

IBM Security Access Manager for Mobile
Version 8.0.0.1

Administration Guide



IBM Security Access Manager for Mobile
Version 8.0.0.1

Administration Guide



Note

Before using this information and the product it supports, read the information in “Notices” on page 115.

Edition notice

Note: This edition applies to version 8.0.0.1 of IBM Security Access Manager for Mobile (product number 5725-L52) and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright IBM Corporation 2013.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures	v
--------------------------	----------

Tables	vii
-------------------------	------------

About this publication	ix
---	-----------

Access to publications and terminology	ix
Accessibility	x
Technical training.	x
Support information.	x
Statement of Good Security Practices	x

Chapter 1. Overview of context-based access	1
--	----------

Business scenarios	1
Features	2
Functional overview.	2
Transaction flow	4

Chapter 2. Risk management overview	9
--	----------

Risk score calculation	12
Risk reports	13
Configuring risk reports	13
Accessing risk reports	15

Chapter 3. Attributes	17
--	-----------

Managing attributes	17
Updating location attributes	18
Attribute properties	21
Predefined attributes	23
Configuring the hash algorithm for attribute storage	32
Dynamic attributes	33
Scenarios for adding and manipulating attributes	34
Coding the dynamic.attributes.js file	34
Updating and deploying the dynamic.attributes.js file	36
Custom attributes for the authorization service	37
[azn-decision-info] stanza.	37
[user-attribute-definitions] stanza	41
Setting the data type or category of a custom attribute	42

Chapter 4. Attribute collection service	45
--	-----------

Configuring the attribute collection service	48
--	----

Chapter 5. Attribute matchers	51
--	-----------

IP reputation	54
Configuring the IBM Security Access Manager for Mobile appliance to contact the license server.	55
Modifying attribute matchers	56

Chapter 6. Obligations	59
---	-----------

Managing obligations	59
Obligation properties	60

Mapping obligations to a URL	61
obligation	62
Predefined obligations.	63

Chapter 7. Risk profiles	65
---	-----------

Managing risk profiles.	65
Predefined risk profiles	73

Chapter 8. Policies	75
--------------------------------------	-----------

Managing policies	75
Creating a policy	76
Managing policy sets	79
Managing policy attachments	81
Policy scenarios	84
Denying access based on a set of conditions	84
Denying access based on a set of conditions with an OR clause	85
Permitting access based on a set of conditions with an AND clause	86
Permitting access after one-time password authentication	87
Registering a device after user consent	88

Chapter 9. Device fingerprints	91
---	-----------

Managing device fingerprints	91
Silent device registration	92
Consent-based device registration	92
Setting the authentication level for consent-based device registration	93
Modifying consent template pages.	93
REST services for device fingerprint registration	95

Chapter 10. Runtime database	97
---	-----------

Managing the runtime database	97
Deploying an external runtime database.	98
Database usage requirements	101
Runtime database tuning parameters	101

Chapter 11. Policy information points	105
--	------------

Managing policy information points.	105
RESTful web service properties	106

Chapter 12. Deploying pending changes	109
--	------------

Chapter 13. User self-administration tasks	111
---	------------

Managing your registered devices	111
Managing OTP secret keys	112

Notices	115
--------------------------	------------

Index	119
------------------------	------------

Figures

- | | | | | | |
|----|---|---|----|--|----|
| 1. | Context-based access architecture | 2 | 3. | The closest points, midpoints, and farthest points on the accuracy ranges of two locations . | 52 |
| 2. | Context-based access transaction flow example | 5 | | | |

Tables

1.	REST services instructions.	95	3.	Runtime database tuning parameters	102
2.	Runtime database deployment scripts.	98			

About this publication

The *IBM Security Access Manager for Mobile Administration Guide* describes how to manage, configure, and deploy an existing IBM Security Access Manager environment.

Access to publications and terminology

This section provides:

- A list of publications in the “IBM Security Access Manager for Mobile library.”
- Links to “Online publications.”
- A link to the “IBM Terminology website.”

IBM Security Access Manager for Mobile library

The following documents are available online in the IBM Security Access Manager for Mobile library:

- *IBM Security Access Manager for Mobile Configuration Guide*, SC27-6205-00
- *IBM Security Access Manager for Mobile Administration Guide*, SC27-6207-00
- *IBM Security Access Manager Appliance Administration Guide*, SC27-6206-00
- *IBM Security Access Manager for Mobile Auditing Guide*, SC27-6208-00
- *IBM Security Access Manager for Mobile Troubleshooting Guide*, GC27-6209-00
- *IBM Security Access Manager for Mobile Error Message Reference*, GC27-6210-00

Online publications

IBM posts product publications when the product is released and when the publications are updated at the following locations:

IBM Security IBM Security Access Manager for Mobile library

The product documentation site (http://pic.dhe.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.ammob.doc_8.0.0/welcome.html) displays the welcome page and navigation for the library.

IBM Security Systems Documentation Central

IBM Security Systems Documentation Central provides an alphabetical list of all IBM Security Systems product libraries and links to the online documentation for specific versions of each product.

IBM Publications Center

The IBM Publications Center site (<http://www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss>) offers customized search functions to help you find all the IBM publications you need.

IBM Terminology website

The IBM Terminology website consolidates terminology for product libraries in one location. You can access the Terminology website at <http://www.ibm.com/software/globalization/terminology>.

Accessibility

Accessibility features help users with a physical disability, such as restricted mobility or limited vision, to use software products successfully. You can use the keyboard instead of the mouse to operate all features of the graphical user interface.

For additional information, see the IBM Accessibility website at <http://www.ibm.com/able/>.

Technical training

For technical training information, see the following IBM Education website at <http://www.ibm.com/software/tivoli/education>.

Support information

IBM Support provides assistance with code-related problems and routine, short duration installation or usage questions. You can directly access the IBM Software Support site at <http://www.ibm.com/software/support/probsub.html>.

IBM Security Access Manager for Mobile Troubleshooting Guide provides details about:

- What information to collect before contacting IBM Support.
- The various methods for contacting IBM Support.
- How to use IBM Support Assistant.
- Instructions and problem-determination resources to isolate and fix the problem yourself.

Note: The **Community and Support** tab on the product information center can provide additional support resources.

Statement of Good Security Practices

IT system security involves protecting systems and information through prevention, detection and response to improper access from within and outside your enterprise. Improper access can result in information being altered, destroyed, misappropriated or misused or can result in damage to or misuse of your systems, including for use in attacks on others. No IT system or product should be considered completely secure and no single product, service or security measure can be completely effective in preventing improper use or access. IBM systems, products and services are designed to be part of a comprehensive security approach, which will necessarily involve additional operational procedures, and may require other systems, products or services to be most effective. IBM DOES NOT WARRANT THAT ANY SYSTEMS, PRODUCTS OR SERVICES ARE IMMUNE FROM, OR WILL MAKE YOUR ENTERPRISE IMMUNE FROM, THE MALICIOUS OR ILLEGAL CONDUCT OF ANY PARTY.

Chapter 1. Overview of context-based access

Context-based access provides access decision and enforcement that is based on a dynamic risk assessment or confidence level of a transaction. Context-based access uses behavioral and contextual data analytics to calculate risk.

Context-based access:

- Improves security during authentication and authorization of business transactions.
- Assesses risk based on static, contextual, and analytically calculated attributes.
- Calculates a risk score based on multiple weighted attributes.
- Provides policy rules that determine whether an access request must be permitted, denied, or challenged.

You can configure context-based access to:

- Silently register or require users to register devices that they commonly use.
- Associate the registered devices with user credentials.
- Present a challenge or request additional authentication, if the user attempts to authenticate with the same credentials from another unregistered device.
- Enforce specific authentication mechanisms to access a particular protected resource.
- Use the behavioral patterns of the user as a factor in risk score calculation. For example, a user might attempt to access a protected resource at a time outside of normal business hours. You can configure the context-based access policy to deny access or force the user access to authenticate with a secondary challenge.

Business scenarios

Business transactions that have an increased security risk factor can benefit by implementing context-based access.

The following examples are some scenarios where you can use context-based access to provide a higher level of confidence for the transaction:

- A user tries to access sensitive information where a simple user ID and password authentication is not sufficient. However, the data is not sensitive enough to use a more complex authentication mechanism, such as token IDs.
- Users require access from remote locations that are not trusted and they use devices such as mobile devices and notebooks. To ensure that mobile users are authenticated sufficiently, the business requires a second factor authentication.
- Users need to access an application that provides sensitive business information. They might access the information outside of their regular work patterns.
- A user accesses a resource from a device that the user previously used and maintains typical usage patterns. Context-based access improves the user experience by limiting secondary authentication mechanisms.

Features

Context-based access provides several capabilities to identify potential risk and limit the ability for an attacker to use stolen credentials.

- Silent device registration where the system does not require any user interaction.
- Ready-to-use, predefined policy attributes that are specific to context-based access.
- Scenario-based, predefined risk profiles.
- A risk-scoring engine that calculates a risk score for the current transaction based on the active risk profile. The risk score is based on configurable weights that are assigned to context attributes and behavior attributes. If the risk score is high, further challenges are presented to the user or access is denied. If the risk score is low, the user is permitted access.

Functional overview

Context-based access includes an external authorization service (EAS), runtime authorization service, and attribute collection service.

The following diagram illustrates the architecture of context-based access. The diagram also shows how the various components plug into WebSEAL. WebSEAL is a component available in the following IBM products:

- IBM Web Gateway Appliance
- IBM Security Access Manager
- Tivoli Access Manager for e-business

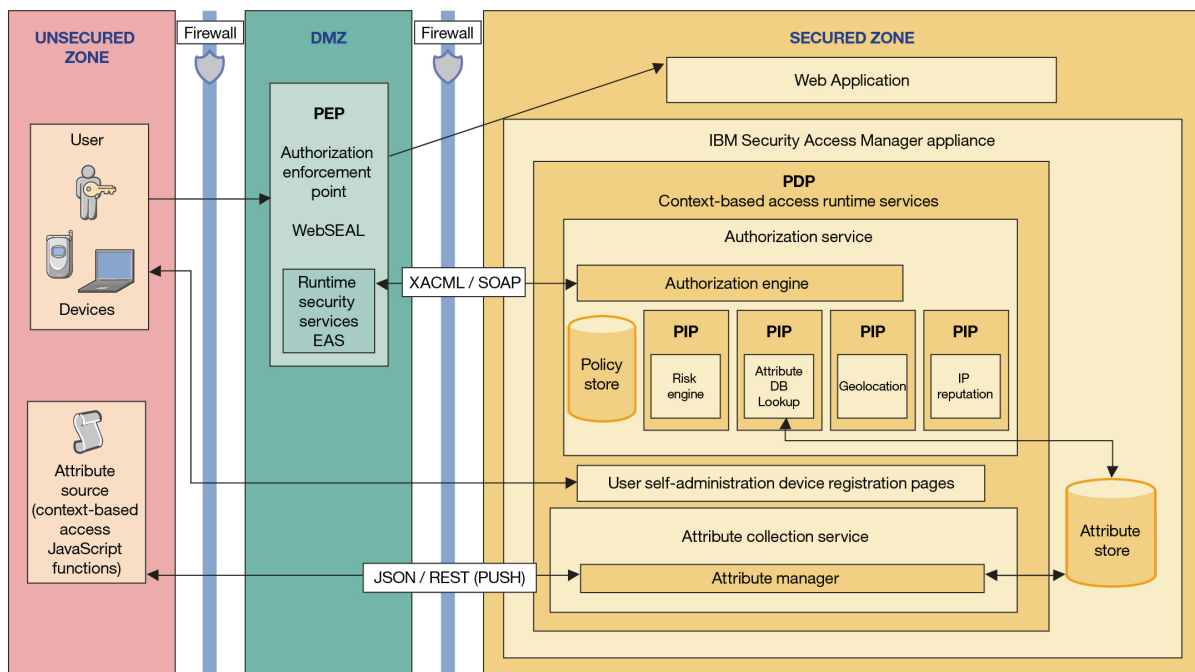


Figure 1. Context-based access architecture

Context-based access runtime services

Context-based access provides the following runtime services:

Authorization service

The context-based access authorization service is a component of the runtime environment. The runtime authorization service stores the policy, calculates the risk score, and makes the access decision. The authorization service exposes an XACML over SOAP web service that third-party enforcement points can call to get authorization decisions.

Attribute collection service

The attribute collection service is a Representational State Transfer (REST) service that collects web browser and location attributes from the user. The attribute collection service is a push service. You can configure the context-based access runtime service to use the collected attributes as the policy attributes for calculating risk. You can also use the Java™ ADK to plug in your custom implementation for a pull service that retrieves attributes from the user.

Risk-scoring engine

The risk-scoring engine calculates the risk or confidence level. It provides a single integer that represents the risk score for the current transaction in the form of a percentage. The risk score is calculated based on the weights that are assigned to one or more of the following policy attributes that are part of the active risk profile:

- Device identification or fingerprint, such as details of hardware, IP address, location information, IP address reputation, operating system, web browser type, web browser version, web browser plug-ins, and screen resolution.
- Behavioral patterns, such as frequency of login, time of access, frequency of access, and type of transactions.
- Custom attributes that you can configure and manage through a pluggable interface. The context-based access authorization service is extensible and can also include external sources for attributes.

The risk engine returns the final risk score as a policy attribute, which is the basis of the final authorization decision.

Policy enforcement point (PEP)

WebSEAL is the policy enforcement point for context-based access. Context-based access integrates with the existing WebSEAL authentication mechanisms, such as cross domain authentication service (CDAS) and external authentication interface (EAI).

External authorization service

The runtime security services EAS plug-in for WebSEAL enforces the policy decision. The EAS takes the request data and sends an authorization decision request to the context-based access authorization service. The authorization service maps the authorization decision response to the appropriate WebSEAL action, such as permit, deny, or step-up authentication. You can manage the EAS with entries in the `webseald.conf` file with the WebSEAL stanza syntax. The **isamcfg** tool automates the configuration of the EAS for the predefined scenarios provided with the product.

Policy information points (PIPs)

Policy information points are components of the context-based access authorization service. They provide all the policy attributes that are not

provided in the initial access request. The risk score and attributes that are pushed to the attribute collection service are provided to the authorization service through PIPs.

Context-based access includes ready-to-use PIP implementations that provide the policy attributes that are required. You can also provide custom policy attributes to the authorization service through a custom PIP.

Policy decision point (PDP)

The runtime authorization service is the policy decision point for context-based access. This service is configured to use the PEP context-based access plug-in. The authorization decision is based on an authorization policy that uses policy attributes and PIPs that are specific to context-based access. The PIPs provide information, such as risk score, user location, and device type.

The policy administration point (PAP)

The IBM Security Access Manager appliance is the policy administration point for context-based access. Context-based access provides an administrative console for configuring and managing the policies, risk profiles, attributes, and weights, which are required for calculating risk.

Transaction flow

In a typical context-based transaction, the user requests access to a protected resource. Context-based access calculates the risk score and determines whether access is permitted, denied, or permitted with an obligation.

In the following example of a context-based access transaction, the risk score indicates that the user must be presented with a challenge. The user successfully completes the challenge and receives permission to access the protected resource.

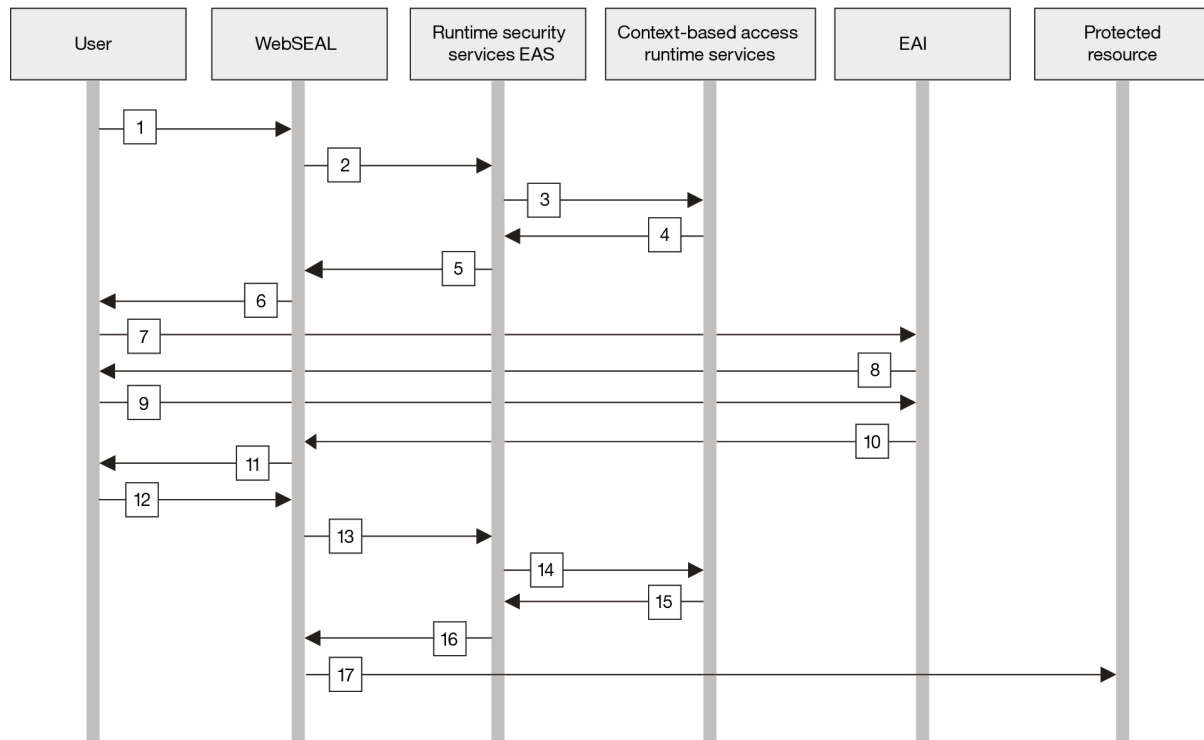


Figure 2. Context-based access transaction flow example

The following process explains the flow of the transaction in the example scenario:

1. The user interacts through a web browser to submit authentication information and requests access to a protected resource. The protected resource is a junctioned web application that WebSEAL protects.
2. WebSEAL inspects the request.
WebSEAL is the reverse proxy server that interacts with all transactions. For the protected resource that the user requests, the WebSEAL policy (POP) is configured to call the runtime security services EAS plug-in to authorize the request. The EAS is a shared library plug-in, which is internal to the WebSEAL process, so there is no on-the-wire callout between WebSEAL and the EAS.
3. The EAS first checks the local access manager policy.
 - a. If the access manager denies access, then the EAS returns the response and does not continue with a forbidden response.
 - b. If the access manager permits access, the EAS collects the context information about the user and the request. The WebSEAL **azn-decision-info** stanza has the specifications to create an XACML over SOAP authorization decision request.
 - c. The EAS sends the request to the context-based access runtime authorization service for the authorization decision.
4. The runtime authorization service (PDP) runs the configured policy and calls the appropriate PIPs based on the current policy.
 - a. If a risk score policy attribute is requested, then the risk engine is called.
 - b. The risk engine takes the context details, requests additional policy attributes, if required, and then calculates a risk score.

- c. The authorization service applies the risk score policy attribute against the authorization policy and returns an appropriate decision response to the runtime security services EAS.
- d. The EAS supports three decision types: permit, deny, and permit with obligation. In this example, a decision to permit with obligation is returned.
 - 1) If the policy decision is a simple permit or deny, the decision is mapped to the WebSEAL permit and not_permit decisions. The WebSEAL decisions are returned from the **azn_svc_decision_access_allowed_ext** call.
 - 2) If the response is to permit access, WebSEAL permits the request to continue to the requested resource.
 - 3) If the response is to deny access, WebSEAL displays an error page, which indicates that access is forbidden.
5. The runtime security services EAS parses the response and returns one of the following EAS responses to WebSEAL:
 - Permit
 - Deny
 - Step-up authentication levels
 - Browser redirect to a location configured on the WebSEAL configuration file

In this example, the response is a redirect to a location on the IBM Security Access Manager for Mobile appliance runtime, so that WebSEAL can enforce the appropriate authentication challenge.

The context-based access EAS also provides a configuration to map the returned obligation to a specific WebSEAL external authentication interface (EAI) mechanism. The EAI can be either a WebSEAL EAI or CDAS external authentication mechanism, which the customer or IBM® Business Partner can implement.
6. WebSEAL specifies the appropriate authentication mechanism to the user.
7. The user is redirected to the authentication mechanism, which builds a challenge response.
8. The challenge response is presented to the user.
9. The user responds to the challenge.
10. If the response of the user to the challenge request is successful, the challenge is processed. The EAI application returns the credentials of the user and a successful step-up level in the HTTP response to WebSEAL.
11. WebSEAL updates the session of the user with the new credential details that the EAI application provides and redirects the user to the protected resource.
12. The request from the user to access the protected resource is sent via 302 redirect. The request does not require any user interaction.
13. WebSEAL inspects the request and directs the request to the runtime security services EAS for authorization.
14. The EAS collects all context information about the user and the request and creates an XACML over SOAP decision request. The EAS sends the request to the runtime authorization service.
15. The context-based access runtime service takes the context and other policy attributes and calculates a risk score. The runtime service applies the risk score against the configured authorization policy and returns a policy decision response to permit access.

16. The runtime security services EAS interprets the response and returns the permit decision to WebSEAL.
17. WebSEAL permits the original request to continue without going back to the web browser of the user. The user is not aware of the transaction process.

Chapter 2. Risk management overview

Context-based access policy decisions can be based on the risk score. The risk score is calculated based on the active risk profile attributes that are retrieved from the user.

The system allows for multiple risk profiles to be defined, but only one is active at run time.

Each attribute included on a risk profile has an assigned weight to be used while calculating the risk score of a given request. The active risk profile attributes are evaluated to determine whether a user should be granted access to a protected resource. A policy author can rely on the risk score to enforce stronger authentication mechanisms or to perform device registration.

To get started setting up context-based access control for your installation, work with:

1. **Attributes:** The product provides a predefined set of attributes that are ready to use without any customization. Optionally, you can add attributes that can come from:
 - Standard HTTP headers
 - HTTP FORM parameters
 - Client-side JavaScript files that are collected into the attribute collection service
 - Custom attributes you define by writing custom JavaScript files
2. **Obligations:** The product provides a predefined set of obligations that are ready to use without any customization. Optionally, you can update or define your own obligations.
3. **Risk profiles:** The product provides a predefined set of risk profiles that are ready to use without any customization. Optionally, you can update or define your own risk profiles to calculate the risk score.

Note: A default risk profile is set as active when you configure the appliance. This risk profile is not intended to be used in a production environment. Set a different risk profile before using risk profiles in your production environment.

4. **Policies:** Create policies that evaluate requests that are based on attributes and obligations that you defined and the risk decisions that you want to make.

