ONE IDENTITY

Cloud Access Manager 8.1.4

Security and Best Practices Guide

Legend

WARNING: A WARNING icon indicates a potential for property damage, personal injury, or death.

CAUTION: A CAUTION icon indicates potential damage to hardware or loss of data if instructions are not followed.

IMPORTANT, NOTE, TIP, MOBILE, or VIDEO: An information icon indicates supporting information.

Cloud Access Manager Security and Best Practices Guide
Updated - November 2018
Version - 8.1.4

Contents

Optimizing Cloud Access Manager for a production environment

Before deploying Cloud Access Manager, we recommend you perform the following steps to ensure Cloud Access Manager is optimized to handle the expected workload. Using this recommended configuration, a Proxy host can typically support up to 7,000 users and a Security Token Service (STS) host can typically support up to 15,000 users. You can add further Proxy and STS hosts to support more users and to provide high availability. For a production environment, we recommend that you deploy an additional proxy host and STS host to provide high availability and protect against a single host failure. For example a company with 20,000 users would typically deploy 4 Proxy hosts (20,000/7,000 +1) and 3 STS hosts (20,000/15,000 +1).

NOTE: If you are not proxying any applications, including the Cloud Access Manager portal, the number of Proxy hosts should match the number of STS hosts.

To confirm you have deployed a sufficient number of hosts to support your users we recommend you use a phased rollout approach. Start by rolling out Cloud Access Manager to a small subset of users and monitor the CPU and memory usage of the Cloud Access Manager hosts to ensure sufficient spare capacity to scale out to the entire user base. For some applications this approach will not be possible, for example with federated applications such as Salesforce or Office 365 where you have to switch over all users at the same time. In this situation, we recommend that you use a phased roll out of applications rather than users. Start with smaller applications and gradually add more when you have verified that sufficient spare resources are available on the Cloud Access Manager hosts.

For further information on how to install additional Cloud Access Manager hosts, please refer to the document entitled *One Identity Cloud Access Manager How To Configure For High Availability*.

Proxy hosts

One Identity Cloud Access Manager contains a reverse proxy to provide Single Sign-On (SSO) to web applications that do not support federation, for example basic, NT LAN Manager (NTLM), header and form authentication. The reverse proxy is also used to allow secure access to internal web applications from the Internet. When you access a proxied

application, all communication between the web browser and the application goes through the proxy for the entire session, not only for the authentication.

For a production environment, we recommend that each proxy host has 9GB of physical memory and 8 processor cores. For example, two quad core processors giving a total of 8 cores spread over two processors.

A single proxy host can handle up to 12,000 concurrent connections. Modern web browsers typically use between 6 and 8 persistent HTTP connections when accessing an application. But during idle periods, such as when a user is reading, they will often reduce the number of connections to just a single connection, or even close all connections until the next user interaction. The browser can use each connection to send multiple HTTP requests to the application. The proxy will close a connection after either processing 100 HTTP requests, or after the connection has been idle for 60 seconds. The browser will establish a new connection the next time it needs to make an HTTP request. So, depending on the application you want to proxy, a single Proxy host will be able to support between 1,500 users (12,000/8) and 12,000 users. Our recommended maximum of 7,000 is an average of the two.

To support up to 12,000 concurrent connections, you must configure the proxy host to increase the number of persistent HTTP connections that it can support. This in turn requires greater memory allocation for the proxy. Please refer to Memory on page 6 and HTTP connections on page 7 for information on how to make the two changes required.

Memory

To support a larger number of persistent HTTP connections, first increase the amount of memory available to the proxy. For a production environment, we recommend that each proxy host has 9GB of physical memory, with 6GB of this memory allocated to the Java virtual machine (JVM) used by the proxy.

NOTE: These figures are intended as guidelines. Different operating systems may require more or less RAM to be allocated to them to function effectively. For instance, 8GB RAM may be sufficient for a proxy running on Windows Server Core OS with 6 GB allocated to the JVM heap.

To configure the maximum amount of memory allocated to the Java virtual machine

Perform the following steps on the proxy host.

