Jenkins instead of showing an IIS error page
3. Last, but not least, go to **Sites Default Web Site Configuration Editor** and change the **Section to system.webServer/rewrite/rules**
4. Now you should see the URL Rewrite 2.1 property `useOriginalURLEncoding` listed, if not install URL Rewrite 2.1 using the x86 or x64 installer, not the WebPI one and resume from here after a reboot.
5. Change `useOriginalURLEncoding` to **False**
   a. As the URL Rewrite 2.1 announcement this will change the value of `{UNENCODED_URL}` to make it RFC3986 and usable for reverse proxy forwarding purposes.
   b. original as in pre 2.1 behaviour.
6. Refresh that tab you were supposed to keep open, or recreate it.
   a. Again, take some time to bask in its glory
7. It should now be white, also the Manage page should no longer complain!

**Continue configuring IIS**

Some of the things you might want but I won't cover:

- Hypertext Strict Transport Security headers
- Redirecting from non canonical URLs to the canonical URL (ok, sort of covered this in the web.config example)
- The X-UA-Comaptibility header so that Internet Explorer 11 (or 9, or ...) won't claim to be IE 7 for intranet sites
- Use IIS Crypto to configure cipher suites
- ...

**A working web.config**
<configuration>
<system.webServer>
<rewrite>
<rules useOriginalURLEncoding="false">
<rule name="CanonicalHostNameRule2" stopProcessing="true">
<match url="(.*)" />
<conditions trackAllCaptures="true">
<add input="{CACHE_URL}" pattern="^(http|ws)://" />
<add input="{HTTP_HOST}" pattern="jenkins|jenkins01\.|acme\.|example$|vrtjenkins01$|vrtjenkins01\.|acme\.|example$" />
</conditions>
<action type="Redirect" url="{C:1}://jenkins.acme.example{UNENCODED_URL}" appendQueryString="false" redirectType="Permanent" />
</rule>
<rule name="CanonicalHostNameRule1" stopProcessing="true">
<match url="(.*)" />
<conditions trackAllCaptures="true">
<add input="{CACHE_URL}" pattern="^(https|wss)://" />
<add input="{HTTP_HOST}" pattern="jenkins|vrtjenkins01\.|acme\.|example$" />
</conditions>
<action type="Redirect" url="{C:1}://jenkins.acme.example{UNENCODED_URL}" appendQueryString="false" redirectType="Permanent" />
</rule>
<rule name="ReverseProxyInboundRule1" stopProcessing="true">
<match url="(.*)" />
<action type="Rewrite" url="{C:1}://jenkins.acme.example:8080{UNENCODED_URL}" appendQueryString="false" />
<serverVariables>
<set name="HTTP_FORWARDED" value="for={REMOTE_ADDR};by={LOCAL_ADDR};host='"{HTTP_HOST}";proto="https"" />
</serverVariables>
<conditions trackAllCaptures="true">
<add input="{CACHE_URL}" pattern="^(http|ws)s://" />
<add input="{HTTP_HOST}" pattern="jenkins\.|vrtjenkins01\.|acme\.|example$" />
</conditions>
</rule>
</rules>
</rewrite>
<security>
:requestFiltering allowDoubleEscaping="true" />
</security>
</system.webServer>
</configuration>

Complain about it still not working
Leave a comment or complain to me over at twitter (@Darsstar)