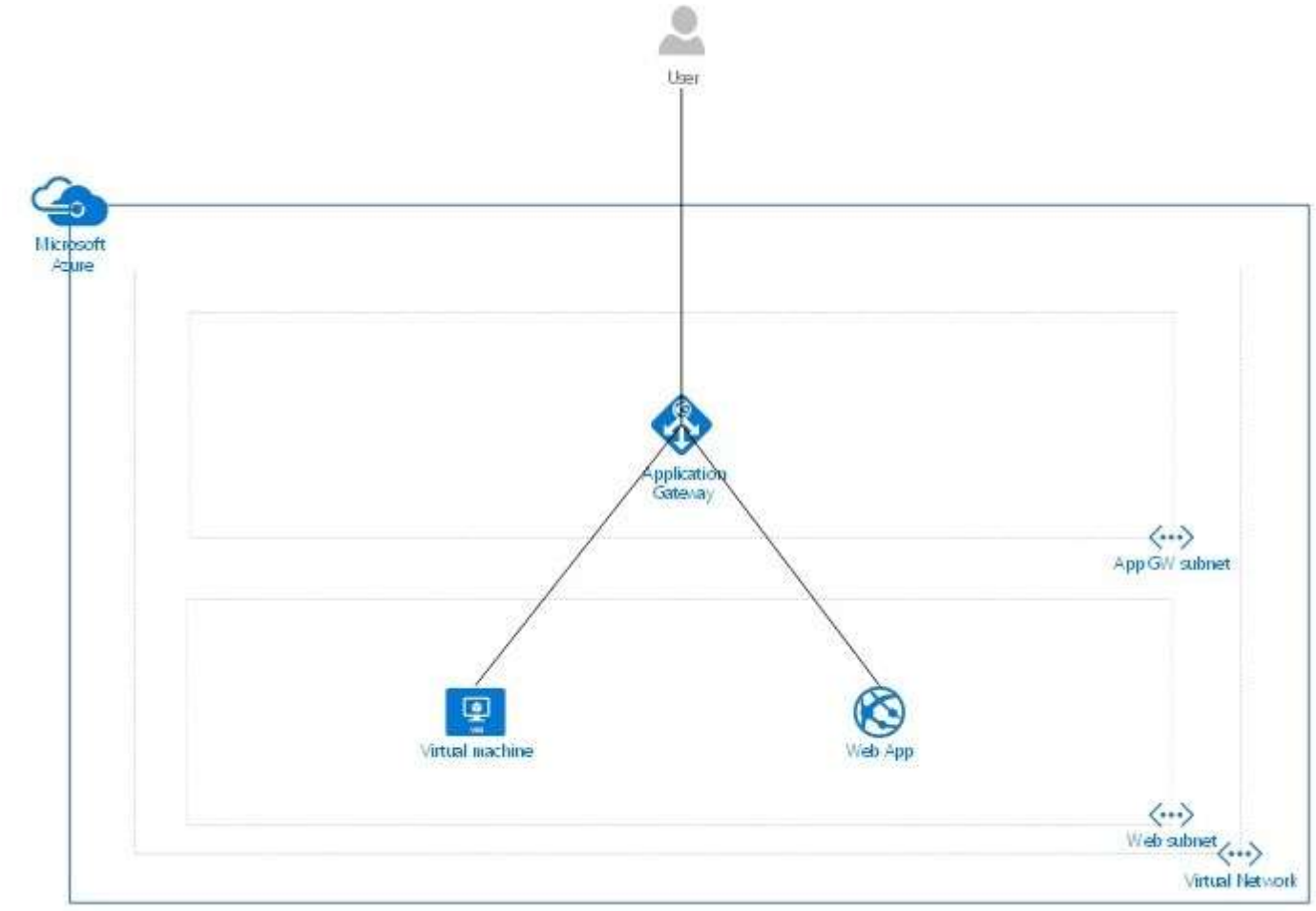
Securing your Web front-end with Azure Application Gateway

I have just completed a project with a customer who were using Azure Application Gateway to secure their web front-end and thought it would be good to post some findings.

