Lab 8: IIS and   
AD Certification Services

Server System Management - Windows Server Labs

SERAFIM CIOBANU

Academic Year: 2023-2024

Table of Contents

[Introduction 3](#_Toc164322609)

[Learning Goals 3](#_Toc164322610)

[Knowledge 3](#_Toc164322611)

[Abilities 3](#_Toc164322612)

[Requirements 3](#_Toc164322613)

[Installing IIS on the Core Server - our first website 4](#_Toc164322614)

[Setting up remote IIS administration from your Win11 Client 5](#_Toc164322615)

[Set up a second website “intranet.serverlabs.be” 7](#_Toc164322616)

[Setting up a (local) Certificate Authority for your Active Directory domain 8](#_Toc164322617)

[Securing the website “intranet.serverlabs.be” 12](#_Toc164322618)

[Write the pester script yourself! 14](#_Toc164322619)

[Optional Assignments: 14](#_Toc164322620)

[References 15](#_Toc164322621)

## Introduction

As hacking and security breaches become more and more common, it is important to implement secure protocols and secure connections on the internet, but also on local networks and within (AD) domains [1]. This requires that both users and devices in the network can authenticate themselves, for example by using digital certificates and public/private encryption key pairs [2]. To ensure that these digital certificates and key pairs are authentic, they are typically issued and maintained by a trusted third party, a so-called Certificate Authority (CA) [3].

In this lab, you’ll setup a local Certificate Authority for your active directory domain and its devices/users. Using certificates from this CA, you’ll then set up a secure (HTTPS) website, hosted on an IIS webserver.

## Learning Goals

# Knowledge

In this lab, you’ll learn about:

* Microsoft IIS and creating/configuring (secure) static websites
* Active Directory Certificate Services (AD CS)
* Public key infrastructures (PKI) based on AD CS

# Abilities

In this lab, you’ll do the following:

* Install and basic administration of a Microsoft IIS webserver
* Host a static website on IIS
* Install and basic administration of a certificate authority based on AD CS
* Issue certificates in AD CS to authenticate devices in the AD domain.
* Apply certificates in IIS to secure the hosted static website.

## Requirements

Your three virtual machines (Windows GUI server, Windows Core server, Windows client), configured as described in the previous labs.

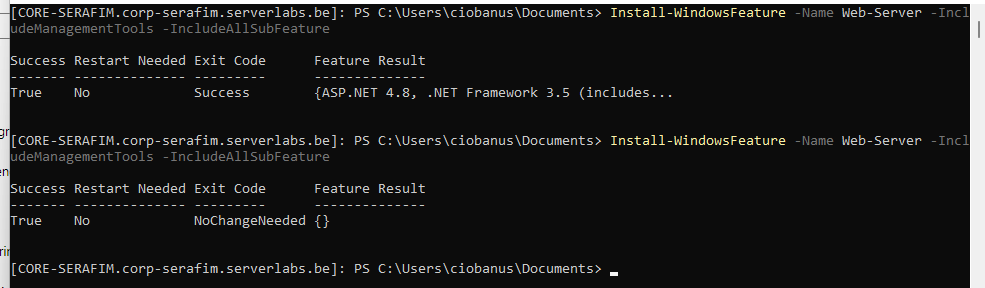
## Installing IIS on the Core Server - our first website

In this lab, we’ll use Microsoft’s IIS (Internet Information Server) webserver to host static websites in our Active Directory environment. Let’s start by installing IIS on the Core Server machine. To do this, at least the core server and Win11 client should be running.

1. Log in to the Win11 VM with your personal domain administrator account, and use Server Manager, Windows Admin Center or Remote PowerShell to install the “Web Server” role/feature (including its management tools) on the Core Server. Also install all the additional “web server security” roles.

Server Manager > All servers > Right-click CORE-SERAFIM > Windows Powershell

Install-WindowsFeature -Name Web-Server -IncludeManagementTools -IncludeAllSubFeature



A screenshot of a computer

Description automatically generated

Seems to have done it correctly

1. IIS comes with its own default website, which is active immediately. Browse to http://core-<name>.corp-<name>.serverlabs.be/ (replace <name> by your first name) to see this page. What protocol is used to host this website? What implications does this have on security? Can we be sure that the website is provided to us by the right server (and not by a rogue server set up by a hacker)?