1. Double click <Installation location>\Cloud Access Manager Proxy\bin\CloudAccessManagerProxyw.exe on each proxy host to open the proxy service configuration tool.
2. Click the **Java** tab.
3. In the **Maximum memory pool** field, enter the value 6144, then click **Apply** to set the maximum amount of memory allocated to the Java Virtual Machine heap to 6GB.
4. You must restart the proxy service for this setting to take effect. To restart the proxy service, click the **General** tab and then click **Restart**.

NOTE: Memory consumption of the proxy can exceed the amount allocated to the JVM heap. This is because Java allocates memory to other processes, such as a stack for each thread. Therefore, it is not unusual for the total memory used by the proxy to exceed the value allocated to the JVM heap by up to 10%.

Disk space

We recommend the following minimum disk space requirements are observed. For further information on installation requirements, please refer to the document entitled *One Identity Cloud Access Manager Installation Guide*.

Table 1: Disk space requirements

Hardware	Requirement	Host
Disk space	25GB	Proxy host.
Disk space	50GB	STS host.
Disk space	50GB	STS host.

NOTE: These recommended disk space values are intended as a general guideline. We suggest that you monitor disk space usage on all your servers to account for usage changes that occur, such as expanding log files (For example, from other applications such as IIS), a life time of Windows updates and system backup data.

HTTP connections

A production environment proxy host should be able to handle up to 12,000 concurrent, persistent HTTP connections.

You may need to configure the host to support this number of connections, by default Microsoft Windows Server 2008 R2 will allow approximately 8,000 connections. To allow a greater number of connections use the `netsh` command to increase the dynamic ports range, please refer to http://en.wikipedia.org/wiki/Ephemeral_port for further information.

The following example will allow approximately 12,000 persistent HTTP connections. Run this command from a command prompt as an administrator, this setting takes immediate effect and does not require a reboot.

```
netsh int ipv4 set dynamicport tcp start=40000 num=25000
```

STS hosts

For a production environment, we recommend that each Security Token Service (STS) host has 8GB of physical memory and 8 processor cores. For example, two quad core processors giving a total of 8 cores spread over two processors.

CPU and memory usage varies between the different authentication methods. Our stress testing has shown a single STS host can support between 12,000 and 18,000 users authenticating over a 30 minute period. Our recommended maximum of 15,000 is an average of the two. No special configuration is required on the STS hosts to support this number of users.

Preventing direct access to applications protected by Cloud Access Manager

When you have added an application to Cloud Access Manager, you can make sure that users only access the application using Cloud Access Manager. This may be required if you are using Cloud Access Manager to enforce strong authentication for the application, or want to use Cloud Access Manager's auditing features to monitor application usage. The procedure described below is only required for non-federated applications, that is, those not using SAML or WS-Federation.

