Predict | Protect | Prevent

ARCON|PAM

App to App Password

arcon

Table of Contents

1 Overview

ARCOS password vault provides secured password storage for privileged user-ids across various applications, technology equipments, operating systems, databases and/or networking devices. However in this interconnected world of technology these equipments and systems have to interface with each other and such interfaces usually requires authentication. This is to ensure that only authorized systems talk to each other. The authentication methodology used by various systems is generally user-id and password. Every system has its own authentication method, storage path and use different encryption algorithms to ensure confidentiality.

The introduction of PAM solutions, that integrate all such technologies and systems now create a need for these solutions to provide the privileged ids and passwords to such interfacing systems. In many cases the access to such equipments or systems for certain activities is only through the privileged user-ids. Example: System monitoring solutions, vulnerability assessment solutions, scripts which take input or provide output to other systems, end-of-day operations, DB links etc. Historically the user-ids and passwords for the target systems are entered into the system/scripts which seeks to connect and they are either encrypted or in clear text.

Similarly there are applications which have hardcoded user-ids and passwords as they need to establish a connection with the datasource. These user-ids and passwords are generally stored in a local service accounts, .ini, .config files and/or embedded (hardcoded) in the .exe.

It is clear from the above that the privileged user-ids, which are hardcoded carry a high degree of risk from being compromised as these user-ids and passwords are known to be in clear text in various files on the application servers or are known to the support team as they insert this in the application configuration files during the any application implementation. Also this create administrative overhead as any change in such passwords have to be replicated in the applications as well as the operating system or databases as the case may be. There are several cases wherein these passwords are in services that need to be restarted again on updation.

There are various scenarios wherein the user-ids and passwords are required i.e. either application to application and/or application to system/databases etc.

ARCOS has a comprehensive framework to deal with various scenarios such that applications can connect seamlessly with datasources/systems by requesting for the privileged passwords from ARCOS vault. The following methodologies are available:

1. In cases where the application provides an API/input methods, ARCOS through its "ARCOS SPC Service" installed on the ARCOS app server connects to the vault and fetches the passwords on a defined frequency or on demand (similar to the password change process used for other systems) and provides the same to the target operating system.
   The following operating systems are supported:

   a. Windows
   b. All flavors of Unix
      On Windows the Windows Password Change Service (ARCOS WPC Service) will receive the various parameters described above and also the password, similarly on Unix flavors these parameters will be passed on through script which gets executed once the ARCOS logs on using the privileged user account already integrated or available with it.
      Note: This method works even if there are multiple applications/scripts on the same application server.
      There are pre and post events, which will get executed based on the application requirements. These are useful depending on how applications input the new passwords.
      Examples:
      --> Target Application will take the password inputs in clear text and thereafter the service needs to be restarted so that the password is encrypted. The service restart can be triggered by either ARCOS (as configured in the pre or post events) or by the application API.

--> Target Application will take the password input and simultaneously encrypt the same and store it in a particular path in .ini or .config files or similar other files.
--> Incase the passwords are to be inserted in the scripts, ARCOS will push the passwords to the "ARCOS SPC Service" or through scripts (depending on the operating system) and thereafter they are updated in the scripts etc.
The approach used above is useful to carry out scheduled password change activity triggered through ARCOS. The passwords of the applications can a also be scheduled to be changed at regular intervals say 30/60/90 days. This approach also takes away the dependency on the ARCOS system once the activity is carried out.
Note: The approach above does not change the conventional application password change process, what it does is automate the inputs. Hence customization (if any) is limited from the perspective of target applications. However only in cases where the passwords are kept in clear text, the same condition continues. However once the systems are integrated in ARCOS the assumption is that any access/change to such files will trigger alerts. This approach works well in case of complex IT Infrastructure as well as applications which are well known products.
The target applications need to be integrated with ARCOS with the requisite information/data, please follow the application integration steps outlined in the ARCOS implementation manual.