A computer screen shot of a blue square

Description automatically generated

It uses http protocol.

That means you can see the data being transferred in clear text.

A computer screen shot of a blue square with a blue square with a blue square with a black square with a white square with a black square with a white square with a black square with a white

Description automatically generated

I decided to open the network tab and there I can see the IP address where it comes from, which is my core server, hence I believe it.

What are these properties about? Meaning information that I can gather from the website or something else?

Okay I found that you need to go into powershell and start looking at

Get-Website to show the websites that are currently set up

A blue screen with white text

Description automatically generated

* + Name: Default Web Site
  + State: Started
  + Physical path: $SystemDrive%\inetpub\wwroot
  + Bindings: http \*:80

What is the meaning of each of these properties?

## Setting up remote IIS administration from your Win11 Client

As is the case for most Microsoft services on Windows Server, the web server and its hosted websites can be configured by using PowerShell cmdlets. In practice however, a GUI interface can be more user-friendly, especially for beginning administrators. Therefore, we will now set up remote IIS management [4] from the Win11 VM:

1. To install the IIS management console through the GUI, open Settings > System > Optional features, and then click More Windows features at the bottom. Scroll down and expand Internet Information Services > Web Management Tools, check the “IIS Management Console” box, and click OK.

A screenshot of a computer

Description automatically generated

Run the newly installed “IIS Manager console”. Can you already manage the IIS server on your core server VM? Close the IIS Manager for now.

No, you cannot, probably because we don’t have services or have not added the server yet.

A screenshot of a computer

Description automatically generated

To allow administration of a **remote** IIS server, the Windows 11 version of IIS manager needs an extra plugin, which you can download from <https://www.microsoft.com/en-us/download/details.aspx?id=41177> . Download the version of this plugin that is suitable for your Win11 VM, and install it.

A screenshot of a computer

Description automatically generated

Installed IIS Manager for Remote Administration 1.2

Now start the “IIS Manager console” again and try to connect to the core server (File -> connect to server). Does this work?

A screenshot of a computer

Description automatically generated  
Maybe its better to initially do all the set up and not do step by step.

Connecting the IIS manager to the server probably still fails, because of two configurations that are missing:

* A special service for remote IIS management should be running on the core server. Check whether the “management service” for IIS (Web-Mgmt-Service feature) is installed, and install it if necessary.

Get-WindowsFeature -Name Web-Mgmt-Service

Also run the following powershell commands on the core server:

• Set-ItemProperty HKLM:\SOFTWARE\Microsoft\WebManagement\Server -Name EnableRemoteManagement -Value 1 -Type DWORD -Force

• $certHash = (Get-ChildItem "Cert:\LocalMachine\My" | ? {$\_.Subject -like "\*WMSvc\*"} | Select -ExpandProperty Thumbprint).ToLower()

• $byteArray = [byte[]]::new($certHash.Length / 2)

• for ($i = 0; $i -lt $certHash.Length; $i += 2) { $byteArray[$i / 2] = [convert]::ToByte($certHash.Substring($i, 2), 16)}

• Set-ItemProperty -Path HKLM:\SOFTWARE\Microsoft\WebManagement\Server -Name SslCertificateHash -Value $byteArray -Type Binary

• Set-Service WMSVC -StartupType Automatic -PassThru | Start-Service -Verbose -PassThru

A screenshot of a computer program

Description automatically generated

* A firewall rule should be added to the core server that allows inbound requests to the WMSVC service. Define this firewall rule using powershell or one of the other applicable (remote) management tools.

In my case it worked without the need for firewall rules.

Then you need to provide user credentials ([ciobanus@corp-serafim.serverlabs.be](mailto:ciobanus@corp-serafim.serverlabs.be) for example), password, and name for connection and it works.

Now the IIS management console should be able to connect to the core server. Paste a screenshot here of the IIS management console that is successfully connected to the core server.

