Journal of Mobile, Embedded and Distributed Systems, vol. VII, no. 1, 2015
ISSN 2067 – 4074

Securing SharePoint Server with Windows Azure Multi-Factor Authentication

Petru-Radu NARITA
Department of Economic Informatics and Cybernetics
The Bucharest University of Economic Studies
ROMANIA
narita_radu@yahoo.com

Abstract: In today's world, people are used to be able to connect anywhere at any time. Users can access their applications whenever they need, with any device. Stealing, phishing or key logging a password becomes easy doable for professional hackers. Many customers already use multi-factor authentication systems to protect their data and applications. However, carrying around tokens or installing software or certificates, can become a headache for both IT and users. This article talks about Windows Azure Multi-Factor Authentication – a Microsoft solution for adding a second factor of authentication using something that users nowadays already have at hand, their phone – which I checked in a SharePoint Server 2013 on premise test environment.

Key-Words: Multi, Factor, Authentication, SharePoint, 2013, Premise, Phone

1. Introduction

Multi-factor authentication is an approach of authentication, which requires two or more of the three authentication factors:

- a knowledge factor (something only the user knows, typically a password)
- a possession factor (something only the user has – a trusted device that is not easily duplicated, like a phone)
- an inherence factor (something only the user is – biometrics).

After presentation, for authentication to occur each factor must be validated by the other party.

The security of multi-factor authentication lies in its layered approach. [1] Compromising multiple authentication factors presents a significant challenge for attackers. Even if an attacker manages to learn the user's password, it is useless without also having possession of the trusted device. Conversely, if the user happens to lose the device, the finder of that device will not be able to use it unless he or she also knows the user's password.

The most common multi-factor methods include hardware tokens like RSA SecurID, certificates, smartcards, and increasingly phone-based authentication methods, which leverage the user's telephone as the trusted device for the second factor of authentication.

Multi-factor authentication is rapidly becoming a key component of how businesses ensure trust in a multi-device, mobile, cloud world. Passwords are easily compromised and the consumerization of IT has only increased the scope of vulnerability. Phone-based authentication is quickly becoming the de facto standard for multi-factor authentication as it provides enhanced security for businesses and consumers in a convenient form factor that the user already has: their phone.

Today many customers use on-premises multi-factor authentication systems to protect critical data in their file servers and their critical business apps. As these workloads move to the cloud, they need an effective and easy-to-use solution for protecting that data in their cloud-based applications.

Multi-factor authentication is no longer optional for many customers; many are required by various governing agencies to strongly authenticate access to sensitive data and applications. [2] Among others, multi-factor authentication is addressed in:

- NIST 800-63 Electronic Authentication Guidelines for Level 3 Assurance
- HIPAA Requirements Relative to Electronic Protected Health Information (EPHI)
- Payment Card Industry Data Security

Standards (PCI DSS)
- Criminal Justice Information System (CJIS) Security Policy
- Authentication in an Internet Banking Environment Guidance (FFIEC)

2. Windows Azure Multi-Factor Authentication

Windows Azure Multi-Factor Authentication is a Windows Azure service that helps safeguard access to data and applications by strengthening traditional sign-in approaches. The service supports both cloud applications that use Windows Azure Active Directory as well as on-premises applications using the Multi-Factor Authentication Server.
According to [3] there are various versions of multi-factor authentication available and resources that can be secured:
- Multi-Factor Authentication for Office 365 – allows you to secure Office 365 resources for users licensed for Office 365.
- Multi-Factor Authentication for Azure Administrators – allows you to secure Azure resources for administrators.
- Azure Multi-Factor Authentication – allows you to secure all Microsoft Online Services, multiple SaaS app resources, resources that span on-premises and cloud including VPN and LOB apps.

Multi-Factor Authentication offers a number of authentication options, allowing users to choose the one that works best for them. Support for multiple methods ensures additional authentication is always available. According to [1] there are following options:
- Multi-Factor Authentication apps are available for Windows Phone, Android, and IOS devices. Users can download the free app from the device store and activate it by using a code that they get during setup. When the user signs in, a notification is pushed to the app on their mobile device. The user taps to approve or deny the authentication request. Cellular or Wi-Fi access is required for installing and setting up the app. After the app is installed, it can operate in the following modes to provide the additional security that a multi-factor authentication service can provide:
  - Notification. In this mode, the Multi-Factor Authentication app prevents unauthorized access to accounts and stops fraudulent transactions. It accomplishes this by using a push notification to the phone or registered device. The user simply views the notification, and if it is legitimate, selects Authenticate. Otherwise, the user can choose to deny, or choose to deny and report, the fraudulent notification.
  - One-Time Passcode**.** In this mode, the Multi-Factor Authentication app can be used as software token to generate an Open Authentication (OATH) passcode. The user can then enter this passcode along with the user name and password to provide the second form of authentication. This option is useful in instances of spotty phone coverage.
- Automated phone calls can be placed by the Multi-Factor Authentication service to any phone, whether landline or mobile. The user simply answers the call and presses the pound key (#) on the phone to complete the sign-in.
- Text messages can be sent by the Multi-Factor Authentication service to any mobile phone. Each text message contains a one-time passcode. The user is prompted to either reply to the text message by using the passcode or enter the passcode on the sign-in screen.

