Predict | Protect | Prevent

ARCON|PAM

SIEM

arcon

Table of Contents

Disclaimer

The handbook of ARCON PAM solution is being published to guide stakeholders and users. If any of the statements in this document are at variance or inconsistent it shall be brought to the notice of ARCON through the support team. Wherever appropriate, references have been made to facilitate better understanding of the PAM solution. ARCON team has made every effort to ensure that the information contained in it was correct at the time of publishing.
Nothing in this document constitutes a guarantee, warranty, or license, expressed or implied. ARCON disclaims all liability for all such guarantees, warranties, and licenses, including but not limited to: Fitness for a particular purpose; merchantability; non-infringement of intellectual property or other rights of any third party or of ARCON; indemnity; and all others. The reader is advised that third parties can have intellectual property rights that can be relevant to this document and the technologies discussed herein, and is advised to seek the advice of competent legal counsel, without obligation of ARCON.

Trademarks
Other product and corporate names may be trademarks of other companies and are used only for explanation and to the owners' benefit, without intent to infringe.

Sales Contact
You can directly contact us with sales related topics at the email address <sales@arconnet.com>, or leave us your contact information and we will call you back.

1 Overview

SIEM as a technology provides real-time analysis of security-alert generated by network hardware and applications. SIEM refers to System Incident and Event Management. It is a solution that tracks all the events from the network and infrastructure and collects the logs and analyzes them. The ARCON PAM SIEM Connector Service provides the logs to the third-party applications. This service allows third-party applications to perform analytic or security incident and event management. The SIEM system collects the logs from different applications, operating systems, and creates a dashboard on the security incidences. Similarly, you can integrate ARCON PAM with different SIEM solutions.

SIEM products provide real-time monitoring and alerts, as well as reports for compliance (primarily log management) and threat management by correlating and analyzing data drawn from network security devices and applications.

2 About Integrations

The following type of logs are available for SIEM Application to access from PAM:

- User Login Attempt
- User Audit Logs
- Failed Login Attempt
- Service Audit Logs
- Service Access
- Service Command
- Password View
- Service Password View
- Service Password Change
- Service Password Envelope Print
- Service ARCOS Log
- Mapping
- Command Logs
- ARCON PAM Logs
- ARCON User Validity Details

Some of the important SIEM solutions in the industry are:

- Symantec SIEM
- McAfee SIEM
- RSA
- QRadar by IBM
- Splunk
- Arcsight by HP
- LogRythm
- Rapid7

ARCON PAM data can be accessed by any SIEM application using the following connection methods:

- SIEM Connector Service
- JDBC Connection
- API Connection
- Syslog Server

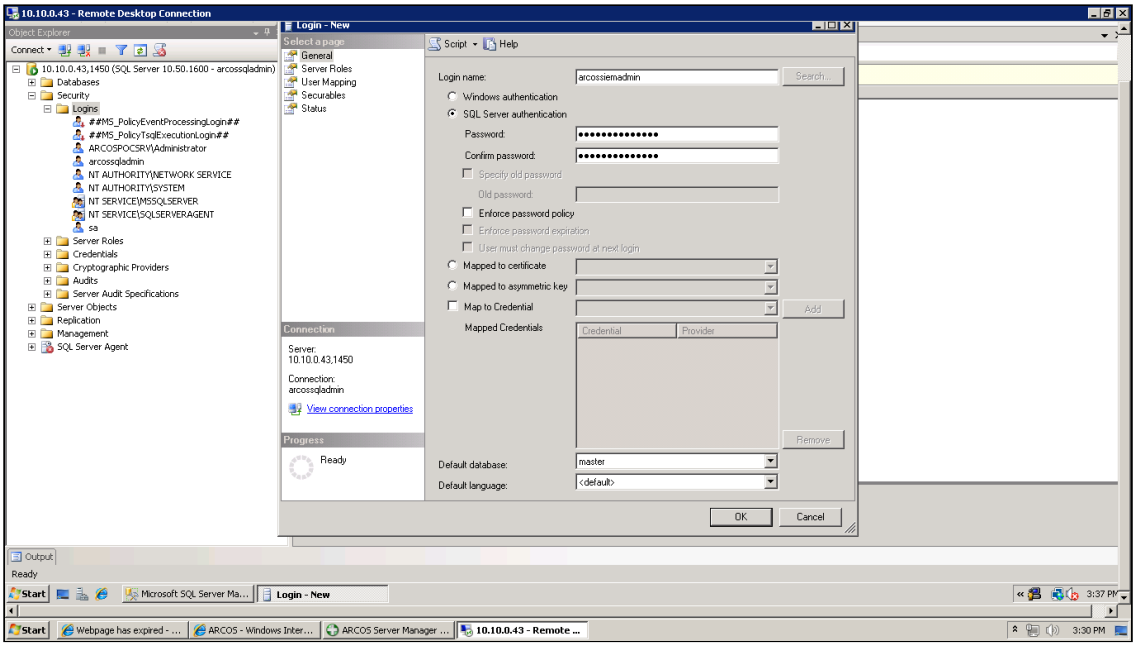
3 SIEM Connector Service

ARCON PAM can integrate with SIEM to send audit logs of privileged account activities in the enterprise SIEM solution. The SIEM rights are granted to the ARCON PAM Administrators to maintain and monitor the SIEM activities. SIEM connector provides data of commands fired in ARCON PAM, User validity status, and activities performed by Admin.

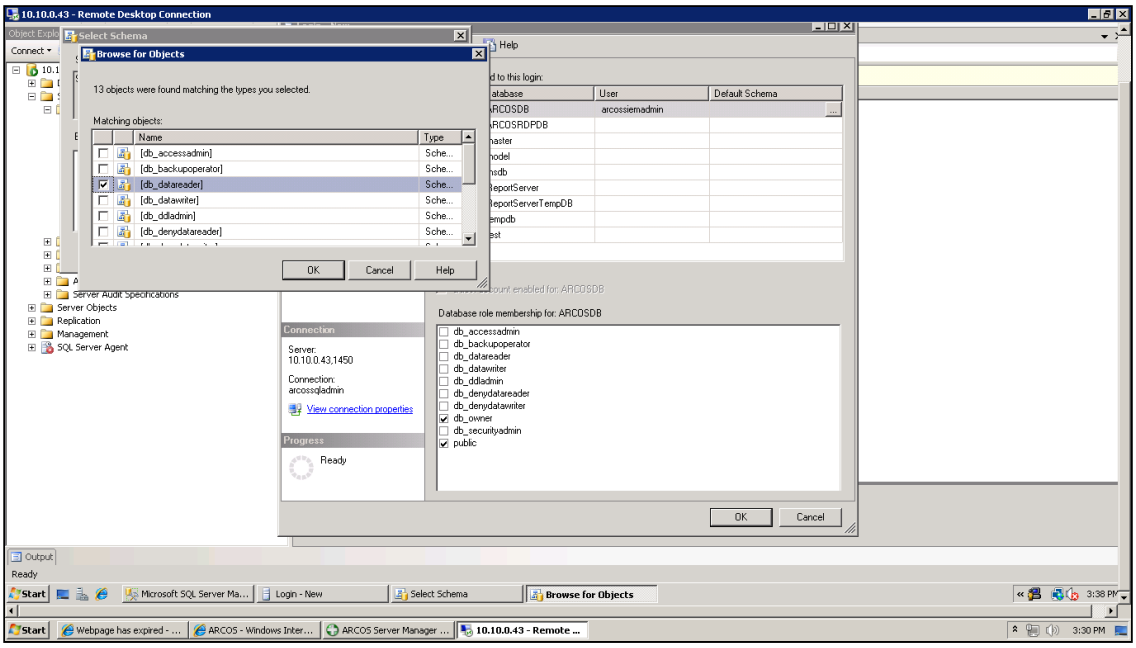
3.1 Pre-requisites for SIEM Connector Service

- Db Settings configurations

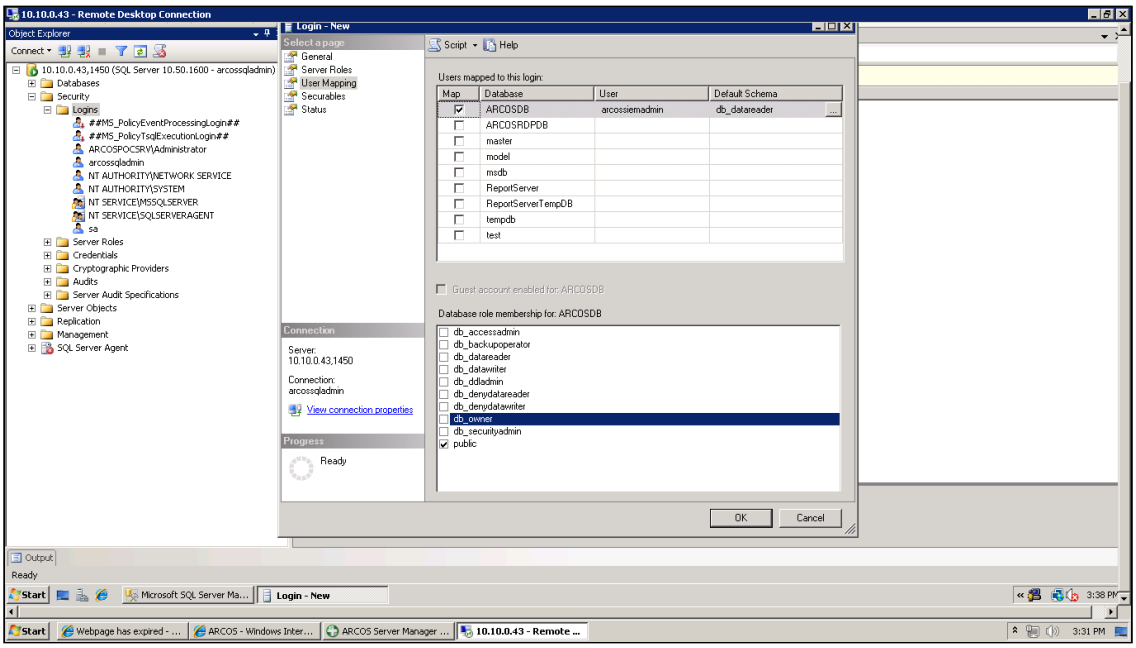
1. User creation for accessing the SIEM through ARCON PAM.
    a. Create a User named arcossiemadmin.



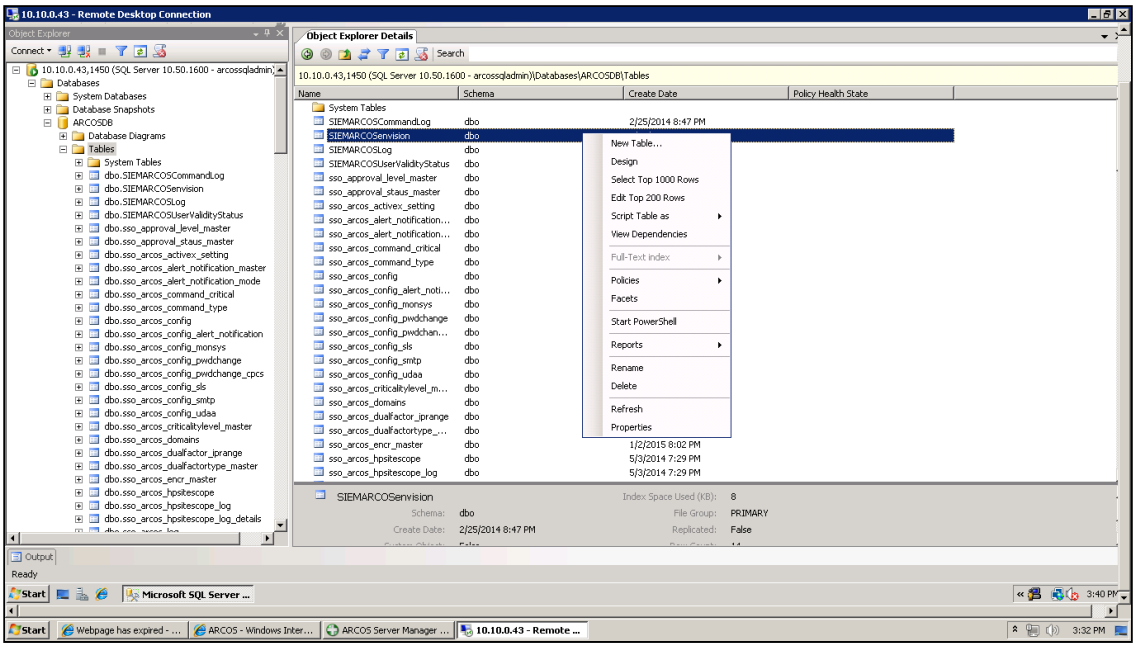
    b. Grant the user db_datareader rights.



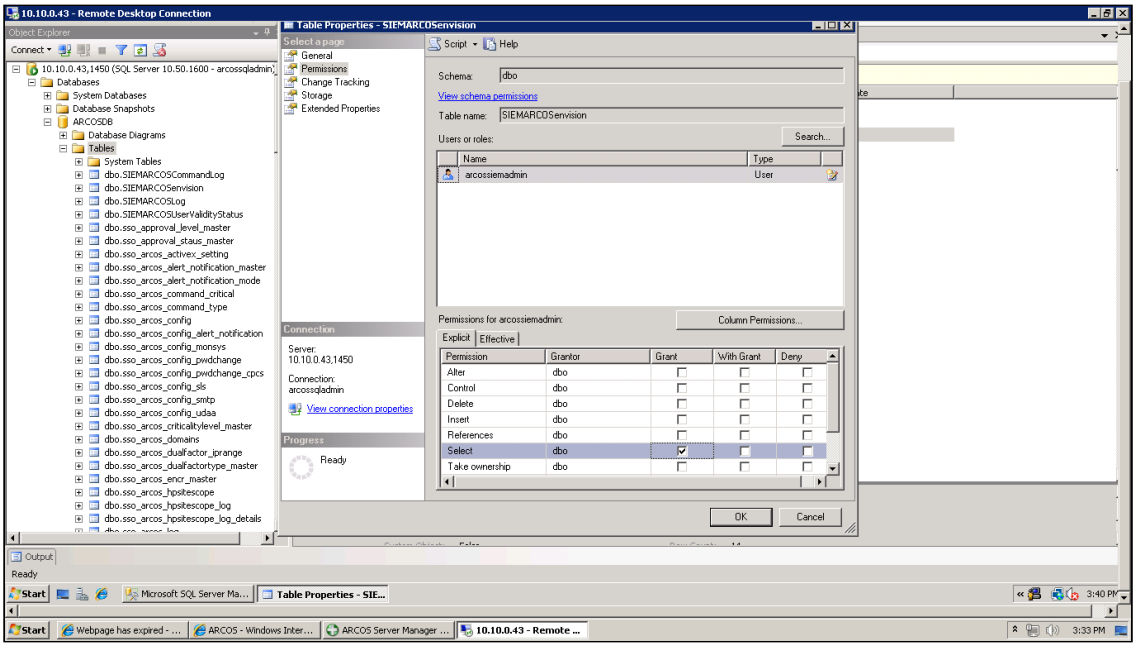
c. Do NOT make the user db_owner.



d. Now, Login via the Admin credentials and select the table under ARCOSDB SIEMARCOSenvision



e. Under Permissions grant the user arcossiemadmin **select** rights only.



3.2 Installations

1. ARCOS SIEM Connector Service Setup is to be installed on ARCON PAM Application or Database Server.
2. Post-installation, dbsetting.ini file of ACMO is to be placed in the installed location, for the service to connect to the database.

3.3 Method 1

In this method, the details are fetched from SIEMARCOSEnvision table of the database where the trigger occurs based on the various events. Each Event along with its table is explained below.