A screenshot of a computer

Description automatically generated

Also search where you can find the binding options of the Default website that is hosted on the server (cf. question 2)

## Setting up a second website “intranet.serverlabs.be”

1. Use the IIS management console to create a second website that is hosted on the core server, but that will be shown when users surf to “http://intranet.serverlabs.be”. Use the “bindings” settings of this website to associate this domain name to the site. Select (and create accordingly) the path c:\InetPub\intranet as the physical location for the site’s files.  
   As website content, you can for example use the files that are provided on Leho.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. For your machines to be able to access the intranet site, they should be able to resolve the FQDN of the site to the IP address of the core server. Next week, we’ll explore how this can be done using the domain’s DNS server, but for now it should be ok to do this name resolution with the “hosts” file. On your Win11 VM, modify the hosts file in such a way that <http://intranet.serverlabs.be> refers to the correct IP address.  
   Paste a screenshot below of the website as seen when you browse from your Win11 VM.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Had to create everything manually on the core server, and transfer the contents manually.

A computer screen with white text

Description automatically generated

## Setting up a (local) Certificate Authority for your Active Directory domain

To secure and authenticate the identity of the web server, https-based websites can use a certificate that is issued by a trusted third-party, called the Certificate Authority or CA. For internet sites, a number of commercial CAs exist where you can buy a certificate for your site (they check you actually own the domain in the process). For our local (intranet) sites however, we will use the CA that is built into active directory, and where the Domain Controller will act as the trusted third-party for the website and the web client.

1. On your domain controller (the GUI machine), set up a CA. To do this in the Server Manager, go to the Roles and Features wizard and install the “Active Directory Certificate Services” role. Click Next a few times until you can also add an extra role service: “Certification Authority Web Enrollment”. This will install a website to the server, where the certificate can be requested (see later).

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. After installation don’t forget to configure the server for Certification Authority and Web Enrollment:

* Select Enterprise CA
* Select Root CA
* Create a new private key and on the next page select the defaults.
* For the next pages, also select the defaults

A screenshot of a computer

Description automatically generated

1. Now start the CA Manager tool and place a screenshot of the manager and the CA root certificate (certificate #0) properties here. Make sure the following properties of the certificate are clearly visible:

certmgr

* The name of the certificate (which includes your name)
* Hash Algorithm
* Issuer
* Valid from & Valid to (it is fine if the hours are not shown but we want to see the dates!)
* Public Key

A screenshot of a computer

Description automatically generated

1. To make our certificates more flexible, we’ll configure the Certificate Authority policy so it accepts Subject Alternative Names (SAN). This will allow the CA to issue a single certificate that is simultaneously valid for multiple domain names that are hosted on the same server. Use the “certutil” command (in the terminal) to configure this setting. Make sure to read the output of this command, and restart some service(s) if required!

certutil -setreg policy\EditFlags +EDITF\_ADDOLDKEYUSAGE

A computer screen shot of a blue screen

Description automatically generatednet stop certsvc

net start certsvc

A screenshot of a computer program

Description automatically generated

## Securing the website “intranet.serverlabs.be”

1. Now our CA is operational, we can apply for a TLS certificate for the “intranet.serverlabs.be” website. First, we’ll create a certificate request. This certificate request needs to be created on the core server itself, which means we’ll work in a remote terminal using the command line and certreq.exe:
   * Create a file “**cert.in”, containing the following information:**

**[Version]**

**Signature = "Windows core"**

**[NewRequest]**

**Subject = "C=BE,S=West-vlaanderen,L=brugge,O=Howest,OU=IT,CN=intranet2.serverlabs.be"**

**Exportable = TRUE**

**KeyLength = 2048**

**KeySpec = 1**

**KeyUsage = 0xa0**

**MachineKeySet = True**

**ProviderName = "Microsoft RSA SChannel Cryptographic Provider"**

**ProviderType = 12**

**Silent = True**

**SMIME = False**

* + certreq.exe -new "cert.in" intranet2.req

A blue screen with white text

Description automatically generated

1. Based on the certificate request, we’ll create the site’s certificate. Now copy the certificate request to your Win11 VM and browse to http://gui-<firstname>/certsrv where gui-<firstname> is the hostname of your domain controller (your GUI machine). On that website, click “request a certificate” and then click “advanced certificate request”. Submit the certificate request that you generated in the previous question (copy&paste the contents) and select the “Web Server” certificate template. You will need to add the san into the attribute field as shown below:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A computer code with black and blue lines