Risk score calculation

Risk score calculation is the process by which the risk engine determines a risk score. The *risk score* demonstrates the level of risk that is associated with permitting a request to access the resource. This risk score is compared to a *threshold score* that is set in a policy. A decision is made based on the result of this comparison.

Overview

The risk engine determines a risk score by comparing sets of attributes that identify devices. These sets of attributes are called *device fingerprints*. Device fingerprint attributes include items such as IP address, location, and screen size.

Each registered device has one device fingerprint. Because the user accesses the resource in different locations and on different devices, the user can have many registered devices.

The following process describes how risk assessment works:

1. The incoming device requests access to the resource.
2. The risk engine collects as many device fingerprint attributes as it can from the request device.
3. After the attributes are collected, the risk engine:
 - Determines the device fingerprint.
 - Calculates the risk score. The risk score
 - Is a number.
 - Represents the amount of risk that is associated with the incoming request.
 - Indicates the likelihood that the incoming request represents the user.
4. The risk engine:
 - Compares the incoming fingerprint with each registered device fingerprint.
 - Uses the attributes that are contained in the larger fingerprint for each comparison.
 - Calculates a risk score for each comparison.
5. To determine the final risk score, the risk engine:
 - Chooses the lowest risk score of the comparisons between the incoming fingerprint and the registered fingerprint.
 - Measures the final risk score against a threshold score or range that the administrator sets in a policy.
6. Depending on the way the administrator writes the policy, one of the following outcomes occurs:

Permit

The risk score for the incoming request is well below the threshold score. The user is granted access to the resource. For example, the risk score is 30, and the threshold score that is set by the administrator is 40.

Permit with obligation or authentication

The user is asked to complete an extra security measure, such as step up authentication. For example, the risk score is 40, and the policy that the administrator wrote requires users that operate devices with scores 30 - 90 to step up.

Deny

The risk score for the incoming request is above the threshold score or range. The user is denied access to the resource. For example, the risk score is 50, and the threshold score that is set by the administrator is 40.

The risk score is calculated through the following formula:

Risk Score = (total weight of mismatched attributes / total weight of all attributes) × 100

When the values that belong to the incoming device fingerprint and the registered device fingerprint are the same, the values are matched. When the values that belong to the incoming device fingerprint and the registered device fingerprint are not the same, the values are mismatched.

Sometimes, the fingerprints contain attributes that are not matched or mismatched. These attributes are called indeterminate attributes. When there are indeterminate attributes present, the following formula is used to calculate the risk score:

$$\text{Risk Score} = \left(\frac{\text{total weight of mismatched attributes}}{\text{total weight of all attributes} - \text{total weight of indeterminate attributes}} \right) \times 100$$

Scenarios

The following example scenarios demonstrate risk score calculation.

All three of the scenarios assume that the administrator

- Wrote a policy that specifies that any risk score at or below 40 is permitted, and any risk score above 40 is denied.
- Gave equal weight values to all of the attributes in the tables.
 - The attributes in the tables have the same weight value of 10.

Scenario 1: Authentication permitted

The total weight of the unequal device fingerprint values that belongs to one attribute is not significant enough to prohibit authentication.

The example information in the table is used to calculate the risk score.

Attribute names	Weight values	Incoming device fingerprint values	Registered device fingerprint values
colorDepth	10	32	32
deviceLanguage	10	en-US	en-US
devicePlatform	10	Win32	Win32
http:userAgent	10	Mozilla/5.0 (Windows NT 6.1; WOW64; rv:15.0) Gecko/20120427 Firefox/15.0a1	Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/28.0.1468.0 Safari/537.36
ipAddress	10	42.29.144.5	42.29.144.5
screenHeight	10	1080	1080
screenWidth	10	1920	1920

- All of the device fingerprint values match except for the incoming device fingerprint value and existing device fingerprint value for http:userAgent.
- Because http:userAgent is the only attribute that has any mismatched values, the total weight of the mismatched attributes is 10.
- The total weight of all of the attributes is 70 because each attribute has a weight value of 10.
- According to the risk score calculation formula: $(10/70) \times 100 = 14$. Therefore, the risk score is 14.
- Because the risk score is below 40, authentication is permitted.

Scenario 2: Authentication denied with multiple significant attributes

The total weight of the unequal device fingerprint values that belongs to 6 out of 7 of the attributes is significant enough to prohibit authentication.

The example information in the table is used to calculate the risk score.

Attribute names	Weight values	Incoming device fingerprint values	Registered device fingerprint values
colorDepth	10	24	32
deviceLanguage	10	en-US	en-US
devicePlatform	10	Linux	Win32
http:userAgent	10	Mozilla/5.0 (X11; Linux i686 (x86_64)) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/27.0.1453.93 Safari/537.36	Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/28.0.1468.0 Safari/537.36
ipAddress	10	9.53.18.164	42.29.144.5
screenHeight	10	1050	1080
screenWidth	10	1680	1920

- None of the device fingerprint values match except for the incoming device fingerprint value and existing device fingerprint value for deviceLanguage.
- Because all of the attributes except for deviceLanguage have mismatched values, the collective weight of the mismatched attributes is 60.
- The total weight of all of the attributes is 70 because each attribute has a weight value of 10.
- According to the risk score calculation formula: $(60/70) \times 100 = 86$. Therefore, the risk score is 86.
- Because the risk score is above 40, authentication is denied.

Scenario 3: Authentication denied with one significant attribute

The total weight of the unequal device fingerprint values that belongs to one attribute is significant enough to prohibit authentication.

The example information in the table is used to calculate the risk score.

Attribute names	Weight values	Incoming device fingerprint values	Registered device fingerprint values
devicePlatform	5	Android	Android
geoLocation	85	51.499444, -0.1275, 10	30.283611, -97.7325, 10
screenHeight	5	800	800
screenWidth	5	480	480

- In addition to the previous assumptions, this scenario prohibits any distance greater than 40 kilometers.
- All of the device fingerprint values match except for the incoming device fingerprint value and the existing device fingerprint value for the geoLocation attribute. The geoLocation attribute contains the values that the risk engine uses to calculate the distance between the incoming device fingerprint and the registered device fingerprint. In this instance, the distance between the two device fingerprints is 7909 kilometers.
- Because the geoLocation attribute is the only attribute with mismatched values, the weight of the mismatched attributes is 85.

- The total weight of all of the attributes is 100 because the geoLocation attribute has a weight value of 85. devicePlatform, screenHeight, and screenWidth each have weight values of 5.
- According to the risk score calculation formula: $(85/100) \times 100 = 85$. Therefore, the risk score is 85.
- Because the risk score is above 40, authentication is denied.

Note: Authentication can be denied if the incoming fingerprint value and registered device fingerprint value for just one attribute indicate a large enough discrepancy. In this scenario, the distance between the incoming device fingerprint value and the registered device fingerprint value is too large for authentication to be permitted.

Risk reports

Risk reports illustrate the comparison between the incoming device fingerprint and the registered device fingerprint for each risk score calculation.

Each risk report contains the incoming device fingerprint that attempts to access a protected resource as compared to the registered device fingerprint. Analysis of each risk report provides the following information about the corresponding risk score calculation:

- Outcome of the risk score calculation.
- Matcher that the risk engine used for the comparison between the device fingerprints.
- Attributes that were compared between the incoming fingerprint and the registered fingerprint.
- Other calculations that were made during the risk score determination.

The administrator can use risk reports to complete the following objectives:

- Write more effective risk profiles.
- Monitor system activity.

Related concepts:

Risk score calculation

Risk score calculation is the process by which the risk engine determines a risk score. The *risk score* demonstrates the level of risk that is associated with permitting a request to access the resource. This risk score is compared to a *threshold score* that is set in a policy. A decision is made based on the result of this comparison.

Related tasks:

Accessing risk reports

The administrator can access the risk reports that are generated during risk score calculation scenarios.

Configuring risk reports

The risk engine generates risk reports to analyze risk score calculation scenarios. The administrator can configure the `riskEngine.reportsEnabled` property and the `riskEngine.reportsMaxStored` property to enable the risk engine to store a specified number of these reports.

About this task

The following properties are necessary for the generation of risk reports:

riskEngine.reportsEnabled

Enables the risk engine to produce risk reports after the risk engine generates the risk score for each incoming request.

The riskEngine.reportsEnabled property defaults to disabled, so the administrator must configure riskEngine.reportsEnabled to enable risk reporting.

riskEngine.reportsMaxStored

Enables the risk engine to store a maximum number of risk reports as specified by the administrator.

The administrator can set the riskEngine.reportsMaxStored property. However, the property defaults to five reports. The risk engine deletes the oldest report when it produces a report that exceeds the maximum number of reports that it is specified to store.

To configure the riskEngine.reportsEnabled property, complete the following steps.

Procedure

1. Log in to the local management interface.
2. Click **Secure Mobile Settings**.
3. Click **Advanced Configuration**.
 - Configure the riskEngine.reportsEnabled property
 - a. Under **Key**, find riskEngine.reportsEnabled.
 - b. Click the edit button.
 - c. Complete one of the following actions:

Enable risk reporting

- 1) Select the **Enabled** box.
- 2) Click **Save**.

Note: Enabling risk reporting sets the riskEngine.reportsEnabled property to **true**.

Disable risk reporting

- 1) Clear the **Enabled** box.
- 2) Click **Save**.

Note: Disabling risk reporting sets the riskEngine.reportsEnabled property to **false**.

- Configure the riskEngine.reportsMaxStored property
 - a. Under **Key**, find riskEngine.reportsMaxStored.
 - b. Click the edit button.
 - c. Specify the number of reports for the risk engine to store.
 - d. Click **Save**.

Related tasks:

Accessing risk reports

The administrator can access the risk reports that are generated during risk score calculation scenarios.

Accessing risk reports

The administrator can access the risk reports that are generated during risk score calculation scenarios.

About this task

The administrator can access the risk reports specific to your business by completing the following steps:

Procedure

1. Log in to the local management interface.
2. Click **Monitor Analysis and Diagnostics**.
3. Click **Application Log Files**.
4. Click the + next to the **mga** folder.
5. Click the + next to the **risk_reports** folder. The administrator can access the risk reports after the risk engine generates the first report and the **risk_reports** folder is created.
6. Find the necessary report, and click **View**.

Related concepts:

Risk reports

Risk reports illustrate the comparison between the incoming device fingerprint and the registered device fingerprint for each risk score calculation.

Related tasks:

Configuring risk reports

The risk engine generates risk reports to analyze risk score calculation scenarios. The administrator can configure the `riskEngine.reportsEnabled` property and the `riskEngine.reportsMaxStored` property to enable the risk engine to store a specified number of these reports.

Chapter 3. Attributes

Attributes specify the context of a request that you want to be evaluated as part of an access decision. For example, an attribute might be information in a request, such as a user name, or information in an external source, such as a user's age in a user registry or an information type in a database.

Requests contain one or more of the following categories of attributes:

Action

Indicates the user action.

Environment

Indicates when and how the user is trying to access the resource.

Resource

Gives information about what the user is trying to access.

Subject

Indicates who is trying to access the resource.

When you author policies or risk profiles, you select attributes that you want to be evaluated. Commonly used attributes are predefined. You can also create your own to meet the needs of your environment.

Managing attributes

Attributes represent unique information about a request. You use attributes in policy rules and risk profiles to match the attributes in a request. You can view, add, modify, and delete attributes.

Procedure

1. Log in to the local management interface.
2. Click **Secure Mobile Settings**.
3. Under **Policy**, click **Attributes**.
4. Perform one or more of the following actions:

View and filter attributes.

Take any of the following actions to filter your view:

- Select the  **Details View** to view attribute name, category, and data type.
- Select the  **List View** to view only the name of the attribute.
- Type a term, such as an attribute name, category, or data type in the **Filter** field to list attributes that use that term.

Note: The filter searches all attribute properties fields, including descriptions, for the alphanumeric characters you type in the **Filter** field. For example, if you type header in the **Filter** field, all attributes that contain header in their properties are shown in the attributes list. Click x to clear the **Filter** field.

- Sort the attribute list by column with the up or down arrow on each column. For example, you can view the list of attributes that are sorted by the **Data type** column in ascending order by clicking the up arrow.

Add an attribute.

- Click .
- Complete the properties for the attribute.

Note: The attribute name must begin with an alphabetic character. Do not use control characters, leading and trailing blanks, and the following special characters ~ ! @ # \$ % ^ & * () + | ` = \ ; " ' < > ? , [] { } / anywhere in the name.

- Click **Save**.

Modify an attribute.

Attention: Ensure that the modification does not affect a current policy or configuration. If you modify an attribute that is in-use, the policy or configuration that uses the attribute might stop working.

- Click .
- Complete the properties for the attribute.

Note: The attribute name must begin with an alphabetic character. Do not use control characters, leading and trailing blanks, and the following special characters ~ ! @ # \$ % ^ & * () + | ` = \ ; " ' < > ? , [] { } / anywhere in the name.

- Click **Save**.

Delete an attribute.

- Select an attribute or press and hold the Ctrl key and select multiple attributes to remove.
- Click . Confirm the deletion. Click **OK** to continue or click **Cancel**.

The attribute is removed.

- When you add, modify or delete an attribute, a message indicates that there are changes to deploy. If you are finished with the changes, deploy them.

For more information, see Chapter 12, “Deploying pending changes,” on page 109.

Related reference:

“Predefined attributes” on page 23

The IBM Security Access Manager for Mobile appliance uses attributes to provide information about users and devices that try to access a protected resource. The appliance also includes a set of commonly used attributes called *predefined attributes*.

Updating location attributes

To define policy that is based on geolocation, you must update the geolocation database with appropriate location and IP data.

About this task

When a request is received, a GeoLocator policy information point (PIP) determines the location of the device that made the request. The device IP address as determined by the point of contact server is the input to the PIP. The PIP reads the geolocation database to determine the device location.

All location attributes stored in the database are shown as environment attributes that you can use to author policies.

Location attributes include:

- Country
- State or region
- City

Attention: Sample data is included in the geolocation database. However, this sample data cannot be used in a production environment. Use the sample files for IPv4, IPv6, or both to create your own file. To locate the files, log in to the local management interface and click **Manage System Settings > Tools - File Downloads**. Then expand **mga > cba > geolocation**.

Procedure

1. Obtain or create an appropriate geolocation data file in ZIP format. The file or files you must use depend on whether you want support for IPv4 addresses, IPv6 addresses, or both.
 - **For IPv4:** The file must contain two CSV files. One file contains all of the possible locations and the other contains the IP blocks and their corresponding locations.

Locations file

Each line in the locations file corresponds to one location and is in the following format:

```
location id,country,region,city,,,,,
```

Attention: Note the 5 commas after city. You must include these commas in your locations file.

location id

A unique integer value that identifies the location.

country

A two-letter country code. For assistance with locating a country code, see geoCountryCode in “Predefined attributes” on page 23.

region A two-character region code. For assistance with locating a region code, see geoRegionCode in “Predefined attributes” on page 23.

city The name of a city.

The locations file must have Location in its file name. The sample provided is named: GeoLiteCity-Location.csv

IP blocks file

Each line in the IP blocks file corresponds to one IP block and is in the following format:

startip, endip, location id

startip

The first IP address in the block that is represented as an integer.

endip

The last IP address in the block that is represented as an integer.

location id

The integer that is defined in the locations file that corresponds with the IP block.

The IP blocks file must have Blocks in its name. The sample provided is named: GeoLiteCity-Blocks.csv

Attention: Ensure that the CSV files contain all of the data that you want to load in the database. When you import the file, the existing data is removed and replaced with the data in the file.

- **For IPv6:** The file must contain one CSV file that contains all of the location and IP block information.

Each line in the file corresponds to one location and IP block combination:

```
startip string, endip string, startip int, endip int, country,  
region, city,,,,,
```

Attention: Note the 5 commas after city. You must include these commas in your locations file.

startip string

The first IP address in the block that is represented as a hexadecimal string. For example, a IPv6 string might be
2001:200:ffff:ffff:ffff:ffff:ffff:ffff

endip string

The last IP address in the block that is represented as a hexadecimal string.

startip integer

The first IP address in the block that is represented as an integer. For example, the IPv6 integer that corresponds to startip
2001:200:ffff:ffff:ffff:ffff:ffff:ffff might be
42540528806023212578155541913346768895.

endip integer

The last IP address in the block that is represented as an integer.

country

A two-letter country code. For assistance with locating a country code, see geoCountryCode in “Predefined attributes” on page 23.

region

A two-character region code. For assistance with locating a region code, see geoRegionCode in “Predefined attributes” on page 23.

city

The name of a city.

The file must have v6 in its file name. The sample provided is named: GeoLiteCityv6.csv

2. Create a file in ZIP format that contains the files you want to upload. For example, if you want to use both IPv4 and IPv6, include all the CSV files that you created in one ZIP formatted file.

3. Log in to the local management interface.
4. Click **Manage System Settings**.
5. Under **Updates and Licensing**, click **Geolocation Database**.
6. Click **Import**.
7. Select the geolocation file in ZIP format.
8. Click **Import**.

Note: Importing the data can take more than 20 minutes to complete.

9. Click **Refresh Status** to check the import process. When the process is complete, the status says **Loaded**.

Attribute properties

When you add or modify an attribute, you specify properties that make that attribute unique.

Adding attributes

Specify the following properties when you add an attribute:

Name A unique name for the attribute.

Description

A description of the attribute.

Identifier

The internal name of the attribute that is used in the generated XACML policy.

Issuer The identifier of the policy information point from which the value of the attribute is retrieved. If an attribute can be returned from multiple policy information points, the issuer property specifies which policy information point to use.

Note: Use this field only if you are using a policy information point. Otherwise, leave this field blank.

Type Indicates whether the attribute is used for policies or risk profiles or both. If neither check box is selected, the attribute is not available for policies or risk profiles.

Category

The part of the XACML request that the attribute value comes from.

Data type

The type of values that the attribute can handle. In a policy rule with an attribute, the data type indicates how the attribute can be compared to a value. In a risk profile, the risk matchers compare attribute values that have the same data type.

Matcher

An attribute matcher compares the values of a specified attribute in the incoming device fingerprint with the existing device fingerprint of the user.

Storage Domain

The storage domain indicates whether the attribute is stored as a device, session, or behavior attribute.

Device fingerprint data

Consists of attributes that are stored when a device is registered. The incoming device fingerprint is compared against this stored repository of trusted device fingerprints.

Session data

Consists of the session attributes of the user that are stored temporarily until the session times out. However, if the device is registered, the session attributes are also stored as part of the device fingerprint. If session is selected, the attribute is collected in the user's session.

Behavior data

Is historic data that is stored in the database and used for behavior-based attribute matching. For example, the login timestamps of the user over the previous three months. If an attribute is included in a risk profile configuration and the storage domain is not specified, the default storage domain is device.

Modifying attributes

All the properties for an attribute are displayed. However, you can modify only some of attribute properties. Also, if an attribute is included in a policy, you cannot make further updates to the attribute.

You can modify the following properties:

Editable properties of predefined attributes**Storage Domain**

The storage domain indicates whether the attribute is stored as a device, session, or behavior attribute. If session is selected, the attribute is collected in the user's session. If an attribute is included in a risk profile configuration and the storage domain is not specified, the default storage domain is device.

Editable properties of custom attributes

Name A unique name for the attribute.

Description

A description of the attribute.

Identifier

The internal name of the attribute that is used in the generated XACML policy.

Issuer The identifier of the policy information point from which the value of the attribute is retrieved. If an attribute can be returned from multiple policy information points, the issuer property specifies which policy information point to use.

Note: Use this field only if you are using a policy information point. Otherwise, leave this field blank.

Type Indicates whether the attribute is used for policies or risk profiles or both. If neither check box is selected, the attribute is not available for policies or risk profiles.

Category

The part of the XACML request that the attribute value comes from.

Data type

The type of values that the attribute can handle. In a policy rule with an attribute, the data type indicates how the attribute can be compared to a value. In a risk profile, the risk matchers compare attribute values that have the same data type.

Matcher

An attribute matcher compares the values of a specified attribute in the incoming device fingerprint with the existing device fingerprint of the user.

Storage Domain

The storage domain indicates whether the attribute is stored as a device, session, or behavior attribute. If session is selected, the attribute is collected in the user's session. If an attribute is included in a risk profile configuration and the storage domain is not specified, the default storage domain is device.

Related tasks:

"Managing attributes" on page 17

Attributes represent unique information about a request. You use attributes in policy rules and risk profiles to match the attributes in a request. You can view, add, modify, and delete attributes.

Predefined attributes

The IBM Security Access Manager for Mobile appliance uses attributes to provide information about users and devices that try to access a protected resource. The appliance also includes a set of commonly used attributes called *predefined attributes*.

Attribute categories

Categories indicate the types of information that attributes convey. Attribute categories include:

Action

Indicates the user action.

Environment

Indicates when and how the user is trying to access the resource.

Resource

Gives information about what the user is trying to access.

Subject

Indicates who is trying to access the resource.

Types of predefined attributes

The predefined attribute types include:

Access policy

The administrator uses policy attributes to create policies.

Risk profile

The administrator uses risk attributes to create risk profiles.

Data types of predefined attributes

Each predefined attribute has a data type. Data types are classifications that identify the possible values for each type of attribute. Possible data types include the following values:

Boolean

Condition that refers to two possible values:

- True
- False

Date Date of the request.

Integer

Number that can be written without a fractional or decimal component.

String Sequence of characters.

Time Time of the request.

X500Name

Values with distinguished names.

Source types of predefined attributes

Source types indicate the source of each attribute. The source types include:

Active Collected by the attribute collection service. The administrator must add JavaScript to the application so that active attributes can be collected. For example: system fonts.

Derived

Generated by a policy information point (PIP). For example: risk score.

Passive

Collected from the browser by the external authorization service (EAS) and placed into an XACML request. Attributes with this source type are collected by the policy enforcement point (PEP) without installing more software or challenging the client to provide more details. For example: user-agent HTTP header and client IP address.

Sources of predefined attributes

Sources indicate where the attributes originate from. The sources include:

Attribute collection service

Collects information about the user device such as browser information, the operating system of the device, and the language of the device.

Consent external authentication interface

Asks the user for a device registration decision.

Device fingerprint count PIP

Counts the number of devices that are associated with the user.

Geolocation PIP

Looks up the location of the user that is based on the IP address.

HTTP headers

Provides information about the request.

IP reputation PIP

Generates the IP reputation. See IP reputation.

Risk engine

Generates the risk score. See Risk score calculation.

System time

Keeps the time of the system.

Security Access Manager credential

Collects information about the user from Security Access Manager.

Action attributes**action**

Indicates the action that is performed on the resource and includes one of the following values:

- CONNECT
- DELETE
- GET
- HEAD
- OPTIONS
- PATCH
- POST
- PUT
- TRACE

Type	Data type	Source type	Source
Access policy	String	Passive	HTTP headers

Environment attributes**accessTime**

Contains the record of the past login times that belongs to the user.