SIEM Connector Service Configurations

ARCOSSIEMConnectorService - ARCOS Event ID From SIEM	This configuration sets the ARCON PAM Event ID that is defined in ARCON PAM SIEM Connector Service.
User Login Attempts (Event ID: 9100)	This Event ID represents the successful attempts in the ARCON PAM.
Failed Login Attempts (Event ID: 9101)	This Event ID represents the invalid logon attempts in ARCON PAM.
Service Access (Event ID: 9200)	This Event ID represents the individual servers accessed by any ARCON PAM user.
Service Command (Event ID: 9300)	This Event ID represents the command and informs you about the particular command that is fired in the session.
Password View (Event ID: 9400)	This Event ID contains the information about the level of user who has requested and approved the password, time of the password request, the device for which the password request is sent, and so on.
Service Password Change (Event ID: 9500)	This Event ID is a password change service. It consists of information about the password that is printed on the server.
Service Password Envelope Print (Event ID: 9600)	This Event ID consists of information about the envelope printing and password printing. There are 10 levels of printing status.
ARCOS Log (Event ID: 9700)	This Event ID represents the all generic ARCON PAM logs (They are Object type and Operation Type).
(Event ID: 9800)	This Event ID shows the Service Password is in Open or Closed status (Check In or Check Out).
Process Log (Event ID: 9900)	This Event ID represents all the ARCON PAM Process logs.
SSM (Event ID: 9901)	This Event ID represents all the ARCON PAM SSM (Smart Session Monitoring) logs.

SIEM table connects to these tables and fetches the data.

The Table Structure for **User Login Attempts** is as follows

Field Name	Description
EventID	Event Id for User Login attempt is 9100.
EventTimeStamp	The date and time value when the event was triggered

Field Name	Description
UserID	User Login ID / Domain ID
UserName	User Display Name
UserDomain	User Domain Name
UserMAC [IP]	User Desktop's foot print i.e the IP address
UserSessionLoginTimeStamp	User Login date & time
UserSessionLogoutTimeStamp	User Logout date & time
Status	User Login status (Success)

The Table structure for **Failed Login Attempts** is as follows

Field Name	Description
EventID	Event Id for User Login attempt is 9101.
EventTimeStamp	The date and time value when the event was triggered
UserID	User Login ID / Domain ID
UserName	User Display Name
UserDomain	User Domain Name
UserMAC [IP]	User Desktop's foot print i.e the IP address
UserSessionLoginTimeStamp	User Login date & time
UserSessionLogoutTimeStamp	User Logout date & time
Status	Failed/ Unknown

The Table structure for **Service Access** is as follows

Field Name	Description
EventID	Event Id for Service Access is 9200.
EventTimeStamp	The date and time value when the event was triggered.
UserID	User Login ID / Domain ID
UserName	User Display Name
UserMAC [IP]	User Desktop's foot print i.e the IP address

Field Name	Description
UserSessionID	User SessionID
ServerID	ServerID
ServerIPAddress	Server IP Address
ServerUserID	Server user account name (i.e name of the privileged account)
ServeConnectionType	ServeConnectionType
ServerSessionID	Server SessionID
ServerSessionLoginTimeStamp	Server session start date & time
ServerSessionLogoutTimeStamp	New session end date & time
LOBProfile	LOB/Profile Name

The Table structure for **Service Command** is as follows

Field Name	Description
EventID	Event Id for Service Command is 9300.
EventTimeStamp	The date and time value when the event was triggered
UserID	User Login ID / Domain ID
UserName	User Display Name
UserMAC [IP]	User Desktop's foot print i.e the IP address
UserSessionID	User SessionID
ServerID	ServerID
ServerIPAddress	Server IP Address
ServerUserID	Server user account name (i.e name of the privileged account)
ServeConnectionType	ServeConnectionType
ServerSessionID	Server SessionID
Command	Commands fired on any server by the user
CommandResponse	Response after the Command
CommandTimeStamp	Command fired on date & time
LOBProfile	LOB/Profile Name

The Table structure for **Password View** is as follows

Field Name	Description
EventID	Event Id for Password View is 9400.
EventTimeStamp	The date and time value when the event was triggered
ServerID	ServerID
ServerIPAddress	Server IP Address
ServerUserID	Server user account name (i.e name of the privileged account)
ServeConnectionType	ServeConnectionType
UserName	User Display Name
UserDomain	User Domain Name
ViewOnTimeStamp	Password viewed on date & time
ViewForHours	Password viewed for hours
ViewDescription	Password view/ open description
AuthbyUserName1	Password view/open authorized by user name1
AuthbyDomainName1	Password view/open authorized by user domain name1
AuthbyUserName2	Password view/open authorized by user name2
AuthbyDomainName2	Password view/open authorized by user domain name2
LOBProfile	LOB/Profile Name

The Table structure for **Service Password Change** is as follows

Field Name	Description
EventID	Event Id for Service Password Change is 9500.
EventTimeStamp	The date and time value when the event was triggered
ServerID	ServerID
ServerIPAddress	Server IP Address
ServerUserID	Server user account name (i.e name of the privileged account)
ServeConnectionType	ServeConnectionType
PasswordChangedByUserName	Password changed by the user name

Field Name	Description
IsPasswordChangedManually	Is password manually changed (Yes / No)
PasswordChangedOnTimeStamp	Password viewed on date & time
Status	Failed/Success/Unknown
LOBProfile	LOB/Profile Name

The Table structure for **Service Password Envelope Print** is as follows

Field Name	Description
EventID	Event Id for Service Password Change is 9600.
EventTimeStamp	The date and time value when the event was triggered
ServerID	ServerID
ServerIPAddress	Server IP Address
ServerUserID	Server user account name (i.e name of the privileged account)
ServeConnectionType	ServeConnectionType
PrintedByUserName	Password envelope printed by user name
PrintedOn	Password envelope printed on date & time
VerifiedByUserName1	Password envelope print verified / Authorized by user name1
VerifiedByUserName2	Password envelope print verified / Authorized by user name2
PrintingStatus	Password envelope status • Generated • Printed • First Reprint • Second Reprint • Third Reprint • Fourth Reprint • Fifth Reprint • Sixth Reprint • Seventh Reprint • Eighth Reprint • Ninth Reprint • Tenth Reprint • Can Not Print Any More
LOBProfile	LOB/Profile Name

The Table Structure for **ARCOS Log** is as follows

Field Name	Description
EventID	Event Id for Service Password Change is 9700.
EventTimeStamp	The date and time value when the event was triggered
UserName	User Display Name
ObjectType	• ARCOS Log Object Types are as follows • Group Transactions • User Transactions • Services Transactions • Transaction Between User And User Group • Transaction Between Service And Service Group • Transaction Between User And Service • Transaction Between User Group And Service Group • Transaction Between User And Restrict Command • Users Privileges • LOB / Profile Transactions • Transaction Between LOB/Profile And Users • Transaction Between LOB/Profile And Service • Transaction Between LOB/Profile And Service Group • Transaction Between LOB/Profile And User Group • Transaction Between Service And Windows Services • Transaction Between Service And Windows DCOM • Transaction Between Service And Windows Task • Transaction Between Service And Depended Service • Command Profiler • User Security Setting
OperationType	ARCOS Log Operation Types are as follows • Created • Modified • Deleted • Assigned • Revoked • CheckerApproved • CheckerNotApproved • Accessed • Viewed • Shared • Un-Shared
TrasactionFor	ARCOS Log transaction for value
OldValue	ARCOS Log transaction for old value
NewValue	ARCOS Log transaction for new value

Field Name	Description
LogTimeStamp	ARCOS Log transaction date & time

The Table structure for **Process Log** is as follows

Field Name	Description
EventID	Event Id for Service Password Change is 9900.
EventTimeStamp	The date and time value when the event was triggered
LogID	Log ID
UserName	User Display Name
UserDisplayName	User Display Name
UserMAC	User Desktop's foot print i.e the IP address
ServeConnectionType	ServeConnectionType
ServerSessionID	Server SessionID
ServerIPAddress	Server IP Address
ServerUserID	Server user account name (i.e name of the privileged account)
ProcessName	Process Name
ProcessTitle	Process Title
LogType	Log type i.e (Video log downloaded, Log details viewed)
LogTimeStamp	ARCOS Log transaction date & time
HostName	Hostname

The Table structure for **SSM log** is as follows

Field Name	Description
EventID	Event Id for Service Password Change is 9901.
EventTimeStamp	The date and time value when the event was triggered
LogID	Log ID
UserName	User Display Name
UserDisplayName	User Display Name
UserMAC	User Desktop's foot print i.e the IP address

Field Name	Description
ServerSessionID	Server SessionID
ServerIPAddress	Server IP Address
ServerUserID	Server user account name (i.e name of the privileged account)
ServeConnectionType	ServeConnectionType
ProcessName	Enter Process Name
Action	It gives the details where the user clicked or performed an activity.
LogType	Log type i.e (Video log downloaded, Log details viewed)
LogTimeStamp	ARCOS Log transaction date & time

3.4 Method 2

In this method, the details are fetched from SIEMARCOSCommandLog, SIEMARCOSLog, and SIEMARCOSUserValidityStatus tables of the database where the trigger occurs based on the various events. Each Event along with its table is explained below.

SIEM Tool can connect to this table and fetch the data.

Table Structure for **Command Logs** Table

Field Name	Description
UserID	User Domain ID
UserSessionLogID	ARCON PAM Session ID
UserLoggedIn	Logged in time
UserLoggedOut	Logged out time
IPMAC	IP,MAC Address of System Logged in From
ServiceType	Service Type (Linux,Windows,SQL)
ServiceDescription	Service IP and Username of Service
Command	Command Fired
CommandTimeStamp	Time when the command was fired
CommandResponse	Response after command.

Field Name	Description
ServiceLoggedin	Service Logged in Time
ServiceLoggedOut	Service Logged Out Time
ServiceLogID	ARCON PAM Service Log ID
PasswordAge	Password Age of Service
PasswordLastChangedDate	Date on which Password was last changed
PasswordNextChangeDate	Date on which Next Password change is scheduled
PasswordStatus	Password Status (Open / Close)
PasswordOpenBy	User ID if password is open
PasswordOpenDate	Password open date if password is open

Table Structure of **ARCON PAM Log** Table

Field Name	Description
UserID	User Domain ID
ObjectType	Operation Performed Type (User, Group Transaction)
OperationType	Operation Type (Assigned, Revoked, Created)
TrasactionFor	User ID for which transaction is performed
OldValue	Old Value before modify
NewValue	New Value after modify
TimeStamp	Time of activity when it was performed

Table Structure of **ARCON PAM User Validity Status** Table

Field Name	Description
UserID	User Domain ID
DomainName	Domain Name of User ID
UserType	User Type (Client or Administrator)

Field Name	Description
ValidUpto	User Valid up to date
Status	Status of user (Active)
Hours	No of Hours user is active
Days	No of Days Still left for the user
ForthNights	No of Forth Nights left for the user
Months	No of Months left for the user
Years	No of Years left for the user

Sample Data in **Command Logs** Tables

UserID	ARCOSADMIN1
UserSessionLogID	123123
UserLoggedIn	04-12-2009 16:39
UserLoggedOut	04-12-2009 16:53
IPMAC	192.168.0.239[0017A4E4953E][BFEBFBFF000006F
ServiceType	SSH LINUX
ServiceDescription	192.168.0.240@root
Command	[root@ARCONLinuxSRV ~]# passwd
CommandTimeStamp	16-10-2009 13:32
CommandResponse	
ServiceLoggedin	16-10-2009 13:32
ServiceLoggedOut	16-10-2009 13:32
ServiceLogID	3529
PasswordAge	30
PasswordLastChangedDate	16-09-2009 13:32
PasswordNextChangeDate	16-11-2009 13:32
PasswordStatus	Open

PasswordOpenBy	UserID
PasswordOpenDate	20-09-2009 13:32

Sample Data in **ARCON PAM Log** Tables

UserID	ARCOSADMIN
ObjectType	Users Privileges
OperationType	Revoked
TrasactionFor	ARCON PAM ADMIN 1
OldValue	
NewValue	ARCON PAM Group Admin -> ALLSERVER
TimeStamp	01-10-2009 12:52

Sample Data in **ARCON User Validity Status** Tables

UserID	ARCOSADMIN
DomainName	ARCON
UserType	Administrator
ValidUpto	01-01-2058 00:00
Status	Active
Hours	421420
Days	17560
ForthNights	1254
Months	577
Years	49

4 JDBC Connection for SIEM (IBM QRadar)

4.1 IBM QRadar Configuration

Follow below steps before configure IBM QRadar with ARCOS Database:

1. Create MS – SQL User on ARCOS Database
2. Grant Permission only on table "dbo.SIEMARCOSenvision"
   GRANT SELECT, INSERT ON dbo.SIEMARCOSenvision To MS-SQL User
3. Run an ARCOS SIEM Connector Service Setup (On Windows Server)
4. Copy dbsetting.ini file to the installed location
5. Start "ARCOSSIEMConnectorService" from services.msc

4.2 Configuration of JDBC Protocol

Log sources configured with Java Database Connectivity (JDBC) protocol can remotely poll databases for events. The JDBC protocol enables QRadar to collect information from tables or views that contain event data from several database types.

The following table describes the parameters for JDBC protocol:

Parameter	Description
Log Source Name	Type a unique name of the log source.
Log Source Description	Optional. Type a description for the log source.
Log Source Type	From the list, select the type of log source to add.
Protocol Configuration	From the list, select JDBC.
Log Source Identifier	Type the log source identifier in one of the following formats:
	database@hostname
	table name|database@hostname
	The database name must match the value of the Database Name parameter. The database name is a required parameter.
	The hostname is the hostname or IP address for the device that hosts the database. The hostname must match the parameter in the IP or Hostname field. The hostname is a required parameter.
	Optional. The table name is the name of the table or view on the database which contains the event records. If you define the name of a table or view, you must include a pipe (|) character as a separator. The name of the view or table must match the Table Name field.
Database Type	From the list box, select the type of database that contains the events.

Parameter	Description
Database Name	Type the name of the database to which the protocol can connect. The database name must match the database name specified in the Log Source Identifier field.
IP or Hostname	Type the IP address or hostname of the database server.
Port	Type the port number used by the database server. The default displayed depends on the selected Database Type. The valid range is 0 to 65536. The defaults include:
	MSDE – 1433
	Postgres – 5432
	MySQL – 3306
	Sybase – 1521
	Oracle – 1521
	Informix – 9088
	The JDBC port must match the listen port configured on the remote database. The database must permit incoming TCP connections.
	If a Database Instance is used with the MSDE database type, administrators must leave the Port parameter blank in the log source configuration.
Username	Type the database username. The username can be up to 255 alphanumeric characters in length and can include underscore (_) characters.
	To track access to database access for audit purposes, administrators can create a specific user on the database for QRadar.
Password	Type the database password. The password can be up to 255 characters in length.
Confirm Password	Confirm the password to access the database.
Authentication Domain	Type a domain for the database.
	A domain must be configured for MSDE databases that are within a Windows domain. If your network does not use a domain, leave this field blank.
Database Instance	Type the database instance, if required. MSDE databases can include multiple SQL server instances on one server.

Parameter	Description
	When a non-standard port is used for the database or administrators have blocked access to port 1434 for SQL database resolution, the Database Instance parameter must be blank in the log source configuration.
Predefined Query	Optional. Select a predefined database query for the log source. If a predefined query is not available for the log source type, administrators can select none.
Table Name	Type the name of the table or view that includes the event records.
	The table name can include the following special characters: dollar sign ($), number sign (#), underscore (_), en dash (-), and period(.).
Select List	Type the list of fields to include when the table is polled for events. Administrators can use a comma-separated list or type * to select all fields from the table or view.
	If a comma-separated list is defined, the list must contain the field defined in the Compare Field.
Compare Field	Type a numeric value or timestamp field from the table or view that can identify new events added between queries to the table.
	This field enables the protocol to identify events that were previously polled by the protocol to ensure that the duplicate events are not created.
Use Prepared Statements	Select this checkbox to use prepared statements.
	Prepared statements enable the JDBC protocol source to setup the SQL statement, and then execute the SQL statement numerous times with different parameters. For security and performance reasons, most JDBC protocol configurations can use prepared statements.
	Clear this checkbox to use an alternative method of querying that do not use pre-compiled statements.
Start Date and Time	Optional. Configure a start date and time for when the protocol can start to poll the database.
	If a start time is not defined, the protocol attempts to poll for events after the log source configuration is saved and deployed.
Polling Interval	Type the polling interval, which is the amount of time between queries to the database. The default polling interval is 10 seconds.

