



Application Note: Securing WebView

This application note describes how to secure the North ObServer WebView server, on an Internet-facing network, using a reverse proxy and HTTPS.

Contents

Introduction	3
What is a reverse proxy?	3
Internet Information Services	3
HTTPS and certificates	4
How to implement a reverse proxy	5
Configure WebView	5
Create a reverse proxy in IIS	5
Enable HTTPS	6
Redirect HTTP to HTTPS.....	6
Optimising IIS Performance	8
Compression	8
Cache.....	8
IIS Buffer Size	9
ObServer as a Windows Service	10
Document Versions.....	11

Introduction

WebView, the web server component to North's ObServer, uses HTTP to serve its pages. To increase the security of WebView sites, particularly on Internet-facing networks, we recommend using a reverse proxy.

What is a reverse proxy?

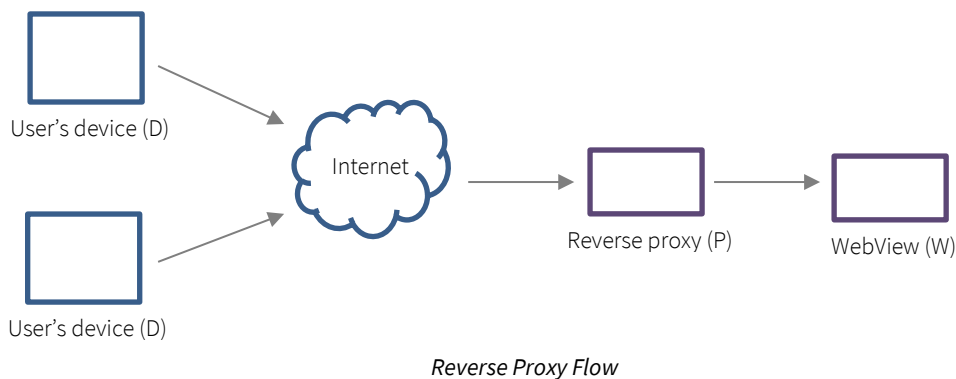
A reverse proxy is a server that sits in front of web servers and forwards client (e.g. web browser) requests to those web servers. Reverse proxies are typically implemented to help increase security, performance, and reliability.

For example, let's name the computers involved in a reverse proxy communication:

D: A web browser on a user's computer or device;

P: The reverse proxy server;

W: The web server, in this case a WebView server.



Usually all requests from a user's device (D) would go directly to the WebView server (W), and W would send responses directly to D. With a reverse proxy, all requests from a user's device (D) will go directly to the reverse proxy (P), and P will send its requests to and receive responses from WebView server (W). P will then pass along the appropriate responses to D.

Some of the benefits of using a reverse proxy include:

- HTTPS – a reverse proxy can be configured to decrypt all incoming requests and encrypt all outgoing responses, protecting the integrity and confidentiality of data to the user's device.
- Protection from attacks – with a reverse proxy in place, a web site never needs to expose the IP address of the origin server. This makes it much harder for attackers to leverage a targeted attack against the server. Instead the attackers will only be able to target the reverse proxy, which will have tighter security.
- Caching & compression – a reverse proxy can also cache (or temporarily save) and compress content, resulting in faster performance.

Internet Information Services

WebView operates on a Windows operating system, so this document will describe configuring Windows Internet Information Services (IIS) as a reverse proxy. For simplicity, let's assume both are installed on the same Windows Server.

Other reverse proxies may be used, including Stunnel, Nginx, AWS CloudFront, etc., but are outside the scope of this document.

HTTPS and certificates

HTTPS (Hypertext Transfer Protocol Secure) is an Internet communication protocol that protects the integrity and confidentiality of data between the user's computer and a web site. Data sent using HTTPS is secured via Transport Layer Security protocol (TLS).

You must obtain a security certificate as a part of configuring a reverse proxy and enabling HTTPS for your WebView site. The certificate is issued by a certificate authority (CA), which takes steps to verify that your web address actually belongs to your organisation. So first you'll need a web address to access the site, like 'webview.example.com'.

A certificate will need to be renewed with a CA regularly, at least every year. For larger organisations, your IT department should manage Windows Server updates and renew security certificates. For smaller organisations, a free certificate from Let's Encrypt (certifytheweb.com) that automatically renews may be preferred.