2. Applications which do not publish APIs for password inputs, will have to be customized to request for making a web call to "ARCOS PWD.API Online", which in-turn executes the "GetServicePassword" function. The target applications, whenever they need the privileged password will connect to the ARCOS web- server by making a web call to "ARCOS PWD.API Online". The target application needs to pass on credentials used during the application integration steps. However in order to avoid passing information which can then be misused, ARCOS provides a ARCOSSharedKey during the integration process. This SharedKey is dynamically generated based on various inputs like User-id, IP address, Application Path, Port No. etc. This SharedKey is matched during runtime to ensure that the application has been registered with ARCOS. The "GetServicePassword" thereafter collects the password from the vault and provides the same to the target application through the "ARCOS PWD.API Online".
The following operating systems are supported:

   a. Windows
   b. All flavors of Unix
   The following are the parameters:

Order No	Parameter Name	Description	Data type	Default Value
1	ARCOSWebAPIURL	Web URL of ARCOS App Server	String	""
2	ARCOSSharedKey	Shared Key / Identification No	String	""

Note: The "ARCOS PWD.API Online" provides resilience as the same is automatically available in HA mode as well.

However in organizations which would like to avoid bottlenecks such as network issues etc. would then have to chose a hybrid mode i.e. the "ARCOS WPC Service" installed on the target application server. In such cases the call will be made to the "ARCOS WPC Service" wherein the passwords are stored in encrypted format in memory. The parameters for the call remains the same and they are matched by "ARCOS WPC Service" at runtime and the success flag is the password.

In this case the "ARCOS WPC Service" receives the password from the "ARCOS SPC Service" installed on the ARCOS application server. There are two methods for updating the "ARCOS WPC Service".

Application password change is scheduled through ARCOS as per organization policy and the "ARCOS SPC Service" pushes the password to the "ARCOS WPC Service". The password is now stored in encrypted format along-with the ARCOSSharedKey/Identifier in the memory.

In case the service needs to be restarted, the memory is cleared and hence "ARCOS WPC Service" on every restart will trigger the password request via a Web Call to the"ARCOS PWD.API Online" by passing the ARCOSSharedKey/ Identifier which is stored within the "ARCOS WPC Service".

Important: The above hybrid method is supported for the Windows platform and for .net, C/C++ and java applications. ARCON recommends the web call approach as it is independent of the programing language and it avoids two way communication with agents especially in large and complex IT infrastructure.

3. Applications which require the privileged passwords and as well as channel to connect to the various systems. The channel request is mainly if the target applications are not able to make direct connection to the target systems. ARCOS not only provides the privileged password through the "ARCOS PWD.API Online" as described above but also helps in creating a secured connection via the ARCOS secured server to the target systems. In this scenario, target Application needs to invoke 3 functions with the required parameters.

→ **Web Call to "ARCOS PWD.API Online" and internal call to the "GetServicePassword"**
GetServicePassword have 2 parameters which are as follows:

Order No	Parameter Name	Description	Data type	Default Value
1	ARCOSWebAPIURL	Web URL of ARCOS App Server	String	""
2	ARCOSSharedKey	Shared Key / Identification No	String	""

→ **Internal Call to function "OpenARCOSGateway"**
OpenARCOSGateway have 9 parameters which are as follows:

Order No	Parameter Name	Description	Data type	Default Value
1	ARCOSWebAPIURL	Web URL of ARCOS App Server	String	""
2	ARCOSSharedKey	Shared Key / Identification No	String	""
3	LOBProfile	Profile of Service	String	""
4	ServerIP	IP Address of Server	String	""
5	ServiceType	Service Type of Server in ARCOS	String	""
6	UserName	Privilege User Name of Server	String	""
7	DBInstanceName	Database Instance Name of Server	String	""
8	IsUseCustomPort	Is Use Customer Port	String	“false”
9	CustomPort	Custom Port	String	“”

Return data of **OpenARCOSGateway** is array of string which contains following 4 values with ‘~’ (tilt) separated format.