Parameter	Description
	Administrators can define a longer polling interval by appending H for hours or M for minutes to the numeric value. The maximum polling interval is 1 week in any time format. Numeric values without an H or M designator poll in seconds.
EPS Throttle	Type the number of Events Per Second (EPS) that you do not want this protocol to exceed. The default value is 20000 EPS.
Use Named Pipe Communication	If MSDE is configured as the database type, administrators can select this checkbox to use an alternative method to a TCP/IP port connection.
	Named pipe connections for MSDE databases require the username and password field to use a Windows authentication username and password and not the database username and password. The log source configuration must use the default named pipe on the MSDE database.
Database Cluster Name	If the Use Named Pipe Communication checkbox, the Database Cluster Name parameter is displayed.
	If you use your SQL server in a cluster environment, define the cluster name to ensure that named pipe communications function properly.
Use NTLMv2	Select the Use NTLMv2 checkbox to force MSDE connections to use the NTLMv2 protocol when communicating with SQL servers that require NTLMv2 authentication. The default value of the checkbox is selected.
	The Use NTLMv2 checkbox does not interrupt communications for MSDE connections that do not require NTLMv2 authentication.
Use SSL	Select this checkbox to enable SSL encryption for the JDBC protocol.
Enabled	Select this checkbox to enable the log source.
	When this checkbox is clear, the log source does not collect events and the log source is not counted in the license limit.
Credibility	Select the credibility of the log source. The range is 0 (lowest) - 10 (highest). The default credibility is 5.
	Credibility is a representation of the integrity or validity of events that are created by a log source. The credibility value that is assigned to a log source can increase or decrease based on incoming events or adjusted as a response to user-created event rules. The credibility of events from log sources contributes to the calculation of the offense magnitude and can increase or decrease the magnitude value of an offense.

Parameter	Description
Target Event Collector	Select the target for the log source. When a log source actively collects events from a remote source, this field defines which appliance polls for the events.
	The target event collector enables administrators to poll and process events on the target event collector, instead of the Console appliance. Distributing events across target event collectors can improve performance in distributed deployments.
Coalescing Events	Select this checkbox to enable the log source to coalesce (bundle) events.
	Coalescing events increase the event count when the same event occurs multiple times within a short time interval. Coalesced events provide administrators a way to view and determine the frequency with which a single event type occurs on the Log Activity tab.
	When this checkbox is clear, events are viewed individually and events are not bundled.
	New and automatically discovered log sources inherit the value of this checkbox from the System Settings configuration on the Admin tab. Administrators can use this checkbox to override the default behavior of the system settings for an individual log source.
Store Event Payload	Select this checkbox to enable the log source to store the payload information from an event.
	New and automatically discovered log sources inherit the value of this checkbox from the System Settings configuration on the Admin tab. Administrators can use this checkbox to override the default behavior of the system settings for an individual log source.
Log Source Language	Select the language of the events that are generated by the log source.
	The log source language helps the system parse events from external appliances or operating systems that can create events in multiple languages.
Log Source Extension	Optional. Select the name of the extension to apply to the log source.
	This parameter is available after a log source extension is uploaded. Log source extensions are XML files that contain regular expressions, which can override or repair the event parsing of a device support module (DSM).

Parameter	Description
Extension Use Condition	From the list box, select the use condition for the log source extension. The options include:
	Parsing enhancement - Select this option when most fields parse correctly for the log source.
	Parsing override - Select this option when the log source is unable to correctly parse events.
Groups	Select one or more groups for the log source.

4.3 Procedure

1. Click the **Admin** tab.
2. Click the **Log Sources** icon.
3. Click **Add**.
4. Configure the parameters for the log source. Click **Save**.
5. On the **Admin** tab, click **Deploy Changes**

5 API Connection for SIEM

5.1 Get User Login Attempt

Get User Login Attempt details by From Date and To Date API.

Type of Request		POST			
Endpoint		/api/SIEM/GetUserLoginAttempt			
Available in API Version		All versions			
Release Version		U4			
Pre Endpoint Call		/arconToken			
Request	Params	-	-	-	-
	Headers	Content-Type	application/x-www-form-urlencoded	*Mandatory*	-
		Authorization	Bearer eyJ0[... Removed for brevity ....]INjTU8	*Mandatory*	-
		x-PAM-Version	-	*Not Applicable*	-
	Body	**Request JSON** 1 { 2 "FromDate":"2019-06-22T02:37:00", 3 "ToDate":"2019-06-26T12:37:00" 4 }			

Response	**Response JSON**
```
{
  "Program": "ARCON PAM API",
  "Version": "1.0",
  "DateTime": "25/Oct/2019 01:05:16",
  "Success": true,
  "ErrorCode": null,
  "ErrorMessage": null,
  "Message": "50 Records Found",
  "Result": [
    {
      "EventID": 9100,
      "EventTimeStamp": "25/Oct/2019 01:05:16",
      "UserID": 35,
      "UserName": "ARCXXXDMIN",
      "UserDomain": "ARXXXAUTH",
      "UserMAC": "10.00.0.00[E0D55E783515]
[Defaultstring][BFEBxxxF000906E9][ACM4.8.5.0]",
      "UserSessionLoginTimeStamp": "06/22/2019
15:40:13",
      "UserSessionLogoutTimeStamp": "06/26/2019
11:26:48",
      "Status": "Success"
    }]
}
```
 |
Response Time	181 milliseconds
Post Endpoint Call	None
Supported Features	NA

5.1.1 Data Types of Request and Response

Type	Parameters	Data Type
Request	FromDate	Date Time ('2012-04-23T18:25:43.511Z' format)
	ToDate	Date Time ('2012-04-23T18:25:43.511Z' format)
Response	EventID	int
	EventTimeStamp	Date Time
	UserID	string
	UserName	string

Type	Parameters	Data Type
	UserDomain	string
	DomainName	string
	UserMAC	string
	UserSessionLoginTimeStamp	Date Time
	UserSessionLogoutTimeStamp	Date Time
	Status	string

5.1.2 Code Blocks for 'SIEM/GetUserLoginAttempt'

C# Sample Code - (Using RestSharp)

```
var client = new RestClient("http://10.00.0.00:0000/api/SIEM/GetUserLoginAttempt");
var request = new RestRequest(Method.POST);
request.AddHeader("cache-control", "no-cache");
request.AddHeader("Connection", "keep-alive");
request.AddHeader("Cookie", "ASP.NET_SessionId=lxeqeq4xhhk113gzrml2siap");
request.AddHeader("Content-Length", "73");
request.AddHeader("Accept-Encoding", "gzip, deflate");
request.AddHeader("Host", "10.00.0.00:0000");
request.AddHeader("Postman-Token", "4c3c87e3-55e8-461c-af0d-67exxxx6cdf8,c3d76ac2-7c96-4526-8499-d0ddb334cb1a");
request.AddHeader("Cache-Control", "no-cache");
request.AddHeader("Accept", "*/*");
request.AddHeader("User-Agent", "PostmanRuntime/7.19.0");
request.AddHeader("Authorization", "Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJyb2xlIjoxxxxmYXVsdCIsImlzcyI6Imh0dHA6Ly9sb2NhbGhvc3Q6ODg5MC8iLCJhdWQiOiI0MTRlMTkyN2EzODg0ZjY4YWJjNzlmNzI4MzgzN2ZkMSIsImV4cCI6MTU3MjAwOTA3OSwibmJmIjoxNTcxOTIyNjc5fQ.pf83SwGr3lbej6fub9TJ8o94ATS0_oSRJpmGOZMMRXs");
request.AddHeader("Content-Type", "application/json");
request.AddParameter("undefined", "{\r\n\"FromDate\":\"2019-06-22T02:37:00\",\r\n\"ToDate\":\"2019-06-26T12:37:00\"\r\n}\r\n", ParameterType.RequestBody);
IRestResponse response = client.Execute(request);
```

Java Code Sample - (Using OK HTTP)

```
OkHttpClient client = new OkHttpClient();
```

```
2
3   MediaType mediaType = MediaType.parse("application/json");
4   RequestBody body = RequestBody.create(mediaType, "{\r\n\"FromDate\":
    \"2019-06-22T02:37:00\",\r\n\"ToDate\":\"2019-06-26T12:37:00\"\r\n}\r\n");
5   Request request = new Request.Builder()
6     .url("http://10.00.0.00:0000/api/SIEM/GetUserLoginAttempt")
7     .post(body)
8     .addHeader("Content-Type", "application/json")
9     .addHeader("Authorization", "Bearer
    eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJyb2xlIjoiRGVmYXVsdCIxxxxcyI6Imh0dH
    A6Ly9sb2NhbGhvc3Q6ODg5MC8iLCJhdWQiOiI0MTRlMTkyN2EzODg0ZjY4YWJjNzlmNzI4Mzgz
    N2ZkMSIsImV4cCI6MTU3MjAwOTA3OSwibmJmIjoxNTcxOTIyNjc5fQ.pf83SwGr3lbej6fub9T
    J8o94ATS0_oSRJpmGOZMMRXs")
10    .addHeader("User-Agent", "PostmanRuntime/7.19.0")
11    .addHeader("Accept", "*/*")
12    .addHeader("Cache-Control", "no-cache")
13    .addHeader("Postman-Token", "4c3c87e3-55e8-461c-
    af0d-67xxxxd6cdf8,8d6e6b83-99a8-4d1d-a8fb-76c1375bcdd5")
14    .addHeader("Host", "10.00.0.00:0000")
15    .addHeader("Accept-Encoding", "gzip, deflate")
16    .addHeader("Content-Length", "73")
17    .addHeader("Cookie", "ASP.NET_SessionId=lxeqeq4xhhk113gzrml2siap")
18    .addHeader("Connection", "keep-alive")
19    .addHeader("cache-control", "no-cache")
20    .build();
21
22  Response response = client.newCall(request).execute();
```

Python Sample Code - (Using http.client Python 3)

```
1   import http.client
2
3   conn = http.client.HTTPConnection("10,00,0,00")
4
5   payload = "{\r\n\"FromDate\":\"2019-06-22T02:37:00\",\r\n\"ToDate\":
    \"2019-06-26T12:37:00\"\r\n}\r\n"
6
7   headers = {
8       'Content-Type': "application/json",
9       'Authorization': "Bearer
    eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJyb2xlIjoiRGVmYXVsdCXXXmlzcyI6Imh0d
    HA6Ly9sb2NhbGhvc3Q6ODg5MC8iLCJhdWQiOiI0MTRlMTkyN2EzODg0ZjY4YWJjNzlmNzI4Mzg
    zN2ZkMSIsImV4cCI6MTU3MjAwOTA3OSwibmJmIjoxNTcxOTIyNjc5fQ.pf83SwGr3lbej6fub9
    TJ8o94ATS0_oSRJpmGOZMMRXs",
10      'User-Agent': "PostmanRuntime/7.19.0",
11      'Accept': "*/*",
12      'Cache-Control': "no-cache",
13      'Postman-Token': "4c3c87e3-55e8-461c-af0d-67e904d6cdf8,29235d3c-
    a913-4366-bac8-8a921d98336c",
14      'Host': "10.00.0.00:0000",
15      'Accept-Encoding': "gzip, deflate",
```

```
    'Content-Length': "73",
    'Cookie': "ASP.NET_SessionId=lxeqeq4xhhk113gzrml2siap",
    'Connection': "keep-alive",
    'cache-control': "no-cache"
    }

conn.request("POST", "api,SIEM,GetUserLoginAttempt", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

Python Sample Code - (Using Python Request)

```
import requests

url = "http://10.00.0.00:0000/api/SIEM/GetUserLoginAttempt"

payload = "{\r\n\"FromDate\":\"2019-06-22T02:37:00\",\r\n\"ToDate\":
\"2019-06-26T12:37:00\"\r\n}\r\n"
headers = {
    'Content-Type': "application/json",
    'Authorization': "Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJyb2xlIjoiRGVmYXVsdCIsIXXXXyI6Imh0d
HA6Ly9sb2NhbGhvc3Q6ODg5MC8iLCJhdWQiOiI0MTRlMTkyN2EzODg0ZjY4YWJjNzlmNzI4Mzg
zN2ZkMSIsImV4cCI6MTU3MjAwOTA3OSwibmJmIjoxNTcxOTIyNjc5fQ.pf83SwGr3lbej6fub9
TJ8o94ATS0_oSRJpmGOZMMRXs",
    'User-Agent': "PostmanRuntime/7.19.0",
    'Accept': "*/*",
    'Cache-Control': "no-cache",
    'Postman-Token': "4c3c87e3-55e8-461c-af0d-67e9xxxxcdf8,7ff86099-
cbd6-4f40-b0a8-1612aeb91913",
    'Host': "10.00.0.00:0000",
    'Accept-Encoding': "gzip, deflate",
    'Content-Length': "73",
    'Cookie': "ASP.NET_SessionId=lxeqeq4xhhk113gzrml2siap",
    'Connection': "keep-alive",
    'cache-control': "no-cache"
    }

response = requests.request("POST", url, data=payload, headers=headers)

print(response.text)
```

5.2 Get Failed Login Attempt

Get Failed Login Attempt details by From Date and To Date API.

Type of Request		POST			
Endpoint		/api/SIEM/GetFailedLoginAttempt			
Available in API Version		All versions			
Release Version		U4			
Pre Endpoint Call		/arconToken			
Request	Params	-	-	-	-
	Headers	Content-Type	application/x-www-form-urlencoded	*Mandatory*	-
		Authorization	Bearer eyJ0[... Removed for brevity ....]INjTU8	*Mandatory*	-
		x-PAM-Version	-	*Not Applicable*	-
	Body	**Request JSON** 1 { 2 "FromDate":"2019-06-22T02:37:00", 3 "ToDate":"2019-06-26T12:37:00" 4 }			

Response	**Response JSON**
```
{
  "Program": "ARCON PAM API",
  "Version": "1.0",
  "DateTime": "25/Oct/2019 12:43:33",
  "Success": true,
  "ErrorCode": null,
  "ErrorMessage": null,
  "Message": "30 Records Found",
  "Result": [
    {
      "EventID": 9101,
      "EventTimeStamp": "25/Oct/2019 12:43:33",
      "UserID": 35,
      "UserName": "ARCOSXXXIN",
      "UserDomain": "AR..XXTH",
      "UserMAC": "100.000.00.000[LAP101]
[7440BB80XXC1][BFEBFBFF000406E3][49SKXQ2]",
      "UserSessionLoginTimeStamp": "06/22/2019
15:58:40",
      "UserSessionLogoutTimeStamp": "06/22/2019
15:58:40",
      "Status": "Authentication Failed"
    }]
}
```
 |
Response Time	4 seconds
Post Endpoint Call	None
Supported Features	NA

5.2.1 Data Types of Request and Response

Type	Parameters	Data Type
Request	FromDate	Date Time ('2012-04-23T18:25:43.511Z' format)
	ToDate	Date Time ('2012-04-23T18:25:43.511Z' format)
Response	EventID	int
	EventTimeStamp	Date Time
	UserID	string
	UserName	string

Type	Parameters	Data Type
	UserDomain	string
	DomainName	string
	UserMAC	string
	UserSessionLoginTimeStamp	Date Time
	UserSessionLogoutTimeStamp	Date Time
	Status	string

5.2.2 Code Blocks for 'SIEM/GetFailedLoginAttempt'

C# Sample Code - (Using RestSharp)