It uses the ISO 8601 standard: *YYYY-MM-DD'T'hh:mm:ss'Z'*.

For example: 2013-06-07T04:02:33Z

Type	Data type	Source type	Source
Risk profile	Time	Passive	System time

browserPlugins

Indicates all of the installed plug-ins in the browser in a comma-separated list.

For example: Shockwave Flash,Chrome Remote Desktop Viewer,Widevine Content Decryption Module,Native Client,Chrome PDF Viewer,Java(TM) Plug-in 1.7.0,Citrix Receiver for Linux

Type	Data type	Source type	Source
Risk profile	String	Active	Attribute collection service

colorDepth

Indicates the bit depth of the color palette of the device.

For example: 32

Type	Data type	Source type	Source
Risk profile	Integer	Active	Attribute collection service

currentDate

Indicates the date of the request.

It uses the format: *YYYY-MM-DDzzzzzz*.

For example: 2013-05-20-06:00

Type	Data type	Source type	Source
Access policy	Date	Passive	System time

currentTime

Indicates the time of the request.

It uses the format: *hh:mm:sszzzzzz*.

For example: 12:15:26+01:00

Type	Data type	Source type	Source
Access policy	Time	Passive	System time

deviceFonts

Indicates all of the installed fonts on the device in a comma-separated list.

For example: Arial, Dingbats, Georgia, Tahoma, Times New Roman, Verdana

Type	Data type	Source type	Source
Risk profile	String	Active	Attribute collection service

deviceLanguage

Indicates the language of the device.

It uses the Internet Engineering Task Force (IETF) language tag standard.

For example: en-US

Type	Data type	Source type	Source
Risk profile	String	Active	Attribute collection service

devicePlatform

Indicates the operating system of the device.

For example: Linux x86_64

Type	Data type	Source type	Source
Risk profile	String	Active	Attribute collection service

geoCity

Indicates the city in which the device is located.

For example: Austin

Type	Data type	Source type	Source
Access policy, Risk profile	String	Derived	Geolocation PIP

geoCountryCode

Indicates the country in which the device is located.

It uses the two letter country code according to the ISO-3166-1 standard.

For example: The country code for the United States is US.

For a list of country codes, see the International Organization for Standardization homepage. To find your two letter country code:

1. On the right side of the page, click **ISO 3166**.
2. In the **HTML, Text and XML versions** section, choose a format for the list of country codes.

Type	Data type	Source type	Source
Access policy, Risk profile	String	Derived	Geolocation PIP

geoLocation

Indicates the latitude, longitude, and accuracy of the requesting device that is obtained with the W3C geolocation standard.

It uses the format: *latitude, longitude, accuracy*

For example: 60.170833, 24.9375, 98

Type	Data type	Source type	Source
Risk profile	String	Active	Attribute collection service

geoRegionCode

Indicates the state or province in which the device is located.

It uses the two letter code for the region.

For the United States and Canada, the region codes are according to the ISO-3166-2 standard.

For example: The region code for the province of Nova Scotia in Canada is NS.

For all other countries, the region codes are according to the FIPS 10-4 standard. Your region code is the last pair of characters in the four-character code that is assigned to your region. For a list of region codes, see the National Geospatial Intelligence Agency homepage. To find your two-letter region code:

1. On the left side of the page, click **Country codes**.
2. In the **Geopolitical Entities and Codes (Formerly FIPS PUB 10-4)** section, find the most recent update and choose a format for a list of countries and region codes.

For example: The region code for Queensland in Australia is 04.

Type	Data type	Source type	Source
Access policy, Risk profile	String	Derived	Geolocation PIP

http:accept

Indicates the acceptable content types.

For example: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Type	Data type	Source type	Source
Risk profile	String	Passive	HTTP headers

http:acceptEncoding

Indicates the acceptable encoding types.

For example: gzip, deflate

Type	Data type	Source type	Source
Risk profile	String	Passive	HTTP headers

http:acceptLanguage

Indicates the acceptable human languages for response.

For example: en-US,en;q=0.8

Type	Data type	Source type	Source
Risk profile	String	Passive	HTTP headers

http:host

Indicates the host that is requested.

Type	Data type	Source type	Source
Access policy	String	Passive	HTTP headers

http:uri

Indicates the requested resource.

It includes the query string.

For example: /MGA/sps/ac/rest

Type	Data type	Source type	Source
Access policy	String	Passive	HTTP headers

http:userAgent

Indicates the user agent.

For example: Mozilla/5.0 (X11; Linux i686 (x86_64)) AppleWebKit/537.17 (KHTML, like Gecko) Chrome/24.0.1312.57 Safari/537.17

Type	Data type	Source type	Source
Risk profile	String	Passive	HTTP headers

scheme

Indicates the protocol that is used to access the resource. For example: http, https, or ftp.

Type	Data type	Source type	Source
Access policy	String	Passive	HTTP headers

screenAvailableHeight

Indicates the height of the screen of the requesting device without the toolbar.

For example: 1025

Type	Data type	Source type	Source
Risk profile	Integer	Active	Attribute collection service

screenAvailableWidth

Indicates the width of the screen of the requesting device without the toolbar.

For example: 1920

Type	Data type	Source type	Source
Risk profile	Integer	Active	Attribute collection service

screenHeight

Indicates the height of the screen of the requesting device.

For example: 1080

Type	Data type	Source type	Source
Risk profile	Integer	Active	Attribute collection service

screenWidth

Indicates the width of the screen of the requesting device.

For example: 1920

Type	Data type	Source type	Source
Risk profile	Integer	Active	Attribute collection service

userConsent

Indicates the registration decision of the user.

A registered device is true. An unregistered device is false.

Type	Data type	Source type	Source
Access policy	Boolean	Active	Consent external authentication interface

Resource attributes**resource**

Indicates the resource on which the action must be taken.

It does not include the query string and is usually the same as the **http:uri** attribute.

For example: /MGA/sps/ac/rest

Type	Data type	Source type	Source
Access policy	String	Passive	HTTP headers

Subject attributes

authenticationLevel

Numeric value that specifies the authentication level of a user.

It increases as the levels of authentication that belong to the user increase.

For example: A possible authentication level is 2.

Type	Data type	Source type	Source
Access policy	Integer	Passive	Security Access Manager credential

authenticationMechanism

Indicates the location of the user registry that belongs to the requesting user.

For example: LDAP Registry

Type	Data type	Source type	Source
Access policy, Risk profile	String	Passive	Security Access Manager credential

authenticationMethod

Indicates the method of authentication that is used, such as password, and includes the following values:

- ext-auth-interface
- password
- ssl
- token-card
- unauthenticated

Type	Data type	Source type	Source
Access policy, Risk profile	String	Passive	Security Access Manager credential

authenticationTypes

Indicates the types of authentication that the user has completed during the current authenticated session. The authenticationTypes values consist of the unique identifiers for the predefined obligations that perform user authentication. Only the following values are valid:

- urn:ibm:security:obligation:register_device
- urn:ibm:security:authentication:asf:otp
- urn:ibm:security:authentication:asf:hotp
- urn:ibm:security:authentication:asf:totp
- urn:ibm:security:authentication:asf:rsa
- urn:ibm:security:authentication:asf:email
- urn:ibm:security:authentication:asf:sms
- urn:ibm:security:authentication:asf:consent_register_device

Type	Data type	Source type	Source
Access policy	String	Passive	Security Access Manager credential

groups

Lists the groups of which the user is a member.

The **groups** attribute is a multivalued attribute. See Attribute properties.

For example: engineering

Type	Data type	Source type	Source
Access policy	String	Passive	Security Access Manager credential

groupsDN

Lists the groups of which the user is a member.

The **groupsDN** attribute has a distinguished name format and is a multivalued attribute.

For example: CN=engineering, O=ibm, C=us

Type	Data type	Source type	Source
Access policy	X500Name	Passive	Security Access Manager credential

ipAddress

Indicates the string representation of the IP address.

Specify one of the following IP address formats depending on the network type:

IPv4 format

192.9.200.1

IPv6 format

FEDC:BA98:7654:3210:FEDC:BA98:7654:3210

Type	Data type	Source type	Source
Access policy, Risk profile	String	Passive	Security Access Manager credential

ipReputation

Lists any combination of the following classifications whose IP reputation score is at or above the threshold:

- Anonymous Proxies
- Botnet Clients
- Dynamic IPs
- Malware
- Scanning IPs
- Spam

The user can either configure this threshold or use the default threshold of 50.

Type	Data type	Source type	Source
Access policy	String	Derived	IP reputation PIP

qop

Indicates the quality of protection information and includes the following valid values:

- None
- Integrity
- Privacy

Type	Data type	Source type	Source
Access policy, Risk profile	String	Passive	Security Access Manager credential

registeredDeviceCount

Indicates the number of devices that the user registers.

Type	Data type	Source type	Source
Access policy	Integer	Derived	Device fingerprint count PIP

riskScore

Indicates the calculated risk score, which is a number 0 - 100.

The risk score indicates the level of risk that is associated with authenticating a request device. See Risk score calculation.

Type	Data type	Source type	Source
Access policy	Integer	Derived	Risk engine

userDN

Indicates the distinguished name for the user in a registry.

For example: CN=Mark Smith,o=IBM, L=Austin, S=Texas, C=US

Type	Data type	Source type	Source
Access policy, Risk profile	X500Name	Passive	Security Access Manager credential

username

Identifies the user.

For example: sec_master

Type	Data type	Source type	Source
Access policy, Risk profile	String	Passive	Security Access Manager credential

Configuring the hash algorithm for attribute storage

Hashing encodes a character string as a fixed-length bit string for comparison. Context-based access hashes certain attributes by default. You can change the hash algorithm and specify additional attributes that you want to hash.

About this task

By default, when attributes are stored in the context-based access database, the attributes that exceed the maximum length according to the schema are hashed. You can also specify any other attribute that you require to be hashed. For example, you might want to hash values that are considered confidential or private.

The default hash algorithm that context-based access uses for storing these attributes is SHA256. Context-based access also uses the default when the hash algorithm is not configured properly. You can specify any other hash algorithm that Java Security supports.

Procedure

1. Log in to the local management interface.
2. Click **Secure Mobile Settings**.
3. Under **Manage**, click **Advanced Configuration**.
4. Under **Key**, find the name of the property that you must work with.
5. Take one of the following actions:
 - Configure the `attributeCollection.attributesHashEnabled` property.

- a. Click the edit icon .
- b. Enter the Identifier names of the attributes that you want the `attributeCollection.attributesHashEnabled` property to hash. For example: `urn:ibm:security:environment:http:userAgent`, `urn:ibm:security:environment:deviceFonts`, `urn:ibm:security:environment:browserPlugins`

To find the list of attributes that context-based access can hash, complete the following steps:

- 1) Log in to your local management interface
- 2) Click **Secure Mobile Settings**.
- 3) Under **Policy**, click **Attributes**.
- 4) Select the name of an attribute, and click the modify attributes icon



- 5) Under **Modify Attribute**, find the Identifier of the attribute.
- 6) Use the Identifier of the attribute in your list of attributes that you want context-based access to hash.
- 7) Click **Cancel** to exit.

- Configure the `attributeCollection.hashAlgorithm` property.
 - a. Click the edit icon .
 - b. Set the value for the `attributeCollection.hashAlgorithm` property to one of the following values:
 - SHA1
 - SHA512
 - SHA256
 - c. Click **Save**.

6. When you make changes to the properties, the appliance displays a message that there are undeployed changes. If you are finished making changes, deploy them.

Dynamic attributes

You can capture additional context-based access information in dynamic attributes that you create or customize.

Attributes used to calculate risk score come from various sources. Context-based access can capture and use many default attributes. However, you might require more information to be captured. This data can come from:

- Data that is gathered when a session is established that is sent to the Attribute Collector Service. This data persists during the session and is stored in the session table in context-based access. This type of information is gathered by the `info.js` file.

- Data that is derived from a particular request for which authorization is being performed. This data can be stored in either the session or behavior tables.
- Data that is provided by a PIP during the authorization decision. This data can be stored in either the session or behavior tables.

This type of data can be captured in *dynamic attributes*.

Related concepts:

“Scenarios for adding and manipulating attributes”

You can dynamically add attributes or manipulate the values of attributes for various purposes.

Scenarios for adding and manipulating attributes

You can dynamically add attributes or manipulate the values of attributes for various purposes.

The following topics provide specific instructions to create or update the JavaScript file for dynamic attributes:

1. “Coding the dynamic.attributes.js file”
2. “Updating and deploying the dynamic.attributes.js file” on page 36

The following scenarios show examples of the data you can dynamically capture:

Creating a device.name attribute

In this scenario, you can create a device name attribute that a user can easily identify. While configuring a system that only has hybrid applications, you can gather a few more attributes while using the `info.js` active collection library. For example, if you combine three separate device attributes that the client sends, you can uniquely identify the device. The attributes are:

- Device Model
- Device Model Version
- Device UUID (unique ID)

Combining these attributes into a single attribute with a zero device weight results in a meaningful attribute name that you can use to display the device. You can accomplish this by providing a server-side JavaScript rule that runs when an attribute collection service request arrives.

Adding the last risk score as a session attribute to be used by a third party

The risk score is a derived attribute using a combination of other attributes. It can be useful for third party authentication services.

In this scenario, you can implement a JavaScript rule that captures the current risk score and stores it as a session attribute. Later, when a third-party application is determining possible further action based on the current context, it can make a remote call to the attribute collection service using the REST API call to retrieve the sessions attributes, such as risk score.

Coding the dynamic.attributes.js file

Add JavaScript code to the `dynamic.attributes.js` file to dynamically add attributes or modify the values of attributes.

Before you begin

- Review the “Scenarios for adding and manipulating attributes” on page 34.
- Determine which attributes you want to modify and how to store them. Create a function for each one, which you will add to the `dynamic.attributes.js` file in the steps below.
- Deploy context-based access.

About this task

The `modifySessionAttributes` and `modifyBehaviorAttributes` functions are the only functions called by the context-based access processing. Therefore, any function you add must be called within one of these functions in order to be executed.

Procedure

1. Create a JavaScript file or edit the sample `dynamic.attributes.js` file.
2. Define the necessary import packages for the APIs you use. For example:

```
importPackage(com.tivoli.am.rba.extensions);
importClass(Packages.com.tivoli.am.rba.attributes.AttributeIdentifier);
```
3. Use the `modifySessionAttributes` function to add or manipulate the session attributes.

```
function modifySessionAttributes(attributes, username, session)
```
4. Add code to work with session attributes.
 - a. Create an attribute identifier. For example, to use the risk score value, the following line is required:

```
var riskScoreIdentifier = new AttributeIdentifier("riskScore",
    "urn:ibm:security:subject:riskScore","Integer",
    "urn:ibm:security:issuer:RiskCalculator");
```
 - b. Use the attribute identifier to get the value of this attribute. For example, to use the risk score, the following line is required:

```
var riskScoreValue = session.getValue(riskScoreIdentifier);
```
 - c. Call the function you created to process the attributes. Define the function either at the beginning or at the end of the `dynamic.attributes.js` file.
 - d. Store your attribute to session storage. For example, to store the risk score, the following line is required:

```
attributes.put(riskScoreIdentifier, riskScoreValue);
```
5. Use the `modifyBehaviorAttributes` function to add or manipulate behavior attributes.

```
function modifyBehaviorAttributes(attributes, username, session)
```
6. Add code to work with behavior attributes.
 - a. Create an attribute identifier.
 - b. Use the attribute identifier to get the value of this attribute.
 - c. Call the function you created to process the attributes. Define the function either at the beginning or at the end of the `dynamic.attributes.js` file.
 - d. Store your attribute to behavioral storage.
7. Save the file.

Example

The following `dynamic.attributes.js` file is a sample that is available with the IBM Security Access Manager for Mobile product. The code in this example JavaScript file captures the risk score. This is the only way you can save the risk score value.

```
/**
 * This script is executed after each request is processed by risk engine.
 * The intent is to allow users to capture attributes that don't get captured
 * automatically by the system. The attributes captured here will be stored
 * in either the session storage or the behavior storage (i.e., usage data, historical)
 * area of RBA, or both. The risk profile configuration dictates where the
 * attributes will be stored by the system.
 *
 * For any RBA specific API, necessary packages need to be imported as shown in this example.
 */

/**
 * Import RBA packages necessary for the script to execute.
 */
importPackage(com.tivoli.am.rba.extensions);
importClass(Packages.com.tivoli.am.rba.attributes.AttributeIdentifier);

/**
 * @param username - current user's name
 * @param attributes - java.util.Map where the 'dynamic' values need to be captured by
 *                    this javascript file.
 * @param session - object containing current values visible to incoming request context
 */
function modifySessionAttributes(attributes, username, session) {

    // creates an identifier with the attribute's name, URI, datatype, and the issuer
    var riskScoreIdentifier = new AttributeIdentifier("riskScore", "urn:ibm:security:subject:riskScore",
        "Integer", "urn:ibm:security:issuer:RiskCalculator");

    // retrieve the risk score
    var riskScoreValue = session.getValue(riskScoreIdentifier);

    // set the risk score to be stored as a session attribute
    attributes.put(riskScoreIdentifier, riskScoreValue);
}

/**
 * @param username - current user's name
 * @param attributes - java.util.Map where the 'dynamic' values need to be captured
 *                    by this javascript file.
 * @param session - RBA's com.tivoli.am.rba.fingerprinting.IValueContainer object
 *                  containing current values visible to incoming request context
 */
function modifyBehaviorAttributes(attributes, username, session) {

    // store any behavior attributes here
}
```

What to do next

Update and deploy the `dynamic.attributes.js` file.

Updating and deploying the `dynamic.attributes.js` file

To implement the `dynamic.attributes.js` file on your appliance, update the file and then deploy the change.

Before you begin

- Code the `dynamic.attributes.js` file to create the customizations you require.
- To use a custom attribute in a policy, create the attribute. See “Managing attributes” on page 17.

You must set the following properties in the local management interface for the custom attribute:

- Specify a name for the attribute.
- Issuer: `urn:ibm:security:issuer:AttributeSession`
- Category: `Environment`

- Type: Policy
- Storage Domain: Session

Then, you must use the attribute you created in the rule section of a policy. See “Creating a policy” on page 76.

Procedure

1. Log in to the local management interface.
2. Click **Secure Mobile Settings**.
3. Under **Manage**, click **Template Files**.
4. Locate the existing `dynamic.attributes.js` file in the `C/ac/javascript_rules` directory.
5. Select the `dynamic.attributes.js` file to edit or import:
 - Click **Manage** > **Edit** to update the existing file in an editing window by using copy and paste from the file you created. Click **Save**.
 - Click **Manage** > **Import** to import the script file you created and replace the existing file. Click **Browse** to select the location of the file, and click **Import** to replace the existing `dynamic.attributes.js` file.
6. Deploy the changes.

Custom attributes for the authorization service

You can modify the `[azn-decision-info]` stanza of the WebSEAL configuration file to make other data available to the external authorization service (EAS).

You can define your own attributes, called *custom attributes*, to specify in policies and for decision making. Update the WebSEAL configuration file with the attribute information so that the EAS can use these attributes. The WebSEAL EAS uses this attribute information to create the runtime security services request.

To update the WebSEAL configuration file for custom attributes:

1. Update the `[azn-decision-info]` stanza with the custom attribute information.
2. Define the type of data and category that the attribute represents using the `[user-attribute-definitions]` stanza.

[azn-decision-info] stanza

Add extra information from the HTTP request to the authorization decision information.

Syntax

```
<attr_ID> = <http_info>
```

Description

This stanza defines any extra information that is available to the authorization framework when it makes authorization decisions. This extra information can be obtained from various elements of the HTTP request, namely:

- HTTP method
- HTTP scheme
- HTTP cookies
- Request URI
- HTTP headers

- POST data
- Client IP address

If the requested element is not in the HTTP request, no corresponding attribute is added to the authorization decision information.

Options

attr_ID

The identifier of the attribute that contains the HTTP information.

http_info

The source of the information. It can be one of the following values:

- method
- scheme
- uri
- client_ip
- header:<header-name>

where:

<header-name>

The name of the header that contains information for WebSEAL to add to the authorization decision information. For example, Host.

- cookie:<cookie-name>

where:

<cookie-name>

The name of the cookie that contains information for WebSEAL to add to the authorization decision information.

- post-data:<values>,

where the content and format of the <values> depends on the type of POST data.

See "WebSEAL support for POST data."

WebSEAL support for POST data

WebSEAL supports two types of POST data:

- Normal FORM data, which is the application/x-www-form-urlencoded content-type.

To add normal FORM data to the HTTP request, use the following format for this entry:

post-data:<post-data-name>

where the <post-data-name> is the name of the selected form data field in the request. WebSEAL adds the corresponding value for this field to the authorization decision information.

- JavaScript Object Notation (JSON) data, which is the application/json content-type. For more information about the JSON syntax, see <http://www.json.org>.

To search for a key in the JSON data and add its value to the HTTP request, use the following format:

post-data:/"<JSON-node-id>"[/"<JSON-node-id>"] [<JSON-array-idx>]]...

where:

"<JSON-node-id>"

The name of a node in the JSON data.

JSON data is essentially a hierarchy of name-value pairs. The forward slash character (/) that precedes each "<JSON-node-id>" identifies a level of the JSON hierarchy. You can repeatedly add [/<JSON-node-id>] elements to move through the JSON data hierarchy and identify the node that contains the value that you want WebSEAL to add to the authorization decision information.

Each <JSON-node-id> must be:

- Enclosed in double quotation marks.
- Preceded by a forward slash character (/).
- A case-sensitive match with a node in the JSON data hierarchy.

If WebSEAL does not find a matching node name in the POST data, no corresponding attribute is added to the authorization decision information.

<JSON-array-idx>

The contents of a node in the JSON data might be a JSON array. If you configure WebSEAL to search for a JSON node that contains an array, specify the array index of the value that you want WebSEAL to use. Use a base of 0. In other words, the first entry in the array has an index of 0.

Note: The <JSON-array-idx> is not enclosed in double quotation marks.

Usage notes:

- The square brackets ([]) in this syntax indicate an optional element. Do not include square brackets in your configuration entry. Similarly, the ellipsis (...) indicates that you can repeat the optional elements that precede it. Do not include the ellipsis in your configuration entry.
- WebSEAL returns only node values of the following JSON types:
 - String
 - Number
 - true or false
 - null

If the value of the selected node is not one of the types in this list, WebSEAL does not return it as authorization decision information.

Object and Array types cannot be added to the authorization decision information.

Usage

This stanza entry is optional.

Default value

None.