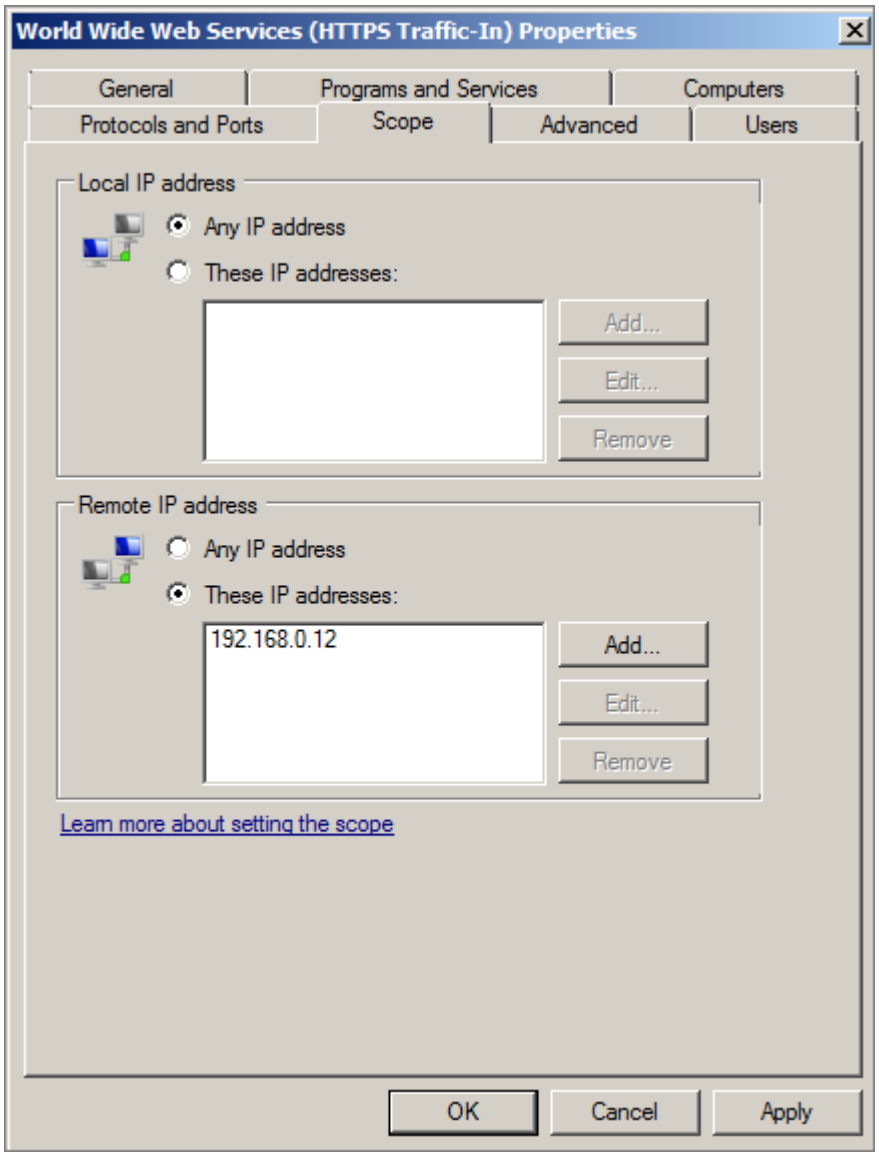
NOTE: To prevent direct access to an application, we recommend that you use the firewall on the application's host. For example, you need to update the HTTP/HTTPS firewall rules to only allow access from the internal IP addresses of the Cloud Access Manager proxy hosts.

To configure the Windows Firewall to prevent direct access to the application

Perform the following steps on the application's host.

NOTE: If the application is installed on multiple hosts, repeat these steps on each host running the application.

1. From the **Start** menu, open **Windows Firewall with Advanced Security**.
2. Ensure the firewall is turned on.
3. Locate the Inbound Firewall rule for Secure HTTP (HTTPS) requests. For applications using Internet Information Services (IIS), this is called **World Wide Web Services (HTTPS Traffic-In)**.
4. Ensure the rule is enabled.
5. Click **Properties**.
6. Select the **Scope** tab.



7. In the **Remote IP address** section, select **These IP addresses:** and add the internal IP address of each proxy host.
8. Repeat Step 3 through Step 7 for the HTTP firewall rule.
9. Using a browser on another host, verify that you cannot directly access the application running on this host.

When you have updated the firewall configuration on each host running the application, verify that users can still access the application using Cloud Access Manager.

NOTE: If the host running the application contains multiple applications and you do not wish to prevent direct access to all of them, you can configure the Web Server to restrict access to the proxy hosts on a per application basis. Please refer to http://support.microsoft.com/kb/324066 for instructions on how to configure IIS to use the **IP Address and Domain Restrictions** feature.

2

Operations

Backup

It is strongly recommended that you take a backup of your Cloud Access Manager environment at regular intervals, and immediately prior to upgrading or carrying out maintenance. Please refer to the *One Identity Cloud Access Manager Installation Guide* for full instructions to backup and restore Cloud Access Manager.

Fallback

The Cloud Access Manager fallback administration account allows you to bypass the directory-based authentication mechanism to:

- Configure Cloud Access Manager for the first time
- Investigate issues with directory-based authentication
- Perform certain activities such as setting up certificate-based authentication.

We recommend you choose a password for the fallback administration account which is complex enough that it cannot be guessed, and that you change it regularly. You can change the fallback password using the Cloud Access Manager administration interface.