```
var client = new RestClient("http://10.00.0.00:0000/api/SIEM/GetFailedLoginAttempt");
var request = new RestRequest(Method.POST);
request.AddHeader("cache-control", "no-cache");
request.AddHeader("Connection", "keep-alive");
request.AddHeader("Cookie", "ASP.NET_SessionId=lxeqeq4xhhk113gzrml2siap");
request.AddHeader("Content-Length", "73");
request.AddHeader("Accept-Encoding", "gzip, deflate");
request.AddHeader("Host", "10.00.0.00:0000");
request.AddHeader("Postman-Token", "4xxxx37c-5e97-4543-a57f-2e571ec4d54e,0b1bb71b-00f2-492a-8706-7d6d9592f0b8");
request.AddHeader("Cache-Control", "no-cache");
request.AddHeader("Accept", "*/*");
request.AddHeader("User-Agent", "PostmanRuntime/7.19.0");
request.AddHeader("Authorization", "Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJybxxxxjoiRGVmYXVsdCIsImlzcyI6Imh0dHA6Ly9sb2NhbGhvc3Q6ODg5MC8iLCJhdWQiOiI0MTRlMTkyN2EzODg0ZjY4YWJjNzlmNzI4MzgzN2ZkMSIsImV4cCI6MTU3MjAwOTA3OSwibmJmIjoxNTcxOTIyNjc5fQ.pf83SwGr3lbej6fub9TJ8o94ATS0_oSRJpmGOZMMRXs");
request.AddHeader("Content-Type", "application/json");
request.AddParameter("undefined", "{\r\n\"FromDate\":\"2019-06-22T02:37:00\",\r\n\"ToDate\":\"2019-06-26T12:37:00\"\r\n}\r\n", ParameterType.RequestBody);
IRestResponse response = client.Execute(request);
```

Java Code Sample - (Using OK HTTP)

```
OkHttpClient client = new OkHttpClient();
```

```
2
3   MediaType mediaType = MediaType.parse("application/json");
4   RequestBody body = RequestBody.create(mediaType, "{\r\n\"FromDate\":
    \"2019-06-22T02:37:00\",\r\n\"ToDate\":\"2019-06-26T12:37:00\"\r\n}\r\n");
5   Request request = new Request.Builder()
6     .url("http://10.00.0.00:0000/api/SIEM/GetFailedLoginAttempt")
7     .post(body)
8     .addHeader("Content-Type", "application/json")
9     .addHeader("Authorization", "Bearer
    eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJyb2xlIjoiRGVmYXVsdCIsImlxxxx6Imh0d
    HA6Ly9sb2NhbGhvc3Q6ODg5MC8iLCJhdWQiOiI0MTRlMTkyN2EzODg0ZjY4YWJjNzlmNzI4Mzg
    zN2ZkMSIsImV4cCI6MTU3MjAwOTA3OSwibmJmIjoxNTcxOTIyNjc5fQ.pf83SwGr3lbej6fub9
    TJ8o94ATS0_oSRJpmGOZMMRXs")
10    .addHeader("User-Agent", "PostmanRuntime/7.19.0")
11    .addHeader("Accept", "*/*")
12    .addHeader("Cache-Control", "no-cache")
13    .addHeader("Postman-Token", "40a7d37c-5e97-4543-
    a57f-2e57xxx4d54e,e520c2f4-e130-4852-a1ad-6d0ab4b902fc")
14    .addHeader("Host", "10.00.0.00:0000")
15    .addHeader("Accept-Encoding", "gzip, deflate")
16    .addHeader("Content-Length", "73")
17    .addHeader("Cookie", "ASP.NET_SessionId=lxeqeq4xhhk113gzrml2siap")
18    .addHeader("Connection", "keep-alive")
19    .addHeader("cache-control", "no-cache")
20    .build();
21
22  Response response = client.newCall(request).execute();
```

Python Sample Code - (Using http.client Python 3)

```
1   import http.client
2
3   conn = http.client.HTTPConnection("10,00,0,00")
4
5   payload = "{\r\n\"FromDate\":\"2019-06-22T02:37:00\",\r\n\"ToDate\":
    \"2019-06-26T12:37:00\"\r\n}\r\n"
6
7   headers = {
8       'Content-Type': "application/json",
9       'Authorization': "Bearer
    eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJyb2xlIjoiRGxxxxsdCIsImlzcyI6Imh0dH
    A6Ly9sb2NhbGhvc3Q6ODg5MC8iLCJhdWQiOiI0MTRlMTkyN2EzODg0ZjY4YWJjNzlmNzI4Mzgz
    N2ZkMSIsImV4cCI6MTU3MjAwOTA3OSwibmJmIjoxNTcxOTIyNjc5fQ.pf83SwGr3lbej6fub9T
    J8o94ATS0_oSRJpmGOZMMRXs",
10      'User-Agent': "PostmanRuntime/7.19.0",
11      'Accept': "*/*",
12      'Cache-Control': "no-cache",
13      'Postman-Token': "40a7d37c-5e97-4543-a57f-2e5xxxx4d54e,f9ca02f0-
    f594-4293-b36c-2c56696669c1",
14      'Host': "10.00.0.00:0000",
15      'Accept-Encoding': "gzip, deflate",
```

```
    'Content-Length': "73",
    'Cookie': "ASP.NET_SessionId=lxeqeq4xhhk113gzrml2siap",
    'Connection': "keep-alive",
    'cache-control': "no-cache"
    }

conn.request("POST", "api,SIEM,GetFailedLoginAttempt", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

Python Sample Code - (Using Python Request)

```
import requests

url = "http://10.00.0.00:0000/api/SIEM/GetFailedLoginAttempt"

payload = "{\r\n\"FromDate\":\"2019-06-22T02:37:00\",\r\n\"ToDate\":
\"2019-06-26T12:37:00\"\r\n}\r\n"
headers = {
    'Content-Type': "application/json",
    'Authorization': "Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJyb2xlIjoiRGVxxxxsdCIsImlzcyI6Imh0d
HA6Ly9sb2NhbGhvc3Q6ODg5MC8iLCJhdWQiOiI0MTRlMTkyN2EzODg0ZjY4YWJjNzlmNzI4Mzg
zN2ZkMSIsImV4cCI6MTU3MjAwOTA3OSwibmJmIjoxNTcxOTIyNjc5fQ.pf83SwGr3lbej6fub9
TJ8o94ATS0_oSRJpmGOZMMRXs",
    'User-Agent': "PostmanRuntime/7.19.0",
    'Accept': "*/*",
    'Cache-Control': "no-cache",
    'Postman-Token': "40a7d37c-5e97-4543-
a57f-2xxxx1ec4d54e,07797332-6ce2-4c89-8bd8-2c5941b856c6",
    'Host': "10.00.0.00:0000",
    'Accept-Encoding': "gzip, deflate",
    'Content-Length': "73",
    'Cookie': "ASP.NET_SessionId=lxeqeq4xhhk113gzrml2siap",
    'Connection': "keep-alive",
    'cache-control': "no-cache"
    }

response = requests.request("POST", url, data=payload, headers=headers)

print(response.text)
```

5.3 Get Service Access

Get Service Access details by From Date and To Date API.

Type of Request		POST			
Endpoint		/api/SIEM/GetServiceAccess			
Available in API Version		All versions			
Release Version		U4			
Pre Endpoint Call		/arconToken			
Request	Params	-	-	-	-
	Headers	Content-Type	application/x-www-form-urlencoded	*Mandatory*	-
		Authorization	Bearer eyJ0[... Removed for brevity ....]INjTU8	*Mandatory*	-
		x-PAM-Version	-	*Not Applicable*	-
	Body	**Request JSON** 1 { 2 "FromDate":"2019-06-22T02:37:00", 3 "ToDate":"2019-06-26T12:37:00" 4 }			

Response	**Response JSON**
```
{
"Program": "ARCON PAM API",
"Version": "1.0",
"DateTime": "28/Jun/2019 06:52:54",
"Success": true,
"ErrorCode": null,
"ErrorMessage": null,
"Message": "10000 Records Found Out of 18437",
"Result": [
{
"EventID": 9200,
"EventTimeStamp": "28/Jun/2019 06:52:55",
"UserID": 35,
"UserName": "AR..XXADMIN",
"UserMAC": "100.000.0.000[1C659D65AE65]
[BFEBFBFF000206A7][LXRPV0C02312740B371601]
[ACM4.4.1.00000]",
"UserSessionID": 1xx20,
"ServerID": 10449,
"ServerIPAddress": "10.00.0.000",
"ServerUserID": "sxxxt",
"ServerConnectionTypeID": 33,
"ServerConnectionType": "App SQL Developer - Oracle",
"ServerSessionID": 1xx697,
"ServerSessionLoginTimeStamp": "04/24/2019 17:09:08",
"ServerSessionLogoutTimeStamp": "04/24/2019
17:20:51",
"LOBProfile": "DEFAULT LOB 1"
}]
}
```
 |
Response Time	4 seconds
Post Endpoint Call	None
Supported Features	NA

5.3.1 Data Types of Request and Response

Type	Parameters	Data Type
Request	FromDate	Date Time ('2012-04-23T18:25:43.511Z' format)
	ToDate	Date Time ('2012-04-23T18:25:43.511Z' format)

Type	Parameters	Data Type
Response	EventID	int
	EventTimeStamp	Date Time
	UserID	string
	UserName	string
	UserMAC	string
	UserSessionID	string
	ServerID	int
	ServerIPAddress	string
	ServerUserID	string
	ServerConnectionTypeID	int
	ServerConnectionType	string
	ServerSessionID	int
	ServerSessionLoginTimeStamp	Date Time
	ServerSessionLogoutTimeStamp	Date Time
	LOBProfile	string

5.3.2 Code Blocks for 'SIEM/GetServiceAccess'

C# Sample Code - (Using RestSharp)

```
var client = new RestClient("http://10.00.0.00:0000/api/SIEM/GetServiceAccess");
var request = new RestRequest(Method.POST);
request.AddHeader("cache-control", "no-cache");
request.AddHeader("Connection", "keep-alive");
request.AddHeader("Cookie", "ASP.NET_SessionId=lxeqeq4xhhk113gzrml2siap");
request.AddHeader("Content-Length", "73");
request.AddHeader("Accept-Encoding", "gzip, deflate");
request.AddHeader("Host", "10.00.0.00:0000");
request.AddHeader("Postman-Token", "99cb0add-7cba-4018-8bb6-c128fcxxx466,990f3a50-8e59-4143-ab50-2dd90191223e");
request.AddHeader("Cache-Control", "no-cache");
request.AddHeader("Accept", "*/*");
request.AddHeader("User-Agent", "PostmanRuntime/7.19.0");
```

```
request.AddHeader("Authorization", "Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJyb2xlIjoiRGXXXXVsdCIsImlzcyI6Imh0d
HA6Ly9sb2NhbGhvc3Q6ODg5MC8iLCJhdWQiOiI0MTRlMTkyN2EzODg0ZjY4YWJjNzlmNzI4Mzg
zN2ZkMSIsImV4cCI6MTU3MjAwOTA3OSwibmJmIjoxNTcxOTIyNjc5fQ.pf83SwGr3lbej6fub9
TJ8o94ATS0_oSRJpmGOZMMRXs");
request.AddHeader("Content-Type", "application/json");
request.AddParameter("undefined", "{\r\n\"FromDate\":
\"2011-01-26T02:37:00\",\r\n\"ToDate\":\"2019-06-26T12:37:00\"\r\n}\r\n",
ParameterType.RequestBody);
IRestResponse response = client.Execute(request);
```

Java Code Sample - (Using OK HTTP)

```
OkHttpClient client = new OkHttpClient();

MediaType mediaType = MediaType.parse("application/json");
RequestBody body = RequestBody.create(mediaType, "{\r\n\"FromDate\":
\"2011-01-26T02:37:00\",\r\n\"ToDate\":\"2019-06-26T12:37:00\"\r\n}\r\n");
Request request = new Request.Builder()
  .url("http://00.00.0.00:0000/api/SIEM/GetServiceAccess")
  .post(body)
  .addHeader("Content-Type", "application/json")
  .addHeader("Authorization", "Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJyb2xlIjoiRGVmYXVsdCIsImlzcyI6Imh0d
HA6Ly9sb2NhbGhvc3Q6ODg5MC8iLCJhdWQiOiI0MTRlMTkyN2EzODg0ZjY4YWJjNzlmNzI4Mzg
zN2ZkMSIsImV4cCI6MTU3MjAwOTA3OSwibmJmIjoxNTcxOTIyNjc5fQ.pf83SwGr3lbej6fub9
TJ8o94ATS0_oSRJpmGOZMMRXs")
  .addHeader("User-Agent", "PostmanRuntime/7.19.0")
  .addHeader("Accept", "*/*")
  .addHeader("Cache-Control", "no-cache")
  .addHeader("Postman-Token", "99cb0add-7cba-4018-8bb6-
c128fc014466,ce1ad4a9-5f58-40cc-bffd-6ed63b086009")
  .addHeader("Host", "10.00.0.00:0000")
  .addHeader("Accept-Encoding", "gzip, deflate")
  .addHeader("Content-Length", "73")
  .addHeader("Cookie", "ASP.NET_SessionId=lxeqeq4xhhk113gzrml2siap")
  .addHeader("Connection", "keep-alive")
  .addHeader("cache-control", "no-cache")
  .build();

Response response = client.newCall(request).execute();
```

Python Sample Code - (Using http.client Python 3)

```
import http.client

conn = http.client.HTTPConnection("10,00,0,00")

```

```
payload = "{\r\n\"FromDate\":\"2011-01-26T02:37:00\",\r\n\"ToDate\":
\"2019-06-26T12:37:00\"\r\n}\r\n"

headers = {
    'Content-Type': "application/json",
    'Authorization': "Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJyb2xlIjoiRxxxYXVsdCIsImlzcyI6Imh0d
HA6Ly9sb2NhbGhvc3Q6ODg5MC8iLCJhdWQiOiI0MTRlMTkyN2EzODg0ZjY4YWJjNzlmNzI4Mzg
zN2ZkMSIsImV4cCI6MTU3MjAwOTA3OSwibmJmIjoxNTcxOTIyNjc5fQ.pf83SwGr3lbej6fub9
TJ8o94ATS0_oSRJpmGOZMMRXs",
    'User-Agent': "PostmanRuntime/7.19.0",
    'Accept': "*/*",
    'Cache-Control': "no-cache",
    'Postman-Token': "99cb0add-7cba-4018-8bb6-c1xxxc014466,67e46e8b-
b3e3-42b7-9327-3cc53c1f356c",
    'Host': "10.00.0.00:0000",
    'Accept-Encoding': "gzip, deflate",
    'Content-Length': "73",
    'Cookie': "ASP.NET_SessionId=lxeqeq4xhhk113gzrml2siap",
    'Connection': "keep-alive",
    'cache-control': "no-cache"
    }

conn.request("POST", "api,SIEM,GetServiceAccess", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

Python Sample Code - (Using Python Request)

```
import requests

url = "http://10.00.0.00:0000/api/SIEM/GetServiceAccess"

payload = "{\r\n\"FromDate\":\"2011-01-26T02:37:00\",\r\n\"ToDate\":
\"2019-06-26T12:37:00\"\r\n}\r\n"
headers = {
    'Content-Type': "application/json",
    'Authorization': "Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJyb2xlIjoiRGVmYXVsxxxsImlzcyI6Imh0d
HA6Ly9sb2NhbGhvc3Q6ODg5MC8iLCJhdWQiOiI0MTRlMTkyN2EzODg0ZjY4YWJjNzlmNzI4Mzg
zN2ZkMSIsImV4cCI6MTU3MjAwOTA3OSwibmJmIjoxNTcxOTIyNjc5fQ.pf83SwGr3lbej6fub9
TJ8o94ATS0_oSRJpmGOZMMRXs",
    'User-Agent': "PostmanRuntime/7.19.0",
    'Accept': "*/*",
    'Cache-Control': "no-cache",
    'Postman-Token': "99cb0add-7cba-4018-8bb6-
c128xxx14466,4c047fde-3dcf-450f-b66a-d50345f7fb9c",
    'Host': "10.00.0.00:0000",
```