Example 1: Standard HTTP elements

```
HTTP_REQUEST_METHOD = method
HTTP_HOST_HEADER= header:Host
```

If these example configuration entries are set in the **[azn-decision-info]** stanza, WebSEAL adds the following attributes to the authorization decision information:

HTTP_REQUEST_METHOD

Contains the HTTP method.

HTTP_HOST_HEADER

Contains the data from the Host header.

Example 2: JSON POST data

For this example, consider the following JSON form data:

```
{ "userid": "jdoe",
  "transactionValue": "146.67",
  "accountBalances": {
    "chequing": "4345.45",
    "savings": "12432.23",
    "creditLine": "19999.12"
  }
}
```

The following configuration entries in the **[azn-decision-info]** stanza extract information from this JSON form data.

USERID = post-data:/"userid"

SAVINGS = post-data:/"accountBalances"/"savings"

The first entry prompts WebSEAL to search for the JSON node called **"userid"**. In this example, the value that is associated with the **"userid"** node is **jdoe**. WebSEAL adds this value to the HTTP request in an attribute called **USERID**.

When WebSEAL processes the second entry, it searches for a top-level JSON node called **"accountBalances"**. Under the **"accountBalances"** hierarchy, WebSEAL locates the **"savings"** JSON node. In the example data, the value that is associated with this node is **12432.23**. WebSEAL adds this value to the HTTP request in an attribute called **SAVINGS**.

WebSEAL adds the following attributes to the authorization decision information:

USERID

Contains the value **jdoe**.

SAVINGS

Contains the value **12432.23**.

Example 3: JSON POST data with a JSON array value

For this example, consider the following JSON form data:

```
{
  "userid": "jdoe",
  "transactionValue": "146.67",
  "accounts": [
    { "name": "chequing", "balance": "4350.45" },
    { "name": "savings", "balance": "4350.46" }
  ]
}
```

The following configuration entry is included in the **[azn-decision-info]** stanza:

SAVINGSBAL = post-data:/"accounts"/1/"balance"

WebSEAL processes this entry as follows:

1. Searches for a top-level node in the JSON data called **"accounts"**.
2. Locates the element in position 1 of the JSON array (base 0).
3. Searches for the **"balance"** name-value pair in this array element.
4. Adds the associated value to the authorization decision information.

In this example, WebSEAL adds the following attribute to the authorization decision information:

SAVINGSBAL

Contains the value 4350.46.

[user-attribute-definitions] stanza

Use the [user-attribute-definitions] stanza to modify the data type, the category, or both of a custom attribute.

Syntax

```
attr_ID.datatype = data_type
attr_ID.category = category_name
```

Description

Use the appropriate stanza entry syntax depending on if you want to set the data type or category of a custom attribute from the default values.

Options

attr_ID

Specify the attribute identifier for which you want to set the data type or category. The attribute ID must match the name that exists in the [azn-decision-info] stanza entry.

datatype *data_type*

Set the data type of an attribute from the default of string to a specified *data_type*. Your choices are:

- string
- boolean
- integer
- double
- time
- date
- x500name

category *category_name*

Set the category of an attribute from the default of Environment to a specified *category_name*. Your choices are:

- Environment
- Subject
- Action
- Resource

Usage

This stanza entry is not required.

Default value

The default value for data type is string.

The default value for category is Environment.

Example: Updating the data type for JSON data

If you defined a custom attribute in the [azn-decision-info] stanza as:

```
urn:example:company:txn:value = post-data:/"accountBalances"/"savings"
```

Then, you can set the data type of urn:example:company:txn:value to double by using the following stanza and entry:

```
[user-attribute-definitions]
urn:example:company:txn:value.datatype = double
```

Example: Updating the category for form data

If you defined a custom attribute in the [azn-decision-info] stanza as:

```
urn:example:company:txn:userid = post-data:userid
```

Then, you can set the category of urn:example:company:txn:userid to Subject by using the following stanza and entry:

```
[user-attribute-definitions]
urn:example:company:txn:userid.category = Subject
```

Setting the data type or category of a custom attribute

Set the data type or category of a custom attribute that is being passed to the runtime security services. Setting the data type or category ensures that the runtime security services use the data accurately when evaluating the policy.

About this task

The default data type of an attribute is string. The default category of an attribute is Environment. Use the following procedure to change the type and the category.

Procedure

1. Open the WebSEAL configuration file.
2. Optional: Create one or more entries in the [azn-decision-info] stanza for custom attributes, if they do not already exist.
3. Set the data type or category of a custom attribute by using the [user-attribute-definitions] stanza and entry. Add the stanza if it does not already exist.

data type

Define the data type by using the following syntax:

```
[user-attribute-definitions]
attr_ID.datatype = data_type
```

category

Define the category by using the following syntax:

```
[user-attribute-definitions]
attr_ID.category = category_name
```

The value you use for *attr_ID* must match the attribute identifier you defined in the [azn-decision-info] stanza.

See “[user-attribute-definitions] stanza” on page 41 for the list of values you can use for data type and category.

4. Save the file.
5. Restart the WebSEAL server for the changes to take effect.

Related reference:

“[user-attribute-definitions] stanza” on page 41

Use the [user-attribute-definitions] stanza to modify the data type, the category, or both of a custom attribute.

“[azn-decision-info] stanza” on page 37

Add extra information from the HTTP request to the authorization decision information.

Chapter 4. Attribute collection service

The attribute collection service is a Representational State Transfer (REST) service. It can collect web browser and location information from the user for calculating the risk score.

Process overview

The following process describes the attribute collection service and how to use it:

1. Make REST calls to store and delete attributes in the database. The initial request to the service receives a correlation ID. The correlation ID is used to make further REST calls.
2. Use JavaScript to collect the web browser attributes. You can place the HTML page that calls the JavaScript functions on any server.
 - Ajax collects information in the background. It does not slow down page loading.
 - You can make standard Ajax requests only to the same domain. With Cross Origin Resource Sharing (CORS), you can make Ajax requests across domains.
 - The CORS response header contains the settings for the following specifications:
 - The server from which requests are accepted.
 - The types of requests that are accepted.

Attributes that are configured as session attributes are collected automatically by the `info.js` file for risk score calculation.

Request types

GET and POST requests create a correlation ID to identify the session in the database. A correlation ID is a UUID that is stored in a cookie. The attribute collection service process uses the following request types:

GET Retrieves information about an attribute session from the database. GET requests are disabled by default. Requests use a URL with a REST path, such as: `https://webseal/mga/sps/ac/rest/UUID`.

POST Creates an attribute session in the database. POST requests use a URL such as `https://webseal/mga/sps/ac/UUID`.

The session attributes are sent as a JSON string with the request. In a response, the server sets a cookie that contains the correlation ID. For example, the `POST /sps/ac/9d37e806-24cf-4398-a3b9-d7f13fb2231f` request creates a session in the database with a UUID of `9d37e806-24cf-4398-a3b9-d7f13fb2231f`.

You can also configure the risk-based access properties to use an existing cookie

DELETE Deletes an attribute session from the database.

Risk-based access runtime properties

Use the local management interface to configure the risk-based access properties that are required for attribute collection service.

The following properties specify information about the attribute collection service:

attributeCollection.cookieName

Correlation ID used by the attribute collector.

Data type: String

Example:

ac.uuid

attributeCollection.requestServer

Request server for attribute collector. A list of the allowable hosts where the ajaxRequest can be sent from.

Data type: String List

Example:

https://rbademo.example.com,https://rbaemo2.example.com

attributeCollection.serviceLocation

Location of the attribute collector.

Data type: String List

Example:

http://rbademo.example.com/mga

attributeCollection.sessionTimeout

Number of seconds in which sessions stored in context-based access will automatically expire, unless updated. If any attribute in the session is updated, the session expiry is extended by the specified number of seconds configured in this property. The default is 3600 seconds.

Data type: Integer

Example:

3600 seconds

attributeCollection.enableGetAttributes

Enables the REST GET method to return attributes.

Data type: Boolean

Example:

False

attributeCollection.getAttributesAllowedClients

A comma-separated list of clients that are allowed to access the ACS REST GET method.

If this property is not set and attributeCollection.enableGetAttributes is set to true, anyone can access the GET method. If this property is set but attributeCollection.enableGetAttributes is set to false, this property is ignored.

Data type: String List

Example:

hostname1, hostname2

attributeCollection.hashAlgorithm

The algorithm that is used to create the hash.

Data type: String

Example:

SHA256

attributeCollection.attributesHashEnabled

A comma-separated list of attribute URI values that have been configured for hashing.

Data type: String List

Example:

urn:ibm:security:environment:http:userAgent,
urn:ibm:security:environment:deviceFonts,
urn:ibm:security:environment:browserPlugins

attributeCollection.authenticationContextAttributes

Comma-separated lists of attribute names to be collected when performing an authentication service obligation.

Data type: String List

Example:

authenticationLevel, http:host

JavaScript functions

Use the JavaScript functions in the C/ac/info.js file to make requests to the server. Include the info.js JavaScript file in the HTML landing page of your application. When info.js is loaded, it calls the following functions:

sendSession()

Makes a POST request to the delegate service.

The sendSession() function collects the web browser attributes and sends them to the server. They are stored in the database. Call this function when a user logs in.

deleteSession()

Makes a DELETE request for a specified correlation ID.

The POST request from the sendSession() returns a correlation ID. Based on the correlation ID, the deleteSession() function deletes the attributes from the database. Call this function when the user logs out or when the current session times out.

getLocation()

Detects the location of the device from which the requests are made. If the location information is sent to the server, call the getLocation() function before the sendSession() function. The following web browsers support the detection of location: Mozilla Firefox, Google Chrome, Opera, Apple Safari, and Microsoft Internet Explorer 9 and 10.

Note: For the JavaScript functions to work in Microsoft Internet Explorer, include the following statement in the HTML page from which you call the function. The following statement forces Microsoft Internet Explorer to use the standards mode:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

For configuration steps and examples, see “Configuring the attribute collection service.”

Configuring the attribute collection service

Before you can collect risk calculation information, you must specify the server and location of the collection service. You also must specify a JavaScript file to collect the session attributes.

Before you begin

Run the **isamcfg** tool to configure the runtime security services EAS. See the **isamcfg** command line reference information in the *IBM Security Access Manager for Mobile Configuration Guide*.

Review Chapter 4, “Attribute collection service,” on page 45.

Procedure

1. Optional: Configure the context-based access properties that you require for the attribute collection service. These properties are set by the **isamcfg** tool, but if you need to change them, use the following instructions:
 - a. Configure `attributeCollection.requestServer` to specify the server from which requests are received using the local management interface:
 - 1) Select **Secure Mobile Settings > Manage > Advanced Configuration**
 - 2) Find the `attributeCollection.requestServer` key in the list and click the edit icon. A new window displays the name and the current value.
 - 3) Edit the value of the request servers. The value is a space-separated list of WebSEAL host names from which requests are permitted. Host names must begin with `http://` or `https://`. For example, type `http://mywebsealhost.company.com`.
 - b. Configure `attributeCollection.serviceLocation` to specify the location using the local management interface:
 - 1) Select **Secure Mobile Settings > Manage > Advanced Configuration**
 - 2) Find the `attributeCollection.serviceLocation` key in the list and click the edit icon. A new window displays the name and the current value.
 - 3) Edit the value of the location. Specify the location as:
`https://host_name/webseal-junction-name`

For example, type `https://mywebsealhost.company.com/mga`.

You can configure any other attribute collection service attributes in a similar fashion, as necessary.

2. Add the URL of `info.js` to the `<head>` block in the HTML landing page of your application. The `info.js` file calls functions that are required to collect session attributes. Follow this format:

```
<script src="https://host_name/webseal-junction-name/sps/ac/js/info.js"></script>
```

Note: When the `info.js` file is included on an HTML page, attribute collection by Ajax calls can take time to complete. To avoid issues, attribute collection must end before moving away from the page. For example, if the attribute collection is still running, and a link is clicked, the policy fails to resolve session attributes. To prevent this issue, modify the JavaScript file to prevent the user from continuing until after the Ajax call completes.

Results

The basic configuration of the attribute collection service for context-based access is complete.

Chapter 5. Attribute matchers

An attribute matcher compares the values of a specified attribute in the incoming device fingerprint with the existing device fingerprint of the user. Context-based access uses the information that is returned by the attribute matchers to calculate the risk score.

In some scenarios, multiple attributes or a set of composite attributes must be matched. For example, longitude, latitude, and accuracy are three attributes that are related to location. In a given scenario, two device fingerprints are considered a match if the distance between two location points is not greater than a specified threshold value. In this scenario, the comparison of only the longitude attribute does not provide accurate results. The matcher must do a more complex comparison or composite matching, where it matches multiple attributes from both fingerprints.

The matcher either returns a match or a mismatch result. A mismatch increases the risk score that is based on the assigned weight of the attributes.

The matcher might not be used in the risk calculation in the following situations:

- The incoming device fingerprint does not contain the required attributes.
- The historical data is not available for a matcher to make a match or mismatch decision.

Risk-based access provides ready-to-use attribute matchers that compare composite attributes or analyze a range of attribute values. You can configure one or more of the attribute matchers that are described in the following sections.

Exact match matcher

The **exact_match** matcher checks whether the values of an attribute in a registered device and an incoming request exactly equal each other. Use this matcher if the more specialized matchers are not appropriate for the attribute.

Location matcher

The location matcher (**location_matcher**) checks whether the location of a device is within a specific distance from the previous known locations of the device. Configure the location matcher properties to specify the accuracy range and how to compare the location information.

Limitation: The retrieval of location attributes depends on the web browser and the settings that the user specifies in the web browser. The web browser must support the Geolocation API. An error might occur in some web browsers if a user tries to access a protected resource from a device with a wired internet connection.

The location-based analysis processes all three location attributes (longitude, latitude, and accuracy) collectively when it determines the match for the location. Though weights are assigned to all three attributes, the weight for only the longitude attribute is considered. The weights that are assigned to the supporting latitude and accuracy attributes are ignored.

The location matcher has two properties:

Comparison

Indicates how you want the attribute matcher to calculate the accuracy range of the location coordinates.

The following figure illustrates the closest points, midpoints, and farthest points of the accuracy ranges of two locations. In this figure, the circle represents the accuracy range and the center of the circle represents the location.

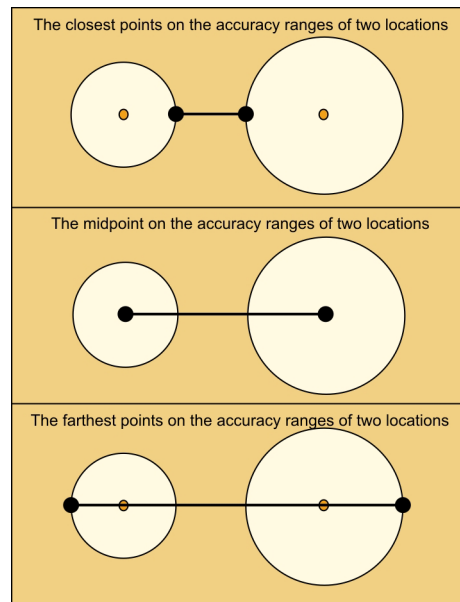


Figure 3. The closest points, midpoints, and farthest points on the accuracy ranges of two locations

Set the Comparison property to one of the following values:

- Specify the value as **closest** to calculate the distance between the closest points on the accuracy range of two locations. This calculation is the most restrictive calculation.
- Specify the value as **midpoint** to calculate the distance between the midpoints of the circles without considering accuracy.
- Specify the value as **farthest** to calculate the distance between the farthest points on the accuracy ranges of the two locations. This calculation is the least restrictive calculation.

Distance

The maximum distance between the new location and the historic locations. The unit of the numeric value is in kilometers. The default value is 40.

Login time matcher

The login time matcher (**login_time_matcher**) compares and analyzes the historical login time data of the user with the current login time of the user. You must configure the attributes and properties that are required for login time analysis. The login time matcher primarily detects the logins per session. The first of the

several access times that are captured within the session is considered the login time of the user. The result of the analysis determines the probability of a fraudulent user.

The login time matcher has one property:

Threshold

Indicates the probability that a user might log in at a particular time. Valid values are 0 to 1. The default value is .3. This default value indicates the probability that the user logs in approximately within an hour of the previous login times. If you set a lower value, the odds of the matcher returning true are higher and the risk score is lower. If you set a higher value, the odds of the matcher returning true are lower and risk score is higher. For example, if you set a value of 0.5, the matcher almost always returns false. The login time analysis collects data for eight login times before it provides input for risk score calculation.

IP address matcher

The IP address matcher (**ipaddr_matcher**) compares the IP address of a request with:

- A trusted list (inclusion list) of IP addresses
- An untrusted list (exclusion list) of IP addresses
- The historical IP addresses of the device
- The IP reputation of the device

The IP address matcher has the following properties:

Trusted addresses

IPv4 addresses

IP and Netmask: Specifies the IP address and its netmask to include. Include X.X.X.X as a value to compare the incoming IP address with the IP address with which the device is registered.

IPv6 addresses

IP and Prefix: Specifies the IP address and its prefix to include. Include X:X:X:X:X:X:X as a value to compare the incoming IP address with the IP address with which the device is registered.

Untrusted addresses

IPv4 addresses

IP and Netmask: Specifies the IP address and its netmask to exclude. Include X.X.X.X as a value to compare the incoming IP address with the IP address with which the device is registered.

IPv6 addresses

IP and Prefix: Specifies the IP address and its prefix to exclude. Include X:X:X:X:X:X:X as a value to compare the incoming IP address with the IP address with which the device is registered.

The IP address matcher returns one of the following decisions after it compares the incoming IP address with the IP address that belongs to the registered device:

MISMATCHED

The decision that the matcher returns if either of the following conditions are true:

- The incoming IP address is in the list of untrusted IP addresses.

- The incoming IP address is not in the list of trusted IP addresses, and the IP address has a reputation other than Dynamic IPs.

MATCHED

The decision that the matcher returns if the matcher finds the incoming IP address in the list of trusted IP addresses.

INDETERMINATE

The decision that the matcher returns if the following conditions are true:

- The IP address is not in the list of untrusted IP addresses.
- The IP address is not in the list of trusted IP addresses.
- The IP address qualifies for one of the following conditions:
 - Does not have a reputation.
 - Has a Dynamic IPs reputation.

Related concepts:

IP reputation

The IP reputation policy information point (PIP) uses the IP reputation database to determine the reputation of the IP address of a request. Based on the IP address reputation, you can write a policy to grant or prohibit access to the requesting IP address. The IP reputation PIP is pre-configured with IBM Security Access Manager for Mobile.

Related tasks:

Modifying attribute matchers

Attribute matchers match incoming attributes to attributes in a device fingerprint. The predefined matchers are set to default values. You can modify those values to customize the risk calculations for your policies.

IP reputation

The IP reputation policy information point (PIP) uses the IP reputation database to determine the reputation of the IP address of a request. Based on the IP address reputation, you can write a policy to grant or prohibit access to the requesting IP address. The IP reputation PIP is pre-configured with IBM Security Access Manager for Mobile.

Possible IP reputations include the following classifications:

- Anonymous proxies
- Botnet Command and Control Server
- Dynamic IPs
- Malware
- Scanning IPs
- Spam

IP addresses that have reputations engage in certain activities that qualify them for reputations. The IP reputation database contains classification information about IP addresses. When the IP reputation PIP requests information, the database returns a score for each classification that the IP address might have. The score ranges from 0 - 100. As the value increases, the likelihood that the IP address has a reputation corresponding with that score increases.

The score for each classification is compared to a threshold score that you can configure. You can also use the default score of 50. The IP address has a reputation if the score that belongs to any number of classifications is greater than or equal to

the threshold score. For example, if an IP address is suspected to be a spammer, the database may return a value of 95 for the spam classification. If this value is greater than or equal to the threshold score you choose to use, the IP reputation PIP returns a Spam classification for the requesting IP address.

You can write a policy to either grant access to or prohibit access from IP addresses with specified classifications. To use the IP reputation PIP, use the `ipReputation` attribute in a policy.

IP reputation can also be used to influence an access decision when the IP matcher is used.

Related tasks:

Modifying attribute matchers

Attribute matchers match incoming attributes to attributes in a device fingerprint. The predefined matchers are set to default values. You can modify those values to customize the risk calculations for your policies.

Attribute matchers

An attribute matcher compares the values of a specified attribute in the incoming device fingerprint with the existing device fingerprint of the user.

Context-based access uses the information that is returned by the attribute matchers to calculate the risk score.

Configuring the IBM Security Access Manager for Mobile appliance to contact the license server

You can configure the IBM Security Access Manager for Mobile appliance to contact the license server so that the IP reputation database can update.

About this task

When the IP reputation database cannot contact the license server, it cannot update. Additionally, the appliance cannot resolve the domain name for the license server. The inability for the IP reputation database to contact the license server causes the database to contain outdated IP reputation data or lack IP reputation data.

The IP reputation database cannot contact the license server if the administrator did not complete one of the following tasks:

- Configure the domain name system (DNS) so that the appliance can access the license server directly.
- Add the license server to the hosts file so that the appliance can access the license server directly.
- Use a proxy so that the appliance can access the license server indirectly.

If an administrator writes a policy that denies access based on IP reputation and the database cannot contact the license server, the policy information point (PIP) might grant access to any IP address. The IP reputation database does not contain the necessary data that is required to determine IP address classifications.

Procedure

Confirm that the appliance can access the license server by completing one of the following tasks:

- Adding the license server to the hosts file

1. Log in to the local management interface.
 2. Under **Manage System Settings**, click **Hosts File**.
 3. Click **New**.
 4. Enter 206.253.225.13 for **Address**. Enter license.cobion.com for the **Hostname**.
 5. Click **Save**.
- Using a proxy to access the license server
 1. Log in to the local management interface.
 2. Under **Manage System Settings**, click **Application Database Settings**.
 3. Under **Proxy Settings**, complete the following steps:
 - a. Select the **Use Proxy** check box if you want to use a proxy to access the update server.
 - b. For **Server Address**, enter the address of the central repository that contains the most recent IP reputation data.
 - c. For **Port**, enter the port to access the central repository that contains the most recent IP reputation data.
 - d. If the central repository requires authentication, select the **Use Authentication** check box and also enter the user name and password for authentication.

Note: **Server Address**, **Port**, and **Use Authentication** are only required if **Use Proxy** is selected.

Modifying attribute matchers

Attribute matchers match incoming attributes to attributes in a device fingerprint. The predefined matchers are set to default values. You can modify those values to customize the risk calculations for your policies.

About this task

Each predefined matcher uses specific properties.

exact_match

The **exact_match** matcher checks whether the values of an attribute in a registered device and an incoming request exactly equal each other. Use this matcher if the more specialized matchers are not appropriate for the attribute. This matcher cannot be modified.

location_matcher

The location matcher checks whether the location of a device is within a specific distance from the previous known locations of a device.