3. Securing IIS authentication – SharePoint Server 2013

For Windows Azure Multi-Factor Authentication to work with on-premises apps, the Multi-Factor Authentication Server is installed on existing on-premises hardware. It can be configured to secure on-premises applications using built-in support for RADIUS, LDAP, Windows, and IIS authentication. [4] The Multi-Factor Authentication Server can synchronize with an Active Directory or another LDAP

ISSN 2067 – 4074

directory for centralized user management and automated provisioning. A multi-factor authentication provider will be created within the Active Directory category on the Windows Azure Management Portal, which will allow you to download and activate the Azure Multi-Factor Authentication Server.

The IIS Authentication section of the Azure Multi-Factor Authentication Server allows enabling and configuring IIS authentication for integration with Microsoft IIS web applications, as SharePoint. The Azure Multi-Factor Authentication Server installs a plug-in, which can filter requests being made to the IIS web server in order to add Azure Multi-Factor Authentication. At the time of this writing, the IIS plug-in provides support for Form-Based Authentication and Integrated Windows HTTP Authentication.

While in SharePoint 2010 you could select from the user interface in Central Administration classic or claims-based authentication, in SharePoint 2013 claims-based authentication is the default and preferred method of user authentication and is required to take advantage of server-to-server authentication and app authentication. [5] In Central Administration, you can only configure claims-based authentication when you create web applications, but you can create and configure web applications with classic authentication using PowerShell:

```
New-SPWebApplication -Name <Name> -
ApplicationPool <ApplicationPool> -
AuthenticationMethod
<WindowsAuthType> -
ApplicationPoolAccount
<ApplicationPoolAccount> -Port
<Port> -URL <URL>
```

Where:

- <Name> is the name of the new web application.
- <ApplicationPool> is the name of the application pool.
- < WindowsAuthType > is "NTLM".
- <ApplicationPoolAccount> is the user account that this application pool will run as.
- <Port> is the port on which the web application will be created in IIS.
- <URL> is the public URL for the web application.

The Azure Multi-Factor Authentication Server will be installed on the SharePoint Web Front End server, which is handling the authentication. Here the IIS Authentication has to be enabled, as presented in figure 1.