```
    'Accept-Encoding': "gzip, deflate",
    'Content-Length': "73",
    'Cookie': "ASP.NET_SessionId=lxeqeq4xhhk113gzrml2siap",
    'Connection': "keep-alive",
    'cache-control': "no-cache"
    }

response = requests.request("POST", url, data=payload, headers=headers)

print(response.text)
```

5.4 Get Service Command

Get Service Command details by From Date and To Date API.

Type of Request		POST			
Endpoint		/api/SIEM/GetServiceCommand			
Available in API Version		All versions			
Release Version		U4			
Pre Endpoint Call		/arconToken			
Request	Params	-	-	-	-
	Headers	Content-Type	application/x-www-form-urlencoded	*Mandatory*	-
		Authorization	Bearer eyJ0[... Removed for brevity ....]INjTU8	*Mandatory*	-
		x-PAM-Version	-	*Not Applicable*	-

	Body	**Request JSON**
		{
		"FromDate":"2019-06-22T02:37:00",
		"ToDate":"2019-06-26T12:37:00"
		}			

Response	**Response JSON**
```
{
  "Program": "ARCON PAM API",
  "Version": "1.0",
  "DateTime": "25/Oct/2019 02:34:47",
  "Success": true,
  "ErrorCode": null,
  "ErrorMessage": null,
  "Message": "4 Records Found",
  "Result": [
    {
      "EventID": 9300,
      "EventTimeStamp": "25/Oct/2019 02:34:47",
      "UserID": 35,
      "UserName": "ARCOSADMIN",
      "UserMAC": "10.10.2.84[DSK016][E0D55E09B283][BFEBFBFF000906E9][Defaultstring][ACM4.8.5.0]",
      "UserSessionID": 6854,
      "ServerID": 114150,
      "ServerIPAddress": "10.10.0.000",
      "ServerUserID": "sxxxx_sshk",
      "ServerConnectionTypeID": 140,
      "ServerConnectionType": "App WinSCP",
      "ServerSessionID": 2x94,
      "Command": "~$ date",
      "CommandResponse": "Critical Command with Approval : Workflow Not Set",
      "CommandTimeStamp": "05/07/2019 11:45:01",
      "LOBProfile": "DEFAULT LOB 1"
    }]
}
```
 |
Response Time	5 seconds
Post Endpoint Call	None
Supported Features	NA

5.4.1 Data Types of Request and Response

Type	Parameters	Data Type
Request	FromDate	Date Time ('2012-04-23T18:25:43.511Z' format)
	ToDate	Date Time ('2012-04-23T18:25:43.511Z' format)

Type	Parameters	Data Type
Response	EventID	int
	EventTimeStamp	Date Time
	UserID	string
	UserName	string
	UserMAC	string
	UserSessionID	string
	ServerID	int
	ServerIPAddress	string
	ServerUserID	string
	ServerConnectionTypeID	int
	ServerConnectionType	string
	ServerSessionID	int
	Command	Date Time
	CommandResponse	string
	CommandTimeStamp	string
	LOBProfile	string

5.4.2 Code Blocks for 'SIEM/GetServiceCommand'

C# Sample Code - (Using RestSharp)

```
var client = new RestClient("http://10.00.0.00:0000/api/SIEM/GetServiceCommand");
var request = new RestRequest(Method.POST);
request.AddHeader("cache-control", "no-cache");
request.AddHeader("Connection", "keep-alive");
request.AddHeader("Cookie", "ASP.NET_SessionId=lxeqeq4xhhk113gzrml2siap");
request.AddHeader("Content-Length", "73");
request.AddHeader("Accept-Encoding", "gzip, deflate");
request.AddHeader("Host", "10.00.0.00:0000");
request.AddHeader("Postman-Token", "cfe7b089-3c13-404e-abe2-5dxxxx8d0460,12a35b9d-aae9-473b-928a-f3908a06355f");
request.AddHeader("Cache-Control", "no-cache");
```

```
request.AddHeader("Accept", "*/*");
request.AddHeader("User-Agent", "PostmanRuntime/7.19.0");
request.AddHeader("Authorization", "Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJyb2xlIjoiRGVmYXVsdCIsImlzxxxx6Imh0
dHA6Ly9sb2NhbGhvc3Q6ODg5MC8iLCJhdWQiOiI0MTRlMTkyN2EzODg0ZjY4YWJjNzlmNzI4Mz
gzN2ZkMSIsImV4cCI6MTU3MjAwOTA3OSwibmJmIjoxNTcxOTIyNjc5fQ.pf83SwGr3lbej6fub
9TJ8o94ATS0_oSRJpmGOZMMRXs");
request.AddHeader("Content-Type", "application/json");
request.AddParameter("undefined", "{\r\n\"FromDate\":
\"2019-01-26T02:37:00\",\r\n\"ToDate\":\"2019-06-26T12:37:00\"\r\n}\r\n",
ParameterType.RequestBody);
IRestResponse response = client.Execute(request);
```

Java Code Sample - (Using OK HTTP)

```
OkHttpClient client = new OkHttpClient();

MediaType mediaType = MediaType.parse("application/json");
RequestBody body = RequestBody.create(mediaType, "{\r\n\"FromDate\":
\"2019-01-26T02:37:00\",\r\n\"ToDate\":\"2019-06-26T12:37:00\"\r\n}\r\n");
Request request = new Request.Builder()
  .url("http://10.00.0.00:0000/api/SIEM/GetServiceCommand")
  .post(body)
  .addHeader("Content-Type", "application/json")
  .addHeader("Authorization", "Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJyb2xlIjoiRGVmYXVsdCIsImlzcyI6Imh0d
HA6Ly9sb2NhbGhvc3Q6ODg5MC8iLCJhdWQiOiI0MTRlMTkyN2EzODg0ZjY4YWJjNzlmNzI4Mzg
zN2ZkMSIsImV4cCI6MTU3MjAwOTA3OSwibmJmIjoxNTcxOTIyNjc5fQ.pf83SwGr3lbej6fub9
TJ8o94ATS0_oSRJpmGOZMMRXs")
  .addHeader("User-Agent", "PostmanRuntime/7.19.0")
  .addHeader("Accept", "*/*")
  .addHeader("Cache-Control", "no-cache")
  .addHeader("Postman-Token", "cfe7b089-3c13-404e-
abe2-5d02c28d0460,005fbc1e-1489-4ea9-b06c-113b86a39bf8")
  .addHeader("Host", "10.00.0.00:0000")
  .addHeader("Accept-Encoding", "gzip, deflate")
  .addHeader("Content-Length", "73")
  .addHeader("Cookie", "ASP.NET_SessionId=lxeqeq4xhhk113gzrml2siap")
  .addHeader("Connection", "keep-alive")
  .addHeader("cache-control", "no-cache")
  .build();

Response response = client.newCall(request).execute();
```

Python Sample Code - (Using http.client Python 3)

```
import http.client

```

```
conn = http.client.HTTPConnection("10,00,0,00")

payload = "{\r\n\"FromDate\":\"2019-01-26T02:37:00\",\r\n\"ToDate\":
\"2019-06-26T12:37:00\"\r\n}\r\n"

headers = {
    'Content-Type': "application/json",
    'Authorization': "Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJyb2xlIjoiRGVmYXVsdCIsImlzcyI6Imh0d
HA6Ly9sb2NhbGhvc3Q6ODg5MC8iLCJhdWQiOiI0MTRlMTkyN2EzODg0ZjY4YWJjNzlmNzI4Mzg
zN2ZkMSIsImV4cCI6MTU3MjAwOTA3OSwibmJmIjoxNTcxOTIyNjc5fQ.pf83SwGr3lbej6fub9
TJ8o94ATS0_oSRJpmGOZMMRXs",
    'User-Agent': "PostmanRuntime/7.19.0",
    'Accept': "*/*",
    'Cache-Control': "no-cache",
    'Postman-Token': "cfe7b089-3c13-404e-
abe2-5dxxxx8d0460,e60ff0ad-706b-4dbb-b110-b3ad2ad56652",
    'Host': "10.00.0.00:0000",
    'Accept-Encoding': "gzip, deflate",
    'Content-Length': "73",
    'Cookie': "ASP.NET_SessionId=lxeqeq4xhhk113gzrml2siap",
    'Connection': "keep-alive",
    'cache-control': "no-cache"
    }

conn.request("POST", "api,SIEM,GetServiceCommand", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

Python Sample Code - (Using Python Request)

```
import requests

url = "http://10.00.0.00:0000/api/SIEM/GetServiceCommand"

payload = "{\r\n\"FromDate\":\"2019-01-26T02:37:00\",\r\n\"ToDate\":
\"2019-06-26T12:37:00\"\r\n}\r\n"
headers = {
    'Content-Type': "application/json",
    'Authorization': "Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJyb2xlIjoiRGVmYXVsdCIsImlzcyI6Imh0d
HA6Ly9sb2NhbGhvc3Q6ODg5MC8iLCJhdWQiOiI0MTRlMTkyN2EzODg0ZjY4YWJjNzlmNzI4Mzg
zN2ZkMSIsImV4cCI6MTU3MjAwOTA3OSwibmJmIjoxNTcxOTIyNjc5fQ.pf83SwGr3lbej6fub9
TJ8o94ATS0_oSRJpmGOZMMRXs",
    'User-Agent': "PostmanRuntime/7.19.0",
    'Accept': "*/*",
    'Cache-Control': "no-cache",
```

```
    'Postman-Token': "cfe7b089-3c13-404e-abe2-5d02c28d0460,0e2f2038-0eae-42d7-8d53-7e32d128c5a5",
    'Host': "10.00.0.00:0000",
    'Accept-Encoding': "gzip, deflate",
    'Content-Length': "73",
    'Cookie': "ASP.NET_SessionId=lxeqeq4xhhk113gzrml2siap",
    'Connection': "keep-alive",
    'cache-control': "no-cache"
    }

response = requests.request("POST", url, data=payload, headers=headers)

print(response.text)
```

5.5 Get Password View

Get Service Password View details by From Date and To Date API.

Type of Request		POST			
Endpoint		/api/SIEM/GetPasswordView			
Available in API Version		All versions			
Release Version		U4			
Pre Endpoint Call		/arconToken			
Request	Params	-	-	-	-
	Headers	Content-Type	application/x-www-form-urlencoded	*Mandatory*	-
		Authorization	Bearer eyJ0[... Removed for brevity ....]INjTU8	*Mandatory*	-
		x-PAM-Version	-	*Not Applicable*	-

	Body	**Request JSON**
		{
		"FromDate":"2019-06-22T02:37:00",
		"ToDate":"2019-06-26T12:37:00"
		}			

Response	**Response JSON**
```
{
"Program": "ARCON PAM API",
"Version": "1.0",
"DateTime": "28/Jun/2019 08:07:55",
"Success": true,
"ErrorCode": null,
"ErrorMessage": null,
"Message": "244 Records Found",
"Result": [
{
"EventID": 9400,
"EventTimeStamp": "28/Jun/2019 08:08:03",
"UserName": "MXXX.XXXXXXX",
"UserDomain": "ANB..XXLDC",
"ServerID": 3x002,
"ServerIPAddress": "10.00.0.00",
"ServerUserID": "mxxxx",
"ServerConnectionTypeID": 7,
"ServerConnectionType": "SSH LINUX",
"ViewOnTimeStamp": "06/06/2018 11:41:01",
"ViewForHours": 1,
"ViewDescription":
"ddddddddddddhhhhhhhhhhhhhhhhhhhh",
"AuthbyUserName1": "LAWRENCE",
"AuthbyUserDomain1": "ARCOSAUTH",
"AuthbyUserName2": "+",
"AuthbyUserDomain2": "+",
"LOBProfile": "DEFAULT LOB XX"
}
```
 |
Response Time	151 milliseconds
Post Endpoint Call	None
Supported Features	NA

5.5.1 Data Types of Request and Response

Type	Parameters	Data Type
Request	FromDate	Date Time ('2012-04-23T18:25:43.511Z' format)
	ToDate	Date Time ('2012-04-23T18:25:43.511Z' format)
Response	EventID	int

Type	Parameters	Data Type
	EventTimeStamp	Date Time
	UserName	string
	ServerID	int
	ServerIPAddress	string
	ServerUserID	string
	ServerConnectionTypeID	int
	ServerConnectionType	string
	ViewOnTimeStamp	Date Time
	ViewForHours	int
	ViewDescription	string
	AuthbyUserName1	string
	AuthbyUserDomain1	string
	AuthbyUserName2	string
	AuthbyUserDomain2	string
	LOBProfile	string

5.5.2 Code Blocks for 'SIEM/GetPasswordView'

C# Sample Code - (Using RestSharp)

```
var client = new RestClient("http://10.00.0.00:0000/api/SIEM/
GetPasswordView");
var request = new RestRequest(Method.POST);
request.AddHeader("cache-control", "no-cache");
request.AddHeader("Connection", "keep-alive");
request.AddHeader("Cookie", "ASP.NET_SessionId=lxeqeq4xhhk113gzrml2siap");
request.AddHeader("Content-Length", "73");
request.AddHeader("Accept-Encoding", "gzip, deflate");
request.AddHeader("Host", "10.00.0.00:0000");
request.AddHeader("Postman-Token", "f7b0f06a-7288-4458-8aa3-
f648xxxc1b37,f5d5be24-7c56-486b-aeb1-99bf588cbd64");
request.AddHeader("Cache-Control", "no-cache");
request.AddHeader("Accept", "*/*");
request.AddHeader("User-Agent", "PostmanRuntime/7.19.0");
```

```
request.AddHeader("Authorization", "Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJyb2xlIjoiRGVmYXVsdCIsImlzcXXXXh0dH
A6Ly9sb2NhbGhvc3Q6ODg5MC8iLCJhdWQiOiI0MTRlMTkyN2EzODg0ZjY4YWJjNzlmNzI4Mzgz
N2ZkMSIsImV4cCI6MTU3MjAwOTA3OSwibmJmIjoxNTcxOTIyNjc5fQ.pf83SwGr3lbej6fub9T
J8o94ATS0_oSRJpmGOZMMRXs");
request.AddHeader("Content-Type", "application/json");
request.AddParameter("undefined", "{\r\n\"FromDate\":
\"2018-01-26T02:37:00\",\r\n\"ToDate\":\"2019-06-26T12:37:00\"\r\n}\r\n",
ParameterType.RequestBody);
IRestResponse response = client.Execute(request);
```

Java Code Sample - (Using OK HTTP)

```
OkHttpClient client = new OkHttpClient();

MediaType mediaType = MediaType.parse("application/json");
RequestBody body = RequestBody.create(mediaType, "{\r\n\"FromDate\":
\"2018-01-26T02:37:00\",\r\n\"ToDate\":\"2019-06-26T12:37:00\"\r\n}\r\n");
Request request = new Request.Builder()
  .url("http://10.00.0.00:0000/api/SIEM/GetPasswordView")
  .post(body)
  .addHeader("Content-Type", "application/json")
  .addHeader("Authorization", "Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJyb2xlIjoiRGVxxxsdCIsImlzcyI6Imh0dH
A6Ly9sb2NhbGhvc3Q6ODg5MC8iLCJhdWQiOiI0MTRlMTkyN2EzODg0ZjY4YWJjNzlmNzI4Mzgz
N2ZkMSIsImV4cCI6MTU3MjAwOTA3OSwibmJmIjoxNTcxOTIyNjc5fQ.pf83SwGr3lbej6fub9T
J8o94ATS0_oSRJpmGOZMMRXs")
  .addHeader("User-Agent", "PostmanRuntime/7.19.0")
  .addHeader("Accept", "*/*")
  .addHeader("Cache-Control", "no-cache")
  .addHeader("Postman-Token", "f7b0f06a-7288-4458-8aa3-
f648xxxx1b37,5506b635-a26a-4ece-a7fa-34a3b7633c4a")
  .addHeader("Host", "10.00.0.00:0000")
  .addHeader("Accept-Encoding", "gzip, deflate")
  .addHeader("Content-Length", "73")
  .addHeader("Cookie", "ASP.NET_SessionId=lxeqeq4xhhk113gzrml2siap")
  .addHeader("Connection", "keep-alive")
  .addHeader("cache-control", "no-cache")
  .build();

Response response = client.newCall(request).execute();
```

Python Sample Code - (Using http.client Python 3)

```
import http.client

conn = http.client.HTTPConnection("10,00,0,00")