Comparison

Indicates how you want the attribute matcher to calculate the accuracy range of the location coordinates.

Distance

Specifies the maximum distance between the new location and the historic locations. The value is in kilometers. The default value is 40.

login_time_matcher

The login matcher compares and analyzes the historical login time data for the user with the current login time of the user.

Threshold

Indicates the probability that a user might log in at a particular time. Valid values are 0 to 1. The default value is 0.3. This default value indicates the probability that the user logs in approximately within an hour of the previous login times. If you set a lower value, the odds of a return value of true are higher and the risk score is lower. If you set a higher value, the odds of a return value of true are lower and risk score is higher. For example, if you set a value of 0.5, the matcher almost always returns false. The login time analysis collects data for eight login times before it provides input for risk score calculation.

ipaddr_matcher

The IP address matcher compares an inclusion list (trusted) or exclusion list (not trusted) of IP addresses with the historical IP addresses of the device.

Trusted addresses

IPV4 addresses

IP and Netmask: Specifies the IP address and its netmask to include. Include X.X.X.X as a value to compare the incoming IP address with the IP address with which the device is registered.

IPV6 addresses

IP and Prefix: Specifies the IP address and its prefix to include. Include X:X:X:X:X:X:X:X as a value to compare the incoming IP address with the IP address with which the device is registered.

Untrusted addresses

IPV4 addresses

IP and Netmask: Specifies the IP address and its netmask to exclude. Include X.X.X.X as a value to compare the incoming IP address with the IP address with which the device is registered.

IPV6 addresses

IP and Prefix: Specifies the IP address and its prefix to exclude. Include X:X:X:X:X:X:X:X as a value to compare the incoming IP address with the IP address with which the device is registered.

Use the IP reputation database for classification of IP addresses

Select this box to check the requesting IP address against the addresses in the IP Reputation database. Addresses in the database are associated with one or more classifications. If the requesting address matches an address in the database, the database returns a score for each classification that is associated with the address.

The IP reputation threshold for classifications

The score that is compared to the classification score of an IP address. Select a score between 0 and 100 below the **Untrusted** tab in **IP Address Matcher Properties**. The default value is 50.

Procedure

1. Log in to the local management interface.
2. Click **Secure Mobile Settings**.

3. Under **Policy**, click **Attributes**.
4. Click **Matchers**.
5. Click the  icon for the matcher.
6. Change the properties.
7. Click **Save**.
8. When you modify an attribute matcher, a message indicates that there are changes to deploy. If you are finished with the changes, deploy them.
For more information, see Chapter 12, “Deploying pending changes,” on page 109.

Results

The modified attribute matcher is saved.

Related concepts:

Attribute matchers

An attribute matcher compares the values of a specified attribute in the incoming device fingerprint with the existing device fingerprint of the user. Context-based access uses the information that is returned by the attribute matchers to calculate the risk score.

IP reputation

The IP reputation policy information point (PIP) uses the IP reputation database to determine the reputation of the IP address of a request. Based on the IP address reputation, you can write a policy to grant or prohibit access to the requesting IP address. The IP reputation PIP is pre-configured with IBM Security Access Manager for Mobile.

Chapter 6. Obligations

Obligations are used in policies to inform the enforcement point that more actions are required before access is granted or denied to a protected resource.

Obligations are either of the following two types:

- Actions that require the user to perform an operation, such as logging in a second time or answering a challenge question.
- Actions that occur on the server without user involvement.

Some obligations are specific to authentication actions. In the local management interface, these types of obligations are called *Authentications*.

Predefined obligations are available by default. See “Predefined obligations” on page 63.

You can create, modify, or delete obligations.

When you create a policy, you can choose to add your obligation. If the obligation has parameters, define the parameter details when you create the policy.

You can use the same obligation for different policies. Parameters customize the obligation for each policy. Parameter details are defined in the policy, not the obligation.

Obligations are saved in the local configuration database.

Managing obligations

Obligations are used for authoring policies. You can view, add, modify, and delete obligations.

Procedure

1. Log in to the local management interface.
2. Click **Secure Mobile Settings**.
3. Under **Policy**, click **Obligations**.

Note: If you want to work with only authentication type of obligations, you can click **Authentication** instead of **Obligations**.

4. Perform any of the following actions:
 - Add an obligation:
 - a. Click .
 - b. Complete the properties for the obligation.

Note: The obligation name must begin with an alphabetic character. Do not use control characters, leading and trailing blanks, and the following special characters ~ ! @ # \$ % ^ & * () + | ` = \ ; : " ' < > ? , [] { } / anywhere in the name.

- c. Click **Save**.

- Modify a custom obligation:

Note: You cannot modify predefined obligations or obligations that are in use by a policy.

- Click the obligation that you want to modify to highlight the obligation.
- Click .
- Modify any of the obligation properties.

Note: The obligation name must begin with an alphabetic character. Do not use control characters, leading and trailing blanks, and the following special characters ~ ! @ # \$ % ^ & * () + | ` = \ ; : " ' < > ? , [] { } / anywhere in the name.

- To add a parameter, click , specify the settings. Press Enter.
 - To delete a parameter, highlight it, click , and click **OK**.
 - Click **Save**.
- Delete a custom obligation:
- Note:** You cannot remove predefined obligations or obligations that are in use by a policy.
- Select an obligation from the list. To select multiple obligations, press and hold the Ctrl key and select several obligations
 - Click . A message prompts you to confirm the deletion.
 - Click **Delete**.
- When you add, modify or delete an obligation, a message indicates that there are changes to deploy. If you are finished with the changes, deploy them.
For more information, see Chapter 12, “Deploying pending changes,” on page 109.

What to do next

You can use any obligation in your policy. See Chapter 8, “Policies,” on page 75 for instructions on how to specify an obligation for use in your policy.

Related reference:

“Predefined obligations” on page 63

Predefined obligations are provided for your use in policy authoring.

“Obligation properties”

When you add or modify an obligation, you specify properties that make that attribute unique.

Obligation properties

When you add or modify an obligation, you specify properties that make that attribute unique.

Specify the following properties when you add or modify an obligation:

Name Specify a unique name for the obligation.

Description

Enter a description of the obligation. (Optional)

Identifier

Specify an obligation ID that is used within the XACML policy file to identify the obligation at run time. The identifier must be a known identifier that can be handled by the target enforcement point. This value is required. For example,
urn:ibm:security:obligation:myCustomObligation.

Type This field is read only. The type is **Obligation**.

Parameters**DataType**

Select a data type for the parameter. Valid options include the following options:

Boolean

Represents the XML Boolean data type,
http://www.w3.org/2001/XMLSchema#boolean.

Date Represents the XML date data type, http://www.w3.org/2001/XMLSchema#date.

Double

Represents the XML double data type,
http://www.w3.org/2001/XMLSchema#double.

Integer

Represents the XML integer data type,
http://www.w3.org/2001/XMLSchema#integer.

String Specifies the XML string data type, http://www.w3.org/2001/XMLSchema#string.

Time Represents the XML time data type, http://www.w3.org/2001/XMLSchema#time.

X500Name

Represents the XACML X500 name data type,
urn:oasis:names:tc:xacml:1.0:data-type:x500Name.

Related tasks:

“Managing obligations” on page 59

Obligations are used for authoring policies. You can view, add, modify, and delete obligations.

Mapping obligations to a URL

You can define the mapping between the obligation level that the policy decision point (PDP) returns and the URL used to satisfy the obligation.

Procedure

1. Open the WebSEAL configuration file.
2. Add entries to the [obligations-urls-mapping] stanza. These entries define the mapping between an obligation and the URL that is used to satisfy that obligation. Following is an example of the stanza and entry:

```
[obligations-urls-mapping]
obligation = URL
```

where:

obligation

Defines the obligation string that is returned by runtime security services. This string, or *key*, is case-sensitive.

URL

Defines the URL which must be accessed to satisfy the obligation. The URL must point to an external authentication interface (EAI) application. See the WebSEAL documentation for information about the requirements for the EAI application.

3. Save the file.
4. Restart the WebSEAL server for the changes to take effect.

Results

When the runtime security services returns an obligation, the key is searched for in the configuration file in following order:

1. [obligations-urls-mapping] entries
2. [obligations-levels-mappings] entries

The keys in the [obligations-urls-mapping] stanza must differ from the keys in the [obligations-levels-mappings] stanza. Do not use the same key name in both stanzas.

Example

For example, specify:

```
[obligations-urls-mapping]
auth1 = https://example.com/mga/sps/xauth?AuthenticationLevel=2
```

This entry defines an obligation named auth1 that contains a URL that is used to satisfy the obligation.

obligation

Define a URL to be used to satisfy an obligation.

Syntax

obligation = *URL*

Description

This entry defines the mapping between the obligation that the policy decision point (PDP) returns and the URL that is used to satisfy the obligation.

Options

obligation

Defines the obligation that is returned by runtime security services. This string, or *key*, is case-sensitive.

URL Defines the URL that is used to satisfy the obligation. The URL must point to an external authentication interface (EAI) application. See the WebSEAL documentation for information about the requirements for the EAI application.

Usage

This stanza entry is not required.

Example

The following entry specifies that an obligation named `auth1`. The value of `auth1` is a URL that is used to satisfy the obligation.

```
[obligations-urls-mapping]
auth1 = https://example.com/MGA/sps/xauth?AuthenticationLevel=2
```

Predefined obligations

Predefined obligations are provided for your use in policy authoring.

An obligation consists of an action that must occur to allow or deny access to a resource. Some obligations require user interaction. Others are operations that happen without any user involvement. Choose from the following predefined obligations:

Consent Register Device

The user is presented with a page that asks for consent to register a device. Optionally, the user can assign a name to the device to be registered.

Email One-Time Password

The user is required to authenticate with a one-time password that is delivered by email. The one-time password value is generated and verified with the MAC one-time password.

HOTP One-Time Password

The user is required to authenticate with a counter-based one-time password. No one-time password delivery is required. The one-time password value is verified with the HOTP one-time password provider.

One-Time Password

The user is required to authenticate with a one-time password. The user is allowed to select the one-time password provider to be used for authentication when more than one provider is enabled.

Register Device

The system registers the device that is used by the user to access the resource. No user interaction is required for this obligation.

The administrator can configure the following properties to affect device fingerprint registration:

`deviceRegistration.allowIncompleteFingerprints`

Specifies whether the risk engine registers devices with incomplete device fingerprints.

An *incomplete device fingerprint* is a device fingerprint that does not include all of the attributes that the administrator specifies in the risk profile.

`deviceRegistration.permitOnIncompleteFingerprint`

Specifies whether the risk engine grants access to request devices with incomplete device fingerprints.

If the administrator does not set the configuration properties, then the properties default to **false**. When the properties are set to **false**:

- The risk engine cannot register devices that have incomplete device fingerprints.
- The risk engine denies request devices with incomplete device fingerprints.

RSA One-Time Password

The user is required to authenticate with an RSA one-time password. No one-time password delivery is required. The one-time password value is verified with the RSA one-time password provider. The RSA one-time password provider uses the RSA Authentication Manager to complete one-time password verification.

SMS One-Time Password

The user is required to authenticate with a one-time password that is delivered by SMS. The one-time password value is generated and verified with the MAC one-time password provider.

TOTP One-Time Password

The user is required to authenticate with a time-based one-time password. No one-time password delivery is required. The one-time password value is verified with the TOTP one-time password provider.

Chapter 7. Risk profiles

The risk engine uses the active risk profile to calculate risk scores for incoming requests. The administrator can create risk profiles or use predefined risk profiles that are provided on IBM Security Access Manager for Mobile.

There are two types of risk profiles:

Predefined risk profiles

Default risk profiles that are pre-configured on IBM Security Access Manager for Mobile.

Each predefined risk profile is tailored to a specific scenario.

Risk profiles

Custom risk profiles that the administrator creates.

Risk profiles include:

- Attributes that the administrator specifies. The risk engine collects all of the attributes that the administrator specifies in the risk profile from each incoming request device.
- Weight values for each attribute that the administrator specifies in the risk profile. The weight that an administrator gives to each attribute in the risk profile depends on the importance of each attribute. As the attributes increase in importance, their weight values also increase.

Note: There is not a limit on the number of attributes that the administrator can include in a risk profile. However, the risk engine uses one risk profile at a time.

Related concepts:

Risk score calculation

Risk score calculation is the process by which the risk engine determines a risk score. The *risk score* demonstrates the level of risk that is associated with permitting a request to access the resource. This risk score is compared to a *threshold score* that is set in a policy. A decision is made based on the result of this comparison.

Related reference:

Predefined risk profiles

Predefined risk profiles are pre-configured on IBM Security Access Manager for Mobile. The risk engine uses the active risk profile to calculate risk scores for incoming requests.

Managing risk profiles

A risk profile is a collection of attributes with assigned weights. You can view a list of profiles or to add, delete, or clone profiles or set a profile to active. Only one risk profile can be active at a time.

About this task

Several risk profiles are predefined based on risk assessment for a collection of attributes. Predefined risk profiles are read only and you cannot modify their weights or add attributes to them. You can add custom risk profiles by creating your own or by cloning an existing profile.

By default, a risk profile named Default is set to active. The Default profile includes all the risk profile attributes with weights set to 0. With this profile active, if a user logs in with no devices registered, the risk score of that user is 100. If a user logs in with a device registered, the risk score of that user is 0. The Default profile is a sample profile. It is not intended to be used in a production environment. Before you use Security Access Manager, choose another risk profile or create your own.

Note: Use single-value attributes when you create a risk profile. Multivalue attributes are not supported in risk profiles.

There are two sections on the Risk Profile page:

Risk Profiles

The Risk Profiles section lists all the available risk profiles. An icon indicates which profile is active. Click a profile to select it.

Selected Profile Contents

The Selected Profile Contents section displays the Attribute and Weight for a selected risk profile. If you select a custom risk profile, you can modify the weight by typing a number or selecting a number.

Procedure

1. Log in to the local management interface.
2. Click **Secure Mobile Settings**.
3. Under **Policy**, click **Risk Profiles**.
4. Perform one or more of the following actions:
 - Rename a risk profile:
 - a. Right-click the profile name in the table and click **Rename** to change the name.

Note: The risk profile name must begin with an alphabetic character. Do not use control characters, leading and trailing blanks, and the following special characters ~ ! @ # \$ % ^ & * () + | ` = \ ; : " ' < > ? , [] { } / anywhere in the name.

- b. Press Enter to apply the change.
- Modify a custom profile:

Note: You can modify only custom profiles.

 - a. Select a custom profile.
 - b. Modify the attributes and associated weights in the Selected Profile Contents section.
 - Create a custom profile:
 - a. Click .
 - b. Type a name for the profile. Click **Save**.
 - c. In the Profile Contents section, click  to add one or more attributes.
 - d. In the Add Profile Attributes window, select an attribute to use. Click the **Attribute** column to sort the list in ascending or descending order. To filter the list of available attributes, type one or more characters in the **Filter** field. For example, if you type current in the **Filter** field, all attributes that start with current are shown in the attributes list. The attributes that match those characters are displayed.

- e. Select one or more attributes to add.
- f. Click **Add**.
- g. Click **Close** when you are done with the Add Profile Attributes window.
- h. Change the weights by typing or selecting a number.
- i. Click **Save** to apply the changes.
- j. If you want to make this new profile active, click  Set Active.
- Create a copy of a profile and use it to make a custom profile:
 - a. Select the profile that you want to clone and give the profile a unique name.
 - b. Click .
 - c. Type a new name for the cloned profile.
 - d. Click **Save**.
 - e. Select the clone.
 - f. In the Profile Contents section, select an attribute and take one of the following actions:
 - Type a new value in the **Weight** field.
 - Click  to remove both the attribute and weight.
 - Click  to add one or more attributes. In the Add Profile Attribute window, select the attributes to use. To filter the list of available attributes, type one or more characters in the **Filter** field. The attributes that match those characters are displayed. For example, if you type header in the **Filter** field, all attributes that start with header are shown in the attributes list. Select the attribute that you want to use in the list and click **Add**. Click **Close** when you are done with the Add Profile Attributes window.
 - g. Click **Save** after each change to apply the change.
 - h. If you want to make this new profile active, click  Set Active.
- Delete a risk profile:

Note: You cannot remove predefined risk profiles.

 - a. Select a risk profile.
 - b. Click .
 - c. Respond to the confirmation prompt.

The risk profile is removed and cannot be used.
- Make a profile active:
 - a. Select a profile.
 - b. Click  Set Active.

Note: Only one profile can be active at a time.

5. When you add, modify or delete a risk profile, a message indicates that there are changes to deploy. If you are finished with the changes, deploy them.
For more information, see Chapter 12, “Deploying pending changes,” on page 109.

Predefined risk profiles

Predefined risk profiles are pre-configured on IBM Security Access Manager for Mobile. The risk engine uses the active risk profile to calculate risk scores for incoming requests.

Predefined risk profile types

Predefined risk profiles:

- Are tailored for specific scenarios.
- Can be cloned.
- Cannot be modified.

To choose the most appropriate risk profile, you must determine your security priority. You can also create your own risk profile.

Attention: By default, a risk profile named `Default` is set to active. The `Default` profile includes all the risk profile attributes with weights set to 0. With this profile active, if a user logs in with no devices registered, the risk score of that user is 100. If a user logs in with a device registered, the risk score of that user is 0. The `Default` profile is a sample profile. It is not intended to be used in a production environment. Before you use Security Access Manager, choose another risk profile or create your own.

You can choose one of the following predefined risk profiles for the risk engine to use as a filter when it calculates the risk score:

Behavior

Determines a risk score by comparing the time of the current request with the time that the user usually tries to access the resource.

The following table contains the attributes and corresponding weight values that are included in the Behavior risk profile.

Attribute	Weight
accessTime	50
browserPlugins	10
deviceFonts	10
http:userAgent	10

Browser

Determines a risk score by comparing the attributes from the requesting browser with the browsers that the user is known to use.

The following table contains the attributes and corresponding weight values that are included in the Browser risk profile.

Attribute	Weight
browserPlugins	50
deviceFonts	50
http:accept	30
http:acceptEncoding	50
http:acceptLanguage	50
http:userAgent	50

Device

Determines a risk score by comparing the attributes from the requesting device with the devices that are associated with the user.

The following table contains the attributes and corresponding weight values that are included in the Device risk profile.

Attribute	Weight
browserPlugins	30
colorDepth	50
deviceFonts	50
deviceLanguage	50
devicePlatform	50
screenAvailableHeight	50
screenAvailableWidth	50
screenHeight	50
screenWidth	50

Location

Determines a risk score by comparing the location of the incoming request with the locations that the user is known to log in from.

The following table contains the attributes and corresponding weight values that are included in the Location risk profile.

Attribute	Weight
geoLocation	50
geoCity	10
geoCountryCode	10
geoRegionCode	10

Usage scenarios

The following example usage scenarios demonstrate risk score calculation in predefined risk profiles.

Each scenario assumes that the administrator wrote a policy, which specifies that:

- Any risk score at or below 40 is permitted.
- Any risk score above 40 is denied.

Scenario 1: Behavior risk profile

The risk engine uses risk profile information in the following table to calculate the risk score.

Attributes	Weight values	Incoming device fingerprint values	Registered device fingerprint values
accessTime	50	2013-05-07T03:25:13Z	2013-05-06T04:00:39Z, 2013-05-13T03:05:20Z,2013-05-20T03:15:22,2013-05-27T03:26:05Z, 2013-06-03T03:42:45Z

The risk engine uses risk profile information in the following table to calculate the risk score.

Attributes	Weight values	Incoming device fingerprint values	Registered device fingerprint values
browserPlugins	10	Shockwave Flash,Chrome Remote Desktop Viewer,Widevine Content Decryption Module,Native Client,Chrome PDF Viewer,Java(TM) Plug-in 1.7.0,Citrix Receiver for Linux	Shockwave Flash,Chrome Remote Desktop Viewer,Native Client, Chrome PDF Viewer, Conference Plugin, AmazonMP3DownloaderPlugin, Google Update
deviceFonts	10	Andale Mono,Arial Black,Arial,Bitstream Charter,Century Schoolbook L,Comic Sans MS,Courier 10 Pitch,Courier New,DejaVu Sans Mono,DejaVu Sans,DejaVu Serif,Dingbats,Georgia, Impact,Khmer OS System,Khmer OS,Liberation Mono,Liberation Sans,Liberation Serif,Lohit Bengali,Lohit Gujarati,Lohit Punjabi,Lohit Tamil,Luxi Mono,Luxi Sans,Luxi Serif,Meera,Nimbus Mono L,Nimbus Roman No9 L,Nimbus Sans L,Standard Symbols L,Tahoma,Times New Roman,Trebuchet MS,URW Bookman L,URW Chancery L,URW Gothic L,URW Palladio L,UnBatang,UnDotum,Verdana, Waree,Webdings	Aharoni,Andalus,Angsana New, AngsanaUPC,Aparajita,Arabic Typesetting,Arial Black,Arial, Batang,BatangChe,Browallia New,BrowalliaUPC,Calibri, Cambria Math,Cambria,Candara, Comic Sans MS,Consolas, Constantia,Corbel,Cordia New,CordiaUPC,Courier 10 Pitch,Courier New,David, DilleniaUPC,DokChampa, Dotum,DotumChe,Ebrima, Estrangelo Edessa,EucrosiaUPC, Euphemia,FangSong,FrankRuehl, Franklin Gothic Medium,LilyUPC, Lucida Bright,Lucida Console, Lucida Sans Typewriter,Tahoma, Times New Roman,Traditional Arabic,Wingdings
http:userAgent	10	Mozilla/5.0 (Windows NT 6.2; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/29.0.1547.2 Safari/537.36	Mozilla/5.0 (X11; Linux i686 (x86_64)) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/27.0.1453.110 Safari/537.36

Results:

- None of the device fingerprint values match except for the incoming device fingerprint value and existing device fingerprint value for accessTime.
- Because all of the attributes except for accessTime have mismatched values, the collective weight of the mismatched attributes is 30.
- The total weight of all of the attributes is 80. The accessTime attribute has a weight value of 50. The http:userAgent attribute, browserPlugins attribute, and deviceFonts attribute each have weight values of 10.
- According to the risk score calculation formula: $(30/80) \times 100 = 38$. Therefore, the risk score is 38.
- Authentication is permitted because the risk score is below 40.

Scenario 2: Browser risk profile

The risk engine uses risk profile information in the following table to calculate the risk score.

Attributes	Weight values	Incoming device fingerprint values	Registered device fingerprint values
browserPlugins	50	Shockwave Flash,Chrome Remote Desktop Viewer,Widevine Content Decryption Module,Native Client,Chrome PDF Viewer,Java(TM) Plug-in 1.7.0,Citrix Receiver for Linux	Shockwave Flash,Chrome Remote Desktop Viewer,Native Client, Chrome PDF Viewer, Conference Plugin, AmazonMP3DownloaderPlugin, Google Update

The risk engine uses risk profile information in the following table to calculate the risk score.