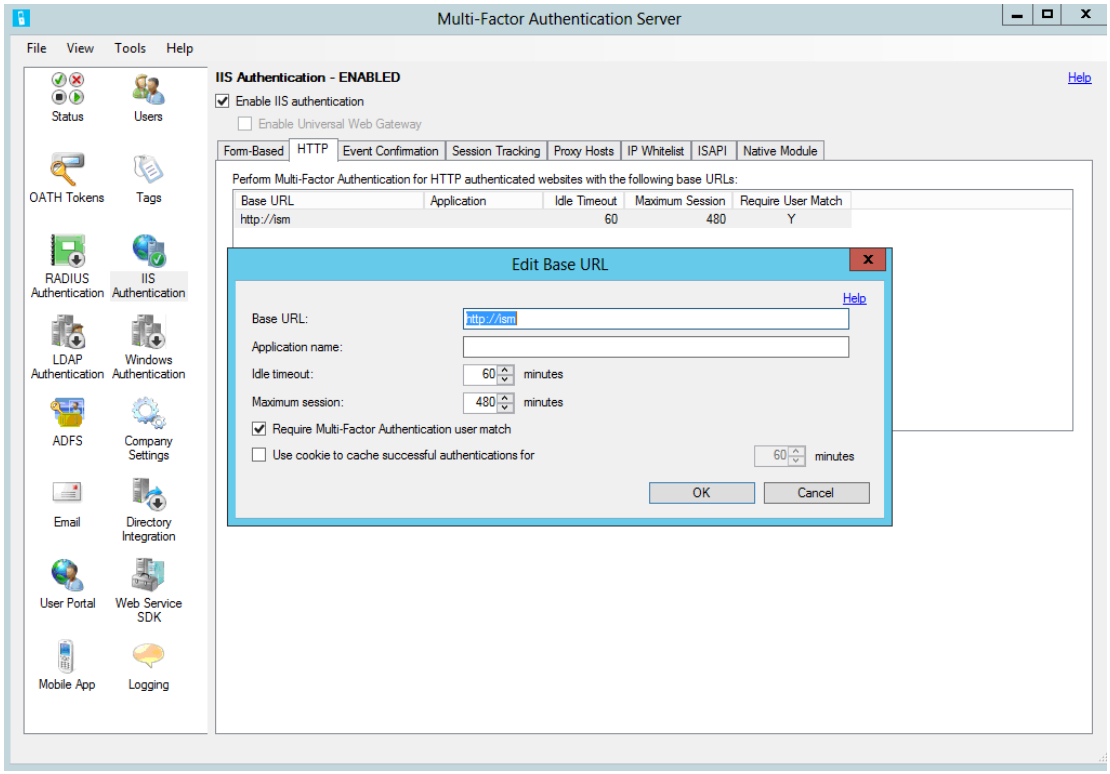

Figure 1. Enabling IIS Authentication and adding the Base URL

Once configured the HTTP authentication URLs and settings, the locations where the Azure Multi-Factor Authentication IIS plug-ins should be loaded and enabled in IIS are to be selected, as shown in Figure 2.

Afterwards the users have to be imported into the server.

The authentication will occur as follows:

1. The user logins with username and password.
2. The server sends a Windows Multi-Factor Authentication Challenge.
3. The user responds to the challenge from the device.
4. If the challenge is successful, the user is authenticated.

The standard way to establish a NTLM HTTP connection consists of 3 HTTP roundtrip requests:

- First Request - The HTTP POST or GET request is sent as an anonymous access request (it will not include the Authorization header) and includes the POST body for POST requests. If the resource requires NTLM authentication, the server will respond with a 401 (Unauthorized) HTTP response and include the WWW-Authorization header(s) listing the authentication schemes the server supports. One of these should be NTLM or Negotiate.
- Second Request - In response, the client creates an NTLM Authorization token that includes the domain and computer name and then sends another HTTP request with the token in the Authorization header. The NTLM token will usually begin with "TIRM..." and will be roughly 60 characters long. For POST requests, the second request will not contain the POST body data (the content-length header will be 0). The HTTP server will respond to this with an NTLM challenge response as a 401 (Unauthorized) HTTP response. It will again include the WWW-Authenticate header.
- Third Request - The client will send the HTTP request with an Authorization header containing a new token that also includes the password of the user encrypted. This token will also begin with "TIRM..." and will be longer than the first token in the second request. For POST requests, this will also include the POST body. The server will respond with the data

ISSN 2067 – 4074

if the token succeeds the authentication.

If we analyze the http(s) network traffic using Fiddler when signing on a SharePoint 2013 page on IIS 8.0 configured with classic Windows NTLM authentication and Windows Azure Multi-Factor Authentication, we can observe the typical NTLM flow:

1. Internet Explorer does not include any authentication information in the first request on a new connection:

```
GET http://ism/ HTTP/1.1
Accept: image/jpeg, application/x-
ms-application, image/gif,
application/xaml+xml, image/pjpeg,
application/x-ms-xbap, */*
Accept-Language: en-US
User-Agent: Mozilla/4.0
(compatible; MSIE 7.0; Windows NT
6.2; WOW64; Trident/6.0; .NET4.0E;
.NET4.0C; .NET CLR 3.5.30729; .NET
CLR 2.0.50727; .NET CLR 3.0.30729)
Accept-Encoding: gzip, deflate
Host: ism
DNT: 1
Connection: Keep-Alive
```

2. If the IIS server is not configured to support Anonymous authentication, the server returns a 401.2 status that tells the client that the client is unauthorized. Together with the error status, the server also sends a list of authentication protocols that the server supports. The response headers that IIS returns in this NTLM-only scenario resemble the following:

```
HTTP/1.1 401 Unauthorized
Server: Microsoft-IIS/8.0
SPRequestGuid: 62e38e9c-6ada-c044-
aabc-07c2a0655b60
request-id: 62e38e9c-6ada-c044-
aabc-07c2a0655b60
X-FRAME-OPTIONS: SAMEORIGIN
SPRequestDuration: 1
SPIisLatency: 0
WWW-Authenticate: NTLM
X-Powered-By: ASP.NET
MicrosoftSharePointTeamServices:
15.0.0.4420
X-Content-Type-Options: nosniff
X-MS-InvokeApp: 1; RequireReadOnly
Date: Thu, 08 May 2014 13:32:50 GMT
Content-Length: 0
Proxy-Support: Session-Based-
Authentication
```