- String [0] = Dynamic IP Address

- String [1] = Dynamic Port No.
- String [2] = Password of requested ARCOS Service
- String [3] = Session ID

→ Internal Call to function "CloseARCOSGateway"
CloseARCOSGateway have 1 parameter which is as follows:

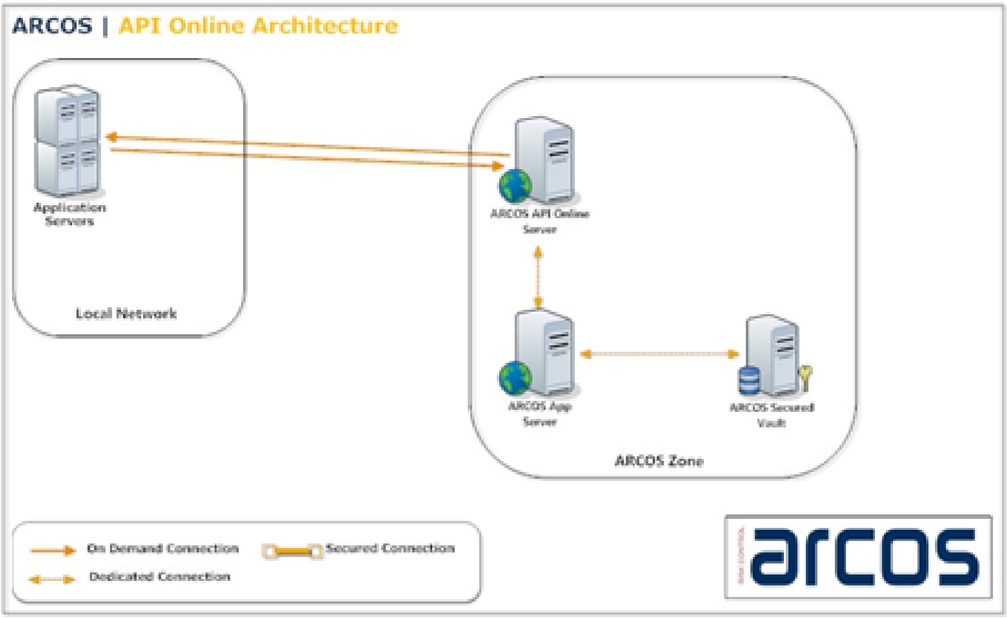
Order No	Parameter Name	Description	Data type	Default Value
1	pSessionID	Returned session ID by OpenARCOSGateway function	String	""

Note: the following are the service types of ARCOS which are currently supported by the "ARCOS PWD.API Online"

- SSHTelnet
- WindowsRDP
- TelnetROUTER
- SSHUNIX
- SSHLINUX
- OracleQA
- MSSQLQA
- SSHRouter
- SSHSwitch
- SSHFirewall
- TelnetSwitch
  Important: ARCOS PWD.API Online can be installed as a separate module on a different application server or on the ARCOS application server.

2 ARCOS pwd.api online Architecture

Following is basic architecture of ARCOS pwd.api online:



ARCOS API Online have 2 different functionalities to provide **Application to Application Password Management** functionality.

- Password Request
- Routing and Password Request

As ARCOS API Online is standby application server and does not have any connectivity preference with implemented ARCOS Server (vault) in same environment. In above both methods **Web URL of ARCOS App Server** and **Shared Key / Identification No** are common parameters which required to be remember by calling application.