How to implement a reverse proxy

In this section, we'll outline the main tasks in implementing a reverse proxy for WebView using IIS. Examples below assume you are using IIS 10.0 on Windows Server 2016.

Configure WebView

You should have already installed North ObSys. This guide assumes both ObSys and IIS are on the same Windows Server, so here we'll block network access to WebView by restricting it to use the local loopback address (127.0.0.1). To stop any port conflict with IIS, we'll also change WebView's port number to 8080.

If you are using a reverse proxy on a separate computer then skip these steps. Instead note the IP address WebView is using.

- 🖥 To set up WebView's network interface, follow these steps:
 - Launch **Start Engineering**
 - Open **ObServer, Configuration, WebView Server, Network Interface**
 - Set **Force IP address** to '127.0.0.1'
 - Set **TCP port** to '8080'.

Create a reverse proxy in IIS

In Windows Server, you should have already added the Web Server (IIS) role.

During these steps you will need the public web address that user's will access WebView, e.g. webview.example.com. This address should be assigned in DNS to the Windows Server.

From IIS Manager, using Get New Web Platform Components, install the 'URL Rewrite' and 'Application Request Routing 3.0' modules, we'll use these below.

- 🖥 To create the reverse proxy in IIS, follow these steps:

- Launch **IIS Manager**
- Under **Sites**, select **Default Web Site** or the site you are using for WebView's reverse proxy
- Open the **URL Rewrite** feature
- Choose **Add Rule(s)...**, select **Reverse Proxy**, then **OK**
- You may be prompted to enable proxy functionality in ARR, click OK
- under **Inbound Rules** – set **IP address** '127.0.0.1:8080' (WebView server IP address and port), and check **Enable SSL Offloading**
- Under **Outbound Rules** – uncheck **Rewrite the domain names**
- Click **OK**

Using a web browser, you should now be able to access the WebView site using the public web address (or IP address of the Windows Server).

Add Reverse Proxy Rules

Inbound Rules

Enter the server name or the IP address where HTTP requests will be forwarded:

127.0.0.1:8080

Example: contentserver1

☒ Enable SSL Offloading

Selecting this option will forward all HTTPS requests over HTTP.

Outbound Rules

☐ Rewrite the domain names of the links in HTTP responses

Responses that are generated by applications that are behind a reverse proxy can have HTTP links that use internal domain names. These links must be updated to use external domain names.

From:

To:

Example: contentserver1

Example: www.contoso.com

OK Cancel

Enable HTTPS

Before you can enable HTTPS, you will need a security certificate. Navigate to the Server Certificates in IIS Manager and import your SSL certificate, or create a self-signed certificate.

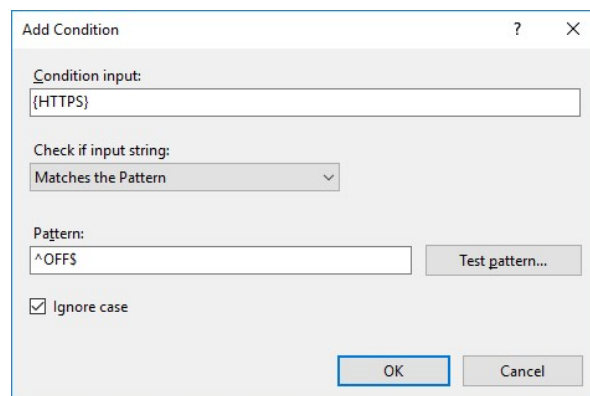
- 🖥️ To enable HTTPS for the reverse proxy, follow these steps:
 - Launch **IIS Manager**
 - Under **Sites**, select **Default Web Site** or the site you are using for WebView's reverse proxy
 - Select **Bindings...** from the Action menu
 - Click **Add**, set **Type** to 'https', select the **SSL certificate**
 - If IIS is hosting multiple sites, set **Host name** to the public web address
 - Click **OK**
 - **Restart** the Website

Using a web browser, you should now be able to access the WebView site using https.

Redirect HTTP to HTTPS

IIS will respond to requests via http and https. To keep the WebView site secure we are now going to redirect any http requests to the more secure https.