3. When the client receives the server's notification that the server supports the NTLM protocol, the client re-sends the request. The client includes authentication information in an Authorization header:

```
GET http://ism/ HTTP/1.1
Accept: image/jpeg, application/x-
ms-application, image/gif,
application/xaml+xml, image/pjpeg,
application/x-ms-xbap, */*
Accept-Language: en-US
User-Agent: Mozilla/4.0
(compatible; MSIE 7.0; Windows NT
6.2; WOW64; Trident/6.0; .NET4.0E;
.NET4.0C; .NET CLR 3.5.30729; .NET
CLR 2.0.50727; .NET CLR 3.0.30729)
Accept-Encoding: gzip, deflate
Host: ism
Authorization: NTLM
TlRMTVNTUAABAAAAB4IIogAAAAAAAAAAAAA
AAAAAAAAGAvAjAAAADw==
Connection: Keep-Alive
DNT: 1
```

4. As part of the NTLM handshake, the server acknowledges that the client has sent authentication information. However, the server needs the client to send more information. Therefore, the server returns another 401 response that resembles the following:

```
HTTP/1.1 401 Unauthorized
Server: Microsoft-IIS/8.0
WWW-Authenticate:                NTLM
TlRMTVNTUAACAAAAFAAU…
SPRequestGuid:   67e38e9c-7a99-c044-
aabc-0a1b6a62e4b9
request-id:      67e38e9c-7a99-c044-
aabc-0a1b6a62e4b9
X-FRAME-OPTIONS: SAMEORIGIN
SPRequestDuration: 1
SPIisLatency: 0
X-Powered-By: ASP.NET
MicrosoftSharePointTeamServices:
15.0.0.4420
X-Content-Type-Options: nosniff
X-MS-InvokeApp: 1; RequireReadOnly
Date: Thu, 08 May 2014 13:33:09 GMT
Content-Length: 0
Proxy-Support:        Session-Based-
Authentication
```

5. The 401 status that IIS sends tells the client that the client must provide the remainder of the valid authentication information. The client receives this challenge. The client then sends one more request that resembles the following:

```
GET http://ism/ HTTP/1.1
Accept: image/jpeg, application/x-
ms-application, image/gif,
application/xaml+xml, image/pjpeg,
application/x-ms-xbap, */*
Accept-Language: en-US
User-Agent: Mozilla/4.0
(compatible; MSIE 7.0; Windows NT
6.2; WOW64; Trident/6.0; .NET4.0E;
.NET4.0C; .NET CLR 3.5.30729; .NET
CLR 2.0.50727; .NET CLR 3.0.30729)
Accept-Encoding: gzip, deflate
Authorization: NTLM
TlRMTVNTUAADAAAAGAA…
Host: ism
Connection: Keep-Alive
DNT: 1
```

6. At this point, the username and password have been verified, but the server does not send a HTTP response yet. Instead, it waits for the second layer of authentication to occur. Assuming the automated phone calls method were configured on the Azur Multi-Factor Authentication Server, the user will receive a phone call – he has to answer and press the pound key. Now that the second method of authentication occurred, the user's authentication is successful and we get the server response with the site.

If the user does not approve the request when prompted or cannot be reached for authentication, access is denied. [3] However, because the user's credentials are verified before Multi-Factor Authentication is triggered, this is an indication that the user's password has been compromised. In some cases, the user will have the option to submit a fraud alert during the authentication request. This will prevent further login attempts and sends a notification to the customer. The customer should work with the user to reset the user's password.

4. Conclusion

Passwords have long been used to authenticate users. They rely on the fact that only the user knows the password.
With the proliferation of systems and resources, it becomes clear that passwords can be compromised in a number of ways: many users write down their passwords; the same password is used for different sites and applications, so if the password is compromised in one place it will affect all sites and services; replay attacks; social engineering and phishing.
Multi-factor authentication using something that users already have, their phone, will rapidly grow and become a key component to ensure trust in a mobile, cloud world.

Acknowledgement

Parts of this paper were presented at The 7th International Conference on Security for Information Technology and Communications (SECITC 2014), Bucharest, Romania, 12-13 June 2014.

References

[1] Azure Multi-Factor Authentication, http://technet.microsoft.com/en-us/library/dn249471.aspx
[2] Leverage Windows Azure Multi-Factor Authentication Server for Windows Azure AD single sign-on with AD FS, http://download.microsoft.com/download/F/C/A/FCA7C6E3-7153-4FB1-9825-0B1BB26F14E0/Leverage-Multi-Factor-Authentication-Server-on-your-premises.docx
[3] Adding Multi-Factor Authentication to Azure Active Directory, http://technet.microsoft.com/en-us/library/dn249466.aspx
[4] Enabling Multi-Factor Authentication for On-Premises Applications and Windows Server, http://msdn.microsoft.com/en-us/library/azure/dn249467.aspx
[5] Troy Lanphier, *Exam Ref 70-331:Core Solutions of Microsoft SharePoint Server 2013*, 2013