This is part one in a two part post looking at how to secure a web front-end using Azure Application Gateway with the WAF component enabled. In this post I will explain the process for configuring the Application Gateway once deployed. You can deploy the Application Gateway from an ARM Template, Azure PowerShell or the portal. To be able to enable the WAF component you must use a Medium or Large instance size for the Application Gateway.

Using Application Gateway allows you to remove the need for your web front-end to have a public endpoint assigned to it, for instance if it is a Virtual Machine then you no longer need a Public IP address assigned to it. You can deploy Application Gateway in front of Virtual Machines (IaaS) or Web Apps (PaaS).

An overview of how this will look is shown below. The Application Gateway requires its own subnet which no other resources can be deployed to. The web server (Virtual Machine) can be assigned to a separate subnet, if using a web app no subnet is required.



The benefits we will receive from using Application Gateway are:

- Remove the need for a public endpoint from our web server.
- End-to-end SSL encryption.
- Automatic HTTPS to HTTPS redirection.
- Multi-site hosting, though in this example we will configure a single site.
- In-built WAF solution utilising OWASP core rule sets 3.0 or 2.2.9.

To follow along you will require the Azure PowerShell module version of 3.6 or later. You can install or upgrade following this link

Before starting you need to make sure that an Application Gateway with an instance size of Medium or Large has been deployed with the WAF component enabled and that the web server or web app has been deployed and configured.

Now open PowerShell ISE and login to your Azure account using the below command.

```
Login-AzureRmAccount
```

Now we need to set our variables to work with. These variables are your Application Gateway name, the resource group where you Application Gateway is deployed, your Backend Pool name and IP, your HTTP and HTTPS Listener names, your host name (website name), the HTTP and HTTPS rule names, your front end (Private) and back end (Public) SSL Names along with your Private certificate password.

NOTE: The Private certificate needs to be in PFX format and your Public certificate in CER format.

Change these to suit your environment and copy both your pfx and cer certificate files to C:\Temp\Certs on your computer.

```
# Application Gateway name.
[string]$ProdAppGw = "PRD-APPGW-WAF"
# The resource group where the Application Gateway is deployed.
[string]$resourceGroup = "PRD-RG"
# The name of your Backend Pool.
[string]$BEPoolName = "BackEndPool"
# The IP address of your web server or URL of web app.
[string]$BEPoolIP = "10.0.1.10"
# The name of the HTTP Listener.
[string]$HttpListener = "HTTPListener"
# The name of the HTTPS Listener.
[string]$HttpsListener = "HTTPSListener"
# Your website hostname/URL.
[string]$HostName = "website.com.au"
# The HTTP Rule name.
[string]$HTTPRuleName = "HTTPRule"
# The HTTPS Rule name.
[string]$HTTPSRuleName = "HTTPSRule"
# SSL certificate name for your front-end (Private cert pfx).
[string]$FrontEndSSLName = "Private_SSL"
# SSL certificate name for your back-end (Public cert cer).
```

```
[string]$BackEndSSLName = "Public_SSL"
# Password for front-end SSL (Private cert pfx).
[string]$sslPassword = "&lt;Enter your Private Certificate pfx password here.&gt;"
```

Our first step is to configure the Front and Back end HTTPS settings on the Application Gateway.