IMPORTANT: We recommend that you keep a hardcopy of the fallback password in a secure place, accessible only to staff with the authority to configure Cloud Access Manager. There is no reset facility for the fallback password. If you forget the password, you will need to completely re-install Cloud Access Manager.

In addition, fallback administration is not automatically exposed by the reverse proxy, so access to this user interface is restricted to internal connections.

Shared secret

The Cloud Access Manager shared secret is used to send information securely between Cloud Access Manager hosts, so that it can be stored securely in the Cloud Access Manager configuration database. In addition you will need to use the same shared secret to add new nodes to your Cloud Access Manager deployment.

IMPORTANT: We recommend that you keep a hardcopy of the shared secret in a secure place, accessible only to staff with the authority to configure Cloud Access Manager as there is no facility to change the shared secret within the software.

Proxy mapping and URL rewriting

In order to relay web content between a web application and a browser, the proxy must:

- Convert inbound public URLs from the browser to internal URLs which resolve to the target application on the private network.
- Convert hypertext links in outbound web content received from the internal web application to public URLs accessible by the browser.

To rewrite the URLs correctly, the proxy maintains an internal mapping table. An application URL can be mapped to its public URL equivalent in one of two ways:

- Folder-to-root mapping
- Root-to-root mapping

Folder-to-root mapping

With a folder-to-root mapping the public URL includes a path component, for example:

Public URL	Private URL
https://www.acme.com/erp	https://erp.acme.prod.local

Folder-to-root mappings allow you to multiplex several applications with a single external hostname, for example:

Public URL	Private URL
https://www.acme.com/erp	https://erp.acme.prod.local
https://www.acme.com/mail	https://owa.acme.prod.local
https://www.acme.com/payroll	https://payroll.acme.secure.net

Root-to-root mapping

Root-to-root mappings associate a single dedicated public URL hostname with a corresponding private URL hostname, for example:

Public URL	Private URL
https://erp.webapps.acme.com	https://erp.acme.prod.local
https://mail.webapps.acme.com	https://owa.acme.prod.local
https:// payroll.webapps.acme.com	https://payroll.acme.secure.net

Choosing the best mapping strategy

An advantage to using folder-to-root mappings is that the Secure Sockets Layer (SSL) certificate that authenticates the public server can cover all web applications with a single hostname. Using folder-to-root mappings is less expensive than root-to-root mappings, this is because an SSL certificate that authenticates a single hostname is generally cheaper than one which protects multiple hostnames.

However, use of folder-to-root mappings is not recommended for complex web sites, particularly those which rely heavily on client-side scripting to generate dynamic content. When using the folder-to-root approach, the proxy must rewrite relative URLs embedded in the content body, and if an embedded URL is built dynamically by the application, the proxy may need special rules (filters) to rewrite it correctly.

Rewriting relative URLs using folder-to-root

Private URL https://er-p.acme.prod.local	Public URL https://www.acme.-com/erp
/images/home.jp	/erp/images/home.jpg
../scripts/login.js	../erp/scripts/login.js
reports/salesfigures2014.pdf	erp/reports/salesfigures2014.pdf
this.href.location = '/register.aspx';	this.href.location = '/erp/register.aspx';

Relative URLs using root-to-root

Private URL https://er-p.acme.prod.local	Public URL https://er-p.webapps.acme.com
/images/home.jpg	Does not need rewriting.
../scripts/login.js	Does not need rewriting.
reports/salesfigures2014.pdf	Does not need rewriting.
this.href.location = '/register.aspx';	Does not need rewriting.

NOTE: As embedded relative URLs do not need rewriting when using the root-to-root approach there is less scope for URL rewrite problems, and the proxy can return the page to the browser more quickly. In general, we strongly recommended the root-to-root approach for both reliability and performance.

Choosing the right SSL certificate

We recommend that you purchase and install an Secure Sockets Layer (SSL) certificate from a Certificate Authority, this ensures Cloud Access Manager users can be confident they are interacting with a genuine service. Please refer to the *One Identity Cloud access Manager Installation Guide* for full instructions on how to request and install an SSL certificate for Cloud Access Manager.