```

```
payload = "{\r\n\"FromDate\":\"2018-01-26T02:37:00\",\r\n\"ToDate\":
\"2019-06-26T12:37:00\"\r\n}\r\n"

headers = {
    'Content-Type': "application/json",
    'Authorization': "Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJyb2xlIjoiRGVmYXxxxCIsImlzcyI6Imh0d
HA6Ly9sb2NhbGhvc3Q6ODg5MC8iLCJhdWQiOiI0MTRlMTkyN2EzODg0ZjY4YWJjNzlmNzI4Mzg
zN2ZkMSIsImV4cCI6MTU3MjAwOTA3OSwibmJmIjoxNTcxOTIyNjc5fQ.pf83SwGr3lbej6fub9
TJ8o94ATS0_oSRJpmGOZMMRXs",
    'User-Agent': "PostmanRuntime/7.19.0",
    'Accept': "*/*",
    'Cache-Control': "no-cache",
    'Postman-Token': "f7b0f06a-7288-4458-8aa3-f648df3c1b37,a910656a-
bb6f-43d5-a2b6-8b6181d6e119",
    'Host': "10.00.0.00:0000",
    'Accept-Encoding': "gzip, deflate",
    'Content-Length': "73",
    'Cookie': "ASP.NET_SessionId=lxeqeq4xhhk113gzrml2siap",
    'Connection': "keep-alive",
    'cache-control': "no-cache"
    }

conn.request("POST", "api,SIEM,GetPasswordView", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

Python Sample Code - (Using Python Request)

```
import requests

url = "http://10.00.0.00:00000/api/SIEM/GetPasswordView"

payload = "{\r\n\"FromDate\":\"2018-01-26T02:37:00\",\r\n\"ToDate\":
\"2019-06-26T12:37:00\"\r\n}\r\n"
headers = {
    'Content-Type': "application/json",
    'Authorization': "Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJyb2xlIjoiRGVmYXVsdxxmlzcyI6Imh0dHA
6Ly9sb2NhbGhvc3Q6ODg5MC8iLCJhdWQiOiI0MTRlMTkyN2EzODg0ZjY4YWJjNzlmNzI4MzgzN
2ZkMSIsImV4cCI6MTU3MjAwOTA3OSwibmJmIjoxNTcxOTIyNjc5fQ.pf83SwGr3lbej6fub9TJ
8o94ATS0_oSRJpmGOZMMRXs",
    'User-Agent': "PostmanRuntime/7.19.0",
    'Accept': "*/*",
    'Cache-Control': "no-cache",
    'Postman-Token': "f7b0f06a-7288-4458-8aa3-
f648df3c1b37,56322f4c-18bb-4735-b9df-274fed4f8cad",
    'Host': "10.00.0.00:00000",
```

```
    'Accept-Encoding': "gzip, deflate",
    'Content-Length': "73",
    'Cookie': "ASP.NET_SessionId=lxeqeq4xhhk113gzrml2siap",
    'Connection': "keep-alive",
    'cache-control': "no-cache"
    }

response = requests.request("POST", url, data=payload, headers=headers)

print(response.text)
```

5.6 Get Service PasswordChange

Get Service Password View details by From Date and To Date API.

Type of Request		POST			
Endpoint		/api/SIEM/GetServicePasswordChange			
Available in API Version		All versions			
Release Version		U4			
Pre Endpoint Call		/arconToken			
Request	Params	-	-	-	-
	Headers	Content-Type	application/x-www-form-urlencoded	*Mandatory*	-
		Authorization	Bearer eyJ0[... Removed for brevity ....]INjTU8	*Mandatory*	-
		x-PAM-Version	-	*Not Applicable*	-
	Body	**Request JSON** `{ "FromDate":"2019-06-22T02:37:00", "ToDate":"2019-06-26T12:37:00" }`			

Response	**Response JSON**
```
{
"Program": "ARCON PAM API",
"Version": "1.0",
"DateTime": "28/Jun/2019 08:33:36",
"Success": true,
"ErrorCode": null,
"ErrorMessage": null,
"Message": "2709 Records Found",
"Result": [
{
"EventID": 9500,
"EventTimeStamp": "28/Jun/2019 08:33:37",
"LOBProfile": "DEFAULT LOB XX",
"Status": "Failed",
"ServerID": 11XX69,
"ServerIPAddress": "10.00.0.00",
"ServerUserID": "sha..xxh",
"ServerConnectionTypeID": 7,
"ServerConnectionType": "SSH LINUX",
"PasswordChangedByUserName": "ARCOSADMIN",
"IsPasswordChangedManually": "NO",
"PasswordChangedOnTimeStamp": "06/05/2019 12:04:06"
}]
}
```
 |
Response Time	500 milliseconds
Post Endpoint Call	None
Supported Features	NA

5.6.1 Data Types of Request and Response

Type	Parameters	Data Type
Request	FromDate	Date Time ('2012-04-23T18:25:43.511Z' format)
	ToDate	Date Time ('2012-04-23T18:25:43.511Z' format)
Response	EventID	int
	EventTimeStamp	Date Time
	LOBProfile	string
	Status	string

Type	Parameters	Data Type
	ServerID	int
	ServerUserID	string
	ServerIPAddress	string
	ServerUserID	string
	ServerConnectionTypeID	int
	ServerConnectionType	string
	PasswordChangedByUserName	string
	IsPasswordChangedManually	string
	PasswordChangedOnTimeStamp	Date Time

5.6.2 Code Blocks for 'SIEM/GetServicePasswordChange'

C# Sample Code - (Using RestSharp)

```
var client = new RestClient("http://10.00.0.00:00000/api/SIEM/GetServicePasswordChange");
var request = new RestRequest(Method.POST);
request.AddHeader("cache-control", "no-cache");
request.AddHeader("Connection", "keep-alive");
request.AddHeader("Cookie", "ASP.NET_SessionId=lxeqeq4xhhk113gzrml2siap");
request.AddHeader("Content-Length", "73");
request.AddHeader("Accept-Encoding", "gzip, deflate");
request.AddHeader("Host", "10.00.0.00:0000");
request.AddHeader("Postman-Token", "4ea52c50-14c7-4811-aa00-59c99d564575,06837443-45c7-48bf-ba1f-8e856192e976");
request.AddHeader("Cache-Control", "no-cache");
request.AddHeader("Accept", "*/*");
request.AddHeader("User-Agent", "PostmanRuntime/7.19.0");
request.AddHeader("Authorization", "Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJyb2xlIjoiRGVmYXVsdCIsImlzcyI6Imh0dHA6Ly9sb2NhbGhvc3Q6ODg5MC8iLCJhdWQiOiI0MTRlMTkyN2EzODg0ZjY4YWJjNzlmNzI4MzgzN2ZkMSIsImV4cCI6MTU3MjAwOTA3OSwibmJmIjoxNTcxOTIyNjc5fQ.pf83SwGr3lbej6fub9TJ8o94ATS0_oSRJpmGOZMMRXs");
request.AddHeader("Content-Type", "application/json");
request.AddParameter("undefined", "{\r\n\"FromDate\":\"2015-01-26T02:37:00\",\r\n\"ToDate\":\"2019-06-26T12:37:00\"\r\n}\r\n", ParameterType.RequestBody);
IRestResponse response = client.Execute(request);
```

Java Code Sample - (Using OK HTTP)

```
OkHttpClient client = new OkHttpClient();

MediaType mediaType = MediaType.parse("application/json");
RequestBody body = RequestBody.create(mediaType, "{\r\n\"FromDate\":
\"2015-01-26T02:37:00\",\r\n\"ToDate\":\"2019-06-26T12:37:00\"\r\n}\r\n");
Request request = new Request.Builder()
  .url("http://10.00.0.00:0000/api/SIEM/GetServicePasswordChange")
  .post(body)
  .addHeader("Content-Type", "application/json")
  .addHeader("Authorization", "Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJyb2xlIjoiRGVmYXVsdCIsImlzcyI6Imh0d
HA6Ly9sb2NhbGhvc3Q6ODg5MC8iLCJhdWQiOiI0MTRlMTkyN2EzODg0ZjY4YWJjNzlmNzI4Mzg
zN2ZkMSIsImV4cCI6MTU3MjAwOTA3OSwibmJmIjoxNTcxOTIyNjc5fQ.pf83SwGr3lbej6fub9
TJ8o94ATS0_oSRJpmGOZMMRXs")
  .addHeader("User-Agent", "PostmanRuntime/7.19.0")
  .addHeader("Accept", "*/*")
  .addHeader("Cache-Control", "no-cache")
  .addHeader("Postman-Token", "4ea52c50-14c7-4811-
aa00-59c99d564575,14bf2cba-b79f-44e5-b6ca-4d8dfbd6cd53")
  .addHeader("Host", "10.00.0.00:0000")
  .addHeader("Accept-Encoding", "gzip, deflate")
  .addHeader("Content-Length", "73")
  .addHeader("Cookie", "ASP.NET_SessionId=lxeqeq4xhhk113gzrml2siap")
  .addHeader("Connection", "keep-alive")
  .addHeader("cache-control", "no-cache")
  .build();

Response response = client.newCall(request).execute();
```

Python Sample Code - (Using http.client Python 3)

```
import http.client

conn = http.client.HTTPConnection("10,00,0,00")

payload = "{\r\n\"FromDate\":\"2015-01-26T02:37:00\",\r\n\"ToDate\":
\"2019-06-26T12:37:00\"\r\n}\r\n"

headers = {
    'Content-Type': "application/json",
    'Authorization': "Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJyb2xlIjoiRGVmYXVsdCIsImlzcyI6Imh0d
HA6Ly9sb2NhbGhvc3Q6ODg5MC8iLCJhdWQiOiI0MTRlMTkyN2EzODg0ZjY4YWJjNzlmNzI4Mzg
zN2ZkMSIsImV4cCI6MTU3MjAwOTA3OSwibmJmIjoxNTcxOTIyNjc5fQ.pf83SwGr3lbej6fub9
TJ8o94ATS0_oSRJpmGOZMMRXs",
    'User-Agent': "PostmanRuntime/7.19.0",
    'Accept': "*/*",
```

```
    'Cache-Control': "no-cache",
    'Postman-Token': "4ea52c50-14c7-4811-
aa00-59c99d564575,414d2c1f-18a1-4fb4-ad57-616ca5bae2f4",
    'Host': "10.00.0.00:0000",
    'Accept-Encoding': "gzip, deflate",
    'Content-Length': "73",
    'Cookie': "ASP.NET_SessionId=lxeqeq4xhhk113gzrml2siap",
    'Connection': "keep-alive",
    'cache-control': "no-cache"
    }

conn.request("POST", "api,SIEM,GetServicePasswordChange", payload,
headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

Python Sample Code - (Using Python Request)

```
import requests

url = "http://10.00.0.00:0000/api/SIEM/GetServicePasswordChange"

payload = "{\r\n\"FromDate\":\"2015-01-26T02:37:00\",\r\n\"ToDate\":
\"2019-06-26T12:37:00\"\r\n}\r\n"
headers = {
    'Content-Type': "application/json",
    'Authorization': "Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJyb2xlIjoiRGVmYXVsdCIsImlzcyI6Imh0d
HA6Ly9sb2NhbGhvc3Q6ODg5MC8iLCJhdWQiOiI0MTRlMTkyN2EzODg0ZjY4YWJjNzlmNzI4Mzg
zN2ZkMSIsImV4cCI6MTU3MjAwOTA3OSwibmJmIjoxNTcxOTIyNjc5fQ.pf83SwGr3lbej6fub9
TJ8o94ATS0_oSRJpmGOZMMRXs",
    'User-Agent': "PostmanRuntime/7.19.0",
    'Accept': "*/*",
    'Cache-Control': "no-cache",
    'Postman-Token': "4ea52c50-14c7-4811-
aa00-59c99d564575,ab47bb9e-8987-4b15-a930-9dd59e87fe69",
    'Host': "10.00.0.00:0000",
    'Accept-Encoding': "gzip, deflate",
    'Content-Length': "73",
    'Cookie': "ASP.NET_SessionId=lxeqeq4xhhkxx3gzrml2siap",
    'Connection': "keep-alive",
    'cache-control': "no-cache"
    }

response = requests.request("POST", url, data=payload, headers=headers)

print(response.text)
```

5.7 Get Service PasswordChange

Get Service Password View details by From Date and To Date API.

Type of Request		POST			
Endpoint		/api/SIEM/GetServicePasswordEnvelopePrint			
Available in API Version		All versions			
Release Version		U4			
Pre Endpoint Call		/arconToken			
Request	Params	-	-	-	-
	Headers	Content-Type	application/x-www-form-urlencoded	*Mandatory*	-
		Authorization	Bearer eyJ0[... Removed for brevity ....]INjTU8	*Mandatory*	-
		x-PAM-Version	-	*Not Applicable*	-
	Body	**Request JSON** 1 { 2 "FromDate":"2019-06-22T02:37:00", 3 "ToDate":"2019-06-26T12:37:00" 4 }			

Response	**Response JSON**
```
{
"Program": "ARCON PAM API",
"Version": "1.0",
"DateTime": "01/Jul/2019 12:21:17",
"Success": true,
"ErrorCode": null,
"ErrorMessage": null,
"Message": "173 Records Found",
"Result": [
{
"EventID": 9600,
"EventTimeStamp": "01/Jul/2019 12:21:17",
"LOBProfile": "DEFAULT LOB XX",
"ServerID": 7xx1,
"ServerIPAddress": "100.000.0.000",
"ServerUserID": "Administrator",
"ServerConnectionTypeID": 1,
"ServerConnectionType": "Windows RDP",
"PrintedByUserName": "ARCOSADMIN",
"PrintedOn": "04/26/2019 11:05:30",
"VerifiedByUserName1": "AN....XXX",
"VerifiedByUserName2": "AN....XXX",
"PrintingStatus": "First Reprint"
}]
}
```
 |
Response Time	500 milliseconds
Post Endpoint Call	None
Supported Features	NA

5.7.1 Data Types of Request and Response

Type	Parameters	Data Type
Request	FromDate	Date Time ('2012-04-23T18:25:43.511Z' format)
	ToDate	Date Time ('2012-04-23T18:25:43.511Z' format)
Response	EventID	int
	EventTimeStamp	Date Time
	LOBProfile	string

Type	Parameters	Data Type
	ServerID	int
	ServerIPAddress	string
	ServerUserID	string
	ServerConnectionTypeID	string
	ServerConnectionType	string
	PrintedByUserName	string
	PrintedOn	Date Time
	VerifiedByUserName1	string
	VerifiedByUserName2	string
	PrintingStatus	string

5.7.2 Code Blocks for 'SIEM/GetServicePasswordEnvelopePrint'

C# Sample Code - (Using RestSharp)

```
var client = new RestClient("http://10.00.0.00:0000/api/SIEM/GetServicePasswordEnvelopePrint");
var request = new RestRequest(Method.POST);
request.AddHeader("cache-control", "no-cache");
request.AddHeader("Connection", "keep-alive");
request.AddHeader("Cookie", "ASP.NET_SessionId=lxeqeq4xhhk113gzrml2siap");
request.AddHeader("Content-Length", "73");
request.AddHeader("Accept-Encoding", "gzip, deflate");
request.AddHeader("Host", "10.00.0.00:0000");
request.AddHeader("Postman-Token", "476d7ab7-74ce-46bd-968b-8c4798092b31,354b8742-49d4-4a32-90f8-da9f45d12ef5");
request.AddHeader("Cache-Control", "no-cache");
request.AddHeader("Accept", "*/*");
request.AddHeader("User-Agent", "PostmanRuntime/7.19.0");
request.AddHeader("Authorization", "Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJyb2xlIjoiRGVmYXVsdCIsImlzcyI6Imh0dHA6Ly9sb2NhbGhvc3Q6ODg5MC8iLCJhdWQiOiI0MTRlMTkyN2EzODg0ZjY4YWJjNzlmNzI4MzgzN2ZkMSIsImV4cCI6MTU3MjAwOTA3OSwibmJmIjoxNTcxOTIyNjc5fQ.pf83SwGr3lbej6fub9TJ8o94ATS0_oSRJpmGOZMMRXs");
request.AddHeader("Content-Type", "application/json");
```

```
request.AddParameter("undefined", "{\r\n\"FromDate\":
\"2015-01-26T02:37:00\",\r\n\"ToDate\":\"2019-06-26T12:37:00\"\r\n}\r\n",
ParameterType.RequestBody);
IRestResponse response = client.Execute(request);
```