Save the Application Gateway as a variable.

```
$AppGw = Get-AzureRmApplicationGateway -Name $ProdAppGw `
         -ResourceGroupName $resourceGroup
```

Add the Front-end (Private) SSL certificate. If you have any issues with this step you can upload the certificate from within the Azure Portal by creating a new Listener.

```
Add-AzureRmApplicationGatewaySslCertificate -ApplicationGateway $AppGw `
-Name $FrontEndSSLName -CertificateFile "C:\Temp\Certs\PrivateCert.pfx" `
-Password $sslPassword
```

Save the certificate as a variable.

```
$AGFECert = Get-AzureRmApplicationGatewaySslCertificate -ApplicationGateway $AppGW `
            -Name $FrontEndSSLName
```

Configure the front-end port for SSL.

```
Add-AzureRmApplicationGatewayFrontendPort -ApplicationGateway $AppGw `
-Name "appGatewayFrontendPort443" `
-Port 443
```

Add the back-end (Public) SSL certificate.

```
Add-AzureRmApplicationGatewayAuthenticationCertificate -ApplicationGateway $AppGW `
-Name $BackEndSSLName `
-CertificateFile "C:\Temp\Certs\PublicCert.cer"
```

Save the back-end (Public) SSL as a variable.

```
$AGBECert = Get-AzureRmApplicationGatewayAuthenticationCertificate -ApplicationGateway $AppGW `
            -Name $BackEndSSLName
```

Configure back-end HTTPS settings.

```
Add-AzureRmApplicationGatewayBackendHttpSettings -ApplicationGateway $AppGW `
```

```
-Name "appGatewayBackendHttpsSettings" `
-Port 443 `
-Protocol Https `
-CookieBasedAffinity Enabled `
-AuthenticationCertificates $AGBECert
```

Apply the settings to the Application Gateway.

```
Set-AzureRmApplicationGateway -ApplicationGateway $AppGw
```

The next stage is to configure the back-end pool to connect to your Virtual Machine or Web App. This example is using the IP address of the NIC attached to the web server VM. If using a web app as your front-end you can configure it to accept traffic only from the Application Gateway by setting an IP restriction on the web app to the Application Gateway IP address.

Save the Application Gateway as a variable.

```
$AppGw = Get-AzureRmApplicationGateway -Name $ProdAppGw `
         -ResourceGroupName $resourceGroup
```

Add the Backend Pool Virtual Machine or Web App. This can be a URL or an IP address.

```
Add-AzureRmApplicationGatewayBackendAddressPool -ApplicationGateway $AppGw
`
-Name $BEPoolName `
-BackendIPAddresses $BEPoolIP
```

Apply the settings to the Application Gateway.

```
Set-AzureRmApplicationGateway -ApplicationGateway $AppGw
```

The next steps are to configure the HTTP and HTTPS Listeners.

Save the Application Gateway as a variable.

```
$AppGw = Get-AzureRmApplicationGateway -Name $ProdAppGw `
         -ResourceGroupName $resourceGroup
```

Save the front-end port as a variable – port 80.

```
$AGFEPort = Get-AzureRmApplicationGatewayFrontendPort -ApplicationGateway
$AppGw `
            -Name "appGatewayFrontendPort"
```

Save the front-end IP configuration as a variable.

```
$AGFEIPConfig = Get-AzureRmApplicationGatewayFrontendIPConfig -
ApplicationGateway $AppGw `
                -Name "appGatewayFrontendIP"
```

Add the HTTP Listener for your website.

```
Add-AzureRmApplicationGatewayHttpListener -ApplicationGateway $AppGW `
```

```
-Name $HttpListener `
-Protocol Http `
-FrontendIPConfiguration $AGFEIPConfig `
-FrontendPort $AGFEPort `
-HostName $HostName

```

Save the HTTP Listener for your website as a variable.

```
$AGListener = Get-AzureRmApplicationGatewayHttpListener -ApplicationGateway
$AppGW `
              -Name $HTTPListener
```

Save the front-end SSL port as a variable – port 443.

```
$AGFESSLPort = Get-AzureRmApplicationGatewayFrontendPort -
ApplicationGateway $AppGw `
               -Name "appGatewayFrontendPort443"
```

Add the HTTPS Listener for your website.

```

Add-AzureRmApplicationGatewayHttpListener -ApplicationGateway $AppGW `
-Name $HTTPSListener `
-Protocol Https `
-FrontendIPConfiguration $AGFEIPConfig `
-FrontendPort $AGFESSLPort `
-HostName $HostName `
-RequireServerNameIndication true `
-SslCertificate $AGFECert

```

Apply the settings to the Application Gateway.

```
Set-AzureRmApplicationGateway -ApplicationGateway $AppGw
```

The final part of the configuration is to configure the HTTP and HTTPS rules and the HTTP to HTTPS redirection.