Description automatically generated

1. The certificate for your website will now be generated by the CA. Download this certificate to the Win11 machine, and copy it to the core server. Then import the certificate in the Webhosting certificate store (do this using a PowerShell cmdlet)

Don’t ask me how I copied that file to the server because I somehow managed to do it without even understanding that it succeeded.

I accessed the old share folders that we had for one of the labs, found Donald, and put the file in there. That is why on screenshots you will be able to see Donald.

A screenshot of a computer program

Description automatically generated

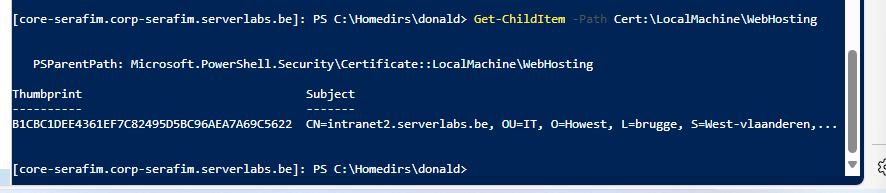
A computer screen shot of a blue screen

Description automatically generated

Import-Certificate -FilePath “.\certnew.cer” -CertStoreLocation Cert:\LocalMachine\WebHosting

1. Check what stores are available for certificates and what webhost certificates are already stored

Get-ChildItem -Path Cert:\LocalMachine\WebHosting



1. Now the certificate has been installed to the server, we can finally use it to secure our website. Go back to the IIS manager, and change the binding of the intranet site to the https protocol on port 443, and associate the https binding with the newly generated certificate.

A screenshot of a computer

Description automatically generated

1. On the Win11 VM, browse to <https://intranet.serverlabs.be> to test the site and place a screenshot here:

A computer screen with a black screen

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

It says not secure, but I believe it is still an issue either because of the browser, or because of some other places that the certificate needs to be put into.

## Write the pester script yourself!

Try to write your own pester script. You are free to add whatever test you want but make sure you test the following things:

* Is IIS installed
* Is IIS hosting a simple webserver

Put a screenshot of your tests here and post the source code of your tests as well.

My implementation looks like this:

Describe 'IIS Web Server Test' {

It 'IIS should be installed' {

Get-WindowsFeature -Name Web-Server | Where-Object { $\_.Installed } | Should Not BeNullOrEmpty

}

It 'IIS should be hosting a simple web server' {

Get-Website -Name 'Default Web Site' | Should Not BeNullOrEmpty | Should Match $True

}

}

A computer screen with text on it

Description automatically generated

## Optional Assignments:

Visit your website on your windows 11. What do you notice? Why do you think this is and are you able to fix this?

Visit <http://intranet.serverlabs.be>. (so no https) What do you notice, could we configure this for a more user friendly experience?

Remember the “execution policies”? There was this thing called “signed”. Now that we have a certificate, is it possible to sign PowerShell scripts with this certificate? If so how?

## References

|  |  |
| --- | --- |
| [1] | K. Schoenmaekers, *Cyber Security Fundamentals,* Howest. |
| [2] | Wikipedia, „Public Key Certificates,” [Online]. Available: https://en.wikipedia.org/wiki/Public\_key\_certificate. [Geopend 25 04 2022]. |
| [3] | Wikipedia, „Certificate Authority,” [Online]. Available: https://en.wikipedia.org/wiki/Certificate\_authority. [Geopend 25 04 2022]. |
| [4] | S. Kumar, „Install and enable IIS Manager for Remote Administration,” [Online]. Available: https://4sysops.com/archives/install-and-enable-iis-manager-for-remote-administration/. [Geopend 16 04 2024]. |