Java Code Sample - (Using OK HTTP)

```
OkHttpClient client = new OkHttpClient();

MediaType mediaType = MediaType.parse("application/json");
RequestBody body = RequestBody.create(mediaType, "{\r\n\"FromDate\":
\"2015-01-26T02:37:00\",\r\n\"ToDate\":\"2019-06-26T12:37:00\"\r\n}\r\n");
Request request = new Request.Builder()
  .url("http://10.00.0.00:0000/api/SIEM/GetServicePasswordEnvelopePrint")
  .post(body)
  .addHeader("Content-Type", "application/json")
  .addHeader("Authorization", "Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJyb2xlIjoiRGVmYXVsdCIsImlzcyI6Imh0d
HA6Ly9sb2NhbGhvc3Q6ODg5MC8iLCJhdWQiOiI0MTRlMTkyN2EzODg0ZjY4YWJjNzlmNzI4Mzg
zN2ZkMSIsImV4cCI6MTU3MjAwOTA3OSwibmJmIjoxNTcxOTIyNjc5fQ.pf83SwGr3lbej6fub9
TJ8o94ATS0_oSRJpmGOZMMRXs")
  .addHeader("User-Agent", "PostmanRuntime/7.19.0")
  .addHeader("Accept", "*/*")
  .addHeader("Cache-Control", "no-cache")
  .addHeader("Postman-Token",
"476d7ab7-74ce-46bd-968b-8c4798092b31,b3eae848-2a39-4d57-be61-
b02191c52a3e")
  .addHeader("Host", "10.00.0.00:0000")
  .addHeader("Accept-Encoding", "gzip, deflate")
  .addHeader("Content-Length", "73")
  .addHeader("Cookie", "ASP.NET_SessionId=lxeqeq4xhhk113gzrml2siap")
  .addHeader("Connection", "keep-alive")
  .addHeader("cache-control", "no-cache")
  .build();

Response response = client.newCall(request).execute();
```

Python Sample Code - (Using http.client Python 3)

```
import http.client

conn = http.client.HTTPConnection("10,00,0,00")

payload = "{\r\n\"FromDate\":\"2015-01-26T02:37:00\",\r\n\"ToDate\":
\"2019-06-26T12:37:00\"\r\n}\r\n"

headers = {
    'Content-Type': "application/json",
```

```
    'Authorization': "Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJyb2xlIjoiRGVmYXVsdCIsImlzcyI6Imh0d
HA6Ly9sb2NhbGhvc3Q6ODg5MC8iLCJhdWQiOiI0MTRlMTkyN2EzODg0ZjY4YWJjNzlmNzI4Mzg
zN2ZkMSIsImV4cCI6MTU3MjAwOTA3OSwibmJmIjoxNTcxOTIyNjc5fQ.pf83SwGr3lbej6fub9
TJ8o94ATS0_oSRJpmGOZMMRXs",
    'User-Agent': "PostmanRuntime/7.19.0",
    'Accept': "*/*",
    'Cache-Control': "no-cache",
    'Postman-Token':
"476d7ab7-74ce-46bd-968b-8c4798092b31,78a8c4ba-2132-425c-
b988-7baaecb41b19",
    'Host': "10.00.0.00:0000",
    'Accept-Encoding': "gzip, deflate",
    'Content-Length': "73",
    'Cookie': "ASP.NET_SessionId=lxeqeq4xhhk113gzrml2siap",
    'Connection': "keep-alive",
    'cache-control': "no-cache"
    }

conn.request("POST", "api,SIEM,GetServicePasswordEnvelopePrint", payload,
headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

Python Sample Code - (Using Python Request)

```
import requests

url = "http://10.00.0.00:0000/api/SIEM/GetServicePasswordEnvelopePrint"

payload = "{\r\n\"FromDate\":\"2015-01-26T02:37:00\",\r\n\"ToDate\":
\"2019-06-26T12:37:00\"\r\n}\r\n"
headers = {
    'Content-Type': "application/json",
    'Authorization': "Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJyb2xlIjoiRGVmYXVsdCIsImlzcyI6Imh0d
HA6Ly9sb2NhbGhvc3Q6ODg5MC8iLCJhdWQiOiI0MTRlMTkyN2EzODg0ZjY4YWJjNzlmNzI4Mzg
zN2ZkMSIsImV4cCI6MTU3MjAwOTA3OSwibmJmIjoxNTcxOTIyNjc5fQ.pf83SwGr3lbej6fub9
TJ8o94ATS0_oSRJpmGOZMMRXs",
    'User-Agent': "PostmanRuntime/7.19.0",
    'Accept': "*/*",
    'Cache-Control': "no-cache",
    'Postman-Token':
"476d7ab7-74ce-46bd-968b-8c4798092b31,9d88488d-4276-414f-96e3-10f7a79f0811
",
    'Host': "10.00.0.00:0000",
    'Accept-Encoding': "gzip, deflate",
    'Content-Length': "73",
```

```
    'Cookie': "ASP.NET_SessionId=lxeqeq4xhhk113gzrml2siap",
    'Connection': "keep-alive",
    'cache-control': "no-cache"
    }

response = requests.request("POST", url, data=payload, headers=headers)

print(response.text)
```

5.8 Get ARCOS Log

Get Service Arcos Log details by From Date and To Date API.

Type of Request		POST			
Endpoint		/api/SIEM/GetARCOSLog			
Available in API Version		All versions			
Release Version		U4			
Pre Endpoint Call		/arconToken			
Request	Params	-	-	-	-
	Headers	Content-Type	application/x-www-form-urlencoded	*Mandatory*	-
		Authorization	Bearer eyJ0[... Removed for brevity ....]INjTU8	*Mandatory*	-
		x-PAM-Version	-	*Not Applicable*	-

	Body	**Request JSON**
		{
		"FromDate":"2019-05-26T02:37:00",
		"ToDate":"2019-06-26T12:37:00"
		}			

Response	**Response JSON**
```
{
  "Program": "ARCON PAM API",
  "Version": "1.0",
  "DateTime": "25/Oct/2019 03:42:05",
  "Success": true,
  "ErrorCode": null,
  "ErrorMessage": null,
  "Message": "609 Records Found",
  "Result": [
    {
      "EventID": 9700,
      "EventTimeStamp": "25/Oct/2019 03:42:05",
      "UserName": "S..XXL",
      "ObjectType": "User Privileges",
      "OperationType": "User Privileges",
      "TrasactionFor": "ARCOSADMIN (ARCOSADMIN)",
      "OldValue": "",
      "NewValue": "ARCON PAM Group Admin -> CONNECTORS",
      "LogTimeStamp": "06/14/2019 16:06:11"
    }]
}
```
 |
Response Time	5 seconds
Post Endpoint Call	None
Supported Features	NA

5.8.1 Data Types of Request and Response

Type	Parameters	Data Type
Request	FromDate	Date Time ('2012-04-23T18:25:43.511Z' format)
	ToDate	Date Time ('2012-04-23T18:25:43.511Z' format)
Response	EventID	int
	EventTimeStamp	Date Time
	UserName	string
	ObjectType	string
	OperationType	string

Type	Parameters	Data Type
	TrasactionFor	string
	OldValue	string
	NewValue	string
	LogTimeStamp	string

5.8.2 Code Blocks for 'SIEM/GetARCOSLog'

C# Sample Code - (Using RestSharp)

```
var client = new RestClient("http://10.00.0.00:0000/api/SIEM/GetARCOSLog");
var request = new RestRequest(Method.POST);
request.AddHeader("cache-control", "no-cache");
request.AddHeader("Connection", "keep-alive");
request.AddHeader("Cookie", "ASP.NET_SessionId=lxeqeq4xhhk113gzrml2siap");
request.AddHeader("Content-Length", "73");
request.AddHeader("Accept-Encoding", "gzip, deflate");
request.AddHeader("Host", "10.00.0.00:0000");
request.AddHeader("Postman-Token", "4bd025ab-4631-44e9-9d72-b91455bc8a99,c1dc85bb-d1d6-45b9-a7e9-2ad5db58f225");
request.AddHeader("Cache-Control", "no-cache");
request.AddHeader("Accept", "*/*");
request.AddHeader("User-Agent", "PostmanRuntime/7.19.0");
request.AddHeader("Authorization", "Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJyb2xlIjoiRGVmYXVsdCIsImlzcyI6Imh0dHA6Ly9sb2NhbGhvc3Q6ODg5MC8iLCJhdWQiOiI0MTRlMTkyN2EzODg0ZjY4YWJjNzlmNzI4MzgzN2ZkMSIsImV4cCI6MTU3MjAwOTA3OSwibmJmIjoxNTcxOTIyNjc5fQ.pf83SwGr3lbej6fub9TJ8o94ATS0_oSRJpmGOZMMRXs");
request.AddHeader("Content-Type", "application/json");
request.AddParameter("undefined", "{\r\n\"FromDate\":\"2019-05-26T02:37:00\",\r\n\"ToDate\":\"2019-06-26T12:37:00\"\r\n}\r\n", ParameterType.RequestBody);
IRestResponse response = client.Execute(request);
```

Java Code Sample - (Using OK HTTP)

```
OkHttpClient client = new OkHttpClient();

MediaType mediaType = MediaType.parse("application/json");
RequestBody body = RequestBody.create(mediaType, "{\r\n\"FromDate\":\"2019-05-26T02:37:00\",\r\n\"ToDate\":\"2019-06-26T12:37:00\"\r\n}\r\n");
```

```
Request request = new Request.Builder()
  .url("http://10.00.0.00:0000/api/SIEM/GetARCOSLog")
  .post(body)
  .addHeader("Content-Type", "application/json")
  .addHeader("Authorization", "Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJyb2xlIjoiRGVmYXVsdCIsImlzcyI6Imh0d
HA6Ly9sb2NhbGhvc3Q6ODg5MC8iLCJhdWQiOiI0MTRlMTkyN2EzODg0ZjY4YWJjNzlmNzI4Mzg
zN2ZkMSIsImV4cCI6MTU3MjAwOTA3OSwibmJmIjoxNTcxOTIyNjc5fQ.pf83SwGr3lbej6fub9
TJ8o94ATS0_oSRJpmGOZMMRXs")
  .addHeader("User-Agent", "PostmanRuntime/7.19.0")
  .addHeader("Accept", "*/*")
  .addHeader("Cache-Control", "no-cache")
  .addHeader("Postman-Token", "4bd025ab-4631-44e9-9d72-
b91455bc8a99,67fcb770-ac17-4225-bb4f-8db6473e11bf")
  .addHeader("Host", "10.00.0.00:0000")
  .addHeader("Accept-Encoding", "gzip, deflate")
  .addHeader("Content-Length", "73")
  .addHeader("Cookie", "ASP.NET_SessionId=lxeqeq4xhhk113gzrml2siap")
  .addHeader("Connection", "keep-alive")
  .addHeader("cache-control", "no-cache")
  .build();

Response response = client.newCall(request).execute();
```

Python Sample Code - (Using http.client Python 3)

```
import http.client

conn = http.client.HTTPConnection("10,00,0,00")

payload = "{\r\n\"FromDate\":\"2019-05-26T02:37:00\",\r\n\"ToDate\":
\"2019-06-26T12:37:00\"\r\n}\r\n"

headers = {
    'Content-Type': "application/json",
    'Authorization': "Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJyb2xlIjoiRGVmYXVsdCIsImlzcyI6Imh0d
HA6Ly9sb2NhbGhvc3Q6ODg5MC8iLCJhdWQiOiI0MTRlMTkyN2EzODg0ZjY4YWJjNzlmNzI4Mzg
zN2ZkMSIsImV4cCI6MTU3MjAwOTA3OSwibmJmIjoxNTcxOTIyNjc5fQ.pf83SwGr3lbej6fub9
TJ8o94ATS0_oSRJpmGOZMMRXs",
    'User-Agent': "PostmanRuntime/7.19.0",
    'Accept': "*/*",
    'Cache-Control': "no-cache",
    'Postman-Token': "4bd025ab-4631-44e9-9d72-
b91455bc8a99,7b0xx0d-0b03-4ebe-ac74-ad532cda319a",
    'Host': "10.00.0.00:0000",
    'Accept-Encoding': "gzip, deflate",
    'Content-Length': "73",
    'Cookie': "ASP.NET_SessionId=lxeqeq4xhhk113gzrml2siap",
    'Connection': "keep-alive",
    'cache-control': "no-cache"
```

```
    }

conn.request("POST", "api,SIEM,GetARCOSLog", payload, headers)

res = conn.getresponse()
data = res.read()

print(data.decode("utf-8"))
```

Python Sample Code - (Using Python Request)

```
import requests

url = "http://10.00.0.00:0000/api/SIEM/GetARCOSLog"

payload = "{\r\n\"FromDate\":\"2019-05-26T02:37:00\",\r\n\"ToDate\":
\"2019-06-26T12:37:00\"\r\n}\r\n"
headers = {
    'Content-Type': "application/json",
    'Authorization': "Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJyb2xlIjoiRGVmYXVsdCIsImlzcyI6Imh0d
HA6Ly9sb2NhbGhvc3Q6ODg5MC8iLCJhdWQiOiI0MTRlMTkyN2EzODg0ZjY4YWJjNzlmNzI4Mzg
zN2ZkMSIsImV4cCI6MTU3MjAwOTA3OSwibmJmIjoxNTcxOTIyNjc5fQ.pf83SwGr3lbej6fub9
TJ8o94ATS0_oSRJpmGOZMMRXs",
    'User-Agent': "PostmanRuntime/7.19.0",
    'Accept': "*/*",
    'Cache-Control': "no-cache",
    'Postman-Token': "4bd025ab-4631-44e9-9d72-
b914xxxbc8a99,c307f936-6e00-4006-af55-7bdb384d4be2",
    'Host': "10.00.0.00:0000",
    'Accept-Encoding': "gzip, deflate",
    'Content-Length': "73",
    'Cookie': "ASP.NET_SessionId=lxeqeq4xhhk113gzrml2siap",
    'Connection': "keep-alive",
    'cache-control': "no-cache"
    }

response = requests.request("POST", url, data=payload, headers=headers)