You can purchase one of three types of SSL certificate:

- Single host certificate
- Wildcard certificate
- Subject Alternative Name (SAN) certificate

Single host certificate

This is typically the cheapest option. It is suitable for organizations who wish to proxy only a single application, or a collection of simple, static web applications with minimal client-side scripting using the folder-to-root method described above.

NOTE: The subject indicated in a single host certificate is a single hostname, for example www.acme.com

Wildcard certificate

This is usually the most costly option, but the most flexible. It allows you to set up unlimited root-to-root proxy mappings by permitting the domain name to be prefixed by any subdomain.

NOTE: The subject indicated in a wildcard certificate is a wildcard hostname, for example *.webapps.acme.com

Subject Alternative Name (SAN) certificate

A SAN certificate authenticates multiple explicitly-defined hostnames, the subjects indicated in a SAN certificate are listed, for example:

DNS Name: erp.acme.com

DNS Name: mail.acme.com

DNS Name: payroll.acme.com

A SAN certificate is widely considered to be more secure than a wildcard certificate. If a wildcard certificate falls into the wrong hands, then an attacker can pose as the legitimate organization through an unlimited number of hostnames. However, a similar compromise of a SAN certificate would only jeopardize the hostnames listed on that particular certificate.

Alternatively, the wildcard certificate has the advantage of flexibility, so you do not need to worry about altering your certificate in the future to accommodate more domain names. For this particular reason, as long as the private key of your wildcard certificate is properly secured, then you may consider the convenience of a wildcard certificate to outweigh the security benefits of a SAN certificate.

Single sign-on methods

Cloud Access Manager offers a variety of ways to automate sign-on to suit all types of web application:

- Security Assertion Markup Language (SAML) federation and WS-Federation
- HTTP Basic and NTLM
- Proxied form-fill
- Unproxied form-fill

Security Assertion Markup Language (SAML) federation and WS-Federation

Many modern web applications support Single Sign-On (SSO) using identity federation protocols. These methods rely on a separate, independent web system, called an identity provider or Security Token Service which performs the task of authenticating the user.

When multiple applications rely on the same identity provider you only need to enter your credentials once, so SSO is achieved. Cloud Access Manager operates as an identity provider for applications which support SAML or WS-Federation SSO.

This method is generally considered the fastest, most cost-effective, reliable, and efficient way to implement single sign-on for those applications which support it. Some Software-as-a-Service (SaaS) providers levy an additional charge for use of federated SSO, however this should be weighed against the significant advantages of this approach.

HTTP Basic and NTLM

Applications which require HTTP Basic or NT LAN Manager (NTLM) authentication rely on the browser to capture your credentials using a pop-up dialog. Information is then passed to the application in HTTP headers which the application uses to check if the supplied credentials are correct. It is typical for web applications which accept HTTP basic or NTLM authentication to run on an internal corporate network.

When such an application is accessed using the One Identity Cloud Access Manager proxy, the proxy can automatically construct the HTTP headers, as the browser would do. By using the username and password previously stored for a user, sign on to the application is automated.

If the application accepts HTTP Basic or NTLM authentication, this approach to Single Sign-On (SSO) is preferred over form-fill techniques, for further information please refer to Proxied form-fill on page 15.

Proxied form-fill

Some web applications do not accept HTTP basic or NT LAN Manager (NTLM) authentication, and instead prompt for your credentials by presenting a login form, this is known as forms-based authentication. Forms-based authentication is a common method of authenticating users for both public software as a service (SaaS) and on-premise web applications.

The Cloud Access Manager proxy can automatically insert JavaScript which detects a username and password entered into a login form, and can save that information in your Cloud Access Manager password wallet over a secure channel. Then, once your credentials have been saved and the application is launched again, Cloud Access Manager can detect

the username and password fields on an application login form, automatically insert your credentials into the correct fields, and submit the form.

NOTE: This technique is appropriate for applications which do not accept HTTP basic, NTLM, SAML or WS-Federation.

Unproxied form-fill

For certain applications where credentials are captured using a login form, you can configure Cloud Access Manager to automate sign on to the application without the need to proxy it. In this case Cloud Access Manager sends a login request directly to the application with your username and password inserted.