Attributes	Weight values	Incoming device fingerprint values	Registered device fingerprint values
deviceFonts	50	Andale Mono,Arial Black,Arial,Bitstream Charter,Century Schoolbook L,Comic Sans MS,Courier 10 Pitch,Courier New,DejaVu Sans Mono,DejaVu Sans,DejaVu Serif,Dingbats,Georgia, Impact,Khmer OS System,Khmer OS,Liberation Mono,Liberation Sans,Liberation Serif,Lohit Bengali,Lohit Gujarati,Lohit Punjabi,Lohit Tamil,Luxi Mono,Luxi Sans,Luxi Serif,Meera,Nimbus Mono L,Nimbus Roman No9 L,Nimbus Sans L,Standard Symbols L,Tahoma,Times New Roman,Trebuchet MS,URW Bookman L,URW Chancery L,URW Gothic L,URW Palladio L,UnBatang,UnDotum,Verdana, Wreee,Webdings	Aharoni,Andalus,Angsana New, AngsanaUPC,Aparajita,Arabic Typesetting,Arial Black,Arial, Batang,BatangChe,Browallia New,BrowalliaUPC,Calibri, Cambria Math,Cambria,Candara, Comic Sans MS,Consolas, Constantia,Corbel,Cordia New,CordiaUPC,Courier 10 Pitch,Courier New,David, DilleniaUPC,DokChampa, Dotum,DotumChe,Ebrima, Estrangelo Edessa,EucrosiaUPC, Euphemia,FangSong,FrankRuehl, Franklin Gothic Medium,LilyUPC, Lucida Bright,Lucida Console, Lucida Sans Typewriter,Tahoma, Times New Roman,Traditional Arabic,Wingdings
http:accept	30	text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8	text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
http:acceptEncoding	50	gzip,deflate,sdch	gzip,deflate,sdch
http:acceptLanguage	50	en-US,en;q=0.8	en-US,en;q=0.8,es;q=0.6
http:userAgent	50	Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/27.0.1453.116 Safari/537.36	Mozilla/5.0 (X11; Linux i686 (x86_64)) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/27.0.1453.110 Safari/537.36

Results:

- None of the device fingerprint values match except for the incoming device fingerprint value and existing device fingerprint value for http:accept and http:acceptEncoding.
- Because all of the attributes except for http:accept and http:acceptEncoding have mismatched values, the collective weight of the mismatched attributes is 200.
- The total weight of all of the attributes is 280. The http:accept attribute has a weight value of 30. The browserPlugins attribute, deviceFonts attribute, http:acceptEncoding attribute, http:acceptLanguage attribute, and http:userAgent attribute each have weight values of 50.
- According to the risk score calculation formula: $(200/280) \times 100 = 71$. Therefore, the risk score is 71.
- Authentication is denied because the risk score is above 40.

Scenario 3: Device risk profile

The risk engine uses risk profile information in the following table to calculate the risk score.

Attribute names	Weight values	Incoming device fingerprint values	Registered device fingerprint values
browserPlugins	30	Shockwave Flash,Chrome Remote Desktop Viewer,Widevine Content Decryption Module,Native Client,Chrome PDF Viewer,Java(TM) Plug-in 1.7.0,Citrix Receiver for Linux	Shockwave Flash,Chrome Remote Desktop Viewer,Native Client, Chrome PDF Viewer, Conference Plugin, AmazonMP3DownloaderPlugin, Google Update
colorDepth	50	24	32
deviceFonts	50	Andale Mono,Arial Black,Arial,Bitstream Charter,Century Schoolbook L,Comic Sans MS,Courier 10 Pitch,Courier New,DejaVu Sans Mono,DejaVu Sans,DejaVu Serif,Dingbats,Georgia, Impact,Khmer OS System,Khmer OS,Liberation Mono,Liberation Sans,Liberation Serif,Lohit Bengali,Lohit Gujarati,Lohit Punjabi,Lohit Tamil,Luxi Mono,Luxi Sans,Luxi Serif,Meera,Nimbus Mono L,Nimbus Roman No9 L,Nimbus Sans L,Standard Symbols L,Tahoma,Times New Roman,Trebuchet MS,URW Bookman L,URW Chancery L,URW Gothic L,URW Palladio L,UnBatang,UnDotum,Verdana, Waree,Webdings	Aharoni,Andalus,Angsana New, AngsanaUPC,Aparajita,Arabic Typesetting,Arial Black,Arial, Batang,BatangChe,Browallia New,BrowalliaUPC,Calibri, Cambria Math,Cambria,Candara, Comic Sans MS,Consolas, Constantia,Corbel,Cordia New,CordiaUPC,Courier 10 Pitch,Courier New,David, DilleniaUPC,DokChampa, Dotum,DotumChe,Ebrima, Estrangelo Edessa,EucrosiaUPC, Euphemia,FangSong,FrankRuehl, Franklin Gothic Medium,LilyUPC, Lucida Bright,Lucida Console, Lucida Sans Typewriter,Tahoma, Times New Roman,Traditional Arabic,Wingdings
deviceLanguage	50	en-US	en-US
devicePlatform	50	Linux x86_64	Win-32
screenAvailable Height	50	1025	870
screenAvailable Width	50	1920	1600
screenHeight	50	1080	900
screenWidth	50	1920	1600

Results:

- None of the device fingerprint values match except for the incoming device fingerprint value and existing device fingerprint value for deviceLanguage.
- Because all of the attributes except for deviceLanguage have mismatched values, the collective weight of the mismatched attributes is 380.
- The total weight of all of the attributes is 430. The browserPlugins attribute has a weight value of 30. The following attributes have weight values of 50:
 - colorDepth
 - deviceFonts
 - deviceLanguage
 - devicePlatform
 - screenAvailableHeight
 - screenAvailableWidth
 - screenHeight

- screenWidth
- According to the risk score calculation formula: $(380/430) \times 100 = 88$. Therefore, the risk score is 88.
- Authentication is denied because the risk score is above 40.

Scenario 4: Location risk profile

The risk engine uses risk profile information in the following table to calculate the risk score.

Attributes	Weight values	Incoming device fingerprint values	Registered device fingerprint values
geoCity	10	Austin	Austin
geoCountryCode	10	US	US
geoLocation	50	30.2861, -97.739321, 10	30.274722, -97.740556, 13
geoRegionCode	10	TX	TX

Results:

- All of the device fingerprint values match. The geoLocation attribute contains the values that the risk engine uses to calculate the distance between the incoming device fingerprint and the registered device fingerprint. In this instance, the distance between the two device fingerprints is 1.27 km.
- Because all of the device fingerprint values match, the total weight of the mismatched attributes is 0.
- The total weight of all of the attributes is 80. The geoLocation attribute has a weight value of 50. The geoCity attribute, geoCountryCode attribute, and geoRegionCode attribute each have weight values of 10.
- According to the risk score calculation formula: $(0/80) \times 100 = 0$. Therefore, the risk score is 0.
- Authentication is permitted because the risk score is below 40.

Risk score calculation

Risk score calculation is the process by which the risk engine determines a risk score. The *risk score* demonstrates the level of risk that is associated with permitting a request to access the resource. This risk score is compared to a *threshold score* that is set in a policy. A decision is made based on the result of this comparison.

Chapter 8. Policies

A policy is a set of conditions that, after they have been evaluated, determine access decisions.

The conditions are a combination of attributes, obligations, and a risk profile.

Before you author a policy, review the available attributes, obligations, and risk profiles in the local management interface to determine if they meet the needs of your policy.

- “Managing attributes” on page 17
- “Managing obligations” on page 59
- “Managing risk profiles” on page 65

For more information, see Chapter 2, “Risk management overview,” on page 9.

When all of the attributes and obligations are available and the risk profile you want to use is set active, use the policy editor to select the conditions for your policy.

Managing policies

An authorization policy is a set of conditions that define whether a user is permitted or denied access to a protected resource.

About this task

You can view, create, edit, or delete a policy.

Procedure

1. Log in to the local management interface.
2. Click **Secure Mobile Settings**.
3. Under **Policy**, click **Access Control**.
4. Click **All Policies**.
5. Perform one or more of the following actions:

Search for and view policies

Type one or more characters in the **Filter** field. The list displays the policies that start with those characters. Click either view icon:



The details view lists the name and description of the policy.



The list view lists only the name of the policy.

Create a policy

Click . The Policy Editor opens. See “Creating a policy” on page 76 for details.

Edit a policy

- a. Select a policy in the policy list.
- b. Click . The Policy Editor opens.

- c. Change the policy. See “Creating a policy” for details.
- d. Click **Save**.

Delete one or more policies

- a. Select a policy or press Ctrl and select multiple policies in the policy list.
- b. Click . A message prompts you to confirm the deletion.
- c. Click **OK**. The policy or policies are deleted.

Add one or more policies to a policy set

A policy set is a group of policies that are used together to protect a resource. To add policies to a policy set, see “Managing policy sets” on page 79.

Creating a policy

Use the Policy Editor on the appliance local management interface to create and configure an authorization policy.

Before you begin

Each policy is a combination of attributes, obligations, and a risk profile.

Before you create a policy:

1. Ensure that the attributes and obligations you want to use in the policy are defined and available in the local management interface:
 - “Managing attributes” on page 17
 - “Managing obligations” on page 59
2. Ensure that the risk profile you want to use is set as active. See “Managing risk profiles” on page 65.

About this task

The Policy Editor page has several sections:

Name and description

Specify a unique name for the policy and optionally include a description of the policy.

Subjects

Optionally specify one or more subjects to which the policy applies. Subjects can be anything in the Subject part of an access request. For example, use this field to specify that the policy applies to subjects who are members of the SystemAdministrators group. Click  **Add Subject** to add subjects to the policy. Click  to remove a subject from the policy. By specifying subjects, you can ensure that the policy rules are evaluated only if they match at least one of the specified subjects.

Rules The Rules section has several settings:

Precedence

Specify an access action to take on the policy.

Deny If any rule in the policy returns deny, the policy returns deny.

Permit

If any rule in the policy returns permit, the policy returns permit.

First

Access is permitted or denied based on the outcome of first rule in the policy that can be evaluated against the access request. The rules in the policy are evaluated in the same order they are listed. The policy returns Not Applicable if none of the rules evaluates to true. To ensure that either a Permit or Deny decision is returned, include in the policy a Permit or Deny rule that does not contain a condition.

Attributes

When a policy is evaluated, the runtime will attempt to retrieve the values for all attributes that are specified in the policy. Attributes that are not found in the incoming request are considered missing. The **Attributes** setting controls how missing attributes are handled.

Optional

If **Attributes** is set to **Optional**, then all attributes specified in the Rule section of the policy are considered optional. With this setting, missing attributes are treated as empty sets and evaluated against the expression. In most cases, a missing attribute will cause the rule expression to return false.

Required

If **Attributes** is set to **Required**, then all attributes specified in the Rule section of the policy are considered required. With this setting, missing attributes are considered an error and will return a decision of Indeterminate when the rule is evaluated. Indeterminate results often cause the access request to be denied.

Add Rule

Click the **Add Rule** drop-down arrow and select either:

- **Conditional rule:** This type of rule contains one or more conditions and an action. Rules are boolean expressions that are applied to a set of context attributes that are passed in the context object of the decision request. Each rule has an If statement and a Then statement. The If statement specifies the conditions that are checked when an access request is received. The Then statement specifies the action to take when the rule conditions are true.
- **Unconditional rule:** This type of rule contains only an action and no conditions.

The rule actions are:

Permit

The request must be permitted to pass.

Permit with Obligation

A specific action must take place before the request is permitted to pass. Specify the action in the adjacent field.

Note: If you select **Permit with Obligation**, all obligation types are displayed in the selection list. If you choose an obligation that is specific to an authentication action and

then save and close the policy, **Permit with Authentication** is the more appropriate label and is displayed in the policy the next time you open it.

Permit with Authentication

A specific authentication action must take place before the request is permitted to pass. Specify the authentication action in the adjacent field.

Note: If you select **Permit with Authentication**, only the authentication types of obligations are displayed in the selection list.

Deny The request must be denied and not permitted to pass.

Deny with Obligation

The request is denied and an obligation is processed.

Procedure

1. Log in to the local management interface.
2. Click **Secure Mobile Settings**.
3. Under **Policy**, click **Access Control**.
4. In the center panel, click .
5. In the **Name** field, type a unique name for the policy.

Note: The name must begin with an alphabetic character. Do not use control characters, leading and trailing blanks, and the following special characters ~ ! @ # \$ % ^ & * () + | ` = \ ; : " ' < > ? , [] { } / anywhere in the name.

6. Optional: In the **Description** field, type a description for the policy.
7. Optional: Specify subjects to which the policy applies.
 - a. Click **Add Subject**.
 - b. In the first box, select a subject attribute. Begin typing the name of the subject to filter the list.
 - c. In the second box, select an operator.
 - d. In the third box, type a value.

For example, if you want the rule to be evaluated only if the access requestor belongs to the SecurityAdministrator group, specify the following selections:

Parameter

groups

Operator

=

Value SecurityAdministrator

Note: If your LDAP root DN is secauthority=default, you can only use the = (equal) operator in policies that use X.500 names userDN and groupsDN. To specify more subjects, click **Add Subject**.

8. In the Rule section, add one or more rules.
 - a. For **Precedence**, select the access action to take for the policy:
 - b. For **Attributes**, select the attribute usage of the policy.
 - c. To add a rule to the policy, click the **Add Rule** drop-down arrow and choose one of the following:

- **Conditional rule:** This type of rule contains one or more conditions and an action.
 - **Unconditional rule:** This type of rule contains no conditions.
- d. If you create an unconditional rule, continue with step 8h.
 - e. If you are creating a Conditional rule, select whether the rules apply if **All** conditions are true or if **Any** of the conditions are true.
 - f. Create a rule by typing or selecting a parameter, operator, and value. To specify a value in the value field, click the drop-down menu on the right and select either **Enter Value** or **Select Attribute**.
 - g. Take one of the following actions:
 - Click **!** to add a NOT operator to the expression. If the expression already has a NOT operator, clicking **!** removes the operator.
 - Click **+** to add another expression. The new expression is added below the preceding expression.
 - Click **-** to remove an expression.
 - Click **()** to create a parenthetical expression. Select the appropriate attributes, operators, and values for the expression. Or, add more expressions to the group. The new expression is added below the preceding expression.
 - h. Specify the action to take when the rule evaluation is completed.
 - i. Click **OK** when the rule is complete.
 - j. To add another rule to the policy, repeat step 8c on page 78.
 - k. If your policy has more than one rule, you can change the sequence of the rules by selecting a rule and clicking  or .

Note: The sequence of the rules is important if you have selected **First** as the action for the policy.

9. Click **Save** when the policy is complete.

What to do next

Attach the policy to a resource. See “Managing policy attachments” on page 81.

Related reference:

“Predefined attributes” on page 23

The IBM Security Access Manager for Mobile appliance uses attributes to provide information about users and devices that try to access a protected resource. The appliance also includes a set of commonly used attributes called *predefined attributes*.

Managing policy sets

A *policy set* is a group of policies that are used together to protect a resource.

Before you begin

You must create policies. See “Creating a policy” on page 76.

About this task

You can view, create, modify, or delete a policy set.

Procedure

1. Log in to the local management interface.
2. Click **Secure Mobile Settings**.
3. Under **Policy**, click **Access Control**.
4. Perform one or more of the following actions:

Create a policy set

- a. Click .
- b. Type a name for the policy set in the **Name** field.

Note: The name must begin with an alphabetic character. Do not use control characters, leading and trailing blanks, and the following special characters ~ ! @ # \$ % ^ & * () + | ` = \ ; : " ' < > ? , [] { } / anywhere in the name.

- c. Optionally, type a description in the **Description** field.
- d. In the Policy Combining Algorithm dialog, set the combined action for the policy set by choosing one of the following options:

Deny access if any policy in the set returns deny

Choose this option if you want the policy set to deny access if any policy in the set returns a response of deny.

Permit access if any policy in the set returns permit

Choose this option if you want the policy set to permit access if any policy in the set returns a response of permit.

Return the decision of the first policy in the set that returns either permit or deny

The policies are evaluated in the order they are listed in the set. Choose this option if you want to use the first policy that returns a response of permit or deny as the result of the policy set.

- e. Click **Save**.
- f. Click **OK**.
- g. Next, add one or more policies to the policy set.

Add one or more policies to a policy set

- a. Click **All Policies**.
- b. Select a policy or press Ctrl and select multiple policies to add to the policy set.
- c. Click  **Add To**.
- d. Select a policy set.
- e. Click **OK**.

Change the order of the policies in a set

If you selected **Return the decision of the first policy in the set that returns either permit or deny**, set the order in which you want the policies to run:

- a. Select a policy set.
- b. Select a policy.
- c. Click  or  to change the position of the policy in the set.

Modify a policy set

- a. Select a policy set in the list of policy sets.
- b. Click .
- c. Change to the name, description, or policy combining algorithm.

Note: The name must begin with an alphabetic character. Do not use control characters, leading and trailing blanks, and the following special characters ~ ! @ # \$ % ^ & * () + | ` = \ ; : " ' < > ? , [] { } / anywhere in the name.

- d. Click **Save**.

Remove one or more policies from a policy set

- a. Select a policy set in the list of policy sets.
- b. Select a policy or press Ctrl and select multiple policies to remove from the policy set.
- c. Click  **Remove**. Confirm the removal.

Note: When you remove a policy from a set, the policy is not deleted from the policy list. To delete a policy from the list, see “Managing policies” on page 75.

- d. Click **OK**. The policy is removed.

Delete a policy set

- a. Select a policy set in the list of policy sets.
- b. Click . Confirm the deletion.
- c. Click **OK**. The policy set is deleted.

Managing policy attachments

Attach policies or API protection definitions to resources so that the policies and definitions can be enforced.

Before you begin

You must create policies, policy sets, or API protection definitions.

When you create policies, policy sets, or API protection definitions you cannot use them until you publish them to resources. Once policies, policy sets, or API protection definitions are published, they are enforced during the evaluation of access requests.

Since policies and API protection definitions are attached to resources, you must first configure resources. This means that policy and API protection definition attachment is dependent on prior installation and configuration of a IBM Security Access Manager for Web runtime component and reverse proxy server. These prior configuration tasks include the creation of a Security Access Manager for Web protected object space. In a Web environment, this configuration includes creation of a junction and resources (objects) that are protected by that junction.

About this task

You can:

- Add a resource

- Add a policy or API protection definition attachment to a resource
- Remove a policy or API protection definition attachment from a resource
- Delete a resource
- Publish a policy or API protection definition attachment

When a deployment is fully configured, the Resources panel displays three levels of entries. The top-level entry is the web container that contains the protected object space for a server instance. The second level shows the resources in the protected object space. The third level lists the policies and API protection definitions that are attached to each resource.

Tip: The user interface provides a quick filter feature for use on the top-level entry. Use the quick filter to search for a specific top-level entry. Enter the first few characters of the web container, and the list displays only the entries that contain the specified characters.

Procedure

1. Log in to the local management interface.
2. Click **Secure Mobile Settings**.
3. Under **Policy**, click **Access Control**.
4. Click **Resources**.
5. Perform one or more of the following actions:

Add a resource

- a. Click .
- b. In the **Web Container** field, click the down arrow icon to display a list of web containers. Select an entry.
For example, the list of web containers is the WebSEAL protected object space that is defined directly under /WebSEAL.
- c. Specify a resource. You can either enter the name of the resource or browse for it:
 - If you know the name of the resource within the protected object space, you can enter it in the **Resource** field.
For example, /myserver-jct/benefits/medical
 - If you select **Browse**, you can select a resource from the tree hierarchy. Expand the hierarchy to see all available resources.
The display of the hierarchy is based on the structure of the WebSEAL protected object space:
 - In some cases, not all resources are displayed because the WebSEAL protected object space is a sparse tree. For example, you might see only the resource /myserver-jct/benefits. You can select this resource and click **OK** to add it to the **Resource**. You can then add /myserver-jct/benefits/medical.
 - In some cases, you cannot view the object space for the web server junction. For example, if the administrator did not install the IBM Security Access Manager **querycontents** script on the application server, you cannot see the junction contents. In these cases, you can enter the resource path manually.
- d. Click **Save**.
- e. Next, attach a policy to the resource.

Attach a policy or API protection definition to a resource

- Select a resource node and click  Attach.
- In the Attach Policies panel, select **Policies** or **Policy Sets** or **API Protection**.
- From the displayed list, select one or more policies or policy sets or API protection definitions.

Tip: You can type the name of the applicable policy or policy set or API protection definition in the quick filter.

Notes:

- You can attach both individual policies, policy sets, or API protection definitions.
 - You cannot attach policies or policy sets to a resource where that resource already has API protection definitions attached.
 - You cannot attach API protection definitions to a resource where that resource already has policies and policy sets attached.
- Click **OK** to save your changes.

Note: The policy or API protection definition remains inactive until you publish it.

Remove a policy or API protection definition attachment

- To remove a policy or API protection definition attachment from a resource, select the policy node and click .
- When prompted, confirm the deletion.

Note: You must publish the change.

Delete a resource

- To delete a resource and all attached policies or API protection definitions, select the resource node and click .
- When prompted, confirm the deletion.
When you delete a resource:
 - You cannot delete the server node.
 - You do not have to manually publish the change. The deletion is automatically published.

Publish a policy or API protection definition

Select a resource in the resource hierarchy and click  Publish. When the publication completes, the status column for the resource indicates the status and time of the publication.

Note: Activation of the published policy or API protection definition could take up to a minute to complete.

Modify Resource

Note: You can only use this function if policy or policy sets are attached to the given resource.

- a. Select a resource node and click .
- b. In the Modify Resource panel, you can modify the **Policy Combining Algorithm**. Choose the preferred algorithm:
 - **Deny access if any attached policy returns deny**
This algorithm means that if multiple policies or API protection definitions are attached to a resource, and any one of those policies or API protection definitions returns Deny, then the access request is denied.
 - **Permit access if any attached policy returns permit**
This algorithm means that if multiple policies or API protection definitions are attached to a resource, and any one of those policies or API protection definitions returns Permit, then the access request is permitted.

Policy scenarios

Several commonly used policy scenarios are provided as examples to help you author policies.

Denying access based on a set of conditions

A common policy scenario is to deny access based on a set of conditions.

About this task

Use the steps in this scenario task to create a policy that denies access if any of the following conditions are true.

- The calculated risk score is higher than a value of 40.
- The reputation of the ipAddress in the request is considered malware.