First configure the HTTPS rule.

Save the Application Gateway as a variable.

```
$AppGw = Get-AzureRmApplicationGateway -Name $ProdAppGw `
         -ResourceGroupName $resourceGroup
```

Save the Backend Pool as a variable.

```
$BEP = Get-AzureRmApplicationGatewayBackendAddressPool -ApplicationGateway
$AppGW `
       -Name $BEPoolName
```

Save the HTTPS Listener as a variable.

```
$AGSSLListener = Get-AzureRmApplicationGatewayHttpListener -
ApplicationGateway $AppGW `
                 -Name $HttpsListener
```

Save the back-end HTTPS settings as a variable.

```
$AGHTTPS = Get-AzureRmApplicationGatewayBackendHttpSettings -
ApplicationGateway $AppGW `
           -Name "appGatewayBackendHttpsSettings"
```

Add the HTTPS rule.

```
Add-AzureRmApplicationGatewayRequestRoutingRule -ApplicationGateway $AppGw
`
-Name $HTTPSRuleName `
-RuleType Basic `
-BackendHttpSettings $AGHTTPS `
-HttpListener $AGSSLListener `
-BackendAddressPool $BEP
```

Apply the settings to the Application Gateway.

```
Set-AzureRmApplicationGateway -ApplicationGateway $AppGw
```

Now configure the HTTP to HTTPS redirection and the HTTP rule with the redirection applied.

Save the Application Gateway as a variable.

```
$AppGw = Get-AzureRmApplicationGateway -Name $ProdAppGw `
         -ResourceGroupName $resourceGroup
```

Save the HTTPS Listener as a variable.

```
$AGSSLListener = Get-AzureRmApplicationGatewayHttpListener -
ApplicationGateway $AppGW `
                 -Name $HttpsListener
```

Add the HTTP to HTTPS redirection.

```
Add-AzureRmApplicationGatewayRedirectConfiguration -Name ProdHttpToHttps `
-RedirectType Permanent `
-TargetListener $AGSSLListener `
-IncludePath $true `
-IncludeQueryString $true `
-ApplicationGateway $AppGw

```

Apply the settings to the Application Gateway.

```
Set-AzureRmApplicationGateway -ApplicationGateway $AppGw
```

Save the Application Gateway as a variable.

```
$AppGw = Get-AzureRmApplicationGateway -Name $ProdAppGw `
         -ResourceGroupName $resourceGroup
```

Save the redirect as a variable.

```
$Redirect = Get-AzureRmApplicationGatewayRedirectConfiguration -Name
ProdHttpToHttps `
            -ApplicationGateway $AppGw
```

Save the HTTP Listener as a variable.

```
$AGListener = Get-AzureRmApplicationGatewayHttpListener -ApplicationGateway
$AppGW `
              -Name $HttpListener
```

Add the HTTP rule with redirection to HTTPS.

```
Add-AzureRmApplicationGatewayRequestRoutingRule -ApplicationGateway $AppGw
`
-Name $HTTPRuleName `
-RuleType Basic `
-HttpListener $AGListener `
-RedirectConfiguration $Redirect
```

Apply the settings to the Application Gateway.

```
Set-AzureRmApplicationGateway -ApplicationGateway $AppGw
```

In this post we covered how to configure Azure Application Gateway to secure a web front-end whether running on Virtual Machines or Web Apps. We have configured the Gateway for end-to-end SSL encryption and automatic HTTP to HTTPS redirection removing this overhead from the web server.