The advantages of this compared to proxied form-fill are:

- Responsiveness of application, as latency is reduced without an intermediate proxy
- Reduced IT costs, with a reduced load on the proxy
- Reliability is increased, as complications caused by URL rewrite are avoided.

The potential disadvantages are:

- No support for capturing credentials as you must enter your credentials into the Cloud Access Manager Password Wallet
- No support for change password forms, the old password field will not be prefilled with your current password, and any new passwords will not be captured
- Does not work with applications which send a pre-authentication cookie to the browser.

Using a reverse proxy or load balancer with Cloud Access Manager

If you use a reverse proxy server or load balancer in front of Cloud Access Manager, you must ensure that all headers required by Cloud Access Manager are maintained at all times. Cloud Access Manager injects JavaScript into app pages to manage session idle timeout and at the same time sets no cache headers on the response. It is essential to maintain the no cache headers at all times for Cloud Access Manager to function as designed. Removing or changing the no cache headers may cause session management issues, for example, when a user uses the **Back** button on their browser.

3

Security

Information stored by Cloud Access Manager

To support Single Sign-On (SSO) to non-federated web applications Cloud Access Manager saves your application passwords, encrypted in a table within the configuration database. The passwords are encrypted with AES-128-CBC, using a key derived from a combination of user ID and the shared secret which you specify during Cloud Access Manager installation.

For SSO to federate SAML and WS-Federation applications, Cloud Access Manager stores signing certificates in its configuration database along with their associated private keys. The private key associated with each signing certificate is encrypted with AES-128-CBC using a key derived from the shared secret.

To allow you to authenticate to Cloud Access Manager with your existing corporate credentials through Cloud Access Manager's built-in Security Token Service (STS), Cloud Access Manager must make an authenticated connection to an Active Directory or Lightweight Directory Access Protocol (LDAP) compliant directory. The credentials used to establish this authenticated connection are also stored in the configuration database and they are encrypted using AES-128-CBC using a key derived from the shared secret.

NOTE: All sensitive information in Cloud Access Manager's database is encrypted using keying material derived from the shared secret. The shared secret is stored on each Cloud Access Manager host in a local file, encrypted using Windows Data Protection API. Please refer to Microsoft documentation at http://msdn.microsoft.com/en-gb/library/ms995355.aspx for a detailed description of Windows DPAPI.

Inter-service communication

Cloud Access Manager transmits information between its services over Secure HTTP (HTTPS). Each connection is authenticated using the shared secret chosen during Cloud Access Manager installation.

User authentication

Cloud Access Manager allows you to access multiple systems without having to supply multiple sets of credentials. However the convenience of Single Sign-On (SSO) comes at the cost of security as an attacker that can hijack your Cloud Access Manager login account has the keys to the kingdom.

User authentication settings should therefore be reviewed thoroughly according to corporate security policy, with attention to:

- Prevailing password complexity rules
- Password expiry interval and password history rules
- Suitability of authentication method.

> NOTE: Consider using two-factor authentication, or smart card authentication, either for access to the application portal or for access to individual applications.

Where you are using a federated identity provider from a third-party organization, we recommend you seek assurances from that organization that their user authentication settings are in agreement with your security policy.

About us

One Identity solutions eliminate the complexities and time-consuming processes often required to govern identities, manage privileged accounts and control access. Our solutions enhance business agility while addressing your IAM challenges with on-premises, cloud and hybrid environments.

Contacting us

For sales or other inquiries, visit https://www.oneidentity.com/company/contact-us.aspx or call +1-800-306-9329.

Technical support resources

Technical support is available to One Identity customers with a valid maintenance contract and customers who have trial versions. You can access the Support Portal at https://support.oneidentity.com/.

The Support Portal provides self-help tools you can use to solve problems quickly and independently, 24 hours a day, 365 days a year. The Support Portal enables you to:

- Submit and manage a Service Request
- View Knowledge Base articles
- Sign up for product notifications
- Download software and technical documentation
- View how-to videos at www.YouTube.com/OneIdentity
- Engage in community discussions
- Chat with support engineers online
- View services to assist you with your product