Procedure

1. Log in to the local management interface.
2. Click **Secure Mobile Settings**.
3. Under **Policy**, click **Access Control**.
4. In the center panel, click .
5. Enter a name for the policy.
6. In the Rules section, set the Precedence property to **Deny**. As a result, access is denied if any rule returns deny.
7. Click **Add Rule**.
8. Select **riskScore** from the attribute list.
9. Select **>** as the operator.
10. Type 40 as the value.
11. In the Decision list, select **Deny**.
12. Click **OK** to complete the rule.
13. Click **Add Rule** to add another rule.
14. Select **ipReputation** from the attribute list.
15. Select **has member** as the operator.
16. Type Malware as the value.
17. In the Decision list, select **Deny**.

18. Click **OK** to complete the rule.
19. Click the arrow next to **Add Rule**.
20. Click **Unconditional rule**.
21. In the Decision list, select **Permit**. The unconditional Permit rule causes the policy to permit access if none of the deny access rules evaluate to true.
22. Click **OK**.

Results

This scenario uses the following settings in the policy editor.

- Precedence: **Deny**
- Attributes: **Optional**
- Rule 1: If **riskScore >40** Then **Deny**
- Rule 2: If **ipReputation has member Malware** Then **Deny**
- Rule 3: **Permit**

Denying access based on a set of conditions with an OR clause

A common policy scenario is to use multiple conditions in a single rule and to join those conditions with And or Or. In this scenario, access is denied if either of the policy conditions that are joined by Or are true.

About this task

Use the steps in this scenario task to create a policy that denies access if either of the following conditions are true:

- The calculated risk score is higher than a value of 40.
- The reputation of the ipAddress in the request is considered malware.

Procedure

1. Log in to the local management interface.
2. Click **Secure Mobile Settings**.
3. Under **Policy**, click **Access Control**.
4. In the center panel, click .
5. Enter a name for the policy.
6. In the Rules section, set the Precedence property to **Deny**. As a result, access is denied if any rule returns deny.
7. Click **Add Rule**.
8. Click **If Any are true**. The rule evaluates to true if any of the conditions in the rule are true.
9. Select **riskScore** from the attribute list.
10. Select **>** as the operator.
11. Type 40 as the value.
12. Click  to add another condition to the rule.
13. Select **ipReputation** from the attribute list.
14. Select **has member** as the operator.
15. Type Malware as the value.

16. In the Decision list, select **Deny**.
17. Click **OK** to complete the rule.
18. Click the arrow next to **Add Rule**.
19. Click **Unconditional rule**.
20. In the Decision list, select **Permit**. The unconditional Permit rule causes the policy to permit access if none of the deny access rules evaluate to true.
21. Click **OK**.

Results

This scenario uses the following settings in the policy editor.

- Precedence: **Deny**
- Attributes: **Optional**
- Rule 1: If **riskScore >40** or **ipReputation has member Malware** Then **Deny**
- Rule 2: **Permit**

Permitting access based on a set of conditions with an AND clause

A common policy scenario is to use multiple conditions in a single rule and to join those conditions with And or Or. In this scenario, access is permitted if both of the policy conditions that are joined by And are true.

About this task

Use the steps in this scenario task to create a policy that permits access if both of the following conditions are true:

- The calculated risk score is less than or equal to a value of 40.
- The reputation of the ipAddress in the request is not considered malware.

Procedure

1. Log in to the local management interface.
2. Click **Secure Mobile Settings**.
3. Under **Policy**, click **Access Control**.
4. In the center panel, click .
5. Enter a name for the policy.
6. In the Rules section, set the Precedence property to **Permit**. As a result, access is permitted if any rule returns permit.
7. Click **Add Rule**.
8. Click **If all are true**. The rule evaluates to true if all of the conditions in the rule are true.
9. Select **riskScore** from the attribute list.
10. Select **<=** as the operator.
11. Type 40 as the value.
12. Click  to add another condition to the rule.
13. Select **ipReputation** from the attribute list.
14. Select **has member** as the operator.
15. Type Malware as the value.

16. Click  to convert the condition to a Not condition.
17. In the Decision list, select **Permit**.
18. Click **OK** to complete the rule.
19. Click the arrow next to **Add Rule**.
20. Click **Unconditional rule**.
21. In the Decision list, select **Deny**. The unconditional Deny rule causes the policy to deny access if none of the permit access rules evaluate to true.
22. Click **OK**.

Results

This scenario uses the following settings in the policy editor.

- Precedence: **Permit**
- Attributes: **Optional**
- Rule 1: If **riskScore <=40** and **not (ipReputation has member Malware)** Then **Permit**
- Rule 2: **Deny**

Permitting access after one-time password authentication

Security Access Manager can prompt users for one-time passwords when they request access to protected resources. You can use a policy to permit access to users who have authenticated with a one-time password. Or, you can prompt them for the password and then permit access when they provide it.

About this task

Use the steps in this scenario task to create a policy that permits access after the user authenticates with a one-time password. This scenario contains the following rules:

Procedure

1. Log in to the local management interface.
2. Click **Secure Mobile Settings**.
3. Under **Policy**, click **Access Control**.
4. In the center panel, click .
5. Enter a name for the policy.
6. In the Rules section, set the Precedence property to **First**. As a result, the policy returns a decision for the first rule in the policy that evaluates to true.
7. Click **Add Rule**.
8. Select **authenticationTypes** from the attribute list.
9. Select **has member** as the operator.
10. Type **urn:ibm:security:authentication:asf:otp** as the value. If this value is present, the request was already authenticated with a one-time password.
11. In the Decision list, select **Permit**.
12. Click **OK** to complete the rule.
13. Click the arrow next to **Add Rule**.
14. Click **Unconditional rule**.
15. In the Decision list, select **Permit with authentication**.

16. In the Authentication list, select **One-time Password**. This selection results in a request for a one-time password from the user.
17. Click **OK**.

Results

This scenario uses the following settings in the policy editor.

- Precedence: **First**
- Attributes: **Optional**
- Rule 1: If **authenticationTypes** has member **"urn:ibm:security:authentication:asf:otp"** Then **Permit**
- Rule 2: **Permit with Authentication One-time Password**

Registering a device after user consent

Device registration is the process that stores the device fingerprint of the user in the risk-based access database. The rules that you specify in a policy determine whether a device is registered silently or only after the user consents to the registration.

About this task

Use the steps in this scenario task to create a policy that registers a device after the user gives consent. This scenario contains the following rules:

- If the calculated risk score is lower than a specified value, then Permit access.
- If the user consents to have the device registered, then register the device. This action causes the riskScore for the request to be lowered when the policy is reevaluated.
- If the user was not yet prompted to register the device, then display the consent registration form to the user.
- Deny access if none of the above rules evaluates to true.

Procedure

1. Log in to the local management interface.
2. Click **Secure Mobile Settings**.
3. Under **Policy**, click **Access Control**.
4. In the center panel, click .
5. Enter a name for the policy.
6. In the Rules section, set the Precedence property to **First**. As a result, the policy returns a decision for the first rule in the policy that evaluates to true.
7. Click **Add Rule**.
8. Click **If all are true**. The rule evaluates to true if all of the conditions in the rule are true.
9. Select **riskScore** from the attribute list.
10. Select **<** as the operator.
11. Type **40** as the value.
12. In the Decision list, select **Permit**. As a result, the policy permits access if the risk score is below the specified value.
13. Click **OK** to complete the rule.

14. Add a second rule that checks whether the **userConsent** attribute is present and that the value of **userConsent** is true. The **userConsent** attribute is present only if the user was previously prompted with the device registration consent form. If the **userConsent** attribute is true, the user granted consent to register the device.
 - a. Click **Add Rule**.
 - b. Select **userConsent** from the attribute list.
 - c. Select **is present** as the operator.
 - d. Click  to add another condition to the rule.
 - e. Select **userConsent** from the attribute list.
 - f. Select **=** as the operator.
 - g. Type **true** as the value.
 - h. In the Decision list, select **Permit with Obligation**.
 - i. In the Obligation list, select **Register Device**.
 - j. Click **OK**.
15. Add a third rule that checks whether the **userConsent** attribute is missing. If the **userConsent** attribute is missing, the user was not yet prompted with the device registration consent form.
 - a. Click **Add Rule**.
 - b. Select **userConsent** from the attribute list.
 - c. Select **is missing** as the operator.
 - d. Type **true** as the value.
 - e. In the Decision list, select **Permit with Authentication**.
 - f. In the Authentication list, select **Consent Register Device**.
 - g. Click **OK**.
16. Click the arrow next to **Add Rule**.
17. Click **Unconditional rule**.
18. In the Decision list, select **Deny**. This rule is evaluated only if none of the previous rules evaluates to true. This situation can occur if:
 - The risk score is higher than the specified value in the first rule (40 in this example).
 - The user decided not to give consent to register the device.
19. Click **OK**.

Results

This scenario uses the following settings in the policy editor.

- Precedence: **First**
- Attributes: **Optional**
- Rule 1: If **riskScore <40** Then **Permit**
- Rule 2: If **userConsent is present** and **userConsent=true** Then **Permit with Obligation Register Device**
- Rule 3: If **userConsent is missing** Then **Permit with Authentication Consent Register Device**
- Rule 4: **Deny**

Chapter 9. Device fingerprints

Device registration is the process that stores the device fingerprint of the user in the context-based access database.

The rules that you specify in the context-based access policy determine whether a device is registered silently or only after the user consents to the registration.

Device fingerprints can be managed by the administrator and also by the user. The administrator can list the device fingerprints that are registered to a user, and can deregister device fingerprints. The administrator can also set the authentication level for consent-based registration, and modify template pages for use with consent-based registration.

For information on user self-administration of registered devices, see “Managing your registered devices” on page 111

Managing device fingerprints

The device fingerprint contains information that is required for risk score calculation. You can list fingerprints or deregister fingerprints.

About this task

To list registered devices, you can search on a user ID or use an asterisk as a wildcard to search on partial matches.

Note: User names are not stored with the fingerprints and you cannot search on user names.

You can also deregister a selected device fingerprint or deregister all of the device fingerprints that belong to a specific user. When you deregister a device fingerprint, it is removed from the database.

Procedure

1. Log in to the local management interface.
2. Click **Secure Mobile Settings**.
3. Under **Manage**, click **Devices**.
4. Take one of the following actions:

List device fingerprints:

- a. In the **Search for the user ID** field, type:
 - One or more characters and an asterisk * in the first or last position to search on partial matches. For example, enter g* to search for all user IDs that begin with g. The search is case-sensitive. Specifying an uppercase letter returns only uppercase results and specifying a lowercase letter returns only lowercase results.
 - A specific user ID.

- b. In the **Maximum results** field, the default value is 100. Use this value or change it to another value. The minimum value is 1. The maximum value is 1000.
- c. Click **Search**. A list of user IDs is displayed.
- d. Click a user ID in the User Results list. The Registered Device Fingerprints table is displayed.

Deregister a registered device fingerprint:

- a. List the device fingerprints for a specific user.
- b. Click **Deregister All** to remove all of the device fingerprints for the user, or select a device fingerprint and click **Deregister**.
- c. Confirm that you want to deregister the selected device fingerprints.

Silent device registration

Silent device registration is the process of registering the device after the user responds successfully to a secondary challenge.

Silent registration does not require any further interaction or consent from the user. The context-based access policy that you configure determines whether a device must be registered silently.

You can optionally configure your system to allow incomplete device fingerprints to be registered. Use the Advanced Configuration menu to specify the necessary properties. For more information, see the Advanced Configuration topic in the *IBM Security Access Manager for Mobile Configuration Guide*.

Consent-based device registration

Consent-based device registration is the process of registering the device fingerprint only after the user consents to the device registration.

A typical scenario that requires consent-based device registration is when a user attempts to access a protected resource from a public access environment. For example, a user might log in from an internet café or airport kiosk. After the user logs in and successfully responds to the secondary challenge, a consent form is presented. The consent form can be an HTML page where users can specify that they consent to the device registration.

- If the user consents to the registration of the device, the device is registered and access is permitted. The next time that the user logs in from the same device, the consent form is not presented because the device is already registered.
- If the user does not consent to the registration of the device, the device is not registered and the access is permitted. If the user logs in from the same device again, the secondary challenge and the consent form are presented again. The process is repeated because the risk score is high when a user logs in from a device that is not registered.

When a user consents to registration of the device, two attributes are automatically set. You can use these attributes when creating policy:

userConsent

Sets the boolean value to true. Specifies that the user has consented to device registration

authenticationLevel

Numeric value that specifies the authentication level of a user. It increases as the levels of authentication that belong to the user increase. For example: A possible authentication level is 2

When a user is granted access, the authentication level is set by the policy enforcement point. This is the default behavior. Optionally, you can control the authentication level for the user, by setting advanced configuration properties.

IBM Security Access Manager for Mobile provides a template page that you can use for the HTML page to display in order to obtain user consent.

Setting the authentication level for consent-based device registration

You can specify the authentication level to grant to a user who consents to device registration.

About this task

By default, a user who consents to device registration is granted access without a specified authentication level. Optionally, you can set properties to specify the authentication level. You might want to do this action if the user needs access to resources that are restricted to users with a higher access level.

Procedure

1. Select **Secure Mobile Settings > Advanced Configuration**.
2. Under **Filter by Category**, select **deviceRegistration**. The advanced configuration properties window lists the properties for device registration.
3. Click  for the property **consentDeviceRegistration.authLevelHeaderEnabled**.
4. Select the **Enabled** check box, and click **Save**.
5. Click  for the property **consentDeviceRegistration.authLevelHeaderValue**.
6. Specify a value for the authentication level to be granted to the user. Click **Save**.
7. When you change configuration settings, the appliance alerts you that there are undeployed changes. If your changes are complete, deploy them.
For more information, see Chapter 12, "Deploying pending changes," on page 109.

Modifying consent template pages

Use the local management interface to manage files and directories in the template files.

About this task

The template page for `consent-form.html` displays the form where the user can indicate consent to device registration and optionally provide a name to the device that is being registered

You can run the following tasks on the template files:

- **Edit**- Use this option if you want to view or modify the template file.

- **Import**- Use this option if you to import a file to the template files root.
- **Export**- Use this option if you want to export a file from the template files root.
- **Import Zip**- Use this option if you want to import the template files from a compressed file.
- **Export Zip**- Use this option if you want to export the template files as a compressed file.

Procedure

1. From the top menu, select **Secure Mobile Settings > Manage > Template Files**.
2. Work with all the management files and directories.

View or update the contents of a file in the template files root

- a. Select the file of interest.
- b. Select **Edit**. You can then view the contents of the file.
- c. Edit the contents of the file.
- d. Click **Save**.

Export a file from the template files root

- a. Select the file of interest.
- b. Select **Manage > Export**.
- c. Confirm the save operation when your browser displays a confirmation window.

Import a file to the template files root

- a. Select the file of interest.
- b. Select **Manage > Import**.
- c. Click **Browse**.
- d. Browse to the file you want to import.
- e. Click **Open**.
- f. Click **Import**.

Export the template file as a compressed file

- a. Select **Manage > Export Zip**.
- b. Confirm the save operation when your browser displays a confirmation window.

Import the template files as a compressed file

Make sure that the .zip file contains files that exist in the document root.

- a. Select **Manage > Import Zip**.
- b. Click **Browse**.
- c. Browse to the file you want to import.
- d. Click **Open**.
- e. Click **Import**.

3. When you edit or import template files, the appliance displays a message that there are undeployed changes. If you have finished making changes, deploy them.

For more information, see Chapter 12, “Deploying pending changes,” on page 109.

Device fingerprint template page for consent-based registration

This consent-form.html template page displays the form where the user can indicate consent to device registration. The user also has the option of providing a name to the device that is being registered.

The consent-form.html template page has the following replacement macro:
@ERROR_MESSAGE@

This macro is replaced with a message that indicates that there was an error in obtaining consent for device registration. For example, the device name is not valid.

Note: The device name must begin with an alphabetic character. Do not use control characters, leading and trailing blanks, and the following special characters: ~ ! @ # \$ % ^ & * () + | ` = \ ; " ' < > ? , [] { } / anywhere in the name.

REST services for device fingerprint registration

You can use the REST services capability to manage your mobile data such as your registered devices.

Note: The user must authenticate to use the REST services capability.

REST services usage scenarios

Depending on your usage scenario, type the following URLs into the web page that calls the REST services:

Table 1. REST services instructions

Method	URL	Request	Response	Response type
GET	https://hostname/mga/sps/mga/user/mgmt/device	There is no JSON payload included in the request.	<pre>{"username": username, "devices":[{"name":device Name, "id":deviceId, "lastUsedTime":lastUsed Time, "isEnabled": {true false}}, ...]}</pre> If the request completes successfully, the HTTP response code is 200. If the request does not complete successfully, the HTTP response is 500.	application/json
GET	https://hostname/mga/sps/mga/user/mgmt/device/{deviceId}	There is no JSON payload included in the request.	<pre>{"username": username, "name":deviceName, "attributes":[{"name": attributeName1, "value": attributeValue1}, ... , {"name":attributeNameN, "value":attributeValueN}]}</pre> If the request completes successfully, the HTTP response code is 200. If the request does not complete successfully, the HTTP response is 500.	application/json

Table 1. REST services instructions (continued)

Method	URL	Request	Response	Response type
DELETE	https://hostname/mga/sps/mga/user/mgmt/device/{deviceId}	There is no JSON payload included in the request.	{"result": message} If the request completes successfully, the HTTP response code is 200. If the request does not complete successfully, the HTTP response is 500. One of the following messages is in the JSON response: - FBTRBA164E The device <i>device name</i> could not be removed. - FBTRBA163I The device <i>device name</i> was removed successfully.	application/json
PUT	https://hostname/mga/sps/mga/user/mgmt/device/{deviceId}	<pre>{ "name": "new_name", "isEnabled": true false }</pre>	{"result": message} If the request completes successfully, the HTTP response code is 200. The following message is in the JSON response: FBTRBA165I The device <Variable formatSpec="{0}">name</Variable> was updated successfully. If you give the device an invalid name, the HTTP response code is 400. The following message is in the JSON response: FBTRBA182E The value <i>device name</i> is invalid. If you give the device a name that belongs to another device, the response code is 400. The following message is in the JSON response: FBTRBA186E A device named '<Variable formatSpec="{0}">device name</Variable>' already exists.	application/json

Managing your device fingerprints

You can modify the registered device fingerprints that the risk engine compares with incoming request device fingerprints in risk score calculation scenarios.

Chapter 10. Runtime database

The runtime database stores user data such as session attributes and device fingerprints.

To manage the disk space usage of this database, you can clear some or all of its stored data.

To optimize performance or increase capacity, you can optionally deploy an external runtime database for the appliance.

Managing the runtime database

To manage the disk space usage of this database, you can clear some or all of its stored data.

About this task

When you remove data, the Security Access Manager Runtime is stopped.

Procedure

1. Log in to the local management interface.
2. Click **Secure Mobile Settings**.
3. Under **Manage**, click **Database Maintenance**.
4. Take one of the following actions:

Remove device fingerprints:

- a. Select **Remove device fingerprints**.
- b. Select a time period in days, months, or years. For example, you might select **1 Year**. When you click **Remove**, device fingerprints that have not been used in more than 1 year are removed from the database. The default is 1 year.
- c. Click **Remove**.

Remove user session data:

- a. Select **Remove user session data**.
- b. Select a time period in hours or days. For example, you might select **12 Hours**. When you click **Remove**, data older than 12 hours is removed from the database. The default is 1 day.
- c. Click **Remove**.

Remove all data from the runtime database:

Attention: Use this selection only if you want to remove *all* data from the runtime database.

- a. Select **Remove all data from the runtime database**.
- b. Click **Remove**.
- c. Respond to the confirmation message. Click **Yes** to remove all data or **No** to cancel the removal.

Note: If the runtime was not stopped before you selected an action, a message is displayed that asks if the runtime can be stopped. The runtime must be stopped before any removal actions can occur.

5. After the selected action is completed, click **Refresh Status** to see the time that processing was completed and the number of records removed. If the status indicates that an error occurred, correct the error and try the action again.
6. When the database record has been removed, restart the runtime by clicking **Go to Runtime Tuning Parameters to restart the runtime**.

Deploying an external runtime database

To optimize performance or increase capacity, you can optionally deploy an external runtime database for the appliance. You can configure the appliance to connect to DB2® or SolidDB on an external server.

About this task

The IBM Security Access Manager for Mobile appliance includes an internal SolidDB database to store user data such as session attributes and device fingerprints. This embedded database is suitable for small environments. If you are working in a production environment, use an external runtime database that can handle the required volume of data.

There are scripts available on the appliance that you can use to deploy the runtime database on an external SolidDB or DB2 server. You can then configure the appliance to use the external database. Using an external database can provide performance benefits and increased storage capacity.

Procedure

1. Use the File Downloads management page in the LMI to access the runtime database deployment files that are relevant to your environment.

Table 2. Runtime database deployment scripts

Database type	Deployment scripts
SolidDB	/mga/database/isam_mobile_soliddb.sql
DB2	/mga/database/isam_mobile_db2.sql

2. Save the deployment script locally on the database server.
3. Run the provided scripts on your SolidDB or DB2 server to create the external database.

SolidDB

- a. Log in to the **solsql** utility. For example:

```
/opt/solidDB/soliddb-7.0/bin/solsql <network_name> <username>  
<password>
```

where:

<network_name>
The network name of the solidDB® server.

<username>
The user name for the database administrator.

<password>
The password for the database administrator.

- b. Run the following command in the SolidDB SQL Editor:
`@<fully_qualified_path_to_script>.`
For example:
`@/tmp/isam_mobile_soliddb.sql`

DB2

- a. Create a DB2 instance to contain the runtime database. For information about creating the DB2 instance, see the DB2 documentation.
- b. Open the `isam_mobile_db2.sql` file in an editor on the DB2 server.
- c. Replace the following macros with the values specific to your environment:

&DBINSTANCE

The name of the DB2 instance.

&DBUSER

The name of the DB2 administrator.

&DBPASSWORD

The password for the DB2 administrator.

- d. Save your changes to the script.
- e. Log in to the DB2 Command utility (Windows) or DB2 host (UNIX) as the DB2 administrator.
- f. Run the following command:

```
db2 -tsvf <fully_qualified_path_to_script>
```

For example:

```
db2 -tsvf /tmp/isam_mobile_db2.sql
```

4. Validate that the tables were successfully created. Ensure that no errors were returned during the creation and log in to the database to manually check that the tables exist.
5. From the top menu of the LMI, select **Manage System Settings > Cluster Configuration** to open the Cluster Configuration management page.
6. Select the **Database** tab.
7. Enter the JDBC connection information. In particular, you must enter the following details:

Type The database type, which is either DB2 or Solid DB.

Address

The IP address of the external database server.

Port The port on which the external database server is listening.

Username

The name of the database administrator.

Password

The password for the database administrator.

If you are using DB2, there are two extra configuration items:

Secure

Select this check box to create a secure connection with the DB2 server.