- 🖥️ To redirect HTTP requests, follow these steps:
 - Launch **IIS Manager**
 - Under **Sites**, select **Default Web Site** or the site you are using for WebView's reverse proxy
 - Open the **URL Rewrite** feature
 - Choose **Add Rule(s)...**, select **Blank**, then **OK**
 - Give your redirect a name, such as 'HTTPS Redirect'
 - Set **Requested URL** to 'Matches the Pattern', **Using** to 'Regular Expressions', **Pattern** to '(.*)', and check **Ignore case**
 - Expand **Conditions**, select 'Match All', then **Add...**



- Set **Condition input** to '{HTTPS}', select 'Matches the Pattern', set **Pattern** '^OFF\$', check **Ignore case**, then **OK**
- Scroll down to the **Action** section

- Set **Action type** to 'Redirect', **Redirect URL** 'https://{HTTP_HOST}/{R:1}', check **Append query string**, **Redirect type** 'Permanent (301)'



The screenshot shows the 'Action' configuration window in IIS Manager. It has a title bar 'Action' with a close button. Inside, there's a section 'Action type:' with a dropdown menu showing 'Redirect'. Below that is a section 'Action Properties' containing a text box for 'Redirect URL' with the value 'https://{HTTP_HOST}/{R:1}'. There's a checkbox labeled 'Append query string' which is checked. At the bottom, there's a section 'Redirect type:' with a dropdown menu showing 'Permanent (301)'.

- Click **Apply** in the Actions menu, then **Back to Rules**
- In the Inbound rules list, **Move Up** the 'HTTPS Redirect' rule to the top
- **Restart** the Website

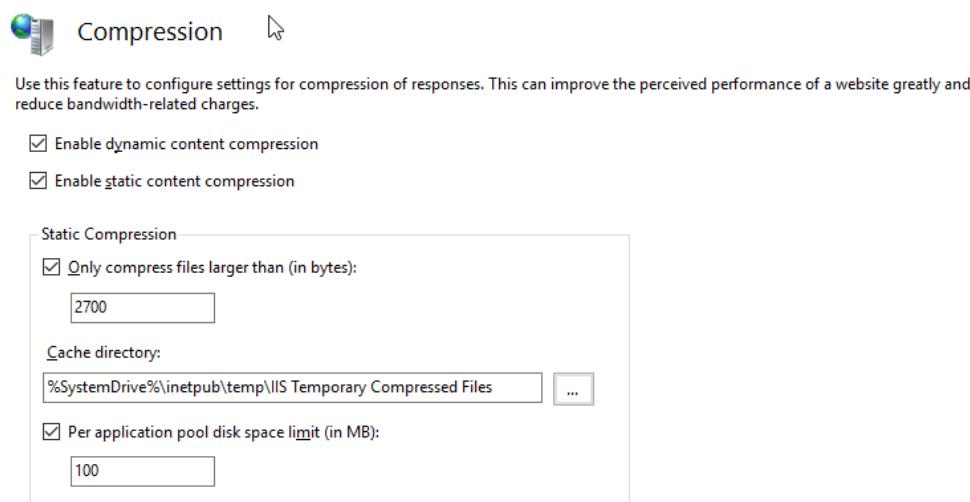
Optimising IIS Performance

To tune the performance of WebView sites, you may wish to optionally enable compression and cache in IIS. Refer to IIS documentation for more information on these features.

Compression

WebView does not provide compression of any responses. Enabling static content compression in IIS may increase performance and reduce the bandwidth of loading images, stylesheets, and other static content. Experiment with enabling dynamic content compression, bandwidth will be reduced but pages may take longer to load.

From IIS Manager, using Get New Web Platform Components, search 'compression' and install the IIS Compression Modules. Enable compression options from the Compression feature at the IIS server home.