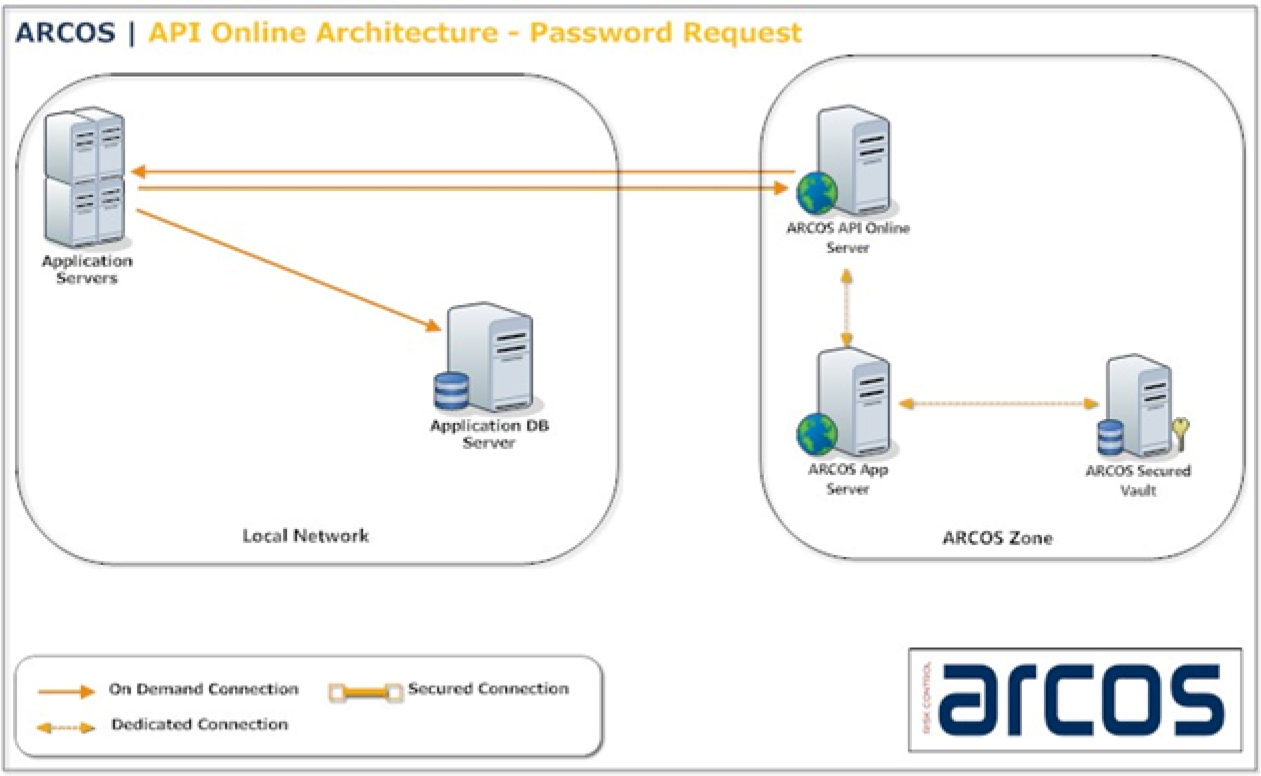
Definitions of common parameters

a. Web URL of ARCOS App Server
This parameters indicates, ARCOS Application server which need to be contacted by ARCOS API Online for retrieving password.

b. Shared Key / Identification No
This parameters indicates, ARCOS API Online is valid server and can connect with ARCOS Vault via ARCOS App Server. Also this parameter helps ARCOS Vault to identify which password is requested by calling application. Which reduces risk of knowing of application Privilege ID, Server IP Address and Database Instance (in case password gets compromised outside of ARCOS).

3 ARCOS API Online – Password Request

ARCOS API Online – Password Request method helps applications to retrieve password from ARCOS Vault. Following is basic architecture of ARCOS API Online for password request:



In this methodology, application need to invoke 1 function with required parameters, which are as follows:

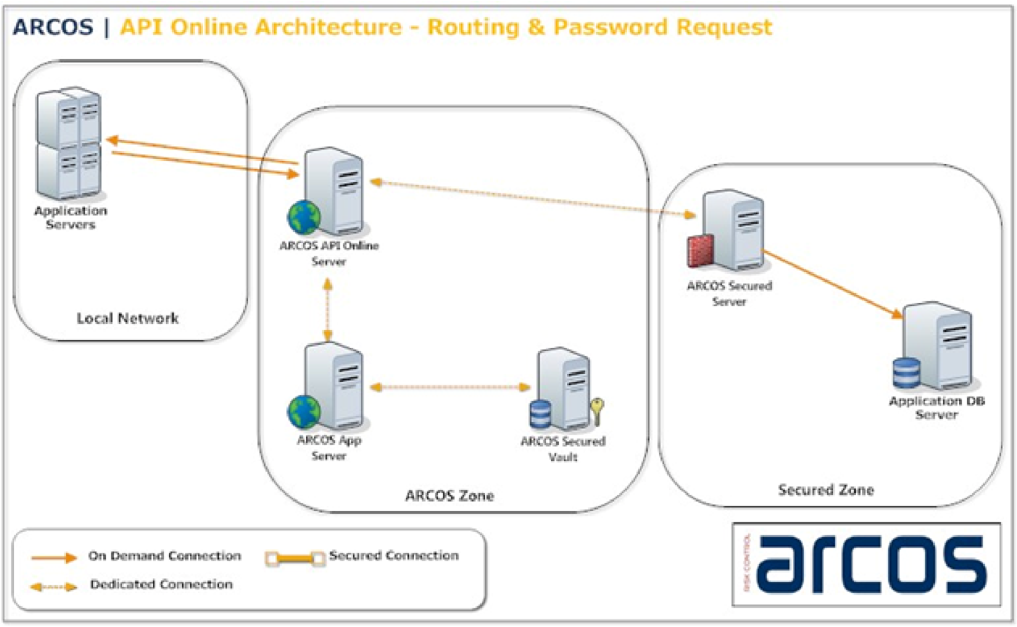
→ **GetServicePassword**

GetServicePassword have 2 parameters which are as follows:

Order No	Parameter Name	Description	Data type	Default Value
1	ARCOSWebAPIURL	Web URL of ARCOS App Server	String	""
2	ARCOSSharedKey	Shared Key / Identification No	String	""

4 ARCOS API Online – Routing and Password Request

ARCOS API Online – Routing and Password Request method helps applications to retrieve password from ARCOS Vault also helps to establish connectivity via ARCOS Secured Server (in case application does not have direct connectivity option to target device). Following is basic architecture of ARCOS API Online for routing and password request.



In this methodology application need to invoke 2 functions with required parameters, which are as follows:

OpenARCOSGateway have 9 parameters which are as follows:

Order No	Parameter Name	Description	Data type	Default Value
1	ARCOSWebAPIURL	Web URL of ARCOS App Server	String	""
2	ARCOSSharedKey	Shared Key / Identification No	String	""
3	LOBProfile	Profile of Service	String	""
4	ServerIP	IP Address of Server	String	""
5	ServiceType	Service Type of Server in ARCOS	String	""
6	UserName	Privilege User Name of Server	String	""
7	DBInstanceName	Database Instance Name of Server	String	""
8	IsUseCustomPort	Is Use Customer Port	String	“false”
9	CustomPort	Custom Port	String	“”

Return data of **OpenARCOSGateway** is array of string which contains following 4 values with '~' (tilt) separated format.

- String [0] = Dynamic IP Address
- String [1] = Dynamic Port No
- String [2] = Password of requested ARCOS Service
- String [3] = Session ID

CloseARCOSGateway

CloseARCOSGateway have 1 parameter which is as follows:

Order No	Parameter Name	Description	Data type	Default Value
1	pSessionID	Returned session ID by OpenARCOSGateway function	String	""

5 Supported ARCOS Service Types

Following are the service types of ARCOS are currently supported by ARCOS API Online for above mentioned method:

1. SSHTelnet
2. WindowsRDP
3. TelnetROUTER
4. SSHUNIX
5. SSHLINUX
6. OracleQA
7. MSSQLQA
8. SSHRouter
9. SSHSwitch
10. SSHFirewall
11. TelnetSwitch

Privileged Access Management Suite

arcon

Predict | Protect | Prevent