Note: Before a secure connection can be established, you must first import the certificate for the appliance to use for communication with the DB2 server. Use the **SSL Certificates** page to import the appropriate certificate.

Database name

The name of the database instance on the external DB2 server.

Complete the following steps to identify and specify the DB2 database name when your DB2 database is remote to the cluster that you are configuring:

- a. Open the `isam_mobile_db2.sql` file that was used to create the database and tables.
- b. In the **CREATE DATABASE** entry, get the name that is specified. For example, in the following entry, `HVDB` is the string that identifies the default database name:

```
CREATE DATABASE HVDB ALIAS HVDB using codeset UTF-8 territory us
COLLATE USING UCA400_NO PAGESIZE 8192 WITH "HVDB Tables";
```

8. Click **Save**.
9. Deploy the changes. The following error message returns in the **Database Maintenance** panel after the location of the runtime database is changed from **Local to the cluster** to **Remote to the cluster** in the **Cluster Configuration** panel:

System Error FBTRBA091E The retrieval failed because the resource cannot be found.

Complete the following steps to restart the local management interface (LMI) and resolve this error:

- a. Use an ssh session to access the LMI. Log in as the administrator.
- b. Log in as the administrator.
- c. Type `lmi`, and press Enter.
- d. Type `restart`, and press Enter.
- e. Type `exit`, and press Enter.

Results

The appliance is configured to use the runtime database that is deployed on the external system.

What to do next

- Tune the external database by setting the configuration parameters that are described in “Runtime database tuning parameters” on page 101.
- If you are using a SolidDB external runtime database, change the **DurabilityLevel** to 3 from the default value of 1 to prevent loss of data. Edit the `solid.ini` file for the runtime database to change the **DurabilityLevel**:

```
DurabilityLevel = 3
```

After you modify and save the `solid.ini` file, shutdown and restart the runtime database.

Database usage requirements

Security Access Manager for Mobile uses the runtime database to store user data such as the attributes with which users are associated.

A range of storage space is available in the runtime database.

The average size per user and the maximum size per user are based on the default settings:

- There are 10 devices per user, and each device has 30 attributes.
- Each user has one session at a time.
- There are 1,000 records of behavior data per user.

Note: The administrator can configure custom database settings or use the default settings.

Space requirements for average-length attributes

100 KB is the average amount of space per user in the runtime database.

Your database can use more than 100 KB of space per user and still run effectively.

100 KB is the suggested amount of space for your database to use per user because databases that use 100 KB per user:

- Allow context-based access to run effectively and efficiently.
- Prevent wasted space in the database.

Do not use less than 100 KB of space per user. Databases that use less than 100 KB of space per user might run out of space. Databases that run out of space experience errors when context-based access runs.

Space requirements for maximum-length attributes

3 MB is the maximum amount of space that is possible per user in the runtime database if every attribute uses 2,000 bytes. 2,000 bytes is the maximum amount of space that each attribute can use.

Runtime database tuning parameters

To improve performance, you can configure the database tuning parameters on the Mobile Advanced Configuration page in the local management interface (LMI).

From the top menu, select **Secure Mobile Settings > Manage > Advanced Configuration** to access these configuration parameters.

Table 3 on page 102 contains the key parameters that you can use to tune the runtime database.

For a full list of the available parameters, see the managing advanced configuration section in the *IBM Security Access Manager for Mobile Configuration Guide*.

Table 3. Runtime database tuning parameters

Parameter	Description
attributeCollection.sessionTimeout	The appliance keeps the collected session data for context-based access in the runtime database tables. In an environment with a high volume of transactions, the tables build quickly. Consider the rate of transactions in your runtime environment to determine an appropriate timeout value.
deviceRegistration.maxRegisteredDevices	The device registration process creates entries across numerous tables in the runtime database. This value limits the maximum number of devices that each user can register. A user can continue to register new devices until this maximum is reached. In a dynamic environment where every user has multiple devices, set this value to a number that represents a reasonable number of devices per user. To limit the volume of data in the database, do not use an excessive number.
deviceRegistration.maxUsageDataPerUser	The number of records each user can have in the runtime database table that holds usage data. If a new usage transaction is received after a user reaches this limit, the oldest record for the user is removed to accommodate the new data. That is, the system retains the most recent usage records for each user. In a large deployment, set this value to a number that retains the necessary usage records without overloading the table with unnecessary data.
distributedMap.getRetryDelay	The amount of time, in milliseconds, to wait before the appliance does another retrieval against the distributed map. In a cluster environment with failover support, you can use this value to cater for failover time. For example, distributedMap.getRetryDelay = 500. Note: Increasing this value might result in longer response times.

Table 3. Runtime database tuning parameters (continued)

Parameter	Description
distributedMap.getRetryLimit	The number of retrievals that are done against the distributed map before the appliance returns that the retrieved data is not in the distributed map. The default value is zero, which means that the retry is disabled. You can use this value with the distributedMap.getRetryDelay value to control the behavior of the appliance when it tries to retrieve data from the distributed map. In a cluster environment with failover support, you might want to permit multiple retrievals by setting a value such as 5. If there is network latency in the environment between cluster members, you can increase this number of retries along with the retry delay. Note: Increasing this value might result in longer response times.

Chapter 11. Policy information points

Typical access control policy scenarios evaluate information that is included in the request and more information from other sources. This additional information, in the form of attributes, is derived or calculated from the request itself or from another source, such as a database. These sources are called *policy information points (PIPs)*.

The appliance provides several policy information points:

Session attribute PIP

Returns attributes captured on the session database entry.

GeoLocation attribute PIP

Returns geographic location attributes.

Risk Calculator PIP

Returns the RiskScore attribute.

IP Reputation PIP

Returns the IP address reputation.

User Fingerprint Count PIP

Returns the number of fingerprints that are registered for a user.

RESTful Web Service PIP

Returns attributes from data that is obtained from a RESTful web service hosted outside of the appliance. You can configure multiple instances of this PIP to access different web services.

Note: This PIP requires configuration before you can use it. For information, see the steps for adding a policy information point in “Managing policy information points.”

Managing policy information points

The appliance provides a specific set of policy information points. The policy information points that use data collected by the appliance are ready to use without additional configuration. However, to use data from sources outside of the appliance, you must add that source as a RESTful web service policy information point.

Before you begin

To use attributes from an external data source, you must add the attributes to the appliance before you add the policy information point. See the steps for adding an attribute in “Managing attributes” on page 17.

Procedure

1. Log in to the local management interface.
2. Click **Secure Mobile Settings**.
3. Under **Policy**, click **Information Points**.
4. Take one of the following actions:

Filter policy information points:

Complete these steps to filter web service policy information points:

- a. In the Quick Filter field, type one or more characters . For example, enter g to search for all that contain g or G.
- b. Press Enter.

Add a policy information point:

Complete these steps to configure a web service policy information point:

- a. Click .
- b. Complete the connection and attribute properties for the information point. See “RESTful web service properties.”

Modify a policy information point:

Complete these steps to modify a web service policy information point that you previously configured:

- a. Select a policy information point.
- b. Click .
- c. Complete the connection and attribute properties for the policy information point. See “RESTful web service properties.”

Delete a policy information point:

Complete these steps to remove a web service policy information point that you previously configured:

- a. Select a policy information point.
- b. Click .

Attention: Do not delete a policy information point if it returns attributes that are used in a policy or risk score.

RESTful web service properties

When you add or modify a RESTful web service policy information point, you must specify its properties.

Connection properties

Name A unique name for the policy information point. Use this name as the Issuer for the attributes that are returned by this policy information point.

Description

A description of the policy information point. (Optional)

Type This field is read only. The type is **RESTful Web Service**.

URL The URL for the RESTful web service starting with http (plain-text) or https (secure HTTP). For example:

`https://example.ibm.com/jaxrs/getApprovedAmount/`

You can also dynamically create the URL by using the values in a request at runtime from the attributes. In the following example, the username for the request is substituted in the URL at runtime:

`https://example.ibm.com/jaxrs/getApprovedAmount/{username}`

In the following example, the username and ipaddress for the request are substituted in the URL at runtime:

`https://example.ibm.com/jaxrs/getApprovedAmount/{username}/{ipAddress}`

Response Format

The format of the response as requested by the service through the URL. This field is read only. The type is **JSON**.

Media Type

The Accept header in the request. The default type is **application/json**.

Certificate Database

If https is used on the RESTful web service URL, specify the key database for the server SSL certificate. For example, `rt_profile_keys.kdb`

Client Authentication

If you require client authentication, select the type of authentication and its appropriate properties:

Basic Authentication

An authentication method that uses a user name and a password.

Username

User name required to authenticate the client.

Password

Password required to authenticate the client.

Client Certificate

An authentication method that requires the client to present an SSL certificate.

Attention: This property is valid only if the RESTful web service is using HTTPS.

Certificate Database

The key database where the client SSL certificate is stored. For example, `rt_profile_keys.kdb`

Certificate Label

The label of the client certificate.

Attributes properties**Attribute**

The attributes that are retrieved from a JSON response and that can be used in a policy or risk score. The selector is used to retrieve the values from the JSON object that is returned by the RESTful web service. The values are mapped to the associated attributes. You can use one or more attributes and you can add, modify, or delete attributes.

Attention:

- You must add the attributes to the appliance before you can use the attributes in this property. See the steps for adding an attribute in “Managing attributes” on page 17.
- Do not delete an attribute that is used in a policy or risk score.

Selector

The JSON selector string. All attribute selectors must return either a primitive type or an array of primitive types. If the selector references complex types, a policy evaluation error occurs and access to the system is denied.

The following JSON example shows various selector formats:

```
{"name": "Bill", "loan": {"amount":100, "rate":0.15, "duration":60}, "accounts":[10000, 2000, 500]}
```

\$.x Returns the value for the property named x in the JSON object.
Example: \$.name returns 'Bill'.

\$.x....z Returns the value for the property within a nested JSON object.
Example: \$.loan.amount returns 100.

\$.x[*] Returns the array found at the specified property. Example:
\$.accounts[*] returns [10000, 2000, 500]. The attribute is multivalued with each object in the array as a value.

Additional selector formats are:

\$.[*] Returns an array contained in the JSON response. The attribute is multivalued with each object in the array as a value.

Example: If the data is ["joe", "bob", "ted"] , \$.[*] returns ["joe", "bob", "ted"]

\$.[x] Returns a value from a JSON array index, where x represents the index on the array of the value you want to return.

Example: If the data is ["joe", "bob", "ted"] , \$.[1] returns ["bob"]

\$.[*].x Returns values from a property within an array of JSON objects.

Example: If the data is:

```
[
  { "name": "joe", "phone": "555-1212" },
  { "name": "bill", "phone": "555-1213" },
  { "name": "ted", "phone": "555-1214" }
]
```

\$.[*].name returns ["joe", "bill", "ted"]

Chapter 12. Deploying pending changes

Some configuration and administration changes require an extra deployment step.

About this task

When you use the graphical user interface on the appliance to specify changes, some configuration and administration tasks take effect immediately. Other tasks require a deployment step to take effect. For these tasks, the appliance gives you a choice of deploying immediately or deploying later. When you must make multiple changes, you can wait until all changes are complete, and then deploy all of them at one time.

When a deployment step is required, the user interface presents a message that says that there is an undeployed change. The number of pending changes is displayed in the message, and increments for each change you make.

Note: If any of the changes require the runtime server to be restarted, the restart occurs automatically when you select **Deploy**. The runtime server will then be unavailable for a period of time until the restart completes.

Procedure

1. When you finish making configuration changes, select **Click here to review the changes or apply them to the system**.

The Deploy Pending Changes window is displayed.

2. Select one of the following options:

Option	Description
Cancel	Do not deploy the changes now. Retain the undeployed configuration changes. The appliance user interface returns to the previous panel.
Roll Back	Abandon configuration changes. A message is displayed, stating that the pending changes were reverted. The appliance user interface returns to the previous panel.
Deploy	Deploy all configuration changes. When you select Deploy , a system message is displayed, stating that the changes were deployed. If any of the changes require the runtime server to be restarted, the restart occurs automatically when you select Deploy . The runtime server will then be unavailable for a period of time until the restart completes.

Chapter 13. User self-administration tasks

Administrators can configure context-based access to enable users to perform certain self-management tasks.

A common user task is to manage registered devices. For example, users can view the attributes for a device. Also, a user can rename a device.

Users can also view, specify and configure secret keys for use with one-time password.

Managing your registered devices

You can modify the registered device fingerprints that the risk engine compares with incoming request device fingerprints in risk score calculation scenarios.

About this task

You can complete the following tasks:

- View a list of your registered devices and the list of attributes that each device has.
- Rename your registered devices.
- Remove your registered devices.
- Disable your registered devices. When you disable a registered device, the device is still registered. However, when a login request comes from the disabled device, the risk engine gives that session the highest level of risk possible, which is 100.
- Enable your registered devices.

Procedure

Take one of the following actions:

View your registered devices and the attributes that each device has

1. Log in to http://hostname/mga/sps/mga/user/mgmt/html/device/device_selection.html.
2. Click the name of a device from the table to view the attributes of that device.

Note: You can also use the following URL to go directly to the attributes page of a specific device: http://hostname/mga/sps/mga/user/mgmt/html/device/device_attributes.html?id=x.

The query string, `?id=x` indicates the device that you are trying to access. The *x* represents the ID of the device.

Rename your registered devices

1. Log in to http://hostname/mga/sps/mga/user/mgmt/html/device/device_selection.html.

Note: If you select a specific device name, `http://hostname/mga/sps/mga/user/mgmt/html/device/device_attributes.html` loads with an attributes page specific to that device.

2. Click the device that you want to rename.
3. Type the name into the device name field, and click **Rename Device**.

Note: The device name must begin with an alphabetic character. Do not use control characters, leading and trailing blanks, and the following special characters: `~ ! @ # $ % ^ & * ( ) + | ` = \ ; " ' < > ? , [] { } /` anywhere in the name.

Remove a registered device

1. Log in to `http://hostname/mga/sps/mga/user/mgmt/html/device/device_selection.html`.

Note: If you select a specific device name, `http://hostname/mga/sps/mga/user/mgmt/html/device/device_attributes.html` loads with an attributes page specific to that device.

2. Click **Remove** next to the device that you want to remove.

Note: You can also click the device that you want to remove and view the attributes that belong to the device before you remove the device. Click **Remove Device** underneath the attributes when you are ready to remove the device.

Disable a registered device

1. Log in to `http://hostname/mga/sps/mga/user/mgmt/html/device/device_selection.html`.
2. Clear the **Enabled** box next to the device that you want to disable.

Enable a registered device

1. Log in to `http://hostname/mga/sps/mga/user/mgmt/html/device/device_selection.html`.
2. Select the **Enabled** box next to the device that you want to enable.

“REST services for device fingerprint registration” on page 95

You can use the REST services capability to manage your mobile data such as your registered devices.

Managing OTP secret keys

To help manage your mobile data, such as your HOTP secret keys and TOTP secret keys, you can use the REST services capability.

About this task

You can use this capability to complete the following tasks:

- View your OTP secret keys.
- Configure your OTP secret keys.
- Reset your OTP secret keys.

Procedure

1. Log in to `http://hostname/mga/sps/mga/user/mgmt/html/otp/otp.html`.
2. Perform one or more of the following actions:

- View your current OTP secret key. When you log in, you see one or both of the following in clear text and as quick response (QR) code:
 - HOTP secret key
 - TOTP secret key
- Configure your secret key in your OTP generator:
 - a. Use the clear text to manually enter either the HOTP secret key or the TOTP secret key into your OTP provider.
 - b. Scan the QR code. When you scan the QR code, the OTP generator:
 - 1) Analyzes the QR code.
 - 2) Acquires the following information from the QR code:
 - Secret key.
 - Account with which the secret key is associated.
 - Type of OTP with which the secret key is associated.
 - 3) Enter a secret key into your OTP generator.
- Reset your OTP secret key by clicking **reset**.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features contained in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM might have patents or pending patent applications that cover subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it to enable: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003
U.S.A.

Such information might be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments might vary significantly. Some measurements might have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements might have been estimated through extrapolation. Actual results might vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements, or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding the future direction or intent of IBM are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing, or distributing application programs that conform to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample

programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows: © (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. 2004, 2012. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations might not appear.

Privacy Policy Considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, See IBM's Privacy Policy at <http://www.ibm.com/privacy> and IBM's Online Privacy Statement at <http://www.ibm.com/privacy/details> the section entitled "Cookies, Web Beacons and Other Technologies" and the "IBM Software Products and Software-as-a-Service Privacy Statement" at <http://www.ibm.com/software/info/product-privacy>.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both: <http://www.ibm.com/legal/copytrade.shtml>

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

The Oracle Outside In Technology included herein is subject to a restricted use license and can only be used in conjunction with this application.

Index

A

- accessibility x
- advanced configuration
 - database tuning parameters 101
- API definition
 - API definition
 - attaching to resource 81
 - publishing 81
- appliance
 - configuring 55
- attribute collection service
 - configuration
 - application login page 48
 - JavaScript functions 48
 - session attributes 48
 - definition 2, 45
 - JavaScript functions
 - deleteSession() 45
 - getLocation() 45
 - sendSession() 45
 - request types
 - DELETE 45
 - GET 45
 - POST 45
 - session attributes 45
- attribute comparison 51
- attribute IDs
 - WebSEAL configuration 42
- attribute matchers
 - definition 51
 - exact matcher 51
 - modifying 56
- attributeCollection.sessionTimeout 101
- attributes
 - category 17, 41
 - data type 41
 - device registration 9
 - dynamic 33
 - managing 17
 - scenarios for dynamic attributes 34
 - update using dynamic.attributes.js file 35
 - updating geolocation 19
- authentication level 93
- azn-decision-info stanza 37
- azn-decision-info stanza entry
 - azn-decision-info stanza 37

C

- category
 - set for attribute 42
 - user-attribute-definitions stanza 41
- configuration
 - application login page 48
 - attribute collection service 48
 - attribute matchers 9
 - hash algorithm 32
 - risk profile 9
 - session attributes 9

- configuration (*continued*)
 - weights of attributes 9
- consent-based device registration 91, 93
- consent-based registration 95
- context-based access
 - architecture 2
 - business scenarios 1
 - components 2
 - database usage requirements 101
 - definition 1
 - dynamic.attributes.js file 35
 - features 2
 - overview 1
 - functional 2
 - process 4
 - runtime database 101
 - transaction flow 4
 - update attributes 35
- custom attributes
 - modify category 41
 - modify data type 41

D

- data type
 - set for attribute 42
 - user-attribute-definitions stanza 41
- database
 - IP reputation 54
- database tuning 101
- database usage requirements
 - context-based access 101
 - runtime database 101
- deploying changes 109
- device
 - consent-based registration 91, 92
 - fingerprint 91, 92
 - registration 91, 92
 - silent registration 91, 92
- device fingerprint
 - risk score calculation 9
- device fingerprint template page 95
- device fingerprint template page for consent-based registration 95
- device fingerprints
 - managing 91, 111
- device registration
 - scenario example 88
- deviceRegistration
 - .maxRegisteredDevices 101
- deviceRegistration
 - .maxUsageDataPerUser 101
- distributedMap.getRetryDelay 101
- distributedMap.getRetryLimit 101
- dynamic attributes
 - overview 33
 - scenarios 34
- dynamic.attributes.js file
 - code to update 35
 - deploy 36
 - update on appliance 36

E

- EAS
 - set category 42
 - set data type 42
- education x
- entries
 - azn-decision-info
 - azn-decision-info stanza 37
- External authorization service (EAS)
 - definition 2

G

- geolocation data
 - updating 19

H

- hash algorithm
 - configuration 32
- HOTP secret keys 112

I

- IBM
 - Software Support x
 - Support Assistant x
- IP reputation
 - database 54
 - policy information point 54
- ip reputation database 55

J

- JavaScript file
 - dynamic.attributes.js 35

L

- license server 55
- location attributes
 - attribute collection service 45

M

- Mobile Advanced Configuration
 - database tuning parameters 101

N

- notices 115

O

- obligations
 - managing 59
 - overview 59

- obligations (*continued*)
 - URL 61
- obligations-urls-mapping stanza
 - obligation entry 62
- one-time password
 - scenario example 87
- online
 - publications ix
 - terminology ix
- OTP secret keys 112

P

- pending changes 109
- policies
 - examples 84
 - managing 75
 - predefined attributes reference 23
 - scenarios 84
- policy
 - creating 76
 - policy
 - attaching to resource 81
 - publishing 81
- Policy administration point (PAP)
 - definition 2
- Policy decision point (PDP)
 - definition 2
- Policy enforcement point (PEP)
 - definition 2
- policy information point
 - IP reputation 54
- policy information points
 - description 105
 - managing 105
 - properties 106
- Policy information points (PIPs)
 - definition 2
- policy scenario
 - deny access using conditions 84
 - deny access with OR clause 85
 - permit access using AND clause 86
 - permit access using one-time password 87
 - register device 88
- policy sets
 - managing 79
- predefined attributes reference 23
 - policies 23
 - risk profiles 23
- predefined obligations 63
 - policy authoring 63
- predefined risk profiles 68
 - risk profiles 65
 - risk score calculation 68
- problem-determination x
- publications
 - accessing online ix
 - list of for this product ix

R

- Representational State Transfer (REST)
 - service
 - attribute collection service 45

- rest services for device fingerprint
 - registration 95
- risk engine
 - risk score calculation 9
- risk profile
 - configuration 9
 - managing 65
- risk profiles 65
 - predefined attributes reference 23
 - predefined risk profiles 65
- risk reports 13
 - accessing 15
 - configuring 13
- risk score
 - risk score calculation 9
- risk score calculation 9
 - device fingerprint 9
 - risk engine 9
 - risk score 9
 - threshold score 9
- risk-based access
 - predefined attributes reference 23
 - risk profiles 23
- risk-scoring engine
 - definition 2
- runtime database
 - managing 97
 - overview 97
 - store user data 101
 - tuning 101

S

- sample
 - silent registration 92
- scenarios
 - dynamic attributes 34
- secret keys
 - management 112
- session attributes
 - attribute collection service 45, 48
 - location attributes 45
 - web browser attributes 45
- silent device registration 91
 - sample policy 92

T

- template files root
 - manage 93
 - modifying
 - consent template pages 93
- terminology ix
- threshold score
 - risk score calculation 9
- TOTP secret keys 112
- training x
- troubleshooting x
- tuning
 - runtime database 101

U

- URL
 - obligation 61
- user self-administration 111

- user-attribute-definitions
 - set category 42
 - set type 42
- user-attribute-definitions stanza
 - category 41
 - data type 41

W

- web browser attributes
 - attribute collection service 45
- WebSEAL configuration file
 - set category 42
 - set data type 42



Product Number: 5725-L52

Printed in USA

SC27-6207-01