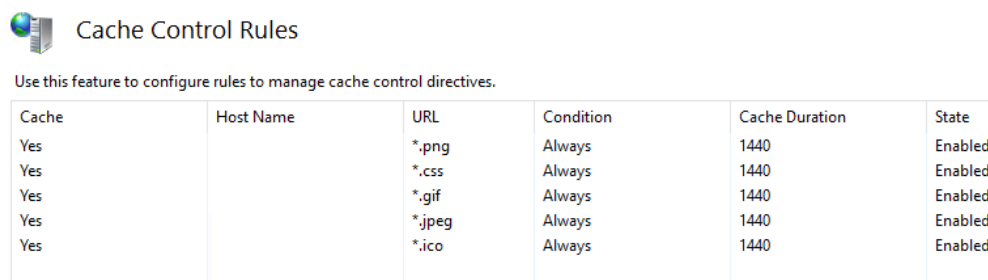


The screenshot shows the 'Compression' feature configuration in IIS Manager. It includes a description: 'Use this feature to configure settings for compression of responses. This can improve the perceived performance of a website greatly and reduce bandwidth-related charges.' There are two checked checkboxes: 'Enable dynamic content compression' and 'Enable static content compression'. Below these is a 'Static Compression' section with a checked checkbox 'Only compress files larger than (in bytes):' with a text box containing '2700'. The 'Cache directory:' is set to '%SystemDrive%\inetpub\temp\IIS Temporary Compressed Files'. Another checked checkbox is 'Per application pool disk space limit (in MB):' with a text box containing '100'.

Cache

WebView generally responds with the no-cache header. To increase performance, enable caching of some static files within IIS.

- 🖥️ To enable caching of static content, follow these steps:
 - Launch **IIS Manager**
 - Select server name
 - Open the **Application Request Routing Cache** feature
 - Choose **Add Drive...**, and enter a **Drive location** to store the cache (e.g. 'C:\cache'), then **OK**
 - Choose **Cache Control Rules...**
 - Choose **Add...**, set **Apply rule** to 'Always', **Cache duration** to '1440' mins (1 day), **URL** to '*.png', then **OK**
 - Repeat the above step, setting **URL** to '*.css', '*.gif', '*.ico', '*.jpeg', and any other image file extensions used in your WebView site



The screenshot shows the 'Cache Control Rules' table in IIS Manager. It has columns for Cache, Host Name, URL, Condition, Cache Duration, and State. There are five rows of rules, all with 'Yes' in the Cache column, 'Always' in the Condition column, and '1440' in the Cache Duration column. The URLs are *.png, *.css, *.gif, *.jpeg, and *.ico, all with 'Enabled' states.

Cache	Host Name	URL	Condition	Cache Duration	State
Yes		*.png	Always	1440	Enabled
Yes		*.css	Always	1440	Enabled
Yes		*.gif	Always	1440	Enabled
Yes		*.jpeg	Always	1440	Enabled
Yes		*.ico	Always	1440	Enabled

- Click **Back to ARR Cache**.

IIS Buffer Size

WebView typically streams the page content, processing each object request which may take several seconds to complete. You may find IIS buffers the content, only sending once the whole page is available, which can make the page appear slow to load. To change this default behaviour to match WebView's native streaming of page content, change the response buffer threshold in IIS.

-  To change the response buffer threshold, follow these steps:
 - Launch **IIS Manager**
 - Select server name
 - Open the **Application Request Routing Cache** feature
 - Choose action **Server Proxy Settings...**, and in the Buffer Settings section set **Response buffer threshold** to '2KB'
 - Click **Apply**, then **Back to ARR Cache**

ObServer as a Windows Service

ObServer, including WebView, typically runs as an application running when a user starts it. In a server environment, it may not be practicable for a user to log on, and with IIS starting automatically at boot, so should WebView.

To enable ObServer as a service, refer to the *ObServer as a Service Application Note*.

Document Versions

Version	Issue Date	Details
1.0.0	10/11/2021	Document released
1.0.1	03/04/2023	Added IIS Buffer Size section

Next Steps...

If you require help, contact support on 01273 694422 or visit www.northbt.com/support



North Building Technologies Ltd
+44 (0) 1273 694422
support@northbt.com
www.northbt.com

This document is subject to change without notice and does not represent any commitment by North Building Technologies Ltd.

ObSys and Commander are trademarks of North Building Technologies Ltd. All other trademarks are property of their respective owners.

© Copyright 2023 North Building Technologies Limited.

Author: JF
Checked by:

Document issued 03/04/2023.