print(response.text)
```

6 Intergration with LogRhythm (SIEM)

6.1 Prerequisites

Follow below steps before configure LogRhythm (SIEM) with ARCON PAM Database:

1. Create MS – SQL User on ARCON PAM Database
2. Grant Permission only on table "dbo.SIEMARCOSenvision"
   GRANT SELECT, INSERT ON dbo.SIEMARCOSenvision To MS-SQL User
3. Run a ARCON PAM SIEM Connector Service Setup (On Windows Server)
4. Copy dbsetting.ini file to installed location
5. Start "ARCOSSIEMConnectorService" from services.msc

6.2 Configuration of JDBC Protocol

Log sources configured with Java Database Connectivity (JDBC) protocol can remotely poll databases for events. The JDBC protocol enables LogRhythm (SIEM) to collect information from tables or views that contain event data from several database type.
The following table describes the parameters for JDBC protocol:

Parameter	Description
Log Source Name	Type a unique name of the log source.
Log Source Description	Optional. Type a description for the log source.
Log Source Type	From the list, select the type of log source to add.
Protocol Configuration	From the list, select JDBC.
Log Source Identifier	Type the log source identifier in one of the following formats:
	database@hostname
	table name|database@hostname
	The database name must match the value of the Database Name parameter. The database name is a required parameter.
	The hostname is the hostname or IP address for the device that hosts the database. The hostname must match the parameter in the IP or Hostname field. The hostname is a required parameter.
	Optional. The table name is the name of the table or view on the database which contains the event records. If you define the name of a table or view, you must include a pipe (|) character as a separator. The name of the view or table must match the Table Name field.
Database Type	From the list box, select the type of database that contains the events.
Database Name	Type the name of the database to which the protocol can connect. The database name must match the database name specified in the Log Source Identifier field.
IP or Hostname	Type the IP address or hostname of the database server.

Parameter	Description
Port	Type the port number used by the database server. The default displayed depends on the selected Database Type. The valid range is 0 to 65536. The defaults include:
	MSDE – 1433
	Postgres – 5432
	MySQL – 3306
	Sybase – 1521
	Oracle – 1521
	Informix – 9088
	The JDBC port must match the listen port configured on the remote database. The database must permit incoming TCP connections.
	If a Database Instance is used with the MSDE database type, administrators must leave the Port parameter blank in the log source configuration.
Username	Type the database username. The username can be up to 255 alphanumeric characters in length and can include underscore (_) characters.
	To track access to database access for audit purposes, administrators can create a specific user on the database for LogRhythm (SIEM).
Password	Type the database password. The password can be up to 255 characters in length.
Confirm Password	Confirm the password to access the database.
Authentication Domain	Type a domain for the database.
	A domain must be configured for MSDE databases that are within a Windows domain. If your network does not use a domain, leave this field blank.
Database Instance	Type the database instance, if required. MSDE databases can include multiple SQL server instances on one server.
	When a non-standard port is used for the database or administrators have blocked access to port 1434 for SQL database resolution, the Database Instance parameter must be blank in the log source configuration.
Predefined Query	Optional. Select a predefined database query for the log source. If a predefined query is not available for the log source type, administrators can select none.
Table Name	Type the name of the table or view that includes the event records.
	The table name can include the following special characters: dollar sign ($), number sign (#), underscore (_), en dash (-), and period(.).
Select List	Type the list of fields to include when the table is polled for events. Administrators can use a comma separated list or type * to select all fields from the table or view.

Parameter	Description
	If a comma-separated list is defined, the list must contain the field defined in the Compare Field.
Compare Field	Type a numeric value or timestamp field from the table or view that can identify new events added between queries to the table.
	This field enables the protocol to identify events that were previously polled by the protocol to ensure that duplicate events are not created.
Use Prepared Statements	Select this check box to use prepared statements.
	Prepared statements enable the JDBC protocol source to setup the SQL statement, and then execute the SQL statement numerous times with different parameters. For security and performance reasons, most JDBC protocol configurations can use prepared statements.
	Clear this check box to use an alternative method of querying that do not use pre-compiled statements.
Start Date and Time	Optional. Configure a start date and time for when the protocol can start to poll the database.
	If a start time is not defined, the protocol attempts to poll for events after the log source configuration is saved and deployed.
Polling Interval	Type the polling interval, which is the amount of time between queries to the database. The default polling interval is 10 seconds.
	Administrators can define a longer polling interval by appending H for hours or M for minutes to the numeric value. The maximum polling interval is 1 week in any time format. Numeric values without an H or M designator poll in seconds.
EPS Throttle	Type the number of Events Per Second (EPS) that you do not want this protocol to exceed. The default value is 20000 EPS.
Use Named Pipe Communication	If MSDE is configured as the database type, administrators can select this check box to use an alternative method to a TCP/IP port connection.
	Named pipe connections for MSDE databases require the username and password field to use a Windows authentication username and password and not the database username and password. The log source configuration must use the default named pipe on the MSDE database.
Database Cluster Name	If the Use Named Pipe Communication check box, the Database Cluster Name parameter is displayed.
	If you use your SQL server in a cluster environment, define the cluster name to ensure that named pipe communications function properly.

Parameter	Description
Use NTLMv2	Select the Use NTLMv2 check box to force MSDE connections to use the NTLMv2 protocol when communicating with SQL servers that require NTLMv2 authentication. The default value of the check box is selected.
	The Use NTLMv2 check box does not interrupt communications for MSDE connections that do not require NTLMv2 authentication.
Use SSL	Select this check box to enable SSL encryption for the JDBC protocol.
Enabled	Select this check box to enable the log source.
	When this check box is clear, the log source does not collect events and the log source is not counted in the license limit.
Credibility	Select the credibility of the log source. The range is 0 (lowest) - 10 (highest). The default credibility is 5.
	Credibility is a representation of the integrity or validity of events that are created by a log source. The credibility value that is assigned to a log source can increase or decrease based on incoming events or adjusted as a response to user created event rules. The credibility of events from log sources contributes to the calculation of the offense magnitude and can increase or decrease the magnitude value of an offense.
Target Event Collector	Select the target for the log source. When a log source actively collects events from a remote source, this field defines which appliance polls for the events.
	The target event collector enables administrators to poll and process events on the target event collector, instead of the Console appliance. Distributing event across target event collectors can improve performance in distributed deployments.
Coalescing Events	Select this check box to enable the log source to coalesce (bundle) events.
	Coalescing events increase the event count when the same event occurs multiple times within a short time interval. Coalesced events provide administrators a way to view and determine the frequency with which a single event type occurs on the Log Activity tab.
	When this check box is clear, events are viewed individually and events are not bundled.
	New and automatically discovered log sources inherit the value of this check box from the System Settings configuration on the Admin tab. Administrators can use this check box to override the default behavior of the system settings for an individual log source.
Store Event Payload	Select this check box to enable the log source to store the payload information from an event.

Parameter	Description
	New and automatically discovered log sources inherit the value of this check box from the System Settings configuration on the Admin tab. Administrators can use this check box to override the default behavior of the system settings for an individual log source.
Log Source Language	Select the language of the events that are generated by the log source.
	The log source language helps the system parse events from external appliances or operating systems that can create events in multiple languages.
Log Source Extension	Optional. Select the name of the extension to apply to the log source.
	This parameter is available after a log source extension is uploaded. Log source extensions are XML files that contain regular expressions, which can override or repair the event parsing of a device support module (DSM).
Extension Use Condition	From the list box, select the use condition for the log source extension. The options include:
	Parsing enhancement - Select this option when most fields parse correctly for the log source.
	Parsing override - Select this option when the log source is unable to correctly parse events.
Groups	Select one or more groups for the log source.

6.2.1 Procedure

1. Click the **Admin** tab.
2. Click the **Log Sources** icon.
3. Click **Add**.
4. Configure the parameters for the log source. Click **Save**.
5. On the **Admin** tab, click **Deploy Changes**

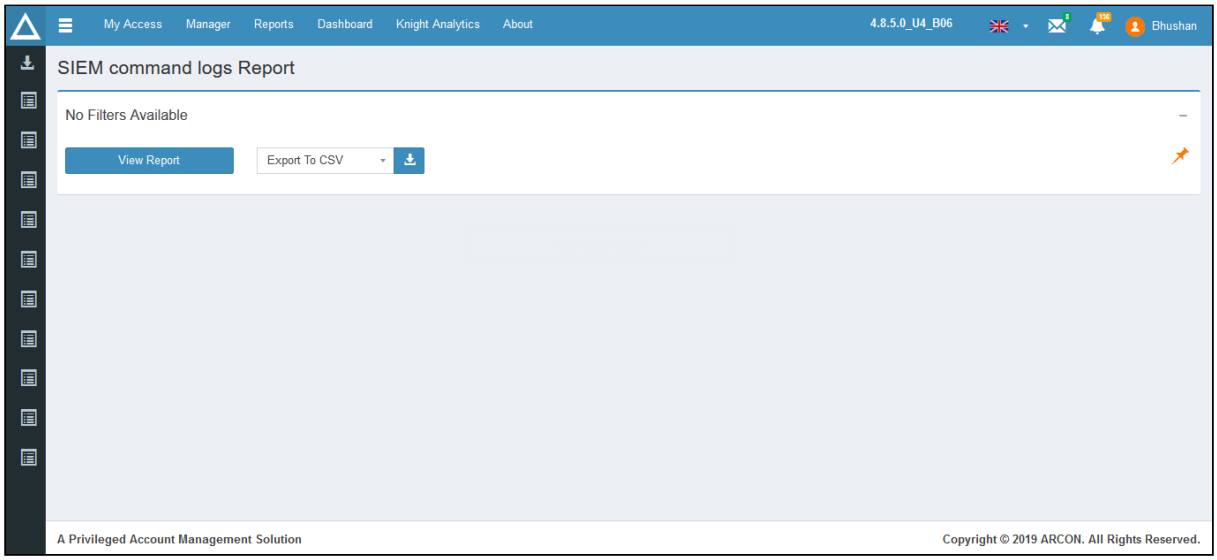
7 SIEM Command Logs Report

SIEM Command Logs Report is used to display command logs fetched from SIEM service. It displays logs of commands executed on Linux service. It displays details such as Login User ID, Session Log ID accessed by User, User Login timestamp details, IP and MAC address, type of service, service description, commands executed, command timestamp, Service login timestamp, Service Log ID, password age, last password change timestamp, password status, name of the User who has tried to view password, and date on which password was viewed.

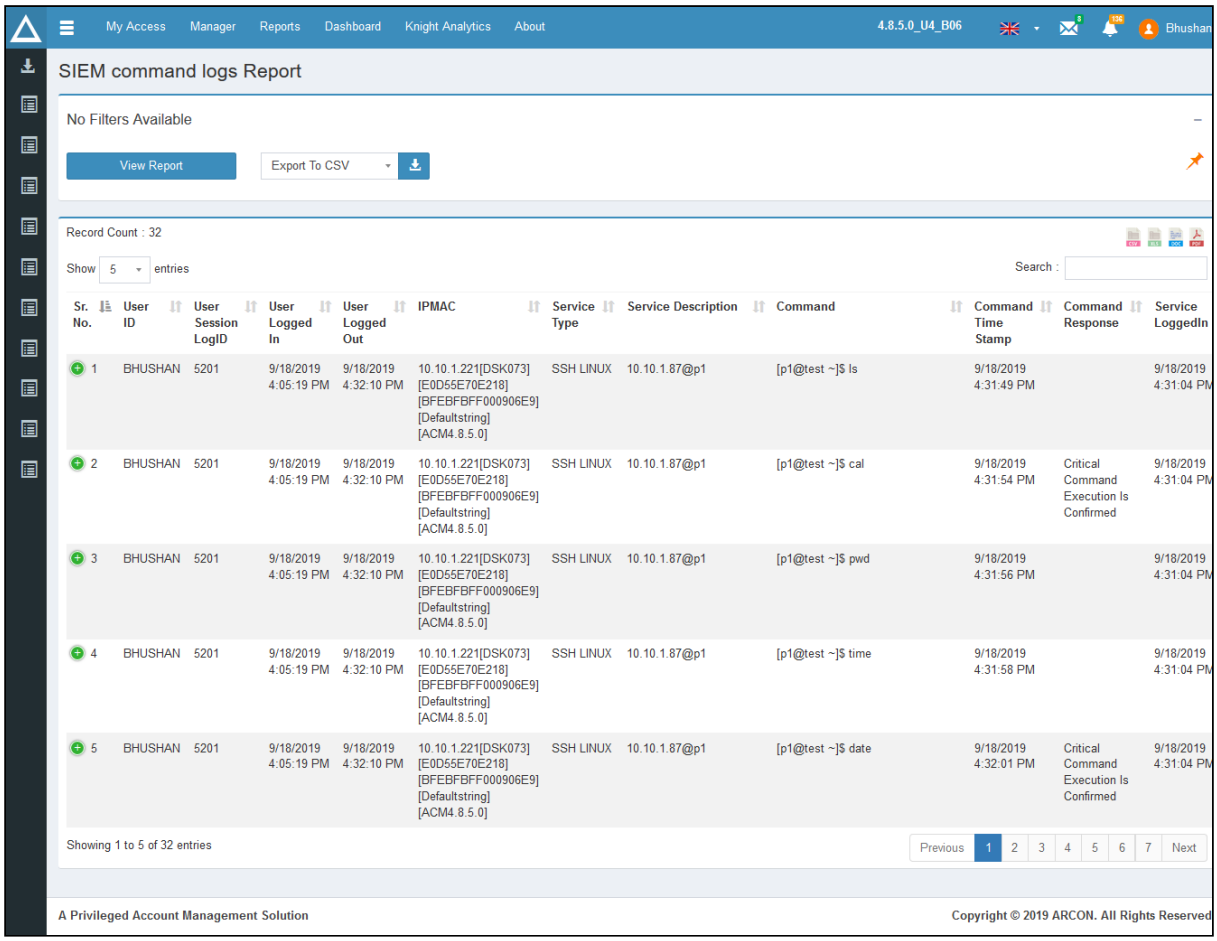
⚠ User having **SIEM Command Logs Report** privilege will be able to view SIEM Command Logs Report.

Process to view SIEM Command Logs:

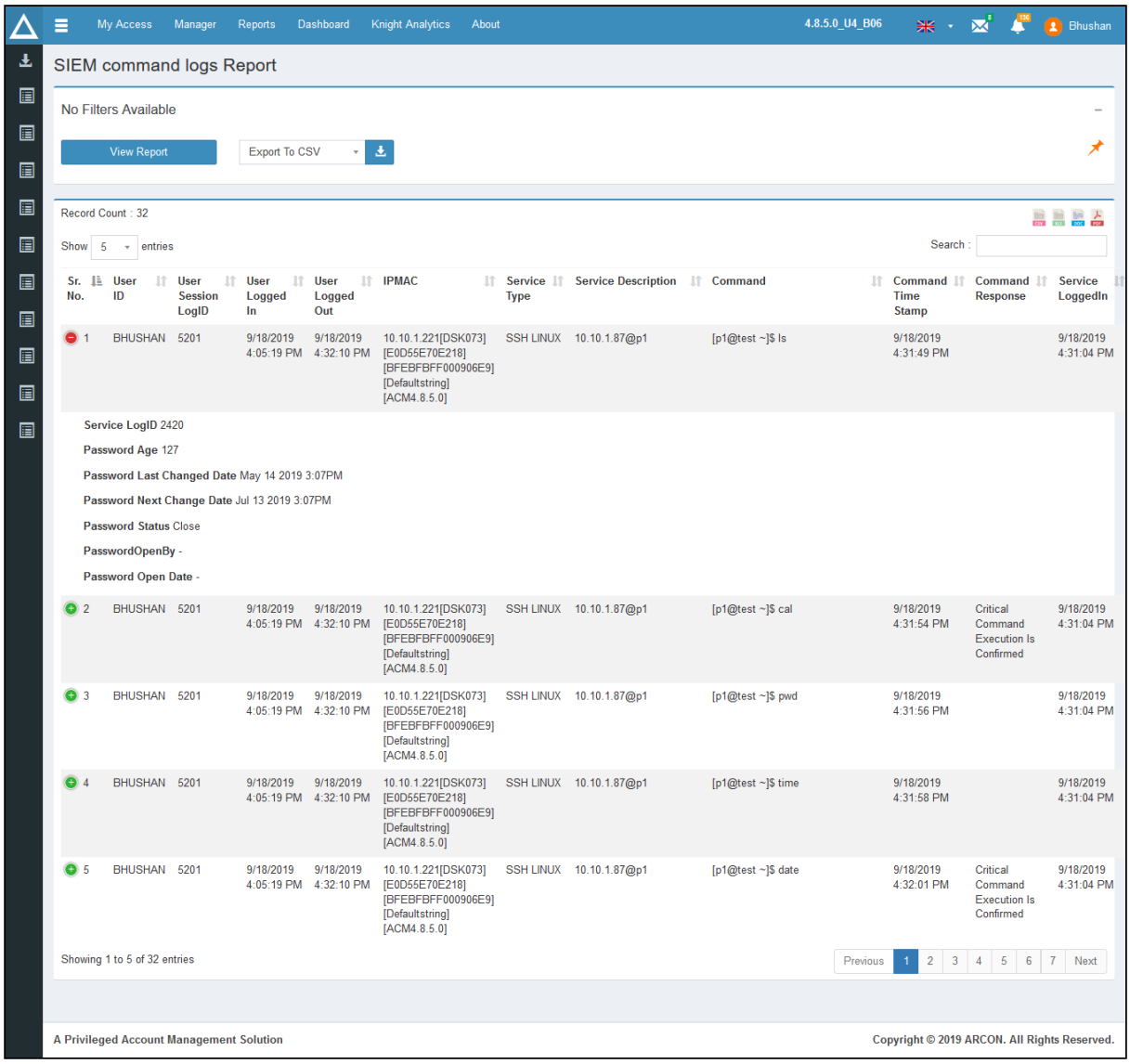
1. Navigate to **Reports → Logs → SIEM Command Logs Report**
2. The **SIEM Command Logs Report** screen is displayed.



3. Click **View Report**. It displays the grid view details of command logs executed on Linux service.



4. Click  icon, to view additional details such as password age, last password change timestamp, password status, name of the User who has tried to view password, and date on which password was viewed.



5. Select the number of entries from **Show entries** drop down list, to display only those numbers of records in the grid.
6. To search for a particular record, enter the required search filter in the **Search** text field, on the right hand side of the screen.
7. To pin the report to **Dashboard**, click  icon.

> ⚠ For downloading reports, refer **Exported Reports**section.

Privileged Access Management Suite